

Lazaros Iliadis
Harris Papadopoulos
Chrisina Jayne (Eds.)

Communications in Computer and Information Science 383

Engineering Applications of Neural Networks

14th International Conference, EANN 2013
Halkidiki, Greece, September 2013
Proceedings, Part I

Part 1

Editorial Board

Simone Diniz Junqueira Barbosa

*Pontifical Catholic University of Rio de Janeiro (PUC-Rio),
Rio de Janeiro, Brazil*

Phoebe Chen

La Trobe University, Melbourne, Australia

Alfredo Cuzzocrea

ICAR-CNR and University of Calabria, Italy

Xiaoyong Du

Renmin University of China, Beijing, China

Joaquim Filipe

Polytechnic Institute of Setúbal, Portugal

Orhun Kara

TÜBİTAK BİLGE and Middle East Technical University, Turkey

Igor Kotenko

*St. Petersburg Institute for Informatics and Automation
of the Russian Academy of Sciences, Russia*

Krishna M. Sivalingam

Indian Institute of Technology Madras, India

Dominik Ślęzak

University of Warsaw and Infobright, Poland

Takashi Washio

Osaka University, Japan

Xiaokang Yang

Shanghai Jiao Tong University, China

Lazaros Iliadis Harris Papadopoulos
Chrisina Jayne (Eds.)

Engineering Applications of Neural Networks

14th International Conference, EANN 2013
Halkidiki, Greece, September 13-16, 2013
Proceedings, Part I

Volume Editors

Lazaros Iliadis

Democritus University of Thrace, Orestiada, Greece

E-mail: liliadis@fmenr.duth.gr

Harris Papadopoulos

Frederick University of Cyprus, Nicosia, Cyprus

E-mail: harris.papadopoulos@gmail.com

Chrisina Jayne

Coventry University, UK

E-mail: ab1527@coventry.ac.uk

ISSN 1865-0929

ISBN 978-3-642-41012-3

DOI 10.1007/978-3-642-41013-0

Springer Heidelberg New York Dordrecht London

e-ISSN 1865-0937

e-ISBN 978-3-642-41013-0

Library of Congress Control Number: Applied for

CR Subject Classification (1998): I.2.6, I.5.1, H.2.8, J.2, J.1, J.3, F.1.1, I.5, I.2, C.2

© Springer-Verlag Berlin Heidelberg 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Artificial Intelligence is a branch of computer science, continuously and rapidly evolving. It is a fact that more and more sophisticated modeling techniques are published in the literature all the time, capable of tackling complicated and challenging problems. Artificial Neural Networks (ANN) and other Soft Computing approaches seek inspiration from the world of biology to enable the development of real world intelligent systems.

EANN is a well established event with a very long and successful history. Eighteen years have passed since the first organization in Otaniemi, Finland, in 1995. For the following years it has a continuous and dynamic presence as a major European scientific event. An important milestone is year 2009, when its guidance by a steering committee of the INNS (*EANN Special Interest Group*) was initiated. Thus, from that moment the conference has been continuously supported technically, by the International Neural Network Society (INNS).

This volume contains the papers that were accepted for oral presentation at the 14th EANN conference and its satellite workshops. This volume belongs to the CCIS Springer Series. The event was held during September 13–16, 2013 at the “Athina Pallas” Resort and Conference Center in Halkidiki, Greece, and was supported by the Aristotle University of Thessaloniki and the Democritus University of Thrace.

Three workshops on timely AI subjects were organized successfully and collocated with EANN’2013:

1. The Second Mining Humanistic Data (MHD) Workshop supported by the Ionian University and the University of Patras. We wish to express our gratitude to Spyros Sioutas and Christos Makris for their common effort towards the organization of the Second MHD Workshop. Also we would like to thank Vassilios Verykios of the Hellenic Open University, Greece, and Evangelia Pitoura of the University of Ioannina, Greece, for their keynote lectures in the MHD workshop

2. The Third Computational Intelligence Applications in Bioinformatics (CIAB) Workshop supported by the University of Patras. We are grateful to Spyros Likothanasis for his kind efforts towards the management of the CIAB Workshop and for his keynote lecture in the frame of this event.

3. The First Innovative European Policies and Applied Measures for Developing Smart Cities (IPMSC) Workshop, supported by the Hellenic Telecommunications Organization. The IPMSC was driven by the hard work of Ioannis P. Chochliouros and Ioannis M. Stephanakis Hellenic Telecommunications Organization - OTE, Greece.

Three keynote speakers were invited and they gave lectures in timely aspects of AI and ANN. Finally, a highly interesting tutorial entitled “Neural Networks for Digital Media Analysis and Description” was delivered by Anastasios Tefas of

the Aristotle University of Thessaloniki, Greece. We wish to express our sincere thanks to the invited keynote speakers and to Anastasios Tefas.

The diverse nature of papers presented, demonstrates the vitality of neural computing and related soft computing approaches and proves the very wide range of ANN applications as well. On the other hand, this volume contains basic research papers, presenting variations and extensions of several approaches.

The Organizing Committee was delighted by the overwhelming response to the call for papers. All papers have passed through a peer review process by at least 2 independent academic referees. Where needed a third referee was consulted to resolve any conflicts. Overall 40% of the submitted manuscripts (totally 91) were accepted to be presented in the EANN and in the three satellite workshops. The accepted papers of the 8th AIAI conference are related to the following thematic topics:

evolutionary algorithms, adaptive algorithms, control approaches, soft computing applications, ANN, ensembles, bioinformatics, classification, pattern recognition, medical applications of AI, fuzzy Inference, filtering, SOM, RBF, image – video analysis, learning, social media applications, community based governance

The authors came from 28 different countries from all over Europe (e.g. Austria, Bulgaria, Cyprus, Czech Republic, Finland, France, Germany, Greece, Holland, Italy, Poland, Portugal, Slovakia, Slovenia, Spain, UK, Ukraine, Russia, Romania, Serbia), Americas (e.g. Brazil, USA, Mexico), Asia (e.g., China, India, Iran, Pakistan,), Africa (e.g. Egypt, Tunisia, Algeria) and Oceania (New Zealand).

September 2013

Lazaros Iliadis
Harris Papadopoulos
Chrisina Jayne

Organization

General Chair

Konstantinos Margaritis University of Macedonia, Greece

Advisory chairs

Nikola Kasabov KEDRI Auckland University of Technology,
Vera Kurkova New Zealand
Mikko Kolehmainen Czech Academy of Sciences, Czech Republic
 University of Eastern Finland, Finland

Honorary Chair

Dominic Palmer Brown Dean London Metropolitan University, UK

Program Committee Co-chairs

Lazaros Iliadis Democritus University of Thrace, Greece
Chrisina Jayne University of Coventry, UK
Haris Papadopoulos Frederick University, Cyprus

Workshop chair

Spyros Sioutas Ionian University, Greece
Christos Makris University of Patras, Greece

Organizing chair

Yannis Manolopoulos Aristotle University of Thessaloniki, Greece

Web chair

Ioannis Karydis Ionian University, Greece

Program Committee Members

Athanasiос Alexiou Ionian University, Greece
Luciano Alonso Renteria Universidad de Cantabria, Spain

VIII Organization

| | |
|---|---|
| Georgios Anastasopoulos | Democritus University of Thrace, Greece |
| Ioannis Andreadis | Democritus University of Thrace, Greece |
| Andreas Andreou | University of Cyprus, Cyprus |
| Costin Badica | University of Craiova, Romania |
| Zorana Bankovic | Technical University of Madrid, Spain |
| Kostas Berberidis | University of Patras, Greece |
| Nick Bessis | University of Derby, UK |
| Monica Bianchini | University of Siena, Italy |
| Ivo Bukovsky | Czech Technical University in Prague, Czech Republic |
| George Caridakis | National Technical University of Athens, Greece |
| Aristotelis Chatzioannou | Institute of Biological Research & Biotechnology, NHRF, Greece |
| Javier Fernandez De Canete Rodriguez | University of Malaga, Spain |
| Ruggero Donida Labati | University of Milano, Italy |
| Anestis Fachantidis | Aristotle University of Thessaloniki, Greece |
| Maurizio Fiasche | Politecnico di Milano, Italy |
| Ignazio Gallo | University of Insubria, Italy |
| Francisco Garcia | University of Oviedo, Spain |
| Christos Georgiadis | University of Macedonia, Greece |
| Efstratios F. Georgopoulos | Technological Educational Institute of Kalamata, Greece |
| Giorgio Gnecco | University of Genova, Italy |
| Petr Hajek | University of Pardubice, Czech Republic |
| Ioannis Hatzilygeroudis | University of Patras, Greece |
| Emmanouil Hourdakis | Forthnet, Greece |
| Raul Jimenez Naharro | Universidad de Huelva, Spain |
| Jacek Kabzinski | Lodz University of Technology, Poland |
| Antonios Kalampakas | Democritus University of Thrace, Greece |
| Ryotaro Kamimura | Hiratsuka Kanagawa, Japan |
| Kostas Karatzas | Aristotle University of Thessaloniki, Greece |
| Kostas Karpouzis | National Technical University of Athens, Greece |
| Ioannis Karydis | Ionian University, Greece |
| Katia Kermanidis | Ionian University, Greece |
| Kyriaki Kitikidou | Democritus University of Thrace, Greece |
| Petia Koprinkova-Hristova | Bulgarian Academy of Sciences, Bulgaria |
| Konstantinos Koutroumbas | National Observatory of Athens, Greece |
| Paul Krause | University of Surrey, UK |
| Pekka Kumpulainen | Tampere University of Technology, Finland |
| Efthyvoulos Kyriacou | Frederick University, Cyprus |
| Sin Wee Lee | University of East London, UK |
| Spyros Likothanasis | University of Patras, Greece |

| | |
|----------------------------|---|
| Ilias Maglogiannis | University of Piraeus, Greece |
| George Magoulas | University of London, UK |
| Mario Malcangi | University of Milano, Italy |
| Francesco Marcelloni | University of Pisa, Italy |
| Avlonitis Markos | Ionian University, Greece |
| Marisa Masvoula | University of Athens, Greece |
| Nikolaos Mitianoudis | Democritus University of Thrace, Greece |
| Haris Mouratidis | University of East London, UK |
| Phivos Mylonas | National Technical University of Athens, Greece |
| | |
| Nicoletta Nicolaou | University of Cyprus, Cyprus |
| Vladimir Olej | University of Pardubice, Czech Republic |
| Mihaela Oprea | Universitatea Petrol-Gaze din Ploiesti, Romania |
| | |
| Ioannis Partalas | Aristotle University of Thessaloniki, Greece |
| Daniel Perez | University of Oviedo, Spain |
| Elias Pimenidis | University of East London, UK |
| Jefferson Rodrigo de Souza | Universidade de Sao Paulo, Brazil |
| Nick Ryman-Tubb | University of Surrey, UK |
| Marcello Sanguineti | University of Genova, Italy |
| Thomas Schack | Bielefeld University, Germany |
| Christos Schizas | University of Cyprus |
| Abe Shigeo | Kobe University, Japan |
| Alexandros Sideridis | Agricultural University of Athens, Greece |
| Luis Silva | University of Aveiro, Portugal |
| Spyros Sioutas | Ionian University, Greece |
| Stephanos Spartalis | Democritus University of Thrace, Greece |
| Ioannis Stamelos | Aristotle University of Thessaloniki, Greece |
| Kathleen Steinhofel | King's College, UK |
| Ioannis Stephanakis | Organization of Telecommunications, Greece |
| Tatiana Tambouratzis | University of Piraeus, Greece |
| Panos Trahanias | Forthnet, Greece |
| Thanos Tsadiras | Aristotle University of Thessaloniki, Greece |
| Nicolas Tsapatsoulis | Technical University of Cyprus, Cyprus |
| George Tsekouras | University of Aegean, Greece |
| Aristeidis Tsitiridis | University of Swansea, UK |
| Grigoris Tsoumacas | Aristotle University of Thessaloniki, Greece |
| Nikolaos Vasilas | TEI of Athens, Greece |
| Panayiotis Vlamos | Ionian University, Greece |
| George Vouros | University of Piraeus, Greece |
| Peter Weller | City University, UK |
| Shigang Yue | University of Lincoln, UK |
| Achilleas Zapranis | University of Macedonia, Greece |
| Rodolfo Zunino | University of Genova, Italy |

Keynotes

Nikola Kasabov: Founding Director and Chief Scientist of the Knowledge Engineering and Discovery Research Institute (KEDRI), Auckland. Chair of Knowledge Engineering at the School of Computing and Mathematical Sciences at Auckland University of Technology. Fellow of the Royal Society of New Zealand, Fellow of the New Zealand Computer Society and a Senior Member of IEEE.

Keynote Presentation Subject: “Neurocomputing for Spatio/Spectro-Temporal Pattern Recognition and Early Event Prediction: Methods, Systems, Applications”

Neurocomputing for Spatio/Spectro-Temporal Pattern Recognition and Early Event Prediction: Methods, Systems, Applications

Nikola Kasabov, Fellow IEEE, Fellow RSNZ

Director, Knowledge Engineering and Discovery Research Institute - KEDRI,

Auckland University of Technology, NZ

nkasabov@aut.ac.nz, www.kedri.aut.ac.nz

Abstract. The talk presents a brief overview of contemporary methods for neurocomputation, including: evolving connections systems (ECOS) and evolving neuro-fuzzy systems [1]; evolving spiking neural networks (eSNN) [2-5]; evolutionary and neurogenetic systems [6]; quantum inspired evolutionary computation [7,8]; rule extraction from eSNN [9]. These methods are suitable for incremental adaptive, on-line learning from spatio-temporal data and for data mining. But the main focus of the talk is how they can learn to predict early the outcome of an input spatio-temporal pattern, before the whole pattern is entered in a system. This is demonstrated on several applications in bioinformatics, such as stroke occurrence prediction, and brain data modeling for brain-computer interfaces [10], on ecological and environmental modeling [11]. eSNN have proved superior for spatio-and spectro-temporal data analysis, modeling, pattern recognition and early event prediction as outcome of recognized patterns when partially presented.

Future directions are discussed. Materials related to the lecture, such as papers, data and software systems can be found from www.kedri.aut.ac.nz and also from: www.theneucom.com and <http://ncs.ethz.ch/projects/evospike/>.

References

- [1] Kasabov, N.: Evolving Connectionist Systems: The Knowledge Engineering Approach. Springer, London (2007), <http://www.springer.de> (first edition published in 2002)
- [2] Wysoski, S., Benuskova, L., Kasabov, N.: Evolving Spiking Neural Networks for Audio-Visual Information Processing. *Neural Networks* 23(7), 819–835 (2010)
- [3] Kasabov, N.: To spike or not to spike: A probabilistic spiking neural model. *Neural Networks* 23(1), 16–19 (2010)
- [4] Mohammed, A., Schliebs, S., Kasabov, N.: SPAN: Spike Pattern Association Neuron for Learning Spatio-Temporal Sequences. *Int. J. Neural Systems* (2011, 2012)
- [5] Kasabov, N., Dhoble, K., Nuntalid, N., Indiveri, G.: Dynamic Evolving Spiking Neural Networks for On-line Spatio- and Spectro-Temporal Pattern Recognition. *Neural Networks* 41, 188–201 (2013)
- [6] Benuskova, L., Kasabov, N.: Computational Neurogenetic Modelling. Springer (2007)
- [7] Defoin-Platel, M., Schliebs, S., Kasabov, N.: Quantum-inspired Evolutionary Algorithm: A multi-model EDA. *IEEE Trans. Evolutionary Computation* 13(6), 1218–1232 (2009)
- [8] Nuzly, H., Kasabov, N., Shamsuddin, S.: Probabilistic Evolving Spiking Neural Network Optimization Using Dynamic Quantum Inspired Particle Swarm Optimization. In: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) ICONIP 2010, Part I. LNCS, vol. 6443. Springer, Heidelberg (2010)
- [9] Soltic, S., Kasabov, N.: Knowledge extraction from evolving spiking neural networks with a rank order population coding. *Int. J. Neural Systems* 20(6), 437–445 (2010)
- [10] Kasabov, N. (ed.): The Springer Handbook of Bio- and Neuroinformatics. Springer (2013)
- [11] Schliebs, S., Platel, M.D., Worner, S., Kasabov, N.: Integrated Feature and Parameter Optimization for Evolving Spiking Neural Networks: Exploring Heterogeneous Probabilistic Models. *Neural Networks* 22, 623–632 (2009)

Erkki Oja: Professor with the Aalto University, Finland, Recipient of the 2006 IEEE Computational Intelligence Society Neural Networks Pioneer Award, Director of the Adaptive Informatics Research Centre, Chairman of the Finnish Research Council for Natural Sciences and Engineering, Visiting Professor at the Tokyo Institute of Technology, Japan, Member of the Finnish Academy of Sciences, IEEE Fellow, Founding Fellow of the International Association of Pattern Recognition (IAPR), Past President of the European Neural Network Society (ENNS), Fellow of the International Neural Network Society (INNS). Author of the scientific books:

- “Subspace Methods of Pattern Recognition”, New York: Research Studies Press and Wiley, 1983, translated into Chinese and Japanese,
- “Kohonen Maps”, Elsevier, 1999,
- “Independent Component Analysis”, Wiley, 2001 translated in Chinese and Japanese.

Machine Learning for Big Data Analytics

Erkki Oja
 Aalto University, Finland
 erkki.oja@aalto.fi

Abstract. During the past 30 years, the amount of stored digital data has roughly doubled every 40 months. Today, about 2.5 quintillion bytes are created very day. This data comes from sensor networks, cameras, microphones, mobile devices, software logs etc. Part of it is scientific data especially in particle physics, astronomy and genomics, part of it comes from other sectors of society such as internet text and documents, web logs, medical records, military surveillance, photo and video archives and e-commerce. This data poses a unique challenge in data mining: finding meaningful things out of the data masses. Central algorithmic techniques to process and mine the data are classification, clustering, neural networks, pattern recognition, regression, visualization etc. Many of these fall under the term machine learning. In the author's research group at Aalto University, Finland, machine learning techniques are developed and applied to many of the above problems together with other research institutes and industry. The talk will cover some recent algorithmic discoveries and illustrate the problem area with case studies in speech recognition and synthesis, video recognition, brain imaging, and large-scale climate research.

Marios Polycarpou is a Fellow of the IEEE and currently serves as the President of the IEEE Computational Intelligence Society. He has served as the Editor-in-Chief of the *IEEE Transactions on Neural Networks and Learning Systems* from 2004 until 2010. He participated in more than 60 research projects/grants, funded by several agencies and industry in Europe and the United States. In 2011, Dr. Polycarpou was awarded the prestigious European Research Council (ERC) Advanced Grant.

Distributed Sensor Fault Diagnosis in Big Data Environments

Marios Polycarpou
 University of Cyprus
 mpolycar@ucy.ac.cy

Abstract. The emergence of networked embedded systems and sensor/actuator networks has given rise to advanced monitoring and control applications, where a large amount of sensor data is collected and processed in real-time in order to achieve smooth and efficient operation of the underlying system. The current trend is towards larger and larger sensor data sets, leading to so called big data environments. However, in situations where faults arise in one or more of the sensing devices, this may lead to a serious degradation in performance or even to an overall system failure. The goal of this presentation is to motivate the need for fault diagnosis in complex distributed dynamical systems and to provide a

methodology for detecting and isolating multiple sensor faults in a class of non-linear dynamical systems. The detection of faults in sensor groups is conducted using robust analytical redundancy relations, formulated by structured residuals and adaptive thresholds. Various estimation algorithms will be presented and illustrated, and directions for future research will be discussed.

We hope that these proceedings will help researchers worldwide to understand and to be aware of new ANN aspects. We do believe that they will be of major interest for scientists over the globe and that they will stimulate further research in the domain of Artificial Neural Networks and AI in general.

Table of Contents – Part I

Invited

- Neural Networks for Digital Media Analysis and Description 1
Anastasios Tefas, Alexandros Iosifidis, and Ioannis Pitas

Evolutionary Algorithms

- Temperature Forecasting in the Concept of Weather Derivatives:
A Comparison between Wavelet Networks and Genetic Programming ... 12
Antonios K. Alexandiris and Michael Kampouridis
- MPEG-4 Internet Traffic Estimation Using Recurrent CGPANN 22
Gul Muhammad Khan, Fahad Ullah, and Sahibzada Ali Mahmud
- SCH-EGA: An Efficient Hybrid Algorithm for the Frequency
Assignment Problem 32
Shaohui Wu, Gang Yang, Jieping Xu, and Xirong Li
- Improving the RACAI Neural Network MSD Tagger 42
Tiberiu Boroş and Stefan Daniel Dumitrescu

Adaptive Algorithms - Control Approaches

- Neural Network Simulation of Photosynthetic Production 52
Tibor Kmet and Maria Kmetova
- A Novel Artificial Neural Network Based Space Vector Modulated DTC
and Its Comparison with Other Artificial Intelligence (AI) Control
Techniques 61
Sadhana V. Jadhav and B.N. Chaudhari

General Aspects of AI Evolution

- Thinking Machines versus Thinking Organisms 71
Petro Gopych

Soft Computing Applications

- Study of Influence of Parameter Grouping on the Error of Neural
Network Solution of the Inverse Problem of Electrical Prospecting..... 81
*Sergey Dolenko, Igor Isaev, Eugeny Obornev, Igor Persiantsev, and
Mikhail Shimelevich*

| | |
|---|-----|
| Prediction of Foreign Currency Exchange Rates Using CGPANN | 91 |
| <i>Durre Nayab, Gul Muhammad Khan, and Sahibzada Ali Mahmud</i> | |
| Coastal Hurricane Inundation Prediction for Emergency Response Using Artificial Neural Networks | 102 |
| <i>Bernard Hsieh and Jay Ratcliff</i> | |
| Crossroad Detection Using Artificial Neural Networks | 112 |
| <i>Alberto Hata, Danilo Habermann, Denis Wolf, and Fernando Osório</i> | |
| Application of Particle Swarm Optimization Algorithm to Neural Network Training Process in the Localization of the Mobile Terminal ... | 122 |
| <i>Jan Karwowski, Michał Okulewicz, and Jarosław Legierski</i> | |
| Modeling Spatiotemporal Wild Fire Data with Support Vector Machines and Artificial Neural Networks | 132 |
| <i>Georgios Karapilafis, Lazaros Iliadis, Stefanos Spartalis, S. Katsavounis, and Elias Pimenidis</i> | |
| Prediction of Surface Texture Characteristics in Turning of FRPs Using ANN | 144 |
| <i>Stefanos Karagiannis, Vassilis Iakovakis, John Kechagias, Nikos Fountas, and Nikolaos Vaxevanidis</i> | |

ANN Ensembles

| | |
|---|-----|
| A State Space Approach and Hurst Exponent for Ensemble Predictors..... | 154 |
| <i>Ryszard Szupiluk and Tomasz Ząbkowski</i> | |

Bioinformatics

| | |
|---|-----|
| 3D Molecular Modelling of the Helicase Enzyme of the Endemic, Zoonotic Greek Goat Encephalitis Virus | 165 |
| <i>Dimitrios Vlachakis, Georgia Tsiliki, and Sophia Kossida</i> | |

Classification - Pattern Recognition

| | |
|---|-----|
| Feature Comparison and Feature Fusion for Traditional Dances Recognition | 172 |
| <i>Ioannis Kapsouras, Stylianos Karanikolos, Nikolaos Nikolaidis, and Anastasios Tefas</i> | |
| Intelligent Chair Sensor: Classification of Sitting Posture | 182 |
| <i>Leonardo Martins, Rui Lucena, João Belo, Marcelo Santos, Cláudia Quaresma, Adelaide P. Jesus, and Pedro Vieira</i> | |

| | |
|--|-----|
| Hierarchical Object Recognition Model of Increased Invariance | 192 |
| <i>Aristeidis Tsitiridis, Ben Mora, and Mark Richardson</i> | |
| Detection of Damage in Composite Materials Using Classification and Novelty Detection Methods | 203 |
| <i>Ramin Amali and Bradley J. Hughes</i> | |
| Impact of Sampling on Neural Network Classification Performance in the Context of Repeat Movie Viewing | 213 |
| <i>Elena Fitkov-Norris and Sakinat Oluwabukonla Folorunso</i> | |
| Discovery of Weather Forecast Web Resources Based on Ontology and Content-Driven Hierarchical Classification | 223 |
| <i>Anastasia Mountzidou, Stefanos Vrochidis, and Ioannis Kompatsiaris</i> | |
| Towards a Wearable Coach: Classifying Sports Activities with Reservoir Computing | 233 |
| <i>Stefan Schliebs, Nikola Kasabov, Dave Parry, and Doug Hunt</i> | |
| Real-Time Psychophysiological Emotional State Estimation in Digital Gameplay Scenarios | 243 |
| <i>Pedro A. Nogueira, Rui Rodrigues, and Eugénio Oliveira</i> | |

Medical Applications of AI

| | |
|--|-----|
| Probabilistic Prediction for the Detection of Vesicoureteral Reflux | 253 |
| <i>Harris Papadopoulos and George Anastassopoulos</i> | |
| Application of a Neural Network to Improve the Automatic Measurement of Blood Pressure | 263 |
| <i>Juan Luis Salazar Mendiola, José Luis Vargas Luna, José Luis González Guerra, and Jorge Armando Cortés Ramírez</i> | |
| An Immune-Inspired Approach for Breast Cancer Classification | 273 |
| <i>Rima Daoudi, Khalifa Djemal, and Abdelkader Benyettou</i> | |
| Classification of Arrhythmia Types Using Cartesian Genetic Programming Evolved Artificial Neural Networks | 282 |
| <i>Arbab Masood Ahmad, Gul Muhammad Khan, and Sahibzada Ali Mahmud</i> | |
| Artificial Neural Networks and Principal Components Analysis for Detection of Idiopathic Pulmonary Fibrosis in Microscopy Images | 292 |
| <i>Spiros V. Georgakopoulos, Sotiris K. Tasoulis, Vassilis P. Plagianakos, and Ilias Maglogiannis</i> | |

Fuzzy Inference - Filtering

- Prediction of Air Quality Indices by Neural Networks and Fuzzy Inference Systems – The Case of Pardubice Microregion 302
Petr Hájek and Vladimír Olej

- Novel Neural Architecture for Air Data Angle Estimation 313
Manuela Battipede, Mario Cassaro, Piero Gili, and Angelo Lerro

- Audio Data Fuzzy Fusion for Source Localization 323
Mario Malcangi

- Boosting Simplified Fuzzy Neural Networks 330
Alexey Natekin and Alois Knoll

SOM-RBF

- A Parallel and Hierarchical Markovian RBF Neural Network: Preliminary Performance Evaluation 340
Yiannis Kokkinos and Konstantinos Margaritis

- Data Mining and Modelling for Wave Power Applications Using Hybrid SOM-NG Algorithm 350
*Mario J. Crespo-Ramos, Iván Machón-González,
Hilario López-García, and Jose Luis Calvo-Rolle*

- Automatic Detection of Different Harvesting Stages in Lettuce Plants by Using Chlorophyll Fluorescence Kinetics and Supervised Self Organizing Maps (SOMs) 360
*Xanthoula Eirini Pantazi, Dimitrios Moshou, Dimitrios Kasampalis,
Pavlos Tsouvaltzis, and Dimitrios Kateris*

- Analysis of Heating Systems in Buildings Using Self-Organizing Maps 370
*Pablo Barrientos, Carlos J. del Canto, Antonio Morán,
Serafín Alonso, Miguel A. Prada, Juan J. Fuertes, and
Manuel Domínguez*

Image-Video Analysis

- IMMI: Interactive Segmentation Toolkit 380
Jan Masek, Radim Burget, and Vaclav Uher

- Local Binary Patterns and Neural Networks for No-Reference Image and Video Quality Assessment 388
*Marko Panić, Dubravko Ćulibrk, Srdjan Sladojević, and
Vladimir Crnojević*

| | |
|--|-----|
| Learning Accurate Active Contours | 396 |
| <i>Adas Gelzinis, Antanas Verikas, Marija Bacauskiene, and Evaldas Vaiciukynas</i> | |
| Pattern Recognition in Thermal Images of Plants Pine Using Artificial Neural Networks | 406 |
| <i>Adimara Bentivoglio Colturato, André Benjamin Gomes, Daniel Fernando Pigatto, Danielle Bentivoglio Colturato, Alex Sandro Roschmidt Pinto, Luiz Henrique Castelo Branco, Edson Luiz Furtado, and Kalinka Regina Lucas Jaquie Castelo Branco</i> | |
| Direct Multi-label Linear Discriminant Analysis | 414 |
| <i>Maria Oikonomou and Anastasios Tefas</i> | |
| Image Restoration Method by Total Variation Minimization Using Multilayer Neural Networks Approach | 424 |
| <i>Mohammed Debakla, Khalifa Djemal, and Mohamed Benyettou</i> | |
| Learning | |
| Algorithmic Problem Solving Using Interactive Virtual Environment: A Case Study | 433 |
| <i>Plerou P. Antonia and Panayiotis M. Vlamos</i> | |
| No-Prop- <i>fast</i> - A High-Speed Multilayer Neural Network Learning Algorithm: MNIST Benchmark and Eye-Tracking Data Classification | 446 |
| <i>André Frank Krause, Kai Essig, Martina Piefke, and Thomas Schack</i> | |
| CPL Criterion Functions and Learning Algorithms Linked to the Linear Separability Concept | 456 |
| <i>Leon Bobrowski</i> | |
| Learning Errors of Environmental Mathematical Models | 466 |
| <i>Dimitri Solomatin, Vadim Kuzmin, and Durga Lal Shrestha</i> | |
| Social Media – Community Based Governance Applications | |
| On Mining Opinions from Social Media | 474 |
| <i>Vicky Politopoulou and Manolis Maragoudakis</i> | |
| Automata on Directed Graphs for the Recognition of Assembly Lines | 485 |
| <i>Antonios Kalampakas, Stefanos Spirtalis, and Lazaros Iliadis</i> | |

XX Table of Contents – Part I

| | |
|---|------------|
| On the Quantification of Missing Value Impact on Voting Advice Applications..... | 496 |
| <i>Marilena Agathokleous, Nicolas Tsapatsoulis, and Ioannis Katakis</i> | |
| Author Index | 507 |

Table of Contents – Part II

| | |
|---|----|
| Evaluating Sentiment in Annual Reports for Financial Distress Prediction Using Neural Networks and Support Vector Machines | 1 |
| <i>Petr Hájek and Vladimír Olej</i> | |
| Identification of All Exact and Approximate Inverted Repeats in Regular and Weighted Sequences | 11 |
| <i>Carl Barton, Costas S. Iliopoulos, Nicola Mulder, and Bruce Watson</i> | |
| Query Expansion with a Little Help from Twitter | 20 |
| <i>Ioannis Anagnostopoulos, Gerasimos Razis, Phivos Mylonas, and Christos-Nikolaos Anagnostopoulos</i> | |
| Recognizing Emotion Presence in Natural Language Sentences | 30 |
| <i>Isidoros Perikos and Ioannis Hatzilygeroudis</i> | |
| Classification of Event Related Potentials of Error-Related Observations Using Support Vector Machines | 40 |
| <i>Pantelis Asvestas, Erricos M. Ventouras, Irene Karanasiou, and George K. Matsopoulos</i> | |
| A Novel Hierarchical Approach to Ranking-Based Collaborative Filtering | 50 |
| <i>Athanasis N. Nikolakopoulos, Marianna Kouneli, and John Garofalakis</i> | |
| Mimicking Real Users' Interactions on Web Videos through a Controlled Experiment | 60 |
| <i>Antonia Spiridonidou, Ioannis Karydis, and Markos Avlonitis</i> | |
| Mining Student Learning Behavior and Self-assessment for Adaptive Learning Management System | 70 |
| <i>Konstantina Moutafi, Paraskevi Vergeti, Christos Alexakos, Christos Dimitrakopoulos, Konstantinos Giotopoulos, Hera Antonopoulou, and Spiros Likothanassis</i> | |
| Exploiting Fuzzy Expert Systems in Cardiology | 80 |
| <i>Efrosini Sourla, Vasileios Syrimpeis, Konstantina-Maria Stamatopoulou, Georgios Merekoulias, Athanasios Tsakalidis, and Giannis Tzimas</i> | |

| | |
|--|-----|
| The Strength of Negative Opinions | 90 |
| <i>Thanos Papaoikonomou, Mania Kardara, Konstantinos Tserpes, and Theodora Varvarigou</i> | |
| Extracting Knowledge from Web Search Engine Using Wikipedia | 100 |
| <i>Andreas Kanavos, Christos Makris, Yannis Plegas, and Evangelos Theodoridis</i> | |
| AppendicitisScan Tool: A New Tool for the Efficient Classification of Childhood Abdominal Pain Clinical Cases Using Machine Learning Tools | 110 |
| <i>Athanasis Mitroulias, Theofilatos Konstantinos, Spiros Likothanassis, and Mavroudi Seferina</i> | |
| Mining the Conceptual Model of Open Source CMS Using a Reverse Engineering Approach | 119 |
| <i>Vassiliki Gkantouna, Spyros Sioutas, Georgia Sourla, Athanasios Tsakalidis, and Giannis Tzimas</i> | |
| Representation of Possessive Pronouns in Universal Networking Language | 129 |
| <i>Velislava Stoykova</i> | |
| Sleep Spindle Detection in EEG Signals Combining HMMs and SVMs | 138 |
| <i>Iosif Mporas, Panagiotis Korvesis, Evangelia I. Zacharaki, and Vasilis Megalooikonomou</i> | |
| Classifying Ductal Trees Using Geometrical Features and Ensemble Learning Techniques | 146 |
| <i>Angeliki Skoura, Tatyana Nuzhnaya, Predrag R. Bakic, and Vasilis Megalooikonomou</i> | |
| Medical Decision Making via Artificial Neural Networks: A Smart Phone-Embedded Application Addressing Pulmonary Diseases' Diagnosis | 156 |
| <i>George-Peter K. Economou and Vaios Papaioannou</i> | |
| A Simulator for Privacy Preserving Record Linkage | 164 |
| <i>Alexandros Karakasidis and Vassilios S. Verykios</i> | |
| Development of a Clinical Decision Support System Using AI, Medical Data Mining and Web Applications | 174 |
| <i>Dimitrios Tsolis, Kallirroi Paschali, Anna Tsakona, Zafeiria-Marina Ioannou, Spiros Likothanassis, Athanasios Tsakalidis, Theodore Alexandrides, and Athanasios Tsamandas</i> | |

| | |
|---|-----|
| Supporting and Consulting Infrastructure for Educators during Distance Learning Process: The Case of Russian Verbs of Motion | 185 |
| <i>Oksana Kalita, Alexander Gartsov, Georgios Pavlidis, and Photis Nanopoulos</i> | |
| Classification Models for Alzheimer's Disease Detection | 193 |
| <i>Christos-Nikolaos Anagnostopoulos, Ioannis Giannoukos, Christian Spenger, Andrew Simmons, Patrizia Mecocci, Hikka Soininen, Iwona Kloszewska, Bruno Vellas, Simon Lovestone, and Magda Tsolaki</i> | |
| Combined Classification of Risk Factors for Appendicitis Prediction in Childhood | 203 |
| <i>Theodoros Iliou, Christos-Nikolaos Anagnostopoulos, Ioannis M. Stephanakis, and George Anastassopoulos</i> | |
| Analysis of DNA Barcode Sequences Using Neural Gas and Spectral Representation | 212 |
| <i>Antonino Fiannaca, Massimo La Rosa, Riccardo Rizzo, and Alfonso Urso</i> | |
| A Genetic Algorithm for Pancreatic Cancer Diagnosis | 222 |
| <i>Charalampos Moschopoulos, Dusan Popovic, Alejandro Sifrim, Grigoris Beligiannis, Bart De Moor, and Yves Moreau</i> | |
| Enhanced Weighted Restricted Neighborhood Search Clustering: A Novel Algorithm for Detecting Human Protein Complexes from Weighted Protein-Protein Interaction Graphs | 231 |
| <i>Christos Dimitrakopoulos, Konstantinos Theofilatos, Andreas Pegkas, Spiros Likothanassis, and Seferina Mavroudi</i> | |
| A Hybrid Approach to Feature Ranking for Microarray Data Classification | 241 |
| <i>Dusan Popovic, Alejandro Sifrim, Charalampos Moschopoulos, Yves Moreau, and Bart De Moor</i> | |
| Derivation of Cancer Related Biomarkers from DNA Methylation Data from an Epidemiological Cohort | 249 |
| <i>Ioannis Valavanis, Emmanouil G. Sifakis, Panagiotis Georgiadis, Soterios Kyrtopoulos, and Aristotelis A. Chatzioannou</i> | |
| A Particle Swarm Optimization (PSO) Model for Scheduling Nonlinear Multimedia Services in Multicommodity Fat-Tree Cloud Networks..... | 257 |
| <i>Ioannis M. Stephanakis, Ioannis P. Chochliouros, George Caridakis, and Stefanos Kollias</i> | |

XXIV Table of Contents – Part II

| | |
|--|------------|
| Intelligent and Adaptive Pervasive Future Internet: Smart Cities for the Citizens | 269 |
| <i>George Caridakis, Georgios Siolas, Phivos Mylonas, Stefanos Kollias, and Andreas Stafylopatis</i> | |
| Creative Rings for Smart Cities | 282 |
| <i>Simon Delaere, Pieter Ballon, Peter Mechant, Giorgio Parladori, Dirk Osstyn, Merce Lopez, Fabio Antonelli, Sven Maltha, Makis Stamatelatos, Ana Garcia, and Artur Serra</i> | |
| Energy Efficient E-Band Transceiver for Future Networking | 292 |
| <i>Evangelia M. Georgiadou, Mario Giovanni Frecassetti, Ioannis P. Chochliouros, Evangelos Sfakianakis, and Ioannis M. Stephanakis</i> | |
| Social and Smart: Towards an Instance of Subconscious Social Intelligence | 302 |
| <i>M. Graña, B. Apolloni, M. Fiasché, G. Galliani, C. Zizzo, G. Caridakis, G. Siolas, S. Kollias, F. Barrientos, and S. San Jose</i> | |
| Living Labs in Smart Cities as Critical Enablers for Making Real the Modern Future Internet | 312 |
| <i>Ioannis P. Chochliouros, Anastasia S. Spiliopoulou, Evangelos Sfakianakis, Evangelia M. Georgiadou, and Eleni Rethimiotaki</i> | |
| Author Index | 323 |

Neural Networks for Digital Media Analysis and Description

Anastasios Tefas, Alexandros Iosifidis, and Ioannis Pitas

Department of Informatics,
Aristotle University of Thessaloniki,
54124 Thessaloniki, Greece
`{tefas, aiosif, pitas}@aiai.csd.auth.gr`

Abstract. In this paper a short overview on recent research efforts for digital media analysis and description using neural networks is given. Neural networks are very powerful in analyzing, representing and classifying digital media content through various architectures and learning algorithms. Both unsupervised and supervised algorithms can be used for digital media feature extraction. Digital media representation can be done either in a synaptic level or at the output level. The specific problem that is used as a case study for digital media analysis is the human-centered video analysis for activity and identity recognition. Several neural network topologies, such as self organizing maps, independent subspace analysis, multi-layer perceptrons, extreme learning machines and deep learning architectures are presented and results on human activity recognition are reported.

Keywords: Neural Networks, Digital Media analysis, Activity recognition, Deep learning.

1 Introduction

Recent advances in technological equipment, like digital cameras, smart-phones, etc., have led to an increase of the available digital media, e.g., videos, captured every day. Moreover, the amount of data captured for professional media production (e.g., movies, special effects, etc) has dramatically increased and diversified using multiple sensors (e.g., 3D scanners, multi-view cameras, very high quality images, motion capture, etc), justifying the digital media analysis as a big data analysis problem. As expected, most of these data are acquired in order to describe human presence and activity and are exploited either for monitoring (visual surveillance and security) or for personal use and entertainment. Basic problems in human centered media analysis are face recognition [1], facial expression recognition [2] and human activity recognition [3]. According to YouTube statistics¹, 100 hours of video are uploaded by the users every minute. Such a data growth, as well as the importance of visual information in many applications, has necessitated the creation of methods capable of automatic processing

¹ <http://www.youtube.com/yt/press/statistics.html>

and decision making when necessary. This is why a large amount of research has been devoted in the analysis and description of digital media in the last two decades.

Artificial Neural Networks (NN), played an important role towards the direction of developing techniques which can be used for digital media analysis, representation and classification. Beyond the methods that will be described in more detail in the rest of the paper we should note recent developments in the area of deep learning neural networks [4]. Deep learning architectures have been successfully used for image retrieval [5], natural language processing [6], large scale media analysis problems [7], and feature learning [8]. Among the architectures used in deep learning are Deep and Restricted Boltzmann Machines, Auto-encoders, Convolutional neural networks, Recurrent neural networks, etc.

As it will be described in the following sections, NN-based techniques can be exploited in order to properly describe digital media, extract semantic information that is useful for analysis and make decisions, e.g., decide in which category (class) a video belongs to. We will discuss these steps in the context of two important applications, i.e., human action recognition and person identification from videos.

2 Problem Statement

Let us assume that a video database \mathcal{U} contains N_T videos depicting human actions. Let us also assume that these videos have been manually annotated, i.e., they have been classified according to the performed action and/or the ID of the persons appearing in each of them. Thus, each video i depicting a human action, called action video hereafter, is accompanied by an action class and a person ID label, α_i and h_i , respectively. We would like to employ these videos, as well as the corresponding labels α_i , h_i , $i = 1, \dots, N_T$, in order to train an algorithm that will be able to automatically perform action recognition and/or person identification, i.e., to classify a new, unknown, video to an action and/or a person ID class appearing in an action class set \mathcal{A} and/or a person ID set \mathcal{P} , respectively.

The above described process is, usually, performed in two steps, as illustrated in Figure 1. The first one exploits an appropriate action/person description in order to determine a convenient video representation. The second one exploits the obtained video representation in order to determine action class and person ID models that will be used for the classification of a new (test) video.

3 Video Representation

Video representations aiming at action recognition and person identification exploit either global human body information, in terms of binary silhouettes corresponding to the human body video frame locations, or shape and motion information appearing in local video locations. In the first case, action videos are

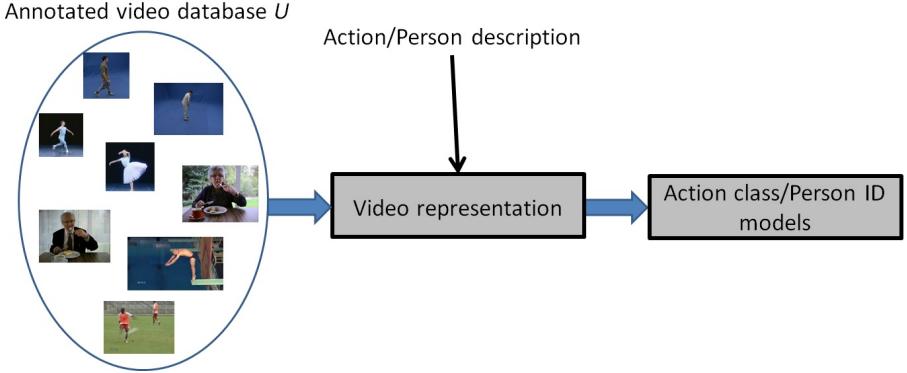


Fig. 1. Processing steps frequently used by action recognition and person identification methods



Fig. 2. Action 'walk' described by using binary human body silhouettes

usually described as sequences of successive human body poses, as illustrated in Figure 2.

By using such an action video representation, it has been shown that, in the case of everyday actions, both action recognition and person identification can be simultaneously performed [9, 10]. The adopted action video representation involves the determination of D human body pose prototypes \mathbf{v}_d , $d = 1, \dots, D$. This is achieved by training a self-organizing NN (Self-Organizing Map) exploiting the human body poses \mathbf{p}_{ij} of all the training action videos appearing in \mathcal{U} . Its training procedure involves two phases:

- **Competition:** For each of the training human body pose \mathbf{p}_{ij} , its Euclidean distance from every SOM neuron \mathbf{v}_d is calculated. Wining neuron is the one providing the smallest distance, i.e.:

$$d^* = \arg \min_d \|\mathbf{p}_{ij} - \mathbf{v}_d\|_2. \quad (1)$$

- **Co-operation:** Each SOM neuron is adapted with respect to its lateral distance from the winning neuron h_d , i.e.:

$$\mathbf{v}_d(n+1) = \mathbf{v}_d(n) + \eta(n)h_d(n)(\mathbf{p}_i - \mathbf{v}_d(n)), \quad (2)$$

where $h_d(n)$ is a function of the lateral distance $r_{d*,d}$ between the winning neuron d^* and neuron d , $\eta(n)$ is an adaptation rate parameter and n refers to the algorithms training iteration. Typical choice of $h_d(n)$ is the Gaussian function $h_d(n) = \exp\left(-\frac{r_{d*,d}^2}{2\sigma^2(n)}\right)$.

An example SOM obtained by using action videos depicting eight persons performing multiple instances of five actions is illustrated in Figure 3. As can be seen, the SOM neurons correspond to representative human body poses during action execution captured from different view angles. Furthermore, it can be observed that each SOM neuron captures human body shape properties of different persons in \mathcal{U} . For example, it can be seen that neuron $\{6, G\}$ depicts a man waving his hand and from a frontal view, while neuron $\{10, I\}$ depicts a woman jumping from a side view.

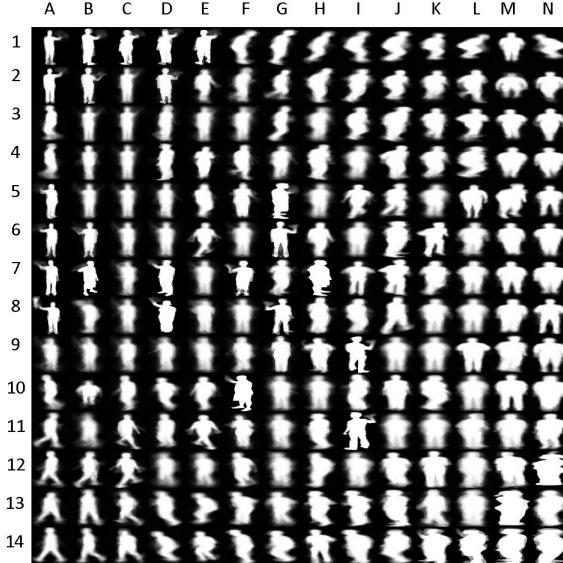


Fig. 3. A 14×14 SOM obtained by using action videos depicting eight persons performing multiple instances of actions walk, run, jump in place, jump forward and wave one hand

After SOM determination, each human body pose \mathbf{p}_{ij} is mapped to the so-called membership vector $\mathbf{u}_{ij} = [u_{ij1}, \dots, u_{ijD}]^T$ encoding the fuzzy similarity between \mathbf{p}_{ij} with all the human body prototypes \mathbf{v}_d , according to a fuzzification parameter $m > 1$:

$$u_{ijd} = (\|\mathbf{p}_{ij} - \mathbf{v}_d\|_2)^{\frac{2}{m-1}}. \quad (3)$$

Finally, each action video i , consisting of N_i video frames, is represented by the so-called action vectors $\mathbf{s}_i \in \mathbb{R}^D$:

$$\mathbf{s}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \frac{\mathbf{u}_{ij}}{\|\mathbf{u}_{ij}\|_1}. \quad (4)$$

Regarding video representations exploiting local video information, a popular choice is to use overlapping 3D video blocks, where the third dimension refers

to time, in order to learn representative 3D blocks describing local shape and motion information. Independent Subspace Analysis (ISA) has been proposed to this end in [11]. An ISA network can be considered to be a neural network consisting of two layers, with square and square-root nonlinearities in the first and second layer respectively. Let us denote by \mathbf{x}_t a given training input pattern. The activation function of each second layer unit is given by:

$$p_i(\mathbf{x}_t; \mathbf{U}, \mathbf{V}) = \left(\sum_{k=1}^m V_{ik} \left(\sum_{j=1}^m U_{kj} x_{tj} \right)^2 \right)^{\frac{1}{2}}. \quad (5)$$

Parameters \mathbf{U} are learned through finding sparse representations in the second layer by solving:

$$\begin{aligned} & \underset{\mathbf{W}}{\text{maximize}} \quad \sum_{t=1}^T \sum_{i=1}^m p_i(\mathbf{x}_t; \mathbf{U}, \mathbf{V}) \\ & \text{subject to : } \mathbf{U}\mathbf{U}^T = \mathbf{I} \end{aligned} \quad (6)$$

\mathbf{U} , \mathbf{V} in (5), (6) are matrices containing the weights connecting the input data to the first layer units and the units of the first layer to the second layer units, respectively. In order to reduce the computational cost of the training process, PCA is performed in order to reduce the dimensionality of the input data. In order to learn high-level concepts, a convolution and stacking technique is employed. According to this, small input patches are employed in order to train an ISA network. Subsequently, the learned network is convolved with a larger region of the input video. The convolved responses are fed to another ISA network. Example filters learned from video frames depicting traditional Greek dances are illustrated in Figure 4 [12].

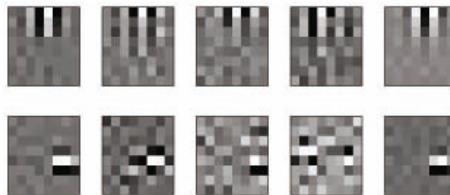


Fig. 4. Filters learned by the ISA algorithm when trained on video frames depicting traditional Greek dances

After training the two-layered ISA network by following the above described procedure, action videos are represented by using the Bag of Features (BoFs) model. That is, the responses of the ISA network corresponding to all the training action videos are clustered in order to determine a set of K representative

ISA features, which form the so-called codebook. Finally, each action video i is represented by the corresponding histogram $\mathbf{s}_i \in \mathbb{R}^K$ calculated by employing the obtained codebook.

4 Video Classification

By following either of the above described procedures, each training action video is represented by the corresponding action vector \mathbf{s}_i . That is, action video classification has been transformed to the corresponding action vector classification task. Feedforward Neural Networks have been widely adopted to this end, due to their ability to universally approximate any continuous target functions in any compact subset \mathcal{X} of the Euclidean space \mathbb{R}^D [13, 14].

A Multi-layer Perceptron (MLP), i.e., a Feedforward NN consisting of D input (equal to action vectors dimensionality) and N_A output neurons (equal to the number of classes forming the classification problem), has been employed to this end in [9] for human action recognition. In the training phase, training action vectors \mathbf{s}_i accompanied by the corresponding action class labels α_i are used in order to define MLP weights \mathbf{W} by using the Backpropagation algorithm [13]. Action class labels α_i are employed in order to set the corresponding network target vectors \mathbf{t}_i . For each of the action vectors, MLP response $\mathbf{o}_i = [o_{i1}, \dots, o_{iN_P}]^T$ is calculated by:

$$o_{ik} = f_{\text{sigmoid}}(\mathbf{w}_k^T \mathbf{s}_i), \quad (7)$$

where \mathbf{w}_k is a vector containing the MLP weights corresponding to output k . The training procedure is performed in an on-line form, i.e., adjustments of the MLP weights are performed for each training action vector. After the feed of a training action vector \mathbf{s}_i and the calculation of the MLP response \mathbf{o}_i the weight connecting neurons i and j follows the update rule:

$$\Delta \mathbf{W}_{ji}(n+1) = c \Delta \mathbf{W}_{ji}(n) + \eta \delta_j(n) \psi_i(n), \quad (8)$$

where $\delta_j(n)$ is the local gradient for neuron j , ψ_i is the output of neuron i , η is the learning rate parameter and c is a positive number, called momentum constant. This procedure is applied until the Mean Square Error (MSE) between the network output vectors \mathbf{o}_i and the network target vectors \mathbf{t}_i falls under an acceptable error rate ϵ .

Single-hidden Layer Feedforward Neural (SLFN) networks have been adopted for action recognition and person identification in [10, 16–18]. A SLFN network consists of D input, L hidden and N_A output neurons, as illustrated in Figure 5. In order to perform fast and efficient network training, the Extreme Learning Machine (ELM) algorithm [15] has been employed in [16].

In ELM, the network's input weights \mathbf{W}_{in} and the hidden layer bias values \mathbf{b} are randomly assigned, while the output weights \mathbf{W}_{out} are analytically calculated. Let \mathbf{v}_j denote the j -th column of \mathbf{W}_{in} and \mathbf{w}_k the k -th column of \mathbf{W}_{out} .

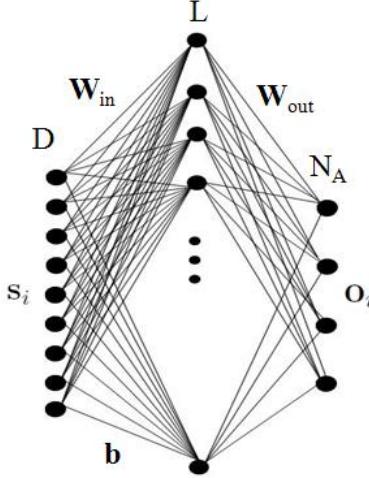


Fig. 5. SLFN network topology

For a given activation function $\Phi()$, the output $\mathbf{o}_i = [o_1, \dots, o_{N_A}]^T$ of the ELM network corresponding to training action vector \mathbf{s}_i is calculated by:

$$o_{ik} = \sum_{j=1}^L \mathbf{w}_k^T \Phi(\mathbf{v}_j, b_j, \mathbf{s}_i), \quad k = 1, \dots, N_A. \quad (9)$$

By storing the hidden layer neurons outputs in a matrix Φ , i.e.:

$$\Phi = \begin{bmatrix} \Phi(\mathbf{v}_1, b_1, \mathbf{s}_1) & \dots & \Phi(\mathbf{v}_1, b_1, \mathbf{s}_{N_T}) \\ \dots & \ddots & \dots \\ \Phi(\mathbf{v}_L, b_L, \mathbf{s}_1) & \dots & \Phi(\mathbf{v}_L, b_L, \mathbf{s}_{N_T}) \end{bmatrix}, \quad (10)$$

Equation (9) can be written in a matrix form as $\mathbf{O} = \mathbf{W}_{out}^T \Phi$. Finally, by assuming that the network's predicted outputs \mathbf{O} are equal to the network's desired outputs, i.e., $\mathbf{o}_i = \mathbf{t}_i$, and using linear activation function for the output neurons, \mathbf{W}_{out} can be analytically calculated by $\mathbf{W}_{out} = \Phi^\dagger \mathbf{T}^T$, where $\Phi^\dagger = (\Phi \Phi^T)^{-1} \Phi$ is the Moore-Penrose generalized pseudo-inverse of Φ^T and $\mathbf{T} = [\mathbf{t}_1, \dots, \mathbf{t}_{N_T}]$ is a matrix containing the network's target vectors.

A regularized version of the ELM algorithm has, also, been used in [10, 17]. According to this, the network output weights \mathbf{W}_{out} are calculated by solving the following optimization problem:

$$\text{Minimize: } L_P = \frac{1}{2} \|\mathbf{W}_{out}^T\|_F + \frac{c}{2} \sum_{i=1}^{N_V} \|\xi_i\|_2^2 \quad (11)$$

$$\text{Subject to: } \phi_i^T \mathbf{W}_{out} = \mathbf{t}_i^T - \xi_i^T, \quad i = 1, \dots, N_T, \quad (12)$$

where ξ_i is the training error vector corresponding to action vector s_i , ϕ_i denotes the i -th column of Φ , i.e., the s_i representation in the ELM space, and c is a parameter denoting the importance of the training error in the optimization problem. By substituting the condition (12) in (11) and solving for $\frac{\partial L_P}{\partial \mathbf{W}_{out}} = 0$, \mathbf{W}_{out} can be obtained by:

$$\mathbf{W}_{out} = \left(\Phi \Phi^T + \frac{1}{c} \mathbf{I} \right)^{-1} \Phi \mathbf{T}^T, \quad (13)$$

or

$$\mathbf{W}_{out} = \Phi \left(\Phi^T \Phi + \frac{1}{c} \mathbf{I} \right)^{-1} \mathbf{T}^T. \quad (14)$$

where \mathbf{I} is the identity matrix.

Exploiting the fact that the ELM algorithm can be considered to be a non-linear data mapping process to a high dimensional feature space followed by linear projection and classification, the Minimum Class Variance ELM (MCVELM) algorithm has been proposed in [18] for action recognition. MCVELM tries to simultaneously minimize the network output weights norm and within-class variance of the network outputs. The network output weights \mathbf{W}_{out} are calculated by solving the following optimization problem:

$$\text{Minimize: } L_P = \frac{1}{2} \|\mathbf{S}_w^{1/2} \mathbf{W}_{out}\|_F + \frac{c}{2} \sum_{i=1}^{N_V} \|\xi_i\|_2^2 \quad (15)$$

$$\text{Subject to: } \phi_i^T \mathbf{W}_{out} = \mathbf{t}_i^T - \xi_i^T, \quad i = 1, \dots, N_T, \quad (16)$$

and the network output weights are given by:

$$\mathbf{W}_{out} = \left(\Phi \Phi^T + \frac{1}{c} \mathbf{S}_w \right)^{-1} \Phi \mathbf{T}^T. \quad (17)$$

\mathbf{S}_w in (15), (17) is the within-class scatter matrix of the network hidden layer outputs, i.e., the representation of s_i in the so-called ELM space. Two cases have been exploited. In the case of unimodal action classes, the within-class scatter matrix is of the form:

$$\mathbf{S}_w = \sum_{j=1}^{N_A} \sum_{i=1}^{N_V} \frac{\beta_{ij}}{N_j} (\phi_i - \mu_j)(\phi_i - \mu_j)^T. \quad (18)$$

In (18), β_{ij} is an index denoting if training action vector s_i belongs to action class j , i.e., $\beta_{ij} = 1$, if $c_i = j$ and $\beta_{ij} = 0$ otherwise, and $N_j = \sum_{i=1}^{N_V} \beta_{ij}$ is the number of training action vectors belonging to action class j . $\mu_j = \frac{1}{N_j} \sum_{i=1}^{N_V} \beta_{ij} \phi_i$ is the mean vector of class j in the ELM space.

In the case of multi-modal action classes, the within-class scatter matrix is of the form:

$$\mathbf{S}_{w,CDA} = \sum_{j=1}^{N_A} \sum_{k=1}^{b_j} \sum_{i=1}^{N_V} \frac{\beta_{ijk} (\phi_i - \mu_{jk})(\phi_i - \mu_{jk})^T}{N_{jk}}. \quad (19)$$

Here, it is assumed that class j consists of b_j clusters, containing N_{jk} , $j = 1, \dots, N_A$, $k = 1, \dots, b_j$ action vectors each. β_{ijk} is an index denoting if action vector \mathbf{s}_i belongs to the k -th cluster of action class j and $\mu_{jk} = \frac{1}{N_{jk}} \sum_{i=1}^{N_V} \beta_{ijk} \phi_i$ denotes the mean vector of the k -th cluster of class j in the ELM space.

By exploiting the fast and efficient ELM algorithm for SLFN network training, a dynamic classification schemes have been proposed for human action recognition in [19]. It consists of two iteratively repeated steps. In the first step, a non-linear mapping process for both the training action vectors and the test sample under consideration is determined by training a SLFN network. In the second step, test sample-specific training action vectors selection is performed by exploiting the obtained network outputs corresponding to both the training action vectors and the test sample under consideration. SLFN-based data mapping and training action vectors selection are performed in multiple levels, which are determined by the test-sample under consideration. At each level, by exploiting only the more similar to the test sample training action vectors, the dynamic classification scheme focuses the classification problem on the classes that should be able to discriminate. A block diagram of the above described dynamic classification scheme is illustrated in Figure 6.

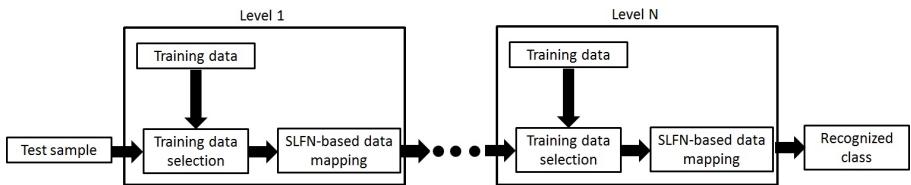


Fig. 6. SLFN-based dynamic classification scheme

Considering the fact that after performing multiple data selections for a level $l > 1$ the cardinality of the training action vectors set that will be used for SLFN network training will be very small compared to the dimensionality of the ELM space, the regularized version of ELM algorithm (13) has been employed in [19]. In order to avoid the determination of the number of hidden layer neurons at each level l , the regularized version of ELM algorithm (14) has been employed. In this case, the network output vector corresponding to \mathbf{s}_i is obtained by:

$$\mathbf{o}_i = \mathbf{W}_{out}^T \phi_i = \mathbf{T} \left(\mathbf{\Omega} + \frac{1}{c} \mathbf{I} \right)^{-1} \mathbf{K}_i, \quad (20)$$

where $\mathbf{K}_i = \Phi^T \phi_i$, $\mathbf{\Omega} = \Phi^T \Phi$ are the kernel matrices corresponding to \mathbf{s}_i and the entire SLFN training set, respectively. Thus, in this case the ELM space dimensionality is inherently determined by exploiting the kernel trick [21] and needs not be defined in advance.

Experimental results in real video data using all the previously presented methods can be found in the corresponding references. The results indicate that

various neural network topologies can be used for solving difficult tasks, such as video analysis and semantic information extraction in digital media. The results obtained indicate that neural networks are among the state-of-the-art solutions for digital media analysis, representation and classification.

5 Conclusion

In this paper a survey on neural networks based methods for digital media analysis and description is presented. Neural Networks are very powerful both on analysing/representing and on classifying digital media content. The semantic information of focus is the human activity and identity and the problem used as case-study is activity recognition and person identification from video data. The presented approaches are generic and can be easily used for other semantic concepts, especially those that involve human presence in digital media content.

Acknowledgment. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 316564 (IMPART). This publication reflects only the authors views. The European Union is not liable for any use that may be made of the information contained therein.

References

1. Kyperountas, M., Tefas, A., Pitas, I.: Dynamic training using multistage clustering for face recognition. *Pattern Recognition*, 894–905 (2008)
2. Kyperountas, M., Tefas, A., Pitas, I.: Salient feature and reliable classifier selection for facial expression classification. *Pattern Recognition*, 972–986 (2010)
3. Gkalelis, N., Tefas, A., Pitas, I.: Combining fuzzy vector quantization with linear discriminant analysis for continuous human movement recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 1511–1521 (2008)
4. Bengio, Y., Courville, A.C., Vincent, P.: Unsupervised Feature Learning and Deep Learning: A Review and New Perspectives. *Arxiv* (2012)
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems* (2012)
6. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing. In: Proceedings of the 15th International Conference on Artificial Intelligence and Statistics, AISTATS (2012)
7. Le, Q.V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G.S., Dean, J., Ng, A.Y.: Building High-level Features Using Large Scale Unsupervised Learning. In: ICML (2012)
8. Goodfellow, I., Courville, A., Bengio, Y.: Large-Scale Feature Learning With Spike-and-Slab Sparse Coding. In: ICML (2012)
9. Iosifidis, A., Tefas, A., Pitas, I.: View-invariant action recognition based on Artificial Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 23(3), 412–424 (2012)

10. Iosifidis, A., Tefas, A., Pitas, I.: Person Identification from Actions based on Artificial Neural Networks. In: Computational Intelligence in Biometrics and Identity Management, Singapore. Symposium Series on Computational Intelligence, SSCI (2013)
11. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3361–3368. IEEE Press, Colorado (2011)
12. Kapsouras, I., Karanikolos, S., Nikolaidis, N., Tefas, A.: Feature Comparison and Feature Fusion for Traditional Dances Recognition. In: 14th Engineering Applications of Neural Networks Conference, Halkidiki (2013)
13. Haykin, S.: Neural Networks and Learning Machines. Upper Saddle River, New Jersey (2008)
14. Huang, G.B., Chen, L., Siew, C.: Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks* 17(4), 879–892 (2006)
15. Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme Learning Machine for Regressiona and Multiclass Classification. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics* 42(2), 513–529 (2012)
16. Minhas, R., Baradarani, S., Seifzadeh, S., Wu, Q.J.: Human action recognition using extreme learning machine based on visual vocabularies. *Neurocomputing* 73(10–12), 1906–1917 (2010)
17. Iosifidis, A., Tefas, A., Pitas, I.: Multi-view Human Action Recognition under Occlusion based on Fuzzy Distances and Neural Networks. In: European Signal Processing Conference, pp. 1129–1133 (2012)
18. Iosifidis, A., Tefas, A., Pitas, I.: Minimum Class Variance Extreme Learning Machine for Human Action Recognition. *IEEE Transactions on Circuits and Systems for Video Technology* (accepted, 2013)
19. Iosifidis, A., Tefas, A., Pitas, I.: Dynamic action recognition based on dynemes and Extreme Learning Machine. *Pattern Recognition Letters* (accepted, 2013)
20. Iosifidis, A., Tefas, A., Pitas, I.: Dynamic Action Classification Based on Iterative Data Selection and Feedfforward Neural Networks. In: European Signal Processing Conference (accepted, 2013)
21. Scholkopf, B., Smola, A.J.: Learning with kernels: Support vector machines, regularization, optimization, and beyond. MIT Press (2001)

Temperature Forecasting in the Concept of Weather Derivatives: A Comparison between Wavelet Networks and Genetic Programming

Antonios K. Alexandiris¹ and Michael Kampouridis²

¹ School of Mathematics, Statistics and Actuarial Science, University of Kent, Canterbury, UK
A.Alexandridis@kent.ac.uk

² School of Computing, University of Kent, Medway, UK
M.Kampouridis@kent.ac.uk

Abstract. The purpose of this study is to develop a model that accurately describes the dynamics of the daily average temperature in the context of weather derivatives pricing. More precisely we compare two state of the art algorithms, namely wavelet networks and genetic programming against the classic linear approaches widely using in the contexts of temperature derivative pricing. The accuracy of the valuation process depends on the accuracy of the temperature forecasts. Our proposed models were evaluated and compared in-sample and out-of-sample in various locations. Our findings suggest that the proposed non-linear methods significantly outperform the alternative linear models and can be used for accurate weather derivative pricing.

Keywords: weather derivatives, wavelet networks, temperature derivatives, genetic programming.

1 Introduction

In this paper, we use a Wavelet Neural Networks (WN) and Genetic Programming (GP) in the context of temperature modeling and weather derivative pricing. Recently, recently a new class of financial instruments, known as “weather derivatives”, has been introduced. Weather derivatives are financial instruments that can be used by organizations or individuals as part of a risk management strategy to reduce risk associated with adverse or unexpected weather conditions, [1]. Just as traditional contingent claims, whose payoffs depend upon the price of some fundamental, a weather derivative has an underlying measure such as: rainfall, temperature, humidity, or snowfall. The difference from other derivatives is that the underlying asset has no value and it cannot be stored or traded while at the same time the weather should be quantified in order to be introduced in the weather derivative. To do so, temperature, rainfall, precipitation, or snowfall indices are introduced as underlying assets. However, in the majority of the weather derivatives, the underlying asset is a temperature index.

According to [2, 3] nearly \$1 trillion of the US economy is directly exposed to weather risk. Today, weather derivatives are being used for hedging purposes by companies and industries, whose profits can be adversely affected by unseasonal weather or for speculative purposes by hedge funds and others interested in capitalising on those volatile markets. Weather derivatives are used to hedge volume risk, rather than price risk. Hence, a model that describes accurate the temperature dynamics, the evolution of temperature, and which can be used to derive closed form solutions for the pricing of temperature derivatives is essential.

In this study two state of the art algorithms are used, namely WN and GP, in order to model the temperature dynamics. WNs were proposed by [4] as an alternative to Neural Networks, which would alleviate the weaknesses associated with Neural Networks and Wavelet Analysis. In [5], various reasons were presented in why wavelets should be used instead of other transfer functions. In particular, first, wavelets have high compression abilities, and secondly, computing the value at a single point or updating the function estimate from a new local measure involves only a small subset of coefficients. WNs have been used in a variety of applications so far, i.e., in short term load forecasting, in time-series prediction, signal classification and compression, signal denoising, static, dynamic and nonlinear modeling, nonlinear static function approximation, [5], to mention the most important and as it was presented in [1], they can constitute an accurate forecasting method in the context of weather derivatives pricing.

On the other hand, GP is a nature-inspired algorithm, which uses the principles of evolution to find computer programs that perform well in a given task, [6-8]. One of the main advantages of GP is its ability to perform well in high-dimensional combinatorial problems, such as the one of weather derivatives pricing. An additional advantage of GP is that it is a white-box technique, which thus allows the traders to visualize the trees and thus the temperature models. To our knowledge GP was applied to weather derivatives only in [9, 10]. In addition the proposed GP in [9, 10] was used for seasonal forecasting. In contrast in this study a GP is used in order to forecast daily average temperatures (DAT) in 3 European cities in which weather derivatives are actively traded.

Using models for daily temperatures can, in principle, lead to more accurate pricing than modelling temperature indices. Daily models very often show greater potential accuracy than the Historical Burn Analysis or seasonal forecasts, [1, 11], since daily modelling makes a complete use of the available historical data. The results produced by the GP and WN are compared to two traditional linear temperature modelling methods proposed by [12] and [13]. Our results are compared in 1-day-ahead forecast and to out-of-sample forecasts.

The rest of the paper is organized as follows. In Section 2 the various methods for forecasting DAT are presented. More precisely in Section 2.1 the linear models are presented while in Sections 2.2 and 2.3 the WN and the GP are discussed respectively. The data set is described in Section 3 while in Section 4 our results are presented. Finally, in Section 5 we conclude.

2 Methodology

According to [1, 14] temperature shows the following characteristics: it follows a predicted cycle, it moves around a seasonal mean, it is affected by global warming and urban effects, it appears to have autoregressive changes, its volatility is higher in winter than in summer. Following [13] a model that describes the temperature dynamics is given by a Gaussian mean-reverting Ornstein-Uhlenbeck (O-U) process defined as follows:

$$dT(t) = dS(t) + \kappa(T(t) - S(t))dt + \sigma(t)dB(t) \quad (1)$$

where $T(t)$ is the average daily temperature, κ is the speed of mean reversion, $S(t)$ is a deterministic function modelling the trend and seasonality, $\sigma(t)$ is the daily volatility of temperature variations and $B(t)$ is the driving noise process. As it was shown in [15] the term $dS(t)$ should be added for a proper mean-reversion towards the historical mean, $S(t)$. For more details on temperature modelling we refer the reader to [1].

2.1 Linear Models

Alaton. In [12] the model given by (1) is used where the seasonality in the mean is incorporated by a sinusoid function

$$S(t) = A + Bt + C \sin(\omega t + \phi) \quad (2)$$

where ϕ is the phase parameter that defines the day of the yearly minimum and maximum temperature. Since it is known that the DAT has a strong seasonality of an one year period, the parameter ω was set to $\omega = 2\pi / 365$. The linear trend caused by urbanization or climate changes is represented by $A + Bt$. The time, measured in days, is denoted by t . The parameter C defines the amplitude of the difference between the yearly minimum and maximum DAT. Another innovative characteristic of the framework presented in [12] is the introduction of seasonalities in the standard deviation modelled by a piecewise function.

Benth. In [13] a mean reverting O-U process where the noise process is modelled by a simple BM as in (1) was suggested. Both seasonal mean and (square of) daily volatility of temperature variations are modelled by truncated Fourier series:

$$S(t) = a + bt + \sum_{i=1}^{I_1} a_i \sin(2\pi i(t - f_i)/365) + \sum_{j=1}^{J_1} b_j \cos(2\pi j(t - g_j)/365) \quad (3)$$

$$\sigma^2(t) = c + \sum_{i=1}^{I_2} c_i \sin(2\pi i t / 365) + \sum_{j=1}^{J_2} d_j \cos(2\pi j t / 365) \quad (4)$$

Using truncated Fourier series a good fit for both the seasonality and the variance component can be obtained while keeping the number of parameters relative low. The above representation simplifies the needed calculations for the estimation of the parameters and for the derivation of the pricing formulas. Equations (3) and (4) allow both larger and smaller periodicities than the classical one year temperature cycle.

2.2 Wavelet Networks

In [1] a more complex model was used by applying WNs. As it was shown in [1] the solution of model (1) can be written as an AR(1) model:

$$\tilde{T}(t+1) = a\tilde{T}(t) + \tilde{\sigma}(t)\varepsilon(t) \quad (5)$$

where $\tilde{T}(t)$ is given by $\tilde{T}(t) = T(t) - S(t)$, $a = e^{-\kappa}$ and $\tilde{\sigma}(t) = a\sigma(t)$.

Intuitively, it is expected that the speed of mean reversion is not constant. If the temperature today is away from the seasonal average (a cold day in summer) then it is expected that the speed of mean reversion is high; i.e. the difference of today and tomorrow's temperature is expected to be high. In contrast if the temperature today is close to the seasonal average we expect the temperature to revert to its seasonal average slowly. To capture this feature the speed of mean reversion is modelled by a time-varying function $\kappa(t)$. Hence the structure to model the dynamics of the temperature evolution becomes:

$$dT(t) = dS(t) + \kappa(t)(T(t) - S(t))dt + \sigma(t)dB(t) \quad (6)$$

Model (5) is a lineal AR(1) model with a zero constant. Since in our analysis the speed of mean reversion is not considered constant but a time-varying function, equation, (5) can be written as follows:

$$\tilde{T}(t) = a(t-1)\tilde{T}(t-1) + \sigma(t)\varepsilon(t) \quad (7)$$

where

$$a(t) = 1 + \kappa(t) \quad (8)$$

The impact of a false specification of a , on the accuracy of the pricing of temperature derivatives is significant, [12]. In this section, we address that issue, by using a WN to estimate non-parametrically relationship (7) and then estimate a as a function of time. Moreover, previous studies [12, 13, 16-19] show that an AR(1) model is not complex enough to completely remove the autocorrelation in the residuals. Alternatively more complex models were suggested, [20, 21].

Using WNs the generalized version of (7) is estimated nonlinearly and non-parametrically, that is:

$$\tilde{T}(t+1) = \phi(\tilde{T}(t), \tilde{T}(t-1), \dots) + e(t) \quad (9)$$

Model (9) uses past temperatures (detrended and deseasonalized) over one period. Using more lags we expect to overcome the strong correlation found in the residuals in models such as in [12], [13] and [18]. However, the length of the lag series must be selected. For additional details on modelling the temperature using WN we refer to [1, 5, 22].

2.3 Genetic Programming

While the previous methods are directly using a functional form for their predictions (e.g., linear), the GP operates in a different manner. It can evolve different arithmetic expressions that can take the form of regression models. This has the advantage of flexibility, since different temperature models can be derived for each city that we are interested in.

In this work, a simple GP was used to evolve trees that predict the temperatures of a given city over a future period. The function set of the GP contained standard arithmetic operators (ADD, SUB, MUL, DIV (protected division)), along with MOD (modulo), LOG(x), SQRT(x) and the trigonometric functions of sine and cosine. The terminal set was composed of the index t representing the current day, $1 \leq t \leq$ size of training and testing set the temperatures of the last three days $\tilde{T}(t-1)$, $\tilde{T}(t-2)$ and $\tilde{T}(t-3)$, the constant π , and 10 random numbers in the range (-10, 10). In this study the GP is based on DAT of the three previous days. Similar, structures were proposed in previous studies, [23]. Nevertheless, our future work will be focused on selecting this window dynamically. The details of the GP is summarized in Table 1¹

Table 1. GP Experimental Parameters

| Parameter | Value |
|-------------------|--|
| Max initial depth | 2 |
| Max depth | 4 |
| Generations | 50 |
| Population size | 500 |
| Tournament size | 4 |
| Subtree crossover | 30% |
| Subtree mutation | 40% |
| Point mutation | 30% |
| Fitness function | Mean Square Error (MSE) |
| Function set | ADD, SUB, MUL, DIV, MOD, LOG, SQRT, SIN, COS |
| Terminal set | Index t corresponding to the current day $Temp_{t-1}$, $Temp_{t-2}$, $Temp_{t-3}$ Constant π 10 random constants in (-10, 10) |

¹ These parameters were selected after careful experimental tuning.

Finally, we should note that traditionally in the GP literature the algorithm is run many times and then statistical results are reported, e.g., the average fitness over the multiple runs, standard deviation, and the best result. This is done in order to get an overall picture of the algorithm's performance. However, because of the fact that the other algorithms tested in this paper are producing a single model only, it is not meaningful for our comparative analysis in Section 4 to use average results. Thus, we obtain the best tree in terms of training fitness (per algorithm), and compare it to the models produced by the two linear methods and the WN.

3 Data Description

For this study DATs for Amsterdam, Berlin and Paris were obtained. Temperature derivatives are actively traded in these cities through the Chicago Mercantile Exchange (CME). The data were provided by the ECAD².

The dataset consists of 4,015 values, corresponding to the DAT of 11 years, (1991–2001). In order for each year to have equal observations the 29th of February was removed from the data. Next the seasonal mean and trend were removed from the data. In order to do so, equation (2) was used in Alaton's method and (3) was used in Benth's and GP methods. In the case of WNs the seasonal mean was captured using wavelet analysis, [1].

In our analysis, the four methods will be used in order to model and then forecast detrended, deseasonalized DATs. This procedure is followed in order to avoid possible over-fitting problems of the WN and the GP in the presence of seasonalities and periodicities. Then, the forecasts are transformed back to the original temperature time-series in order to compare the performance of each algorithm.

The objective is to accurate forecast two temperature indices, namely Heating Degree Day (HDD) and Cumulative Average Temperature (CAT). Temperature derivatives are commonly written on these two temperature indices.

4 Results

In this section our proposed models will be validated out of sample. Our methods are validated and compared against two forecasting methods proposed in prior studies, the Alaton's and Benth's models. The four models will be used for forecasting out-of-sample DATs for different periods. Usually, temperature derivatives are written for a period of a month or a season and sometimes even for a year. Hence, DATs for 1, 2, 3, 6 and 12 months will be forecasted. The out-of-sample period corresponds to the period of 1st January – 31st December 2001 and every time interval starts at 1st January of 2001. Note that the DATs from 2001 were not used for the estimation of the parameters of the four models. Next the corresponding HDDs and CAT indices will be constructed.

² European Climate Assessment & Dataset project: <http://eca.knmi.nl>

The predictive power of the four models will be evaluated using two out-of-sample forecasting methods. First, we will estimate out-of-sample forecasts over a period and then 1-day-ahead forecasts over a period. The first case, in the out-of-sample forecasts, today (time step 0) temperature is known and is used to forecast the temperature tomorrow (time step 1). However, tomorrow's temperature is unknown and cannot be used to forecast the temperature 2 days ahead. Hence, we use the forecasted temperature at time step 1 to forecast the temperature at time step 2 and so on. We call this method the out-of-sample over a period forecast. The second case, the 1-day-ahead forecast, the procedure is as follows. Today (time step 0) temperature is known and is used to forecast the temperature tomorrow (time step 1). Then tomorrow's real temperature is used to forecast the temperature at time step 2 and so on. We will refer to this method as the 1-day-ahead over a period forecast. The first method can be used for out-of-period valuation of a temperature derivative, while the second one for in-period valuation. Naturally, it is expected the first method to cause larger errors.

In the USA, Canada and Australia, CME weather derivatives are based on the HDD index. A HDD is the number of degrees by which the daily temperature is below a base temperature, i.e.

$$\text{Daily HDD} = \max(0, \text{base temperature} - \text{daily average temperature})$$

The base temperature is usually 65 degrees Fahrenheit in the U.S. and 18 degrees Celsius in Europe and Japan. HDDs are usually accumulated over a month or over a season. The accumulated HDD index over a period $[\tau_1, \tau_2]$ is given by

$$\text{HDD} = \int_{\tau_1}^{\tau_2} \max(c - T(s), 0) ds \quad (10)$$

Similarly, the CAT index indicates the cumulative average temperature over a specified period. Hence, over a specified period $[\tau_1, \tau_2]$ the CAT index is given by

$$\text{CAT} = \int_{\tau_1}^{\tau_2} T(s) ds \quad (11)$$

Since we are studying 3 cities and 2 indices for 5 different time periods using two forecasting schemes, the four models are compared in 60 datasets. Our results are very promising. In the 1-day ahead forecasts the WN outperformed the alternative methods in 18 cases out of the 30. The Benth methods gave the best results 8 times while the GP in only 4. On the other hand in out-of-sample forecasts the GP outperformed the other methods in 12 cases out of 30 while the WN was best model in only 4 cases. Due to space limitations the results of the 1-day ahead forecasts for one month (1-31 January 2001) for the HDD index are presented in Table 2. The results for the remaining datasets are similar and are available from the authors upon request. In total the WN had the best predictive performance in 36.67% of the samples while the GP and Benth's method both in 26.67% and Alaton's model in only 10%. A summary of the results is presented in Table 3. More precisely, Table 3 shows the number of samples in which each method outperforms the others, i.e. has the best predictive accuracy. Percentages are reported in parentheses.

Furthermore, we were interested in statistically ranking the 4 algorithms. We thus run the non-parametric Friedman test, with the Holm's post-hoc test [24, 25]. For the out-of-sample tests the WN ranked first with an average ranking of 2.13, then the GP and Alaton rank with 2.33, and lastly Benth had a ranking of 3.19. Holm's test found that WN was significantly better than the remaining 3 algorithms, and also that the GP was significantly better than Benth (at 5% level, where the p -value of the algorithm is compared and found lower than the critical value of the Holm's test). Similarly, the ranks for 1-day-ahead tests, the rankings are as follows: 1. WN (1.46), 2. GP (2.50), 3. Alaton (2.83), 4. Benth (3.19). Holm's post-hoc test showed again that the WN is significantly better than all other 3 algorithms, at 5% significance level. Lastly, we were interested in ranking the 4 algorithms under all 60 datasets tested in this paper (we thus merged the out-of-sample and 1-day-ahead results into a single table). The best overall rank was obtained by WN (1.80), with the GP ranked second with an average rank of 2.41. Alaton and Benth were ranked third and fourth, respectively, with average ranks of 2.58 and 3.20. Holm's post-hoc test also showed that the WN's ranking is significantly better than all other 3 algorithms. In addition, the test showed that the GP's ranking is significantly better than Benth's.

Table 2. Day ahead comparison for a period of 1 month using the HDD index and the relative percentage errors

| <i>HDD/Imonth</i> | <i>Real</i> | <i>Historical</i> | <i>Alaton</i> | <i>Benth</i> | <i>WN</i> | <i>GP</i> |
|----------------------------|-------------|-------------------|---------------|--------------|--------------|-----------|
| Amsterdam | 463.6 | 449.5 | 460.4 | 458.3 | 463.8 | 464.3 |
| Berlin | 522.4 | 517.9 | 524.8 | 523.0 | 523.8 | 524.7 |
| Paris | 378.6 | 394.7 | 381.3 | 379.9 | 380.2 | 384.8 |
| Relative Percentage Errors | | | | | | |
| Amsterdam | | | 0.69% | 1.14% | 0.04% | 0.14% |
| Berlin | | | 0.46% | 0.11% | 0.27% | 0.43% |
| Paris | | | 0.72% | 0.35% | 0.41% | 1.63% |

Real and historical HDDs for the period 1 January – 31 January 2001 and estimated HDDs using the Alaton's, Benth's and the two proposed (WN and GP) methods. The second panel corresponds to the relative absolute percentage errors.

5 Conclusions

The previous analysis indicates that our results are very promising. Modelling the DAT using WNs enhanced the predictive accuracy of the temperature process. The additional accuracy of the proposed model will have an impact on the accurate pricing of temperature derivatives. In addition the GP performed very well in the out-of-sample forecasting method which is very useful for pricing weather contracts before the temperature measuring period.

Our results are preliminary and additional analysis must be contacted. First, the proposed methodologies must be tested in more locations. Second, an extensive analysis of the residuals must be contacted in both in-sample and out-of-sample sets. An understanding of the dynamics that govern the residuals will provide additional information of the validity of the proposed models. The space limitation of this paper prevents us from doing so. Other potential future work could be to further improve the GP models. At the moment, a simple GP was used. However, such GPs are open to criticisms of effective model generalization. A way of tackling this can be by using ensemble learning techniques. We aim to do this next. Also as it was mentioned earlier, the GP is currently based on DAT of the three previous days. Our goal is to allow this window to be changed dynamically through GP operators. We believe that this could lead to even more effective models.

Nevertheless, our preliminary results indicate that the proposed methods can model the dynamics of the temperature very well and they can constitute an accurate method for temperature derivatives pricing.

Table 3. Predictive performance of the four methods

| | 1-day-ahead | Out-of-sample | Total |
|--------|-------------|---------------|------------|
| WN | 18 (60%) | 4 (13%) | 22 (36.6%) |
| GP | 4 (13%) | 12 (40%) | 16 (26.7%) |
| Alaton | 0 (0%) | 6 (20%) | 6 (10.0%) |
| Benth | 8 (27%) | 8 (27%) | 16 (26.7%) |

The number of datasets that each method has the best predictive accuracy. Percentages are reported in parentheses.

References

1. Alexandridis, A., Zapranis, A.: Weather Derivatives: Modeling and Pricing Weather-Related Risk. Springer (2013)
2. Challis, S.: Bright Forecast for Profits. Reactions June edition (1999)
3. Hanley, M.: Hedging the Force of Nature. Risk Professional 1, 21–25 (1999)
4. Pati, Y.C., Krishnaprasad, P.S.: Analysis and Synthesis of Feedforward Neural Networks Using Discrete Affine Wavelet Transforms. IEEE Trans. on Neural Networks 4, 73–85 (1993)
5. Alexandridis, A.K., Zapranis, A.D.: Wavelet neural networks: A practical guide. Neural Networks 42, 1–27 (2013)
6. Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D.: Genetic Programming—An Introduction: On the Automatic Evolution of Computer Programs and Its Applications. Morgan Kaufmann Publishers. Inc., San Francisco (1998)
7. Koza, J.R.: Genetic programming as a means for programming computers by natural selection. Statistics and Computing 4, 87–112 (1994)
8. Poli, R., Langdon, W.W.B., McPhee, N.F.: Field Guide to Genetic Programming. Lulu Enterprises Uk Limited (2008)

9. Agapitos, A., O'Neill, M., Brabazon, A.: Evolving seasonal forecasting models with genetic programming in the context of pricing weather-derivatives. In: Di Chio, C., et al. (eds.) *EvoApplications 2012*. LNCS, vol. 7248, pp. 135–144. Springer, Heidelberg (2012)
10. Agapitos, A., O'Neill, M., Brabazon, A.: Genetic Programming for the Induction of Seasonal Forecasts: A Study on Weather Derivatives. In: *Financial Decision Making Using Computational Intelligence*, pp. 159–188. Springer (2012)
11. Jewson, S., Brix, A., Ziehmann, C.: *Weather Derivative Valuation: The Meteorological, Statistical, Financial and Mathematical Foundations*. Cambridge University Press, Cambridge (2005)
12. Alaton, P., Djehinse, B., Stillberg, D.: On Modelling and Pricing Weather Derivatives. *Applied Mathematical Finance* 9, 1–20 (2002)
13. Benth, F.E., Saltyte-Benth, J.: The volatility of temperature and pricing of weather derivatives. *Quantitative Finance* 7, 553–561 (2007)
14. Cao, M., Wei, J.: Weather Derivatives valuation and market price of weather risk. *Journal of Future Markets* 24, 1065–1089 (2004)
15. Dornier, F., Queruel, M.: Caution to the wind. Weather risk special report. *Energy Power Risk Management*, 30–32 (August 2000)
16. Bellini, F.: The weather derivatives market: Modelling and pricing temperature. Faculty of Economics, Ph.D. University of Lugano, Lugano, Switzerland (2005)
17. Benth, F.E., Saltyte-Benth, J.: Stochastic Modelling of Temperature Variations With a View Towards Weather Derivatives. *Applied Mathematical Finance* 12, 53–85 (2005)
18. Zapranis, A., Alexandridis, A.: Modelling Temperature Time Dependent Speed of Mean Reversion in the Context of Weather Derivative Pricing. *Applied Mathematical Finance* 15, 355–386 (2008)
19. Zapranis, A., Alexandridis, A.: Weather Derivatives Pricing: Modelling the Seasonal Residuals Variance of an Ornstein-Uhlenbeck Temperature Process With Neural Networks. *Neurocomputing* 73, 37–48 (2009)
20. Carmona, R.: Calibrating Degree Day Options. In: 3rd Seminar on Stochastic Analysis, Random Field and Applications (year)
21. Geman, H., Leonardi, M.-P.: Alternative approaches to weather derivatives pricing. *Managerial Finance* 31, 46–72 (2005)
22. Zapranis, A., Alexandridis, A.: Modeling and forecasting cumulative average temperature and heating degree day indices for weather derivative pricing. *Neural Computing & Applications* 20, 787–801 (2011)
23. Šaltytė Benth, J., Benth, F.E., Jalinskas, P.: A Spatial-temporal Model for Temperature with Seasonal Variance. *Journal of Applied Statistics* 34, 823–841 (2007)
24. Demšar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* 7, 1–30 (2006)
25. Garcia, S., Herrera, F.: An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *Journal of Machine Learning Research* 9, 66 (2008)

MPEG-4 Internet Traffic Estimation Using Recurrent CGPANN

Gul Muhammad Khan, Fahad Ullah, and Sahibzada Ali Mahmud

Department of Electrical Engineering, UET Peshawar, Pakistan
`{gk502, fahadullah, sahibzada.mahmud}@nwfpuet.edu.pk`

Abstract. Stretching across the horizon of data communication and networking, in almost every scenario, accurate bandwidth allocation has been a challenging problem. From simple online video streaming to the sophisticated communication network underlying a Smart Grid, efficient management of bandwidth is always desired. One way to achieve such efficiency lies in the science of predication: an intelligent system can be deployed that can estimate the sizes of upcoming data packets by analyzing patterns in the previously received data. This paper presents such a system that implements a fast and robust Neuro-Evolutionary algorithm known as Recurrent-Cartesian Genetic Programming evolved Artificial Neural Network (R-CGPANN). Based on the previously received 10 MPEG-4 video frames, the system estimates the size of the next frame. The simulation results show that the recurrence in CGPANN measurably outperform not only the feed forward version of the said algorithm but other contemporary methods in the field.

Keywords: Neuro-Evolution, frame estimation, recurrent neural networks, bandwidth management, Cartesian Genetic Programming.

1 Introduction

Modern day internet users pervasively use video streaming websites besides the search engine(s) and the social networking sites. For instance YouTube, an online video streaming site, is ranked 3rd globally on the Alexa ranking¹. Today, high speed broadband internet is accessible to users even in the developing countries like Pakistan. Nevertheless, because of the customer demand and the rapid improvement in the video quality, High Definition (HD) videos are commonly streamed by users which can significantly overload the network and hence compromise the overall bandwidth allocation efficiency. Hence an important criterion to achieve that efficiency is to schedule video frames based on the requirements of the streaming users. That way, bandwidth can be properly allocated according to the need of the users.

Recently, a colossal improvement has been seen in the processing power and memory of handheld and portable devices. For instance, contemporary smartphones—such as SAMSUNG Galaxy S-III and HTC One-X+ are powered by quad-core

¹ Alexa, The Web Information Company. (2013). Top Sites: the top 500 sites on the web. [Online]. Available: <http://www.alexa.com/topsites>

processors and gigabyte(s) of memory². Furthermore, an HD camera is a commonplace commodity equipped in most of the modern-day smartphones. Because of the high resolution of images and videos captured by the handheld devices, high data rates are required for transferring and streaming of multimedia across the internet. Real-time streaming comes under the category of Variable Bit Rate (VBR) and it presents a serious issue that must be properly addressed for when the VBR traffic flows, there is a dynamic change in the data rate of the flow. In a bandwidth-constrained wireless network, users tend to compete for the channel time and the required bandwidth to support such multimedia communication. Hence in such scenarios, the need of an efficient bandwidth scheduler becomes inevitable to make sure that an optimum and fair allocation is provided for each user so that to support their respective multimedia flows. Since the multimedia traffic consumes the considerable chunk of bandwidth, at times, there are situations that the required bandwidth exceeds the overall available bandwidth. To counteract such problems, the scheduler must decide and prioritize the bandwidth assignment in a way that the service times and Job Failure Rate are reduced, channel utilization is maximized, and fairness is maintained amongst the flows.

MPEG-4 supports a wide range of video codecs; from 5 Kbps low quality, low bandwidth video for handheld devices to 1 Gbps HDTV formats. Since its video support is spread across such a broad spectrum, MPEG-4 is a good choice to be used in modern day real-time video streaming. The MPEG VBR traffic has a bursty nature, therefore it essentially means that the bandwidth requirements of each streaming user vary dynamically. Peak data rate and mean data rate are the priori information available about each streaming session and their ratio is termed as the peak-to-average ratio. If the peak-to-average ratios are high, the provision of strict QoS guarantees becomes a non-trivial issue because of the strong variations in the required data rates. To ensure the provision a fair degree of acceptable QoS while scheduling MPEG-4 traffic, a traffic estimator is used to predict the upcoming MPEG-4 frames in a certain time window. Based on the output of the estimator, the scheduling system allocates bandwidth to different active MPEG-4 sessions in real time.

This paper presents a Neuro-Evolutionary (NE) algorithm R-CGPANN based MPEG-4 frame estimation system. Previously, a similar scenario has been simulated with the Feed-Forward or F-CGPANN [1]. Based on the historical data of 10 MPEG-4 video frames lying in the streaming user's buffer, the R-CGPANN predicts the upcoming video frame size. The results are obtained from both an extensive and intensive experimentation process—Network size is varied from 50 to 500 nodes with the number of recurrent nodes varying from 1 to 10. Furthermore, the results are compared with the F-CGPANN based estimation system as well as other estimating techniques. The accurate prediction of frame size is crucial because with that information in hand, one can efficiently manage bandwidth amongst a number of users in a range of scenarios [1].

Section 2 of the paper provides the detailed description of the related literature. Section 3 describes the R-CGPANN algorithm. Section 4 shows the experimental

² Alastair Stevenson. (2013, Feb 22). HTC One X+ vs Samsung Galaxy S3 vs Apple iPhone 5 head-to-head review. [Online]. Available: <http://www.v3.co.uk/v3-uk/review/2244615/htc-one-x-vs-samsung-galaxy-s3-vs-apple-iphone-5-head-to-head-review>

setup and the different scenarios to be simulated. Section 5 of the paper explains and analyzes the results obtained from the experiments. Finally the paper is concluded in Section 6.

2 Literature Review

2.1 Neuro-Evolution

Neuro-Evolution (NE) is the artificial evolution of an Artificial Neural Network (ANN) using an evolutionary algorithm. In NE, many aspects of the ANN are evolved: inputs, weights, functions, and even the topology itself. Moreover, the genetic algorithm used to evolve the ANN is the genotype and the network is the corresponding phenotype. The genotype evolves until the phenotype with the desired fitness is obtained. The solution search space for a problem is affected by the various encoded attributes of the genotype.

It has been observed that if only a specific attribute of the network, like weight, is evolved and other attribute, such as topology, is kept constant, the system doesn't lead to a novel solution because the network evolves in a more conservative environment, in that case [2]. A range of neuroevolutionary algorithms are introduced to date targeting various attributes of ANN with different representation schemes. These methods include: Conventional Neural Evolution (CNE), Symbiotic Adaptive Neural Evolution (SANE), Enforced Sub-Population—ESP, Eugenic algorithm based reinforced learning algorithm-EuSANE, Neuro Evolution of Augmented Topologies (NEAT), multi-chromosome based evolved ANNs, IEC, Evolution of recurrent systems with Linear Outputs (ELVINO) for sequence learning [3,4,5,6,7,8,9].

2.2 Related Work: Traffic Prediction and Scheduling

The accuracy of the traffic estimator used in the scheduling system strongly affect the overall efficiency of the scheduling algorithm: more accurate is the estimator, more efficient is the system. In [10] an online traffic estimator known as Variable Step-Size Normalized Least Mean Square (VSSNLMS) algorithm is presented. However, it has been observed that the VSSNLMS algorithm causes large prediction errors and yield slow convergence when a scene change occurs—this is because it causes an abrupt fluctuation in the bit rate near the scene boundaries [11]. A traffic predictor is proposed in [12] that separates the MPEG video sequence into subgroups of I, B and P and estimate each type of frame using the Normalized Least Square (NLMS) algorithm.

In [13] another approach is presented along with a scene change detector metric and it estimates the size of the B-frame and the size of the GOP. It also allows discarding selective B-frames to minimize buffer requirements. Fair-SRPT, an MPEG-4 flow sessions scheduling system is presented in [14] which normalizes backlogged queues by their mean resource reservation and then services them in increasing order of their normalized queue size. Adaptive Neural Networks based estimators are given in [15] and [16]. In [17] a recurrent neural network based MPEG-4 video traffic estimator was proposed. The research undertook three experiments: Single Step Predictors (SSPs) implementation, a down-sampled of the time-series video sequence with

SSPs, and Multi Step Predictors implementation. [18] presented a Kalman filter based video frame size estimator. The proposed research was further improved in [19] that used seasonal ARIMA model and was tested on almost the same data set.

3 Cartesian Genetic Programming Evolved Artificial Neural Network (CGPANN)

CGPANN is a Neuro-Evolutionary technique that uses Cartesian Genetic Programming [20] to evolve an Artificial Neural Network. CGPANN, unlike a conventional ANN, evolves many aspects of a neural network and hence it is fast, robust, and generates more novel solutions.

CGPANN evolves many aspects of an ANN: network weights, functions, connections, and even the topology itself. The network continues to evolve until the best possible solution is obtained. In this paper, Recurrent CGPANN is used as the principle estimator in the scheduling system. In CGPANN, $1 + \lambda$ evolutionary strategy is used to generate the population for next generation. Another unique aspect of CGPANN is that unlike a conventional ANN, the neurons in the network aren't fully connected. Moreover, the systems inputs aren't connected with all of the neurons in the input layer. Such characteristics make CGPANN both fast and efficient in terms of hardware implementation.

3.1 Recurrent CGPANN

To solve problems that are non-linear and dynamic in nature, the use of recurrent networks becomes inevitable. The recurrent form of CGPANN (i.e. RCGPANN) is based on “Jordan Network”. RCGPANN is similar to FCGPANN in many aspects: inputs, weights, connections and functions are the same as in its feed-forward counterpart. The difference is that the input now can be a recurrent input, R as shown in figure 3, fed back from the output. Inputs in the first layer of the network in RCGPANN are recurrent. The following layers might have recurrent connections depending upon whether the feedback input is randomly selected as one of the node input or not. This particular trait of the RCGPANN makes it different from the Jordan Network. However, such an alteration makes the network more flexible.

Similar to FCGPANN, the connections are set to 1 by default. The weights range from -1 to +1 and are multiplied with their corresponding inputs, summed up, and are fed to an activation function. The recurrent input R is obtained by taking the outputs

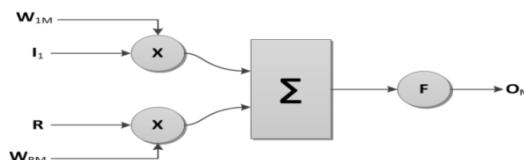


Fig. 1. R-CGPANN Node

of the system, multiplying them with their corresponding weights, summing them up, and finally feeding them to an activation function. The output R from that feedback node is made available for all the network nodes that take it in as one of their inputs, as shown in figure 2.

Figure 2(a) and 2(b) depict the RCGPANN genotype and the corresponding phenotype respectively. The genotype is quite similar to that of the FCGPANN. The only difference is that now instead of the inputs, the recurrent input R can be part of the nodes as shown in the rectangular boxes representing nodes. The phenotype of the network is shown in 2(b) which again—as in direct encoding schemes—is the reflection of the genotype in 2(a). For the recurrent input R, the corresponding node can also be seen in the feedback path of the network. Note that the network in 2(b) only has 1 recurrent node.

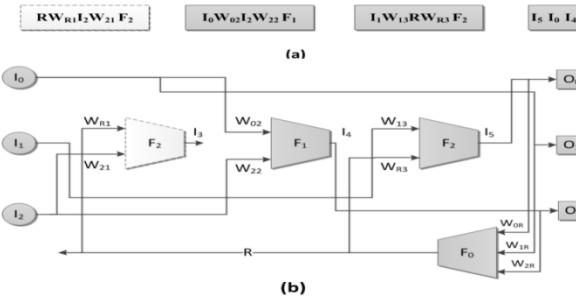


Fig. 2. (a) R-CGPANN genotype (b) phenotype of (a)

4 Experimental Setup

Recurrent CGPANN based estimating system was deployed to predict the next MPEG-4 video frame. Fig 3 depicts the overall skeleton of the system. The network has 10 inputs (10 frames lying in the user buffer) and ten outputs. The average of the ten outputs yields the final output of the estimator that is the estimated size of the next MPEG-4 frame.

Several parameters are varied in different simulation scenarios to obtain the results. For example, for a fixed number of nodes in the network, the number of recurrent

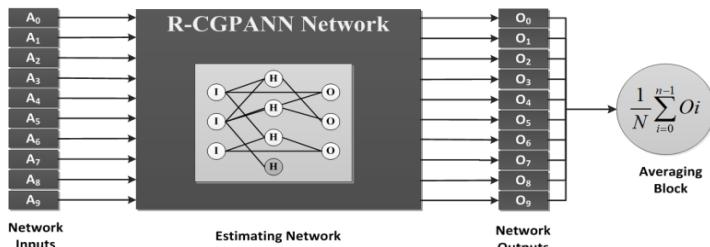


Fig. 3. R-CGPANN based estimator

nodes in the network is varied from 1 to 10 and results are obtained. This process is repeated for a set of 10 different elements representing the number of total nodes in the network. For all the scenarios, the mutation rate is set to 10%. Each neuron (node) has 5 inputs. The evolutionary strategy is set to $1 + \lambda$ with λ set to 9. It means that there is 1 parent and 9 offspring competing for the next generation based on their fitness level.

5 Simulation and Results

MPEG-4 data from eight different sources was selected in which the first source (film) was used as the training set and the remaining 7 served as the testing set for the experiment. Table 1 lists the MPEG-4 sources used for the testing purpose along with their respective number of frames. The table also lists the average to peak frame ratio for the corresponding sources.

Table 1. MPEG-4 Sources: Testing Data

| S.No. | MPEG-4 Source | Number of Frames | Average to Peak Ratio |
|-------|----------------------|------------------|-----------------------|
| 1 | First Contact | 50,712 | 0.13 |
| 2 | Silence of the Lambs | 50,287 | 0.09 |
| 3 | Star Wars IV | 37,536 | 0.12 |
| 4 | The Firm 1 | 65,529 | 0.06 |
| 5 | The Firm 2 | 65,529 | 0.23 |
| 6 | From Dusk till Dawn | 52,520 | 0.21 |
| 7 | Starship Troopers | 65,529 | 0.21 |

Root Mean Square Error (RMSE) is used as an error criterion in this research. Other criteria such as Mean Square Error (MSE) and Mean Absolute Percentage Error (MAPE) have also been calculated but are omitted here for the sake of brevity. Percentage RMSE is given by:

$$RMSE (\%) = \left[\sqrt{\frac{1}{n} \sum_{i=1}^n (P_i - A_i)^2} \right] \times 100$$

Where P_i is the i^{th} predicted frame size and A_i is the actual frame size.

Table 4, 5, and 6 (given at the end) summarize all the results obtained during the simulation process. For all the 7 MPEG-4 sources (films), the tables are obtained each representing three major scenarios 1, 5, and 10—the number of recurrent nodes in the network. And then each major scenario has a set of sub-scenarios (number of nodes in the network, varied from 50 to 500). The results are quite self-explanatory. For instance, it is obvious from the tables that the number of nodes in the network has little or no effect on the efficiency of the system. The error, for example, for MPEG-4 source 2 (Silence of the Lambs) is 2.7% for almost every sub-scenario with only one exception.

Moreover, changing the number of recurrent nodes doesn't significantly impact the performance of the estimator. In all the tables, the corresponding errors for a particular source are quite closer to each other. However, there is one exceptional scenario (10 recurrent nodes, 100 network nodes) where the error is worse and doesn't conform to the afore-mentioned pattern. This anomaly is perhaps due to the intrinsic uncertainty associated with the process of evolution and the network seems to be stuck somewhere at a local minima and hence the unexpected results are observed.

Nevertheless, a closer observation of the three tables suggests that a single recurrent node is the most suitable choice for estimating network. That is true for two reasons. Firstly, the sum of all the errors listed in the tables is lowest for the Table 4. Not only Table 4 has the lowest obtained error—however, that is shared with Table 5 and Table 6—the worst case error is also lower than the remaining two tables. Secondly, using a single recurrent node in RCGPANN yields more stable output for different number of nodes in the network. This can be verified from figure 4(a) which shows the average of all the MPEG-4 source errors for different number of nodes in the network. The line representing one recurrent node is straight and has fewer fluctuations. The remaining two lines are not stable; in fact the error fluctuation aggravates with increasing the number of recurrent nodes as shown in figure 4(a).

Figure 4(b) shows the percent RMSE for all the 7 MPEG-4 sources used for the testing process. The error is actually the average of all the sub-scenarios (changing from 50 to 500 nodes) calculated for each of the sources. Hence, figure 4(b) is essentially related to figure 4(a) but the averaging criterion has changed. All the sources have nearly the same percent RMSE curves. But a closer observation of the figure reveals that for each source, the R-CGPANN used with 1 recurrent node has a better performance.

Table 2. Individual Source Error Comparison

| S.No. | Source | Model | | | |
|-------|----------------------|--------------|------------|-------------------|--------------------|
| | | R-ANN[17] | SARIMA[19] | Kalman Filter[18] | Proposed (RCGPANN) |
| 1 | Silence of the Lambs | RMSE 6.8% | - | - | RMSE 2.7% |
| 2 | Star Wars IV | RMSE 5.1% | - | - | RMSE 4% |
| 3 | The Firm | - | MARE 1.4% | MARE 1.43% | MAPE 1.39% |
| 4 | From Dusk till Dawn | - | MARE 1.6% | MARE 1.8% | MAPE 3.2% |

Table 2 compares the best achieved results taken from various other papers. Note that some of the error criteria are different in other research papers such as Mean Absolute Relative Error (MARE) and Relative Percentage Error (RPE). For MARE, the nearest matching criterion is MAPE and hence for that reason, MARE is represented in its percentage form. As obvious from the table, the proposed estimation method surpasses the performance of the other techniques in terms of the frame prediction error, wherever comparable.

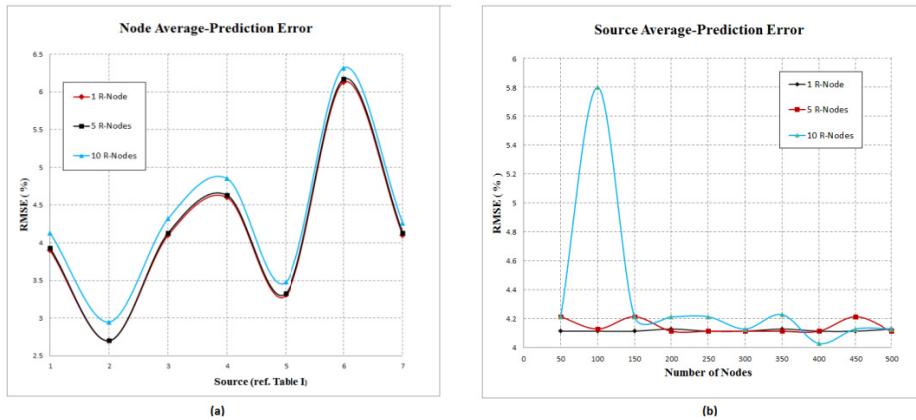


Fig. 4. (a) Average MPEG-4 source RMSE curves (b) Average total Nodes RMSE curves

Table 3. Best Overall Error Comparison

| S.No. | Scheme | Error |
|-------|----------------------------|----------------------------|
| 1 | Recurrent ANN [17] | RMSE=3.0% |
| 2 | F-CGPANN [1] | RMSE=16% |
| 3 | Laetitia et. al model [13] | RPE=7.30% |
| 4 | SARIMA [19] | MARE=1.37% |
| 5 | Kalman Filter [18] | MARE=1.4% |
| 6 | Proposed (RCGPANN) | RMSE=2.7% MAPE=1.2% |

Individual MPEG-4 sources (films) are compared in Table 3 with some of the previously developed frame estimators. For most of the cases, the proposed technique has lower RMSE and MAPE values with only one exception. This clearly indicates that the recurrent CGPANN based MPEG-4 estimator is not only robust but have measurable performance edge over other contemporary and previously proposed models for the said problem.

Table 4. One Recurrent Node: RMSE For The Testing Data

Table 5. Five Recurrent Nodes: RMSE For The Testing Data

| S.No. | Source | RMSE (%) | | | | | | | | | |
|-------|----------------------|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500 |
| 1 | First Contact | 4 | 3.9 | 4 | 3.9 | 3.9 | 3.9 | 3.9 | 3.9 | 4 | 3.9 |
| 2 | Silence of the Lambs | 2.7 | 2.7 | 2.7 | 2.7 | 2.7 | 2.7 | 2.7 | 2.7 | 2.7 | 2.7 |
| 3 | Star Wars IV | 4.2 | 4.1 | 4.2 | 4.1 | 4.1 | 4.1 | 4.1 | 4.1 | 4.2 | 4.1 |
| 4 | The Firm 1 | 4.7 | 4.6 | 4.7 | 4.6 | 4.6 | 4.6 | 4.6 | 4.6 | 4.7 | 4.6 |
| 5 | The Firm 2 | 3.4 | 3.3 | 3.4 | 3.3 | 3.3 | 3.3 | 3.3 | 3.3 | 3.4 | 3.3 |
| 6 | From Dusk till Dawn | 6.3 | 6.2 | 6.3 | 6.1 | 6.1 | 6.1 | 6.1 | 6.1 | 6.3 | 6.1 |
| 7 | Starship Troopers | 4.2 | 4.1 | 4.2 | 4.1 | 4.1 | 4.1 | 4.1 | 4.1 | 4.2 | 4.1 |

Table 6. Ten Recurrent Nodes: RMSE For The Testing Data

| S.No. | Source | RMSE (%) | | | | | | | | | |
|-------|----------------------|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500 |
| 1 | First Contact | 4 | 5.7 | 4 | 4 | 4 | 3.9 | 4 | 3.9 | 3.9 | 3.9 |
| 2 | Silence of the Lambs | 2.7 | 5.2 | 2.7 | 2.7 | 2.7 | 2.7 | 2.7 | 2.7 | 2.7 | 2.7 |
| 3 | Star Wars IV | 4.2 | 5.9 | 4.2 | 4.2 | 4.2 | 4.1 | 4.2 | 4 | 4.1 | 4.1 |
| 4 | The Firm 1 | 4.7 | 6.7 | 4.7 | 4.7 | 4.7 | 4.6 | 4.8 | 4.4 | 4.6 | 4.6 |
| 5 | The Firm 2 | 3.4 | 4.7 | 3.4 | 3.4 | 3.4 | 3.3 | 3.4 | 3.2 | 3.3 | 3.3 |
| 6 | From Dusk till Dawn | 6.3 | 7.1 | 6.3 | 6.3 | 6.3 | 6.2 | 6.3 | 6 | 6.2 | 6.2 |
| 7 | Starship Troopers | 4.2 | 5.3 | 4.2 | 4.2 | 4.2 | 4.1 | 4.2 | 4 | 4.1 | 4.1 |

6 Conclusion

In this paper, a robust and highly efficient MPEG-4 video frame estimator is presented that uses a Neuro-Evolutionary algorithm called R-CGPANN. R-CGPANN uses recurrent nodes in its network and has a measurable performance edge over its feed-forward counterpart. The estimating system is deployed that takes 10 previously received frames as its input and based on that information, the next frame size is predicated. A variety of experiments are conducted to analyze the efficiency of the system. The frame prediction error was calculated in different scenarios. Each scenario was different because of the varying number of recurrent node in the network. Then for each scenario, a sub scenario was simulated by changing the total nodes in the network and thence the error was calculated. The simulation results showed that the overall performance of the estimator aggravated with increasing the number of recurrent nodes in the network. When compared with the previously used techniques, the proposed estimator outperformed almost every contending algorithm with measurable estimation efficiency.

Acknowledgments. The authors acknowledge the support of the **National ICT R & D Fund Islamabad, PAKISTAN** for funding the project "**DESIGN AND IMPLEMENTATION OF A PROTOTYPE FOR A SECURE BILLING FRAMEWORK WITH REAL TIME DETECTION OF MALICIOUS END NODE CONNECTIONS USING WIRELESS SENSOR NETWORKS TO CURB ELECTRICITY THEFT**". The work presented here is part of the project intended to devise an efficient scheduling algorithm for the Smart Grid Traffic.

References

1. Ullah, F., Khan, G.M., Mahmud, S.A.: Intelligent Bandwidth Management Using Fast Learning Neural Networks. In: 9th International Conference on High Performance Computing and Communication (HPCC-ICESS), pp. 867–872 (June 2012)
2. Yao, X.: Evolving Artificial Neural Networks. Proc. IEEE 87, 1423–1447 (1999)
3. Gomez, F., Schmidhuber, J., Miikkulainen, R.: Accelerated Neural Evolution Through Cooperatively Coevolved Synapses. J. Mach. Learn. Res. 9, 937–965 (2008)
4. Moriarty, D.: Symbiotic Evolution of Neural Networks in Sequential Decision Tasks. PhD thesis, University of Texas at Austin, Tech Rep. UT-A197-257 (1997)
5. Polani, D., Miikkulainen, R.: Eugenic Neuro-Evolution for Reinforcement Learning. In: GECCO, pp. 1041–1046 (2000)
6. Stanley, K.O., Miikkulainen, R.: Efficient reinforcement learning through evolving neural networks topologies. In: GECCO 2002 (2002)
7. Mayer, A., Mayer, H.A.: Multi-Chromosomal Representation in NeuroEvolution. In: Computations Intelligence Conference (2006)
8. Takagi, H.: Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation. Proc. of the IEEE 89(9), 1275–1296 (2001)
9. Schmidhuber, J., Wierstra, D., Gomez, F.: Evolino: Hybrid Neuroevolution Optimal Linear Search for Sequence Prediction. In: 19th Int. Conf. on Artificial Intelligence (2005)
10. Tseng, Y.H., Wu, E.H.-K., Chen, G.H.: Scene-Change Aware Dynamic Bandwidth Allocation for Real-Time VBR Video Transmission Over IEEE 802.15.3 Wireless Home Networks. IEEE Transactions on Multimedia 9(3), 642–654 (2007)
11. Kuo, W.K., Lien, S.Y.: Dynamic resource allocation for supporting real-time multimedia applications in IEEE 802.15.3 WPANs. IET Com. 3(1), 1–9 (2009)
12. Adas, A.M.: Using adaptive linear prediction to support real-time VBR video under RCBR network service model. IEEE Trans. on Networking 6(5), 635–644 (1998)
13. Lanfranchi, L.I., Bing, B.K.: MPEG-4 Bandwidth Prediction for Broadband Cable Networks. IEEE Transactions on Broadcasting 54(4), 741–751 (2008)
14. Mangharam, R., Demirhan, M., Rajkumar, R., Raychaudhuri, D.: Size matters: Size-Based Scheduling for MEPG-4 over wireless channels. In: SPIE & ACM Proceedings in Multimedia Computing and Networking, vol. 3020, pp. 110–122 (January 2004)
15. Doulamis, A.D., Doulamis, N.D., Kollias, S.D.: An adaptable neural-network model for recursive nonlinear traffic prediction and modeling of MPEG video sources. IEEE Transactions on Neural Networks 14(1), 150–166 (2003)
16. Park, D.-C., Tran, C.N., Song, Y.-S., Lee, Y.: Prediction of MPEG video source traffic using biLinear recurrent neural networks. In: Yang, Q., Webb, G. (eds.) PRICAI 2006. LNCS (LNAI), vol. 4099, pp. 298–307. Springer, Heidelberg (2006)
17. Aninda, B., Parlos, A.G., Amir, A.F.: Prediction of MPEG-coded video source traffic using recurrent neural networks. IEEE Trans. Signal Processing 51(8), 2177–2190 (2003)
18. Jibukumar, M.G., Datta, R., Kumar, P.: Kalman filter based Variable Bit Rate video frame size prediction. In: 3rd Int. Symp. on Wireless Pervasive Computing, pp. 459–463 (May 2008)
19. Trlin, G.: VBR video frame size prediction using seasonal ARIMA. In: 20th Int. Conference on Software, Telecommunication and Computer Networks (SoftCOM), pp. 1–5 (September 2012)
20. Miller, J.F., Thomson, P.: Cartesian Genetic Programming. In: EuroGP, pp. 121–132 (2000)

SCH-EGA: An Efficient Hybrid Algorithm for the Frequency Assignment Problem*

Shaohui Wu, Gang Yang**, Jieping Xu, and Xirong Li

Multimedia Computing Lab, School of Information,

Renmin University of China, China

{shaohuiwu,yanggang,xjieping,xirong}@ruc.edu.cn

Abstract. This paper proposes a hybrid stochastic competitive Hopfield neural network-efficient genetic algorithm (SCH-EGA) approach to tackle the frequency assignment problem (FAP). The objective of FAP is to minimize the cochannel interference between satellite communication systems by rearranging the frequency assignments so that they can accommodate the increasing demands. In fact, as SCH-EGA algorithm owns the good adaptability, it can not only deal with the frequency assignment problem, but also cope with the problems of clustering, classification, the maximum clique problem and so on. In this paper, we first propose five optimal strategies to build an efficient genetic algorithm(EGA) which is the component of our hybrid algorithm. Then we explore different hybridizations between the Hopfield neural network and EGA. With the help of hybridization, SCH-EGA makes up for the defects in the Hopfield neural network and EGA while fully using the advantages of the two algorithms.

Keywords: Frequency assignment problem, genetic algorithm, neural network, hybrid algorithm.

1 Introduction

The frequency assignment problem (FAP) is a classical problem due to its various applications including satellite communication systems, mobile telephone and TV broadcasting. In satellite communication systems, the reduction of the cochannel interference has become a major factor for determining system design [1],[2]. Furthermore, due to the necessity of accommodating as many satellites as possible in geostationary orbit, this interference reduction has become an even more important issue with the increasing number of geostationary satellites [3]. To deal with interference reduction in practical situations, the rearrangement of frequency assignments is considered as an effective measure [4],[5].

* This research was partially supported by the grants from the Natural Science Foundation of China (No.61003205); the Qianjiang Talent Project of Zhejiang Province (No.2011R10087).

** Corresponding author.

FAP is a NP-complete combinatorial optimization problem and many approaches have been proposed [6]. The application of neural networks in frequency assignment problems was first proposed by Sengoku [7] and Kunz [8]. Then Funabiki and Nishikawa(1997) proposed a gradual neural network(GNN) that consists of $N \times M$ neurons. The disadvantage of GNN is its heavy computation burden, especially in large size problems. Salcedo-Sanz et al. (2004) combined a binary Hopfield neural network with simulated annealing(HopSA) for the FAP. The algorithm also cannot deal with large problems because of the excessive computation time [9]. Recently Wang et al. (2011) proposed a stochastic competitive Hopfield neural network(SCHNN). They introduced the stochastic dynamics to help the network escape from local minima, but utilizing the dynamics only is not efficient. There are also many genetic algorithms(GAs) proposed for FAP. Cuppini, Kim and Lai were among the first papers found in literature to have applied GA to solve the channel allocation problem [10]. Then a lot of new genetic algorithms were proposed for FAP. Ngo et al. and Beckmann et al. proposed a new strategy known as the Combined Genetic Algorith(CGA). CGA starts by estimating the lower bound z on bandwidth and its computation time is highly dependent on the z [11],[12].

The rest of the paper is organized as follows: in the next section we define and analyze the FAP. In section 3, our hybrid algorithm is described. Experiments and results are shown in Section 4. Finally, Section 5 ends the paper with concluding remarks.

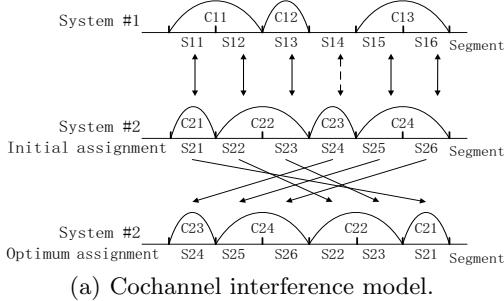
2 Problem Definition

In this section, we describe the FAP in satellite communications systems as a combinatorial optimization problem with three constraints and two objectives [13]. Given two adjacent satellite systems, FAP consists of reducing the inter-system cochannel interference by rearranging the frequency assignment on carriers in system #2(M segments, N carriers), while the assignment in system #1(M segments) remains fixed(Fig.1(a)). Because each carrier usually occupies a different length in a frequency band, Mizuike et al. introduced the segmentation of carriers so that every carrier can be described by a collection of consecutive unit segments. The interference between two M -segment systems is described by a $M \times M$ interference matrix IM (Fig.1(b)), in which the ij th element e_{ij} stands for the cochannel interference when segment i in system #2 uses a common frequency with segment j in system #1.

The three constraints of FAP are:

- (1) Every segment in system #2 must be assigned to a segment in system #1.
- (2) Each segment in system #1 can be assigned by at most one segment in system #2.
- (3) All segments of each carrier in system #2 must be assigned to consecutive segments in system #1 in the same order.

The two objectives are shown as follows: (1)Minimize the largest element of the interference matrix selected in the assignment. (2)Minimize the total interference



(a) Cochannel interference model.

| | | C11 | | C12 | | C13 | |
|-----|-----|----------|-----|-----|-----|-----|----------|
| | | S11 | S12 | S13 | S14 | S15 | S16 |
| C21 | S21 | 20 | 20 | 40 | 0 | 25 | 25 |
| | S22 | 50 | 10 | 30 | 0 | 55 | ∞ |
| C22 | S23 | ∞ | 50 | 30 | 0 | 15 | 55 |
| | S24 | 30 | 30 | 45 | 0 | 35 | 35 |
| C23 | S25 | 45 | 5 | 25 | 0 | 50 | ∞ |
| | S26 | ∞ | 45 | 25 | 0 | 10 | 50 |

(b) Interference matrix IM .**Fig. 1.** Graphs of the cochannel interference model and interference matrix

of all the selected elements. Note that the first objective has a higher priority over the second objective.

3 Our Hybrid SCH-EGA Approach

In this section, we briefly introduce the stochastic competitive Hopfield neural network(SCHNN) proposed by Wang et al. first. Then we propose five optimal strategies to build an efficient genetic algorithm(EGA). Finally, we describe our hybrid algorithm SCH-EGA in detail.

3.1 Stochastic Competitive Hopfield Neural Network

Neural network approaches are a common method for solving the FAP. For FAP with N -carrier- M -segment, Funabiki and Nishikawa(1997) proposed a neural network model which consists of $N \times M$ neurons [13]. In this model, the “one output”($V_{ij}=1$) means that carrier i in system #2 is assigned to segments j -($j+c_i-1$) in system #1. Note that c_i represents the length of carrier i . As this model itself has satisfied the third constraint of FAP, only the first two constraints need to be considered.

Recently Wang et al. proposed a stochastic competitive Hopfield neural network (SCHNN) based on the $N \times M$ model for solving the FAP [9]. In SCHNN, a total energy function proposed for the first two constraints is shown as follows.

$$E = \frac{A}{2} \sum_{i=1}^N \left(\sum_{q=1}^M V_{iq} - 1 \right)^2 + \frac{B}{2} \sum_{i=1}^N \sum_{j=1}^M \sum_{p=1}^N \sum_{\substack{q=j-c_p+1 \\ p \neq i}}^{j+c_i-1} V_{ij} V_{pq} \quad (1)$$

where A and B are coefficients. We believe that it has met all the three constraints when E becomes zero [9].

In SCHNN, updating strategy of v_{ij} is shown in Eq.2 and Eq.3. In the i th row, the $v_{ij} = 1$ if the u'_{ij} is the maximum in the row according to Eq.3. Other $v_{ip}(p \neq j)$ equals to 0. One and only one neuron within each group(row) is fired at every time t.

$$u'_{ij}(t) = \alpha(s) \cdot u_{ij}(t) \quad (2)$$

$$v_{ij}(t+1) = \begin{cases} 1, u'_{ij} = \max_{k=1,\dots,M} \{u'_{ik}(t)\} \\ 0, otherwise \end{cases} \quad (3)$$

where s is the updating step number, $\alpha(s)$ is a random multiplier, and $u'_{ij}(t)$ is the transient variable. The multiplier $\alpha(s)$ in Eq.2 is given by

$$\alpha(s) = \text{random}(h(s), 1) \quad (4)$$

where $h(s) = 1 - T \cdot e^{-s/\lambda}$.

SCHNN satisfies the rule that large carriers with many segments should be assigned as early as possible, or else, it would be difficult to assign them after many carriers have been already assigned [13]. Thus, the updating rule of the input matrix u_{ij} is as shown in Eq.5:

$$u_{ij}(t) = -W_2 \sum_{p=1, p \neq i}^N \sum_{q=\max(j-c_p+1, M)}^{\min(j+c_i-1, M)} c_p v_{pq}(t) - W_3 d' \quad (5)$$

where W_2 and W_3 are weighting factors.

SCHNN obtains good performance for FAP, but it still has many disadvantages. Its ability for escaping from local minima only using the stochastic dynamics is not very strong. At the same time, the two objectives of FAP are not fully synchronized which means that a result with a smaller value of the largest interference element may have a larger total interference. There is no good solution in SCHNN for this phenomenon.

3.2 Efficient Genetic Algorithm for FAP

In our research, five optimal strategies are used to build an efficient genetic algorithm(EGA), which can guarantee both good solutions and the small computational cost. The five strategies for EGA are shown as follows:

Individual Strategy(IS): The first strategy is proposed to define the structures of the individuals in population. Every individual should be an $N \times M$ binary matrix. One and only one element for each row must have one output. Then the individuals have satisfied the first and third constraints of FAP. This strategy IS also guarantees the identical structure with SCHNN.

Calibration Strategy(CS): The calibration strategy is proposed to modify individuals to satisfy the second constraint. For each individual, the position of the "one output" will be checked by columns. If the segment in system #1 is assigned by more than one segment in system #2, the excess "one output" will change to the next position in the row. After this step, all individuals in population will satisfy the three constraints of FAP.

Fitness Function Strategy(FFS): As there are two objectives in FAP, two fitness functions should be built, respectively. The first fitness function with higher priority is about the largest interference element selected in the assignment. And the second fitness function is about the total interference.

Elimination Strategy(ES): Strategy ES is proposed to eliminate the repetitions and those individuals with low fitness values. The detailed elimination step is shown as follows: First, the two fitness values for each individual will be calculated. Then all individuals will be sorted by the two fitness functions, respectively. The parameters L_{num} and T_{num} are set as the numbers of individuals chosen for next generation by the first and second fitness functions. And the sum of L_{num} and T_{num} is the population size I_{num} . When the sorting step ends, the repetitions in population will be eliminated first. Then L_{num} individuals with the least largest interferences will be chosen as the next generation. After that, T_{num} individuals with the least total interferences will be chosen. If the individual has been chosen by the first fitness function, it will be ignored and the next individual will be checked. At last, the individuals kept for next generation are those with the smallest largest interferences or total interferences.

Scale Strategy(SS): The population size I_{num} should not be very large. A proper population size can guarantee good solutions and the small computational cost.

Thus, the basic steps of EGA are shown as follows: First, a small population is randomly generated and all individuals are $N \times M$ binary matrixes. In the crossover operation, each two individuals will be set as a pair to exchange one row which is randomly chosen. This operation guarantees both diversity of the new population and a small computational cost. In the mutation process, one individual will be randomly generated as the mutation results. When the crossover and mutation operations are over, the calibration step starts. All individuals will satisfy the first two constraints after calibration. Then the individuals will be sorted by the two fitness functions, respectively. Last step is the elimination. Only the individuals with the smallest largest or total interferences will be kept. After the steps above, a small scale population with high fitness values exists for next generation. This process repeats again until it finds the final solution.

3.3 Our Hybrid Algorithm

In this section, we explore different hybridizations between SCHNN and EGA. And then we propose our hybrid stochastic competitive Hopfield neural network-efficient genetic algorithm(SCH-EGA).

In order to guarantee the good generality which can be used for hybridizations between Hopfield neural networks and other evolutionary algorithms, our hybrid algorithm is just in the linear structure. Then three hybridizations between SCHNN and EGA are proposed. The first hybridization called N-E for short is designed to improve the ability of searching better solutions for EGA by introducing high quality individuals from SCHNN. The outputs of SCHNN will be added into population and participate in the crossover, mutation and elimination operations with other individuals in EGA. In this hybridization, the outputs of SCHNN are regarded as additional "mutation" operators that inject new good solutions into the evolutionary process. As all outputs of SCHNN will be added to EGA, we just need to consider whether EGA has found the best solution or it runs to the maximum iterations. The second hybridization is that

Algorithm 1. SCH-EGA Algorithm

```

1: Initialize input  $u$ (randomly around zero);
2: Generate a population randomly;
3: Calculate output  $v$  using Eq.3;
4: Set  $T=2$ ,  $t = 0$ ,  $W_2$  and  $W_3$ ;
5: while  $t <$  the max number of iterations or stability criteria are not satisfied do
6:   for  $i = 1$  to  $N$  do
7:     for  $j = 1$  to  $M$  do
8:        $s = \lfloor t/N \rfloor$ ;
9:       Compute the  $u_{ij}$  with Eq.5;
10:      end for
11:    end for
12:    for  $j = 1$  to  $M$  do
13:      Update  $v_{ij}$  using Eq.3;
14:    end for
15:     $t = t + 1$ ;
16:    Add the output of SCHNN to EGA as an individual;
17:    Crossover;
18:    Mutation;
19:    Calibration;
20:    Elimination;
21:    if the output of SCHNN is worse than best solution in EGA then
22:      Set the best solution in EGA as the output of SCHNN;
23:    end if
24: end while

```

the output of SCHNN will be compared with the best solution in EGA in each iteration. If the solution is better than the output, the output will be replaced by the best solution from EGA. If the output of SCHNN does not change in many iterations, or it runs to the maximum iteration, the algorithm terminates. This hybridization is aimed to help SCHNN escape from local minima efficiently by high quality solutions from the genetic algorithm. The last hybridization called N-E-N combines the first two hybrid algorithms. The outputs of SCHNN need to be added to EGA and the best solutions in EGA may replace the outputs of SCHNN.

In our research, the third hybridization obtains the best performance. Then we take the third hybridization as our SCH-EGA algorithm and show it in Algorithm 1. With the help of hybridization, SCH-EGA helps both SCHNN and EGA to escape from the local minima and obtain better solutions.

4 Simulations

In order to assess the performance of SCH-EGA, simulations were implemented in Matlab on a PC (Core(TM) i5-3450 3.10GHz, 8.0G RAM). In this section, we carry out three sets of experiments for different purposes. First, three hybridizations between SCHNN and EGA are compared and the best one is discovered.

Second, in order to show performance of our hybrid algorithm, we compare SCH-EGA with other algorithms on 5 benchmark problems and 12 large problems randomly generated. The five benchmark instances BM1-BM5 are from Funabiki and Nishikawa (1997), where these were called instances 1-5, respectively [13]. For all the problems in our experiments, SCH-EGA obtains better or comparable results. Finally, we show that our SCH-EGA can obtain good performance with a small population which reduces the computational cost a lot.

4.1 Selection of the Best Hybridization

The first experiment is set to compare the three hybridizations between SCHNN and EGA and select the best one for our final SCH-EGA algorithm. In this experiment, the three hybridizations are tested on five benchmark problems and three large problems randomly generated. The results are shown in Table 1.

Table 1. Results of the three hybridizations for five benchmark problems and three large problems randomly generated

| Group | Instance | N-E Algorithm | | | E-N Algorithm | | | N-E-N Algorithm | | |
|---------|----------|---------------|-------|---------|---------------|-------|---------|-----------------|-------------|---------|
| | | Largest | Total | Time(s) | Largest | Total | Time(s) | Largest | Total | Time(s) |
| Group 1 | BM1 | 30 | 100 | 0.116 | 30 | 100 | 0.113 | 30 | 100 | 0.113 |
| | BM2 | 4 | 13 | 0.126 | 4 | 13 | 0.116 | 4 | 13 | 0.116 |
| | BM3 | 7 | 85 | 0.568 | 7 | 88 | 0.539 | 7 | 85 | 0.521 |
| | BM4 | 64 | 866 | 0.524 | 64 | 874 | 0.514 | 64 | 855 | 0.523 |
| | BM5 | 640 | 6925 | 0.514 | 640 | 7006 | 0.532 | 640 | 6721 | 0.524 |
| Group 4 | Case 15 | 87 | 20065 | 292.72 | 91 | 20348 | 293.65 | 83 | 19532 | 291.84 |
| | Case 16 | 85 | 11237 | 570.15 | 89 | 11432 | 573.16 | 81 | 10296 | 569.98 |
| | Case 17 | 79 | 3127 | 565.92 | 75 | 3092 | 567.81 | 71 | 3059 | 563.43 |

From Table 1, we can find that all the three hybridizations obtain the same largest interferences on the five benchmark problems. But some differences exist for the total interferences. On BM3, the total interference of the second hybridization is a little larger than the other two hybridizations. Simplicity of the operations in EGA limits this hybridization to find better solutions and we can improve the ability by introducing high quality individuals from SCHNN. On BM4 and BM5, the third hybridization obtains better results than the first hybridization because the solutions in EGA can also help SCHNN to escape from local minima efficiently.

As the five benchmark problems in Group 1 are not complex enough, the three hybridizations are also tested on three difficult problems in Group 4 (Case 15-17). For large problems, the third hybridization also obtains the best performance. As the advantages of the third hybridization in Group 4 are more obvious than the advantages in Group 1, we believe that our hybrid algorithm will show better performance on large problems. Then we take the third hybridization as our final SCH-EGA algorithm. This experiment shows that our SCH-EGA algorithm makes up for the defects in SCHNN and EGA while fully using the advantages of the two algorithms.

Table 2. Comparison of different algorithms for benchmark problems BM1-BM5

| Group | Instance | HopSA | | GNN | | SCHNN | | EGA | | SCH-EGA | |
|---------|----------|---------|-------|---------|-------|---------|-------|---------|-------|---------|-------|
| | | Largest | Total | Largest | Total | Largest | Total | Largest | Total | Largest | Total |
| Group 1 | BM1 | 30 | 100 | 30 | 100 | 30 | 100 | 30 | 100 | 30 | 100 |
| | BM2 | 4 | 13 | 4 | 13 | 4 | 13 | 4 | 13 | 4 | 13 |
| | BM3 | 7 | 85 | 7 | 85 | 7 | 85 | 7 | 85 | 7 | 85 |
| | BM4 | 64 | 880 | 64 | 880 | 64 | 880 | 67 | 866 | 64 | 851 |
| | BM5 | 817 | 6910 | 640 | 8693 | 640 | 7243 | 640 | 7335 | 640 | 6721 |

4.2 Comparison with Other Algorithms

In order to show the good performance, we test our SCH-EGA algorithm on five benchmark problems(BM1-BM5), which have been solved by other algorithms.

Table 2 shows the results obtained by GNN, HopSA, SCHNN, EGA and SCH-EGA. Each result includes the largest interference and total interference. Since BM1-BM5 are benchmark problems, the results of GNN on BM1-BM5 are directly from Funabiki and Nishikawa(1997) [13], the results of the HopSA are directly from Salcedo-Sanz et al. (2004) [2] and the results of SCHNN are directly from Wang et al. (2010) [9]. Note that only the best values are shown in Table 2.

On BM1-BM3, as the problems are very simple, all algorithms obtain the same results. On BM4, the result of SCH-EGA is better than other algorithms for the total interference and obtains the same largest interference with GNN, HopSA and SCHNN. The largest interference of EGA is a little larger, but the total interference is smaller than HopSA, GNN and SCHNN. On BM5, the total interference of SCH-EGA is much smaller than other algorithms. The total interference of HopSA is smaller than other algorithms except SCH-EGA. But its largest interference is much larger than others. As the objective of decreasing the largest interference has a higher priority, the performance of HopSA is not good. On the five benchmark problems, SCH-EGA obtains better performance than all other compared algorithms. Note that EGA also gets good results.

We also compare SCH-EGA with SCHNN and EGA on 12 large problems(Case 6-17). The detailed results can be found in Table 3. The results of EGA are a little worse than SCHNN, but the computational cost is much less. From Table 3, the advantages of SCH-EGA are more obvious on large problems.

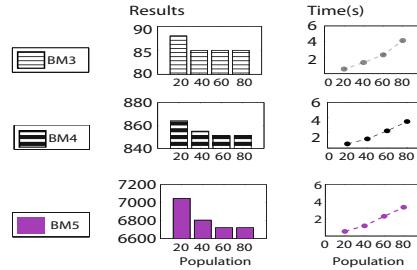
4.3 Evaluation on SCH-EGA with Different Population Sizes

At last, an experiment is conducted to evaluate the effects of different population sizes for SCH-EGA. 20, 40, 60 and 80 are chosen as different population sizes. Fig.2 shows the results on BM3 - BM5:

In Fig.2, three graphs in the left side show the total interferences of SCH-EGA with different population sizes. And the three graphs in the right side depict the running time corresponding to the graphs in the left side. For BM3, the result of SCH-EGA with 20 individuals in population has a larger interference than

Table 3. Comparison of the results obtained by SCHNN, EGA and SCH-EGA for 12 large problems randomly generated

| Group | Instance | SCHNN | | | EGA | | | SCH-EGA | | |
|---------|----------|---------|-------|----------|---------|-------|---------|-----------|--------------|----------|
| | | Largest | Total | time(s) | Largest | Total | time(s) | Largest | Total | time(s) |
| Group 2 | Case6 | 9 | 882 | 18. 601 | 9 | 885 | 5. 186 | 8 | 769 | 24. 830 |
| | Case7 | 92 | 7481 | 18. 323 | 94 | 8263 | 6. 914 | 85 | 6101 | 24. 791 |
| | Case8 | 939 | 78910 | 18. 453 | 960 | 82372 | 6. 703 | 895 | 76317 | 24. 721 |
| Group 3 | Case9 | 92 | 5643 | 40. 147 | 97 | 6013 | 11. 674 | 87 | 4982 | 51. 120 |
| | Case10 | 96 | 11076 | 68. 145 | 98 | 13510 | 22. 348 | 93 | 10852 | 87. 451 |
| | Case11 | 97 | 16839 | 98. 506 | 98 | 18333 | 33. 553 | 95 | 15937 | 139. 333 |
| | Case12 | 98 | 18542 | 118. 348 | 99 | 20056 | 34. 416 | 97 | 17562 | 155. 672 |
| | Case13 | 99 | 22685 | 134. 534 | 99 | 23141 | 39. 637 | 98 | 22076 | 176. 238 |
| | Case14 | 99 | 26398 | 179. 280 | 99 | 28442 | 65. 841 | 98 | 19758 | 251. 293 |
| Group 4 | Case15 | 92 | 20812 | 191. 909 | 95 | 23945 | 37. 911 | 83 | 19532 | 233. 146 |
| | Case16 | 91 | 11551 | 282. 566 | 93 | 18772 | 53. 564 | 81 | 10296 | 335. 180 |
| | Case17 | 84 | 3381 | 331. 220 | 89 | 4037 | 25. 915 | 71 | 3059 | 362. 413 |

**Fig. 2.** Evaluation on our algorithm with different population sizes

others, and the result improves only a little with the increase of population. But in the right side, the running time increases nearly in the linear growth. The running time with 80 individuals is about 5 times larger than it with 20 individuals and 3 times larger than it with 40 individuals. This phenomenon can also be found on BM4 and BM5. Too large population size helps SCH-EGA to search better solutions only a little, but it increases much computational cost. Thus, conclusion can be made that our SCH-EGA algorithm can obtain good performance with a small population.

5 Conclusions

In this paper, a hybrid stochastic competitive Hopfield neural network-efficient genetic algorithm(SCH-EGA) for the frequency assignment problem in the satellite communication systems has been presented. We first propose five optimal strategies to build an efficient genetic algorithm(EGA) which is the component of our hybrid algorithm. Then we explore three hybridizations between SCHNN

and EGA and find the best one. We believe this comparison can also be helpful for hybridizations between Hopfield neural networks and other evolutionary algorithms, such as the artificial bee colony algorithm, the ant colony optimization algorithm and so on. Then SCH-EGA is compared with other algorithms on 5 benchmark problems and 12 large problems randomly generated. On all problems SCH-EGA obtains better or comparable performance. Finally, an experiment is conducted to show that SCH-EGA can obtain good solutions with a small population which means the small computational cost.

References

1. Wang, L., Liu, W., Shi, H.: Noisy Chaotic Neural Network With Variable Thresholds for the Frequency Assignment Problem in Satellite Communications. *IEEE Transactions on Systems, Man, and Cybernetics* 38(2), 209–217 (2008)
2. Salcedo-Sanz, S., Santiago-Mozos, R., Bousoño Calzón, C.: A hybrid hopfield network-simulated annealing approach for frequency assignment in satellite communications systems. *IEEE Transactions on Systems, Man, and Cybernetics* 34(2), 1108–1116 (2004)
3. Liu, W., Shi, H., Wang, L.: Minimizing Interference in Satellite Communications Using Chaotic Neural Networks. In: ICNC 2007 (2007)
4. Mizukjz, T.: Optimization of Frequency Assignment 37(10), 1031–1041 (1989)
5. No-calz, C.B.: A Hybrid Neural-Genetic Algorithm for the Frequency Assignment Problem in Satellite Communications. *Applied Intelligence* 22(3), 207–217 (2005)
6. Cheeneebash, J., Lozano, J.A., Rughooputh, H.C.S.: A Survey on the Algorithms Used to Solve the Channel Assignment Problem. *Recent Patents on Telecommunications* 1(1), 54–71 (2012)
7. Sengoku, M., Nakano, K., Yamaguchi, Y., Abe, T.: Channel Assignment in a Cellular Mobile Communication System and an Application of Neural Networks. *Electronics and Communications in Japan* 75(4) (1992)
8. Kunz, D.: Channel assignment for cellular radio using neural networks. *IEEE Transactions on Vehicular Technology* 40(1), 188–193 (1991)
9. Wang, J., Cai, Y., Yin, J.: Multi-start stochastic competitive Hopfield neural network for frequency assignment problem in satellite communications. *Expert Systems with Applications* 38(1), 131–145 (2011)
10. Cuppini, M.: A Genetic Algorithm for Channel Assignment Problems. *European Transactions on Telecommunications* 5(2), 285–294 (1994)
11. Beckmann, D., Killat, U.: A new strategy for the application of genetic algorithms to the channel-assignment problem. *IEEE Transactions on Vehicular Technology* 48, 1261–1269 (1999)
12. Ngo, C.Y., Li, V.O.K.: Fixed channel assignment in cellular radio networks using a modified genetic algorithm. *IEEE Transactions on Vehicular Technology* 47(1), 163–172 (1998)
13. Funabiki, N., Nishikawa, S.: A gradual neural-network approach for frequency assignment in satellite communication systems. *IEEE Transactions on Neural Networks* 8(6), 1359–1370 (1997)

Improving the RACAI Neural Network MSD Tagger

Tiberiu Boroş and Stefan Daniel Dumitrescu

Research Institute for Artificial Intelligence “Mihai Drăgănescu”,
Romanian Academy (RACAI)
`{tibi, sdumitrescu}@racai.ro`

Abstract. Part-of-speech (POS) tagging is a key process for various natural language processing related tasks, in which each word of a sentence is assigned a uniquely interpretable label (called a POS tag). There are many proposed methodologies for this task, such as Hidden Markov Models, Conditional Random Fields, Maximum Entropy classifiers etc. Such methods are primarily intended for English which, in comparison to highly inflectional languages has a relatively small tagset inventory. One of the well-known methods used for large tagset labeling (referred to as morpho-syntactic descriptors or MSDs) is called Tiered Tagging (Tufiş, 1999), (Tufiş and Dragomirescu, 2006) and it exploits a reduced set of tags from which context irrelevant features (e.g. gender information) which can be deduced through the word form's flectional analysis are stripped. In our previous work we presented an alternative method to Tiered Tagging, in which we performed multi-class classification with a feed-forward neural network. Our methodology has the advantage that it does not require extensive linguistic knowledge as implied by the previously mentioned approach. We extend our work by testing our tool on Czech and successfully experimenting with a genetic algorithm designed to find a better network topology.

Keywords: Large tagset labeling, POS tagging, MSD, Neural Networks, Genetic Algorithms.

1 Introduction

Part-of-speech (POS) tagging is a key process for various natural language processing related tasks, in which each word of a sentence is given a uniquely interpretable label. The labels are called POS tags and the entire inventory of POS tags is called a tagset.

There are several approaches to part-of-speech tagging, such as Hidden Markov Models (HMM) (Brants, 2000), Maximum Entropy Classifiers (Berger et al., 1996; Ratnaparkhi, 1996), Bayesian Networks (Samuelsson, 1993), Neural Networks (Marques and Lopes, 1996) and Conditional Random Fields (CRF) (Lafferty et al., 2001). All these methods are primarily intended for English, which uses a relatively small tagset inventory, compared to highly inflectional languages (such as Romanian, Czech, Slovenian, etc.). For the later mentioned languages the above listed POS tagging methods do not perform that well, mainly because of data sparseness and lack of statistical evidence.

In Boroş et al. (2013) we experimented with large tagset labeling using neural networks. Our solution was an alternative to one of the most successful methods used for POS tagging on the Romanian Language, called Tiered Tagging (Tufiş, 1999). Tiered Tagging exploits reduced set of tags from which context recoverable features have been removed. For instance, the type of noun (common ‘c’ or proper ‘p’) as well as the gender feature (masculine ‘m’ or feminine ‘f’) from the *morpho-sintactic descriptors (MSDs)* ‘Ncfsrn’ and ‘Nemsrn’ are deleted to obtain the *short tag (CTAG)* ‘NSRN’, because the type of noun and gender information can be deterministically recovered based on the CTAG and the wordform itself. After the initial tagging is performed using any tagging method such as HMMs, Maximum Entropy, CRFs etc., the initially removed features are recovered using lexicons, linguistic rules (hand-coded or automatically derived) and, in the cases of unknown words, by employing custom machine learning techniques (Ceauşu, 2006).

The language-dependent process of manually inferring linguistic rules for MSD recovery requires good knowledge of the target language and also extensive amounts of time invested in testing and re-design. By using a Neural Network to perform multi classification (see section 2) our technique avoids any manual intervention in rule-design.

2 A Review of the Framework

A MSD is a vector of attribute values, which can be regarded as a hierachic representation of the POS information. A MSD encodes a part of speech (POS) with the associated lexical attribute values as a string of character codes that have a fixed pre-defined position inside the string. The first character is an upper case character denoting the part of speech (e.g. ‘N’ for nouns, ‘V’ for verbs, ‘A’ for adjectives, etc.) and the following characters (lower letters or ‘-’) specify the instantiations of the individual lexical attributes of the POS. For example, the MSD ‘Ncfsrn’, specifies a noun (the first character is ‘N’) the type of which is common (‘c’, the second character), feminine gender (‘f’), singular number (‘s’), in nominative/accusative case (‘r’) and indefinite form (‘n’). If a specific attribute is not relevant for a language, or for a given combination of feature-values, the character ‘-’ is used in the corresponding position. For a language which does not morphologically mark the gender and definiteness features, the earlier exemplified MSD will be encoded as ‘Nc-sr-’. The complete MSD description can be found in (Erjavec and Monachini, 1997).

The idea of using neural networks for part-of-speech tagging was previously introduced in Schmid (1994) and Marques and Lopes (1996). The main argument of using neural networks was that they are preferable to other methods, when the training set is small. Both previous approaches use a sliding window over the words of the sentence and the network is trained to determine the POS tag of the current word based on the previously assigned tags and the possible following tags that fit within the window. The algorithms use a binary encoding of POS tags in which each tag receives a unique binary vector that has only one bit set to 1 (e.g. 100 tags means the network contains 100 neurons on the output layer). The input vector for predicting the tag of the current

word encodes the probable tags for the current and following two words using MLE (equation 1):

$$P(t|w) = \frac{C(w, t)}{C(w)} \quad (1)$$

- | | | |
|-----------|---|---|
| $P(t w)$ | - | The probability of the word w having tag t |
| $C(w, t)$ | - | The total number of times, the word w appears with tag t in the training corpus |
| $C(w)$ | - | The total number of times, the word w appears in the training corpus |

The previously proposed methods still suffer from the same issue of data sparseness when dealing with large tagset languages. If we would apply the same methodology we would require a vector with at least 687 elements in order to encode each possible MSD in Romanian and around 2,500+ elements for Czech. In our previous work we used a neural framework to perform multi-attribute classification, showing that neural networks are also a better fit for managing large tagsets. We used a custom encoding of MSDs in which each value inside the vectors describing the MSDs uniquely identified a single attribute. In this initial approach we used a simple algorithm to derive these encodings, thus minimizing the size of the MSD vector from an initial 614 elements to only 110. For the tagging itself we trained a fully connected three-layer feed-forward neural network to output a vector of 110 elements (for the current word's MSD) based on the tags assigned to the previous two words and the possible tags for the current and following two words. The final MSD for the current word was chosen based on the minimum Euclidian distance between the output vector and a list of admissible MSDs derived from the training corpus and from a wordform lexicon.

In the case of out-of-vocabulary (OOV) words, we use equation 2, which is inspired after Brants (2000) to encode the possible MSDs based on the word's termination.

$$P(a|w) = \frac{C(w_{2,n}, a) + C(w_{3,n}, a) + \dots + C(w_{n-2,n}, a)}{C(w_{2,n}) + C(w_{3,n}) + \dots + C(w_{n-2,n})} \quad (2)$$

- | | | |
|-----------------|---|---|
| n | - | Length of the word |
| $w_{i,j}$ | - | Word substring from letter at position i to letter at position j |
| $C(w_{i,j}, a)$ | - | The total number of times substring $w_{i,j}$ appears with attribute a in the training corpus |
| $C(w_{i,j})$ | - | The total number of times substring $w_{i,j}$ appears in the training corpus |

This paper presents our improvements and tuning of the framework, as well as complementing our tests on Romanian with tests performed on Czech. We show how a genetic algorithm designed to create a custom topology for the network improves both accuracy and runtime tagging speed.

3 A New MSD Encoding Algorithm

In our previous work we used a simple algorithm for MSD encoding, which neglected the main classes from which the attributes inside the MSD belong to. This initial algorithm worked by sorting all possible attribute-values from all grammatical categories and assigning a unique position in the MSD encoded vector. However, this approach was not suitable for our current experiments and thus we derived a new algorithm for MSD-to-binary encoding.

As presented in section 2, each MSD begins with a letter that uniquely represents its category. Each category has a number of attribute types (ex: number, case) for which a MSD can take a value from (ex: possible values for the number attribute can be singular ‘s’ or plural ‘p’). The attribute types are not always unique to their category. For example, if we compare the Romanian Noun, Verb and Adjective, each of the three categories has a unique Type attribute (for Noun: common and proper, for Verb: main, auxiliary, modal and copula, and for Adjective: only qualificative), but they all share the Number and Clitic attribute types (they have the same possible values).

This observation has led us to the following encoding strategy to minimize the representation space:

1. Automatically identify the minimum number of attribute types and create a set containing them;
2. For each category record its attribute types (either unique or shared);
3. Read a wordform lexicon (containing words and their associated MSDs) and create a unique set of MSDs.
4. For each MSD, parse it letter by letter, applying steps 4.a and 4.b repeatedly to obtain its encoding:
 - a. Represent the MSD encoding space as a binary vector, starting with a number of binary elements of length equal to the number of categories; set the binary value corresponding to the MSD’s category to 1 (first letter in the MSD);
 - b. Continue the encoding space with each of the attribute types in the set obtained at step 1, encoding each as a number of binary elements of length equal to the number of possible values; set to 1 the appropriate element.

For step 3 we have used our latest in-house version of the wordform lexicon for Romanian (containing 1,151,989 wordforms having 687 unique MSDs) and Multext East’s wordform lexicon for Czech (containing 184,643 wordforms having 1,143 unique MSDs) (see table 1 for statistics).

Table 1. Statistics for the MSD encoding strategy

| | Romanian | Czech |
|---|----------|-------|
| Number of categories (distinct parts of speech) | 15 | 13 |
| Number of unique attribute types | 35 | 34 |
| Length of MSD encoding vector | 140 | 135 |
| Number of unique MSDs in lexicon (tag-set size) | 687 | 1,443 |

The table shows that even though Romanian has a slightly larger number of POS categories and unique attribute types yielding a larger MSD encoding vector, Czech is a significantly more inflected language, resulting in more than double the number of possible MSDs. This tag-set size difference between Czech and Romanian directly implies a significantly more difficult task for any POS tagger to obtain similar performance for both languages.

We further present an encoding example for the Romanian MSD ‘Rw-y’ belonging to the Adverb category (Table 2).

Table 2. Example of encoding for MSD ‘Rw-y’

| Category | | | | | | | | | | | | | | | Clitic | | Degree | | | Type-Adverb | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|--------|----|--------|-----|----|-------------|-----|-----|----|----|-----|-----|-----|-----|-----|-----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | ... | 41 | 42 | ... | 61 | 62 | 63 | ... | 98 | 99 | 100 | 101 | 102 | 103 | ... | 139 |
| J | N | V | A | P | D | T | R | S | C | M | Q | I | Y | X | y | n | ... | p | c | s | ... | g | p | z | m | w | c | ... | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | |

Adverbs have three attribute types: Type (unique for Adverbs), Degree (shared attribute type) and Clitic (also shared). MSD ‘Rw-y’ has 3 elements set to the value ‘1’: at position 7 (as positions 0-14 encode the category, index number 7 representing the Adverb ‘R’), at position 102 (as positions 98-103 encode the unique attribute Type for Adverbs with ‘w’ being 5th value out of the possible 6) and at position 41 (as positions 41-42 encode the shared Clitic attribute with 2 possible values). The dash ‘-’ symbol in the MSD specifies that its second attribute type Degree is missing, and thus none of its possible 3 values is set to ‘1’.

4 Romanian and Czech Baseline Experiment

For the first experiment on the Romanian and Czech languages we used a standard feed-forward network. The input and the output layers are fixed and of equal size, each having 140 neurons for Romanian or 135 for Czech, while the hidden layer varies between 50 and 150 neurons, in increments of 20 neurons. The network is fully connected: each neuron links to every neuron on the next layer. We have used the standard sigmoid activation function: $f(t) = 1/(1+e^{-t})$, where t is the weighted sum of all the inputs into the current neuron.

To see what would be the size of the hidden layer that would provide optimum results in our setting, we ran the network several times with different hidden layer sizes. All parameters except the layer size remained equal throughout the tests, stopping the network after 50 iterations (value was chosen empirically, after observing that longer training times yielded no significant score improvements, but taking a longer time, linearly with the number of iterations).

The corpus used for training and testing was George Orwell’s “1984” novel, available in multiple languages, which has been manually annotated with MSDs. The tests

were performed using 10-fold cross validation. The test sentences were randomly extracted from the novel. The Romanian translation of “1984” has 6,424 sentences with 118,357 words (MSDs) while the Czech version has 6,749 sentences with 100,323 words.

Table 3. Fully connected network results for different hidden layer sizes

| Hidden layer size | 50 | 70 | 90 | 110 | 130 | 150 |
|-------------------|--------|--------|---------------|--------|---------------|--------|
| Romanian | 97.96% | 98.05% | 98.19% | 98.14% | 98.12% | 98.18% |
| Czech | 90.70% | 90.48% | 90.90% | 91.14% | 91.30% | 91.22% |

The closest related work with which we can compare our results with was performed by Tufiș and Dragomirescu (2006). We used the same corpus, and we fall within the second scenario presented in Tufiș and Dragomirescu, where the tagger lexicon was computed from the training corpus as well as the wordform lexicon (which contains associated MSDs for each word). This has the direct result that the tagger does not encounter unknown words. However, there are testing differences, as we have 687 unique MSDs (including punctuation) for Romanian versus only 614; a similar difference appears for Czech where we extracted 1,443 MSDs versus 1,428.

The results are comparable: we obtained 98.19% for Romanian, versus the slightly lower score of 97.50% obtained by Tufiș and Dragomirescu (a difference of +0.69%). For Czech, the situation reverses: we obtained 91.30% versus the slightly higher previous score of 91.80% (a difference of -0.5%).

With this experiment we set our first baseline of 98.19% accuracy for Romanian and 91.30% for Czech.

5 Genetic Algorithm Generated Network Topologies

One of the largest problems with neural networks is their tendency for over-fitting especially when using a relatively small corpus. To better understand how the network works for POS tagging, we classified the relation between an input (I_i) and an output (O_j) as belonging to one of three classes: positive (P) – which means that if the feature I_i is non-zero it contributes to increasing the value of O_j ; negative (N) – which means that if the feature I_i is non-zero it contributes to decreasing the value of O_j ; don't care (D) – which means that the value of I_i has little or no impact on the value of O_j . In our pattern analysis process we observed that the feminine gender attribute is positively influenced by the feminine gender attribute of a neighboring noun/adjective, but it is also positively influenced by the presence of a preposition or a main verb, which, from a linguistic perspective should not happen, and it is most likely an effect of over-fitting.

One way to cope with over-fitting is to disable certain links between neurons that would otherwise produce this undesired effect. Genetic algorithms have been successfully applied for finding optimal network topologies (Schaffer et al, 1992; Fischer et al, 1998; Fiszelew et al., 2007). They excel at this task because they can explore many

different parts of a large solution and narrow down the search space in comparison to full-grid optimizations. Starting from an initial random population and based on the fitness of each individual, the best suited candidates are recombined using different methods. If the fitness and recombination methods are properly designed, genetic algorithms have the ability to generate new solutions to a problem while also preserving partial ones.

To assess the performance of genetic algorithms for our task we designed three experiments applied to both Romanian and Czech (see section 6.4 for results). In all our experiments we used the multilingual „1984” annotated corpus (Romanian and Czech versions), maintaining the 10-fold cross validation method for testing. In the first experiment (E1) we used a custom designed topology and in the second (E2) and third (E3) experiments we used two different genetic approaches. As a design choice, the neural network we used in E2 has a 50 neuron hidden layer because the training/testing procedure takes a lot of time on larger networks and also because a smaller hidden layer helps avoid having a sparse network given the fact that roughly half of the synapses will be disabled by the genetic algorithm.

Tagging accuracy is measured as an average after 50 training iterations and 10 random initializations (for each topology).

5.1 Experiment E1: Manually Designed Simplified Network Topology

In the first experiment we used a hand-made custom network topology in which we fully connected only attributes belonging to the same class. For example, the gender attributes (m/f) from all neighboring MSDs were fully connected to a variable number of neurons in the hidden layer, which again was connected to the neurons used to encode the gender attribute of the output MSD in the output layer.

The motivation behind manually designing this topology is that we wanted to have another baseline that was as simple as possible, as opposed to the standard fully connected network generally used.

We experimented with a number of different combinations of the number of neurons in the hidden layer used for each individual attribute group. Because our purpose was to establish a baseline system, we used the topology with the highest yielding results.

5.2 Experiment E2: Unrestricted Genetic Algorithm Generated Topology

In the second experiment (E2) we used a genetic algorithm to find a custom network topology. In our implementation, each individual belonging to a population is represented by a binary vector and each value inside this vector is associated to a synapse inside the neural network, on any layer. The binary values describing the individuals are used to select the state of the synapses (0 - disabled, 1 - enabled). During training and testing, the network disregards the disabled synapses. The genetic algorithm uses a uniform cross-over probability of 50%, with a 10% chance of mutation. Based on the fitness function, the first two best-fit individuals are kept in the new population. The following individuals are generated from the Cartesian product of the first n-best

individuals¹. Each pair of individuals spawns two children. In our early experiments we used the accuracy of the tagger on the cross-validation set as a measurement of fitness.

5.3 Experiment E3: Restricted Genetic Algorithm Generated Topology

In E2 synapses were randomly enabled or disabled both between the input and hidden layer and between the hidden and the output layer. In the third experiment (E3), while similar to E2, we adapted the genetic algorithm to take into account attribute groups so that we enable or disable all synapses that link groups from different layers. As such, an individual is represented by a vector whose bits encode whether a specific group in a layer should connect to another group in the next layer.

The size of the hidden layer was fixed to be equal to the size of a MSD (140 neurons for Romanian and 135 for Czech), to facilitate our segmentation procedure. Each individual layer was segmented into groups of neurons in accordance with the attribute types. The intuition behind E3 was that the links between different attribute types should all be either enabled or disabled at a time rather than just randomly some of them as is the case for E2. This change directly leads to a reduced number of possible variations that can be generated by the genetic algorithm.

The accuracy was also measured as an average after 50 training iterations and 10 random initializations (for each topology).

5.4 Experiment Results

Table 4 summarizes the accuracy figures obtained on Romanian and Czech by our four network topologies: the unmodified, fully connected standard topology (U), the manually designed simplified topology (E1), the topology generated by the unrestricted genetic algorithm (E2) and the topology generated by the restricted genetic algorithm (E3). For experiments E2 and E3 we provide the values for the first two best-fit individuals I1 and I2.

Table 4. Manual and genetic network topology results

| Language | Unmodified topology (U) | Manual topology (E1) | Unrestricted genetic (E2) | Restricted genetic (E3) |
|----------|-------------------------|----------------------|---------------------------|-------------------------|
| Romanian | 97.96% | 97.17% | I1: 98.17% | I1: 98.09% |
| | | | I2: 98.12% | I2: 98.05% |
| Czech | 90.70% | 90.76% | I1: 91.69% | I1: 91.06% |
| | | | I2: 91.64% | I2: 90.04% |

As shown, the manually designed topology (E1) provides slightly better results than the unmodified topology on Czech, but performs worse on Romanian. The intuition behind E1 was that local agreements between attribute values should only appear

¹ In our experiments we used n=5.

between similar attribute groups (e.g. the gender of a noun should be influenced by the gender of a neighboring adjective). However, the results obtained in experiment U and E1 on Romanian show that there are dependencies across different attribute groups. Somewhat surprisingly, the unrestricted genetic algorithm (E2) obtained better candidates than the restricted algorithm (E3). To make sure the results are reliable, we repeated experiments E2 and E3 several times, each time obtaining similar accuracy figures.

As shown in table 5, besides improving the accuracy of the system, a custom network topology also provides increased tagging speeds. The results were computed by averaging the time it took to re-label the entire training sets for both Romanian and Czech. The experiment was repeated three times in order to reduce the effect of system load on tagging speed.

Table 5. Tagging speeds of different network topologies

| Topology | Romanian | Czech |
|-----------------|------------------|------------------|
| | Words/sec | Words/sec |
| U | 7,183 | 6,981 |
| E1 | 14,366 | 13,962 |
| E2-I1 | 14,659 | 14,189 |
| E3-I1 | 11,402 | 12,466 |

6 Conclusions

The Neural MSD tagger is a viable alternative to other language specific large tagset labeling methods. The main advantage of our approach is that it does not require in-depth linguistic knowledge.

Our system allows users to provide their own MSD encodings, which permits them to mask certain features that are not useful for a given NLP application. If one wants to process a large amount of text and is interested only in assigning grammatical categories to words, he or she can use a MSD encoding in which he strips off all unnecessary features. Thus, the number of necessary neurons would decrease, which assures faster training and tagging. This is of course possible in any other tagging approach, but our framework supports this by masking attributes inside the MSD encoding configuration file, without having to change anything else in the training corpus.

We have also showed that using genetic algorithms for selecting a custom network topology offers better results in terms of both accuracy and tagging speed.

The POS tagger is implemented as part of a larger application that is primarily intended for text-to-speech (TTS) synthesis. The system is free for non-commercial use and we provide both web and desktop user-interfaces. It is part of the METASHARE platform and available online².

² <http://ws.racai.ro:9191>

References

1. Berger, A.L., Pietra, V.J.D., Pietra, S.A.D.: A maximum entropy approach to natural language processing. *Computational Linguistics* 22(1), 39–71 (1996)
2. Boros, T., Ion, R., Tufiș, D.: Large tagset labeling using Feed Forward Neural Networks. Case study on Romanian Language. Accepted for publication in ACL, Sofia, Bulgaria (2013)
3. Brants, T.: TnT: a statistical part-of-speech tagger. In: *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pp. 224–231. Association for Computational Linguistics (2000)
4. Calzolari, N., Monachini, M. (eds.): Common Specifications and Notation for Lexicon Encoding and Preliminary Proposal for the Tagsets. *MULTEXT Report* (March 1995)
5. Ceausu, A.: Maximum entropy tiered tagging. In: *Proceedings of the 11th ESSLLI Student Session*, pp. 173–179 (2006)
6. Erjavec, T., Monachini, M. (eds.): Specifications and Notation for Lexicon Encoding. Deliverable D1.1 F. Multext-East Project COP-106 (1997)
7. Fischer, M.M., Leung, Y.: A genetic-algorithms based evolutionary computational neural network for modelling spatial interaction data. *Neural network for modelling spatial interaction data. The Annals of Regional Science* 32(3), 437–458 (1998)
8. Fiszelew, A., Britos, P., Ochoa, A., Merlino, H., Fernández, E., García-Martínez, R.: Finding optimal neural network architecture using genetic algorithms. *Adv. Comput. Sci. Eng. Res. Comput. Sci.* 27 (2007)
9. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
10. Marques, N.C., Lopes, G.P.: A neural network approach to part-of-speech tagging. In: *Proceedings of the 2nd Meeting for Computational Processing of Spoken and Written Portuguese*, pp. 21–22 (1996)
11. Ratnaparkhi, A.: A maximum entropy model for part-of-speech tagging. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, vol. 1, pp. 133–142 (1996)
12. Samuelsson, C.: Morphological tagging based entirely on Bayesian inference. In: *9th Nordic Conference on Computational Linguistics* (June 1993)
13. Schmid, H.: Part-of-speech tagging with neural networks. In: *Proceedings of the 15th Conference on Computational Linguistics*, vol. 1, pp. 172–176. Association for Computational Linguistics (August 1994)
14. Schaffer, J.D., Whitley, D., Eshelman, L.J.: Combinations of genetic algorithms and neural networks: A survey of the state of the art. In: *International Workshop on Combinations of Genetic Algorithms and Neural Networks, COGANN 1992*, pp. 1–37. IEEE (June 1992)
15. Tufiș, D., Barbu, A.M., Pătrașcu, V., Rotariu, G., Popescu, C.: Corpora and Corpus-Based Morpho-Lexical Processing. In: *Recent Advances in Romanian Language Technology*, pp. 35–56. Romanian Academy Publishing House (1997) ISBN 973-27-0626-0
16. Tufiș, D.: Tiered tagging and combined language models classifiers. In: Matoušek, V., Mautner, P., Ocelíková, J., Sojka, P. (eds.) *TSD 1999. LNCS (LNAI)*, vol. 1692, pp. 28–33. Springer, Heidelberg (1999)
17. Tufiș, D., Dragomirescu, L.: Tiered tagging revisited. In: *Proceedings of the 4th LREC Conference* (2004)

Neural Network Simulation of Photosynthetic Production

Tibor Kmet¹ and Maria Kmetova²

¹ Constantine the Philosopher University, Department of Informatics,
Tr. A. Hlinku 1, 949 74 Nitra, Slovakia
tkmet@ukf.sk
<http://www.ukf.sk>

² Constantine the Philosopher University, Department of Mathematics,
Tr. A. Hlinku 1, 949 74 Nitra, Slovakia
mkmetova@ukf.sk

Abstract. A neural network based optimal control synthesis is presented for solving optimal control problems with control and state constraints and discrete time delay. The optimal control problem is transcribed into nonlinear programming problem which is implemented with adaptive critic neural network. The proposed simulation methods is illustrated by the optimal control problem of photosynthetic production described by discrete time delay differential equations. Results show that adaptive critic based systematic approach holds promise for obtaining the optimal control with control and state constraints.

Keywords: adaptive critic synthesis, feedforward neural network, optimal control problem with discrete time delay, mechanistic model of photosynthesis, neural network simulation, numerical solution.

1 Introduction

Optimal control of nonlinear systems with discrete time delays in state and control variables is one of the most active subjects in control theory. There is rarely an analytical solutions [6] although several numerical computation approaches have been proposed e.g. see [7], [9], [15], [19]. The most of the literature dealing with numerical methods for the solution of general optimal control problems focuses on algorithms for solving discretized problems. The basic idea of these methods is to apply nonlinear programming techniques to the resulting finite dimensional optimization problem [1], [7]. Then neural networks are used as universal function approximation to solve finite dimensional optimization problems forward in time with "adaptive critic designs" [12], [13], [20]. For the neural network, a feed forward neural network with one hidden layer, a steepest descent error backpropagation rule, a hyperbolic tangent sigmoid transfer function and a linear transfer function were used.

The paper presented extends adaptive critic neural network architecture proposed by [11] to the optimal control problems with discrete time delays in state

and control variables subject to control and state constraints. This paper is organized as follows. In Section 2, we present a description of the model of adaptive photosynthetic production with discrete time delay. In Section 3, optimal control problems with delays in state and control variables subject to control and state constraints are introduced. We summarize the necessary optimality conditions, give a short overview of the basic results including the iterative numerical methods. Section 4 presents a short description of adaptive critic neural network synthesis for the optimal control problem with delays in state and control variables subject to control and state constraints. We also present new algorithm to solve optimal control problems. We apply the new proposed methods to the model presented to compare short-term and long-term strategies of photosynthetic production. Numerical results and conclusions are being presented in Section 5.

2 A Mechanistic Model of Phytoplankton Photosynthesis

Mathematical models of photosynthesis in bioreactors are important for both basic science and the bioprocess industry [4]. There is a class of models based on the concept of the 'photosynthetic factories' developed by Eilers and Peeters [2]. The dynamic behaviour of the model has also been discussed in [3], [10], [14]. Wu [21] presents an approach to model the kinetic of photosynthetic systems photobioreactor design in an alternating light/dark regime. Assuming that phytoplankton regulates its photosynthetic production rate with a certain strategy which maximize production, two such possible strategies is examined, i.e. instantaneous and the integral maximal production.

Basic for the following consideration is the mechanistic model of phytoplankton photosynthesis. It is based on unit processes concerning the cellular reaction centres called *photo-synthetic factories – PSF*.

It is known from algal physiology [2] that three states of a PSF are possible: x_1 - resting, x_2 - activated and x_3 - inhibited. Photons are captured by a PSF in state x_1 which passes to state x_2 . The PSF in state x_2 can either return to state x_1 at a constant rate γ and with discrete time delay τ_p or pass to the inhibited

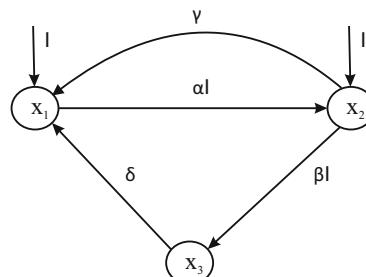


Fig. 1. The structure of the three states model: x_1 - resting, x_2 - activated and x_3 - inhibited

state x_3 . Transitions between states depend both on light intensity and time. The probabilities of the PSF being in the state x_1, x_2 or x_3 , are given as p_1, p_2 and p_3 , respectively. Transitions between states can be expressed as follows:

$$\begin{aligned}\dot{p}_1(t) &= -\alpha I p_1(t) + \gamma p_2(t - \tau_p) + \delta p_3(t) \\ \dot{p}_2(t) &= \alpha I p_1(t) - (\beta I + \gamma) p_2(t) \\ \dot{p}_3(t) &= \beta I p_2(t) - \delta p_3(t).\end{aligned}\quad (1)$$

The parameters $\alpha, \beta, \gamma, \delta$ and τ_p occurring in this model are positive constants and I is a light intensity.

System (1) for all $I \geq 0$ has a unique positive equilibrium $\bar{p}(I)$ with entries

$$\begin{aligned}\bar{p}_1(I) &= \frac{\beta\delta I + \gamma\delta}{F} \\ \bar{p}_2(I) &= \frac{\alpha\delta I}{F} \\ \bar{p}_3(I) &= \frac{\alpha\beta I^2}{F},\end{aligned}\quad (2)$$

where $F = \alpha\beta I^2 + (\alpha + \beta)\delta I + \gamma\delta$. For $\tau = 0$ the equilibrium $\bar{p}(I)$ is globally asymptotically stable, that means for fixed light intensity I all solutions with initial condition $p(0), p_1(0) + p_2(0) + p_3(0) = 1$ converge to $\bar{p}(I)$. For proof see [10]. Based on the result due to Gopalsamy [8] we get that the equilibrium point $\bar{p}(I)$ is asymptotically stable for all $\tau_p \geq 0$.

2.1 Optimization of Photosynthetic Production

Let us assume that phytoplankton regulates its photosynthetic production rate (FP) with a certain strategy which maximizes production. The rate of the photosynthetic production FP is proportional to the number of transitions from x_2 to x_1 . Let us investigate the optimal values of light intensity $I(t)$, for which the photosynthetic production $FP(t) = \gamma p_2(t - \tau_p)$ is maximal under constraints $I \in [I_{min}, I_{max}]$. We will examine two strategies:

(1) instantaneous maximal photosynthetic production with respect to I , (local optimality), i.e.,

$$\dot{p}_2 = f_2(p, I, t) \rightarrow \max$$

for all t , under the constraints $I \in [I_{min}, I_{max}]$.

(2) integral maximal photosynthetic production with respect to I , (global optimality), i.e.

$$J(I) = \int_0^{t_f} \gamma p_2(t - \tau_p) dt \rightarrow \max,$$

under the constraints $I \in [I_{min}, I_{max}]$.

2.2 Local Optimality

In the case of strategy (1), we maximize the following function

$$J(I(t)) = I(t)\alpha p_1(t) - I(t)\beta p_2(t).$$

under the constraints $I \in [I_{min}, I_{max}]$ which attains its maximum \tilde{I} in the manner

$$\tilde{I} := \begin{cases} I_{max} & \text{if } \alpha p_1(t) - \beta p_2(t) > 0 \\ I_{min} & \text{if } \alpha p_1(t) - \beta p_2(t) < 0 \end{cases}$$

By straightforward calculation we can show that $\alpha p_1(t) = \beta p_2(t)$ can occur at isolated point t only. To maximize instantaneous photosynthetic production with respect to light intensity I for steady state solution $p(t) = \bar{p}(I)$ we examine the following function

$$FP(I) = \frac{\alpha\delta I\gamma}{\delta\gamma + (\beta\delta + \alpha\delta)I + \alpha\beta I^2}.$$

By straightforward calculation we get that the optimal light intensity is given by:

$$I^* = \sqrt{\frac{\gamma\delta}{\alpha\beta}}.$$

with corresponding steady state solution $\bar{p}(I^*)$.

2.3 Global Optimality

In case of strategy 2, we have the following optimal control problem: to find a function $\hat{I}(t)$, for which the goal function

$$J(I) = \int_0^{t_f} \gamma p_2(t - \tau_p) dt \quad (3)$$

attains its maximum, where t_f is fixed. We introduce an additional state variable

$$p_0(t) = \int_0^t \gamma p_2(s - \tau_p) ds.$$

We are led to the following optimal control problems: Maximize

$$p_0(t_f) \quad (4)$$

under the constraints

$$\begin{aligned} c_1(p, I) &= I_{min} - I \leq 0 \\ c_2(p, I) &= I - I_{max} \leq 0. \end{aligned}$$

3 Discretization of the Optimal Control Problem

The direct optimization methods for solving the optimal control problem are based on a suitable discretization of (3), see e.g. [1], [7]. Defining $h = \frac{\tau_p}{k}$ gives the interval length for an elementary transformation interval. The grid point number for an equidistant discretization mesh $N = \frac{t_f - t_0}{h}$. Let $t_i \in \langle t_0, t_f \rangle$, $i = 0, \dots, N$, be an equidistant mesh point with $t_i = t_0 + ih$, $i = 0, \dots, N$, where $h = \frac{b-a}{N}$ is a time step and $t_f = Nh + t_0$. Let the vectors $p^i \in R^4$, $I^i \in R$, $i = 0, \dots, N$, be an approximation of the state variable and control variable $p(t_i)$, $I(t_i)$, respectively at the mesh point t_i . Euler's approximation applied to the differential equations yields

$$\text{Minimize } \mathcal{G}(z) = -p_0^N \quad (5)$$

subject to

$$\begin{aligned} p^{i+1} &= p^i + hF(p^i, p^{i-k}, I^i, I^{i-l}) \\ i &= 0, \dots, N-1, \\ p^{-i} &= \phi_p(t_0 - ih), \quad i = k, \dots, 0, \\ c(p^i, I^i) &\leq 0, \quad i = 0, \dots, N-1, \end{aligned} \quad (6)$$

where the vector function

$$F(p, p_{\tau_p}, I) = (-\gamma p_2, f_1(p, p_{\tau_p}, I), \dots, f_3(p, p_{\tau_p}, I))$$

is given by Eq. (4) and by right-hand side of Eq. (1).

Let us introduce the Lagrangian function for the nonlinear optimization problem (5):

$$\begin{aligned} \mathcal{L}(z, \lambda, \mu) &= \sum_{i=0}^{N-1} \lambda^{i+1} (-p^{i+1} + p^i + hf(p^i, p^{i-k}, I^i)) + \mathcal{G}(p^N) + \\ &\quad \sum_{i=0}^{N-1} \mu^i c(p^i, I^i), \end{aligned}$$

where $z := (p^0, p^1, \dots, p^{N-1}, I^0, \dots, I^{N-1}) \in R^{N_s}$, $N_s = 4N$. The first order optimality conditions of Karush-Kuhn-Tucker [15] for the problem (5) are:

$$\begin{aligned} 0 &= \mathcal{L}_{p^i}(z, \lambda, \mu) = \lambda^{i+1} - \lambda^i + h\lambda^{i+1}f_{p^i}(p^i, p^{i-k}, I^i) + \\ &\quad h\lambda^{i+k+1}f_{p_{\tau_p}}(p^{i+k}, p^i, I^{i+k}) + \mu^i c_{p^i}(p^i, I^i), \\ i &= 0, \dots, N-k-1, \end{aligned} \quad (7)$$

$$\begin{aligned} 0 &= \mathcal{L}_{p^i}(z, \lambda, \mu) = \lambda^{i+1} - \lambda^i + h\lambda^{i+1}f_{p^i}(p^i, p^{i-k}, I^i) + \mu^i c_{p^i}(p^i, I^i), \\ i &= N-k, \dots, N-1, \end{aligned}$$

$$0 = \mathcal{L}_{p^N}(z, \lambda, \mu) = \mathcal{G}(p^N) - \lambda^N, \quad (8)$$

$$\begin{aligned} 0 &= \mathcal{L}_{I^i}(z, \lambda, \mu) = h\lambda^{i+1}f_{I^i}(p^i, p^{i-k}, I^i) + \mu^i c_{I^i}(p^i, I^i), \\ i &= 0, \dots, N-1. \end{aligned} \quad (9)$$

4 Adaptive Critic Neural Network for an Optimal Control Problem with Control and State Constraints

It is well known that a neural network can be used to approximate the smooth time-invariant functions and the uniformly time-varying function [5], [18]. Fig. 2 shows a feed forward neural network with n inputs node, one hidden layer of r units and m output units.

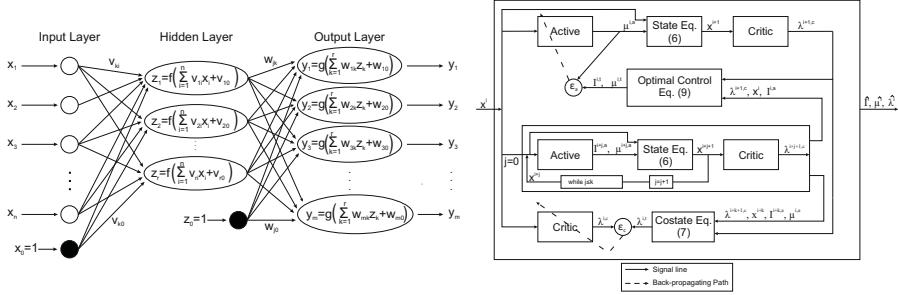


Fig. 2. Feed forward neural network topology with one hidden layer, v_{ki} , w_{jk} are values of connection weights, v_{k0} , w_{j0} are values of bias, $f(\cdot)$, $g(\cdot)$ are activation functions. Architecture of adaptive critic feed forward network synthesis, x^i -input signal to the action and critic network, $\hat{t}^{i,t}$, $\hat{\mu}^{i,t}$ and $\hat{\lambda}^{i,t}$ are output signal from action and critic network, respectively.

Let $x = [x_1, \dots, x_n]'$ and $y = [y_1, \dots, y_m]'$ be the input and output vectors of the network, respectively. Let $V = [v_1, \dots, v_r]'$ be the matrix of synaptic weights between the input nodes and the hidden units, where $v_k = [v_{k0}, v_{k1} \dots, v_{kn}]$; v_{k0} is the bias of the k th hidden unit, and v_{ki} is the weight that connects the i th input node to the k th hidden unit.

Let also $W = [w_1, \dots, w_m]'$ be the matrix of synaptic weights between the hidden and output units, where $w_j = [w_{j0}, w_{j1} \dots, w_{jr}]$; w_{j0} is the bias of the j th output unit, and w_{jk} is the weight that connects the k th hidden unit to the j th output unit. The response of the k th hidden unit is given by $z_k = \tanh(\sum_{i=0}^n v_{ki} x_i)$, $k = 1, \dots, r$, where $\tanh(\cdot)$ is the activation function for the hidden units. The response of the j th output unit is given by $y_j = \sum_{k=0}^r w_{jk} z_k$, $j = 1, \dots, m$. The multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between the input and output vectors. The number of neurons in the input and output layers is given by the number of input and output variables, respectively. The multi-layered feed forward network shown in Fig. 2 is trained using the steepest descent error backpropagation rule. Basically, it is a gradient descent, a parallel distributed optimization technique to minimize the error between the network and the target output [17].

We can state the algorithm to solve the optimal control problem using the adaptive critic and recurrent neural network. In the Pontryagin's maximum principle for deriving an optimal control law, the interdependence of the state, costate and control dynamics is made clear. Indeed, the optimal control \hat{I}^i and multiplier $\hat{\mu}$ is given by Eq. (9), while the costate Eqs. (7) - (8) evolves backward in time and depends on the state and control. The adaptive critic neural network is based on this relationship is shown in Fig. 2. It consists of two networks at each node: an action network, the inputs for which are the current states and its outputs are the corresponding control \hat{u} and multiplier $\hat{\mu}$, and the critic network for which the current states are inputs and current costates are outputs for normalizing the inputs and targets (zero mean and standard deviations). For detail explanation see [17]. From the free terminal condition from Eqs. (7) - (8) we obtain that $\lambda_0^i = -1$, $i = N, \dots, 0$ and $\lambda_j^N = 0$, $j = 1, \dots, N$. We use this observation before proceeding to the actual training of the adaptive critic neural network. Further discussion and detail explanation of these adaptive critic methods can be found in [11], [12], [13] and [20].

Algorithm 1: Algorithm to solve the optimal control problem.

Input: Choose t_0 , t_f , N - number of steps, time step h , $\alpha > 0$, $\beta > 0$, ε_a , and ε_c - stopping tolerance for action and critic neural network, respectively, $p^{-i} = \phi_s(t_0 - ih)$, $i = k, \dots, 0$, $I^{-i} = \phi_c(t_0 - ih)$, $i = l, \dots, 0$ -initial values.

Output: Set of final approximate optimal control $\hat{u}(t_0 + ih) = \hat{I}^i$ and optimal trajectory $\hat{x}(t_0 + (i+1)h) = \hat{p}^{i+1}$, $i = 0, \dots, N-1$, respectively

- 1 Set the initial weight $\mathbb{W}^a = (V^a, W^a)$, $\mathbb{W}^c = (V^c, W^c)$
- 2 **for** $i \leftarrow 0$ **to** $N-1$ **do**
- 3 **while** $err_a \geq \varepsilon_a$ **and** $err_c \geq \varepsilon_c$ **do**
- 4 **for** $j \leftarrow 0$ **to** k **do**
- 5 Compute $I^{i+j,a}$, $\mu^{i+j,a}$ and $\lambda^{i+j+1,c}$ using action (\mathbb{W}^a) and critic (\mathbb{W}^c) neural networks, respectively and p^{i+j+1} by Eq. (6)
- 6 Compute $\lambda^{i,t}$, $I^{i,t}$, and $\mu^{i,t}$ using Eqs. (7), (9)
- 7 **if** $i = N-1$ **then**
- 8 $\lambda^N = \mathcal{G}_{p^N}(p^N)$
- 9 Compute $I^{i,t}$, and $\mu^{i,t}$ using Eqs. (7), (9)
- 10 $err_c = \| \lambda^{i,t} - \lambda^{i,c} \|$
- 11 $err_a = \| (u, \mu)^{i,t} - (u, \mu)^{i,a} \|$
- 12 With the data set p^i , $\lambda^{i,t}$ update the weight parameters \mathbb{W}^c
- 13 With the data set p^i , $(u, \mu)^{i,t}$ update the weight parameters \mathbb{W}^a
- 14 Set $\lambda^{i,c} = \lambda^{i,t}$, $(I, \mu)^{i,a} = (I, \mu)^{i,t}$
- 15 Set $\hat{\lambda}^i = \lambda^{i,t}$, $(\hat{I}^i, \hat{\mu}^i) = (I, \mu)^{i,t}$
- 16 Compute \hat{p}^{i+1} using Eq. (6) and \hat{I}^i
- 17 **return** $\hat{\lambda}^i$, \hat{I}^i , $\hat{\mu}^i$, \hat{p}^{i+1}

5 Numerical Results and Conclusion

In the adaptive critic synthesis, the critic and action network were selected such that they consist of three and two subnetworks, respectively, each having 3-18-1 structure (i.e. three neurons in the input layer, eighteen neurons in the hidden layer and one neuron in the output layer). The results of numerical solutions have

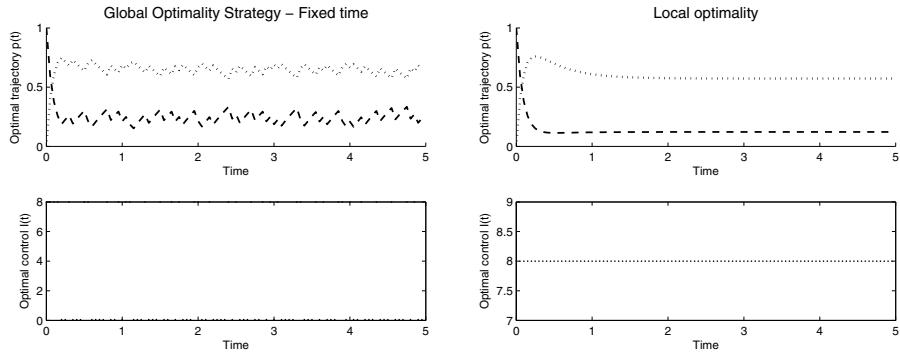


Fig. 3. Adaptive critic neural network simulation of optimal control $\hat{I}(t)$ and \tilde{I} for global and local strategies, respectively with fixed final time, dotted line $\tilde{p}_1(t)$, $\hat{p}_1(t)$, dashed line $\tilde{p}_2(t)$, $\hat{p}_2(t)$, $J(\tilde{I}) = 2.671$, $J(\hat{I}) = 2.920$

shown, see Fig. 3 that the optimal strategies $\tilde{I}(t)$ and $\hat{I}(t)$ based on short or long-term perspective, respectively, have different time trajectory and for long-term strategy optimal trajectory $\hat{p}(t)$ is near to the steady state $\bar{p}(I^*)$. In the case of instantaneous maximal photosynthetic production $\tilde{I}(t) = I_{max}$. In the second case the optimal light intensity $\hat{I}(t)$ is bang-bang and we obtain the light/dark regime. Note that $J(\hat{I}(t)) > J(\tilde{I}(t))$. The results of numerical calculations have shown that the proposed *adaptive critic neural network* is able to meet the convergence tolerance values that we choose, which led to satisfactory simulation results. Simulations, using MATLAB show that proposed neural network is able to solve nonlinear optimal control problem with state and control constraints and discrete time delay.

Acknowledgment. The paper was worked out as a part of the solution of the scientific project number KEGA 004UJS-4/2011 and VEGA 1/0699/12.

References

1. Buskens, C., Maurer, H.: SQP-methods for solving optimal control problems with control and state constraints: adjoint variable, sensitivity analysis and real-time control. *Journal of Computational and Applied Mathematics* 120, 85–108 (2000)

2. Eilers, P.H.C., Peeters, J.C.H.: A model for relationship between light intensity and the rate of photosynthesis in phytoplankton. *Ecol. Modelling* 42, 199–215 (1988)
3. Eilers, P.H.C., Peeters, J.C.H.: Dynamic behaviour of a model for photosynthesis and photoinhibition. *Ecol. Modelling* 69, 113–133 (1993)
4. Garcia-Camacho, F., Sanchez-Miron, A., Molina-Grima, E., Camacho-Rubio, F., Merchuk, J.C.: A mechanistic model of photosynthesis in microalgal including photoacclimation dynamics. *Jour. Theor. Biol.* 304, 1–15 (2012)
5. Hornik, M., Stichcombe, M., White, H.: Multilayer feed forward networks are universal approximators. *Neural Networks* 3, 256–366 (1989)
6. Hrinca, I.: An Optimal Control Problem for the Lotka-Volterra System with Delay. *Nonlinear Analysis, Theory, Methods, Applications* 28, 247–262 (1997)
7. Gollman, L., Kern, D., Mauer, H.: Optimal control problem with delays in state and control variables subject to mixed control-state constraints. *Optim. Control Appl. Meth.* 30, 341–365 (2009)
8. Gopalsamy, K.: Stability and Oscillation in Delay Different Equations in Population Dynamics. Kluwer Academic Publisher, Boston (1992)
9. Kirk, D.E.: Optimal Control Theory: An Introduction. Dover Publications, Inc., Mineola (1989)
10. Kmet, T., Straskraba, M., Mauersberger, P.: A mechanistic model of the adaptation of phytoplankton photosynthesis. *Bull. Math. Biol.* 55, 259–275 (1993)
11. Kmet, T.: Neural Network Solution of Optimal Control Problem with Control and State Constraints. In: Honkela, T. (ed.) ICANN 2011, Part II. LNCS, vol. 6792, pp. 261–268. Springer, Heidelberg (2011)
12. Padhi, R., Unnikrishnan, N., Wang, X., Balakrishnan, S.N.: Adaptive-critic based optimal control synthesis for distributed parameter systems. *Automatica* 37, 1223–1234 (2001)
13. Padhi, R., Balakrishnan, S.N., Randolph, T.: A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems. *Neural Networks* 19, 1648–1660 (2006)
14. Papacek, S., Celikovsky, S., Rehak, R., Styš, D.: Experimental design for parameter estimation of two time-scale model of photosynthesis and photoinhibition in microalgae. *Math. Comp. Sim.* 80, 1302–1309 (2010)
15. Polak, E.: Optimization Algorithms and Consistent Approximation. Springer, Heidelberg (1997)
16. Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., Mischenko, E.F.: The Mathematical Theory of Optimal Process. Nauka, Moscow (1983) (in Russian)
17. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representation by error propagation. In: Rumelhart, D.E., McClelland, D.E. (eds.) PDP Research Group: Parallel Distributed Processing: Foundation, pp. 318–362. The MIT Press, Cambridge (1987)
18. Sandberg, E.W.: Notes on uniform approximation of time-varying systems on finite time intervals. *IEEE Transactions on Circuits and Systems-1: Fundamental Theory and Applications* 45, 305–325 (1998)
19. Sun, D.Y., Huang, T.C.: A solutions of time-delayed optimal control problems by the use of modified line-up competition algorithm. *Journal of the Taiwan Institute of Chemical Engineers* 41, 54–64 (2010)
20. Werbos, P.J.: Approximate dynamic programming for real-time control and neural modelling. In: White, D.A., Sofge, D.A. (eds.) *Handbook of Intelligent Control: Neural Fuzzy, and Adaptive Approaches*, pp. 493–525 (1992)
21. Wu, X., Merchuk, J.C.: A model integrating fluid dynamics in photosynthesis and photoinhibition. *Chem. Ing. Scien.* 56, 3527–3538 (2001)

A Novel Artificial Neural Network Based Space Vector Modulated DTC and Its Comparison with Other Artificial Intelligence (AI) Control Techniques

Sadhana V. Jadhav and B.N. Chaudhari

Department of Electrical Engg.,
College of Engineering Pune, India
{svj.elec,bnc.elec}@coep.ac.in

Abstract. On the basis of Artificial Neural Network (ANN) theory, this paper has put forth a new kind of controller for Space Vector Modulated (SVM) Direct Torque Control (DTC) system for Induction Motor. The controller has features like smooth operation, high dynamics, stable and robust performance. The training algorithm used is Resilient Back Propagation (RBP). The paper also presents the comparison of proposed ANN controller with other intelligent controllers (Fuzzy based control and Fuzzy_Sliding Mode Control) for the drive, based on various control performance criterion at transient state as well as steady state. It is observed that while working with comparatively less control efforts, proposed ANN improves the performance of SVM_DTC in all-round way. Simulation results confirm the superiority and feasibility of the proposed ANN controller.

Keywords: An induction motor, Direct Torque Control, Space Vector Modulation, Artificial Intelligence (AI) Control techniques, Artificial Neural Networks (ANN) based Control, Fuzzy Inference System (FIS), Hybrid Fuzzy-Sliding Mode Control.

1 Introduction

Induction Motors (IM) have some inherent characteristics such as multivariate, parameter indeterminacy, strong coupling and non-linearity. These bring about a lot of complexity to the IM drive system. Thanks to Direct Torque Control (DTC) [1-3] and Space Vector Modulation (SVM) [4-5] technologies, they offer fast and dynamic decoupled control solution. They have gained popularity as they are less dependence on rotor parameter variations, no co-ordinate transformation, constant and controllable switching frequency, etc. SVM_DTC technologies are mature enough and have steered the drive scenario completely. These advanced control techniques for induction motor have more rapid and smoother response with minimum overshoot. Simple and linear PI controllers employed in this system, are unable to give an optimal response for different operating conditions and they are hard to tune. *"There are two central issues and problems in motion control. One is to make*

the resulting drive system, robust against parameter and load variations and disturbances and the other is to make the system intelligent" [6]. Artificial Intelligence (AI) based methods like Artificial Neural Networks (ANN) can be used to identify and control the non-linear dynamic systems since they can approximate a wide range of non-linear functions to any desired degree of accuracy. The advent of fast processors such as DSPs and high frequency power devices like IGBTs has accelerated the research in AI based drives, during last two decades [7-11]. To represent any system accurately by an ANN, the designer has to choose number of hidden layers, number of neurons in each layer of the network. Sigmoid transfer functions or squashing functions are the basic elements of a neuron in the hidden layers. Each neuron has weights and bias to relate mathematically to any other neuron of the next layer. The networks can be "trained" using parallel and distributed processing with massive connections among processing units. While training the network to behave like any practical system, weights and biases of the neurons are tuned using standard input-output mapping. In recent years, the back-propagation learning technique is very popular for the multilayer networks. Standard back propagation is a gradient descent algorithm, in which the network weights are moved along the negative of the gradient of the performance function (error). The rule is:

$$\Delta_{ij}^{(t)} = \Delta_{ij}^{(t-1)} - \eta^+ \frac{\partial E(t-1)}{\partial w_{ij}} \quad (1.1)$$

Where, $\Delta_{ij}^{(t)}$ is the current weight, η is the learning rate, $\frac{\partial E(t-1)}{\partial w_{ij}}$ is the previous

gradient of error. Several high-performance algorithms that can converge faster than the standard algorithm are reported in the literature. [7-12]. Narendra *et al.* [11] proposed the Dynamic Back Propagation (DBP) learning algorithm for identification and control employing a multilayer perception. An example of a powerful adaptive speed controller is based on a nonlinear autoregressive moving average (NARMA-L2) algorithm. While determining the direction of the weight update; the magnitude of the partial derivatives of weights and biases play an important role. When these magnitudes become very small, the problems may occur in the convergence. This is remedied by the Resilient Back Propagation (RBP), which uses only sign of the derivative [12]. Further, ANNs are being used widely in not only the control of drives but also for the other issues in like:

- Sensorless operation without speed encoder, [13-15]
- Stator resistance tuning [16]
- Fault diagnosis, [17-18].

In [19], two separate PI controllers are reported for SVM_DTC. ANN based SVM_DTC Control is presented in [20], where the training algorithm is Levenberg-Marquardt (LM). This paper proposes ANN_SVM_DTC with RBP training algorithm.

This paper is organized as follows: section 1 presents Introduction. RBP algorithm is discussed in section 2. Structure of Proposed ANN_SVM_DTC_System is

presented in section 3. Section 4 presents simulation results for transient and steady state conditions, performance comparison between the intelligent controllers.

2 Space Vector Modulated DTC Based on PI Control Algorithm

The induction motor equations in d - q (stator flux) frame are:

$$\begin{aligned} V_{qs} &= p\psi_{qs} + \omega_e\psi_{ds} + r_s i_{qs} \\ V_{ds} &= p\psi_{ds} - \omega_e\psi_{qs} + r_s i_{ds} \end{aligned} \quad (2.1)$$

The electromagnetic torque is given by

$$T_e = \frac{3}{2} \left(\frac{P}{2} \right) (\psi_{ds}^s i_{qs}^s - \psi_{qs}^s i_{ds}^s) \quad (2.2)$$

If stator flux orientation is assumed,

$$\psi_{sq} = 0 ; \quad \psi_s = \psi_{sd} \quad (2.3)$$

$$V_{ds} = r_s i_{ds} + \frac{d\psi_s}{dt} ; \quad V_{qs} = r_s i_{qs} + \omega_{\psi_s} \psi_s \quad (2.4)$$

$$T_e = \frac{3}{2} \left(\frac{P}{2} \right) (\psi_s^s i_{qs}^s) \quad (2.5)$$

Consider Eq.2.1 with a revolving reference frame aligned with stator flux vector. This stator flux orientation makes the q -axis component of the flux, zero. According to Eq.2.4, it is seen that direct-axis component of stator voltage vector has a strong impact on the rate of change of stator flux. The quadrature -axis component of a stator voltage vector, controls the developed torque. This shows that the core operating principle is to use two controllers in the flux and torque control loops. In the torque loop, the control signal is augmented with the speed emf to improve the dynamic response to the torque command. Also, decoupling of the flux and torque control is achieved. The block diagram of induction motor drive based on SVM_DTC is shown in Fig. 1. It represents a class of DTC techniques developed with control system that calculates reference vector of stator voltage, instead of directly indicating the next state of the inverter. Lascu has proposed proportional integral (PI) controller for the stator flux and the torque regulations, but the dynamic response was limited due to the use of the PI controllers.

The drawbacks of PI control algorithms can be eliminated replacing the PI control algorithms using modern control techniques like Fuzzy, Sliding Mode Control, Hybrid Control. Studies regarding these are carried out in [19-21].

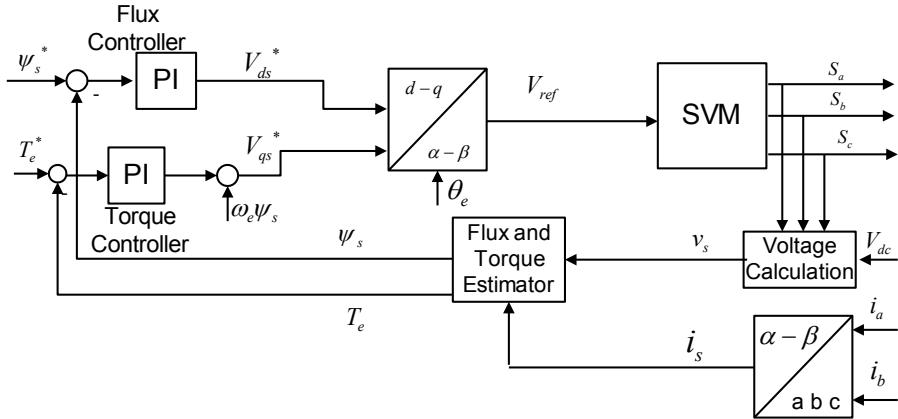


Fig. 1. SVM_DTC based on PI Control Algorithm

3 The Resilient Back Propagation (RBP) Algorithm

Sigmoid functions used in hidden layers have almost zero slope when the input gets large. This gives very small magnitude of partial derivatives of weights and biases and therefore, causes small changes in the updated weights and biases, even when the weights and biases are far from their optimal values. The RBP training algorithm considers only the sign of the derivative and thus eliminates harmful effects of the magnitudes of the partial derivatives [12].

The size of the weight change is determined by a separate update value $\Delta_{ij}^{(t)}$. This update value depends on error function E , and given as

$$\begin{aligned}\Delta_{ij}^{(t)} &= \eta^+ \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial E(t-1)}{\partial w_{ij}} * \frac{\partial E(t)}{\partial w_{ij}} > 0 \\ &= \eta^- \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial E(t-1)}{\partial w_{ij}} * \frac{\partial E(t)}{\partial w_{ij}} < 0 \\ &= \Delta_{ij}^{(t-1)}, & \text{else}\end{aligned}\quad (3.1)$$

Where, $0 < \eta^- < 1 < \eta^+$

The update value for each weight and bias is increased by a factor, whenever the derivative of the performance function has the same sign for two successive iterations. The update value is decreased by a factor, whenever the derivative changes sign from the previous iteration. If the derivative is zero, then the update value remains the same. Once the update-value for each weight is adapted, the weight-update itself given as,

$$\begin{aligned}
 \Delta w_{ij}^{(t)} &= -\Delta_{ij}^{(t)}, & \text{if } \frac{\partial E(t)}{\partial w_{ij}} > 0 \\
 &= +\Delta_{ij}^{(t)}, & \text{if } \frac{\partial E(t)}{\partial w_{ij}} < 0 \\
 &= 0, & \text{otherwise.}
 \end{aligned} \tag{3.2}$$

Whenever the weights are oscillating, the weight change will be reduced. If the weight continues to change in the same direction for several iterations, then the magnitude of the weight change will be increased.

4 Structure of Proposed ANN_SVM_DTC_System

The drive block diagram is shown in Fig.2. It operates with constant rotor flux, direct stator flux, and torque control. The speed controller is a classical PI regulator which produces the reference torque. Only the dc-link voltage and two line currents are measured. The algorithm operates with two ANN Controllers for decoupled flux and torque control. Each of the ANN has input layer consisting of two neurons representing respective error and the time derivative of the error. The structure has nine neurons in the hidden layer. Both the controllers produce a stator voltage vector component, which, forms control voltage vector in rectangular form. This is further synthesized by SVM unit and applied to IM through VSI.

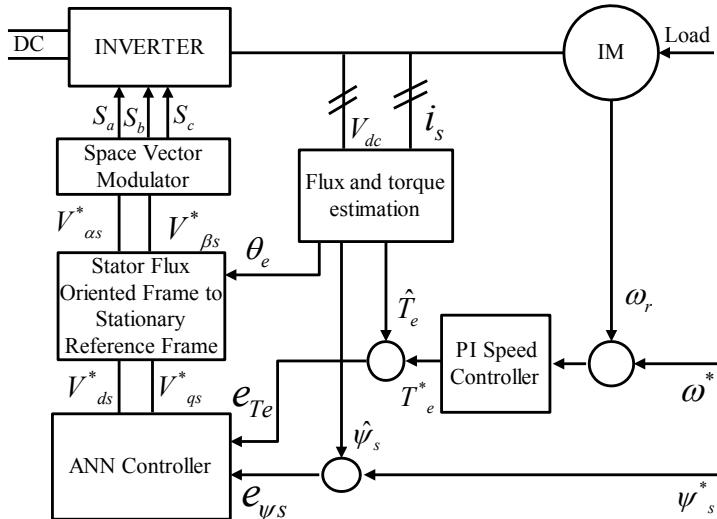


Fig. 2. ANN based SVM_DTC scheme

In this case, ANN uses three inputs, and two outputs and three hidden layers. The training of ANN is done using back propagation algorithm. Training data is obtained from PI controller based SVM_DTC.

5 Simulation Results

The ANN_SVM_DTC scheme is simulated for 3-phase induction motor on MATLAB-SIMULINK platform. The results are compared with other two intelligent control performances reported in [22]. Fig. 3 to Fig. 9 show the comparison of the results for three Intelligent Control Algorithms. Parameters of motor are : 3-phase, 3 kW, 50Hz, 4- poles, $R_s = 7.83 \Omega$, $R_r = 7.55 \Omega$, 440 V, 4.9 A, $L_s = L_r = 0.4751 \text{ H}$, $L_m=0.4535 \text{ H}$, $J=0.013 \text{ Kgm}^2$.

ANN is trained using RBP training algorithm and tested for varying conditions of torque, stator resistance, and disturbance rejection. The speed reference for all these conditions is 382 rpm. Training performance is indicated in Table 1.

Table 1. Training Performance of Proposed System using RBP Algorithm

| Epoch | Time in sec | performance | gradient | Validation checks |
|-------|-------------|-------------|----------|-------------------|
| 160 | 8 | 305 | 231 | 6 |

The transient performance for no load is shown in Fig. 3. ANN takes less settling time (0.042 sec) as compared to Fuzzy (0.048 sec) and Fuzzy_SMC (0.06 sec). Also it gives smoother steady state response. Integral Time Absolute Error (ITAE) is used as performance index for the steady state error. Fig. 4 shows that ANN_SVM_DTC has the least ITAE for no load. When full load of 12 Nm is applied at 0.5 sec., ANN has a dip in speed response, for a short time interval (5 msec.) but afterwards, smooth and steady state response shown in Fig. 5. It may be noted that the control effort (stator current) is also optimum in ANN_SVM_DTC. Further, it has reasonable ITAE for load as shown in Fig. 6. For testing the robustness of the proposed control algorithm, the stator resistance is changed by 150 % of the nominal value. It is seen that proposed ANN gives robust performance for speed (Fig.7) and ITAE (Fig.8), respectively. Similarly, disturbance of $0.001 \sin(100t)$ added to the input channel to observe the disturbance rejection for speed and ITAE (Fig.9. and Fig. 10, respectively).

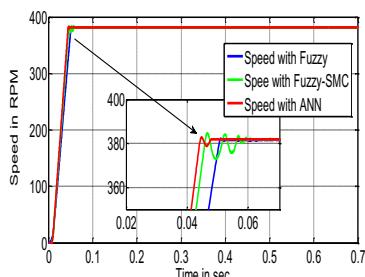


Fig. 3. No load transients for the three intelligent controllers

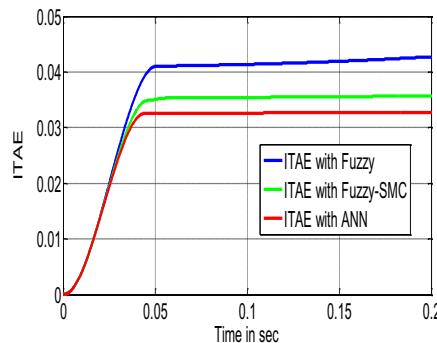


Fig. 4. ITAE for the three intelligent controllers in no load condition

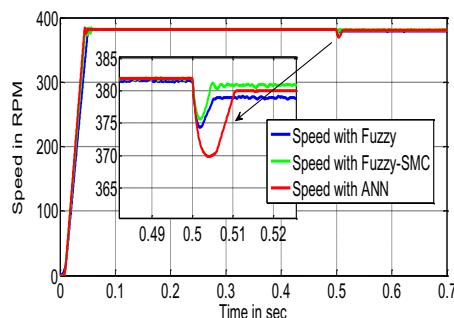


Fig. 5. Performance of three intelligent controllers after application of load at 0.5 sec

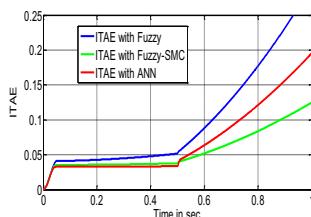


Fig. 6. ITAE for the three intelligent controllers after application of load at 0.5 sec

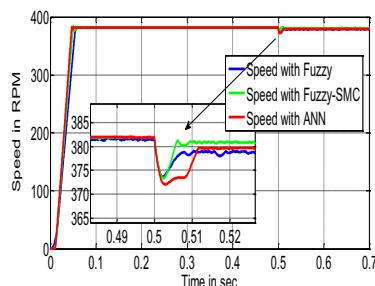


Fig. 7. Performance of three intelligent controllers with stator resistance 150%

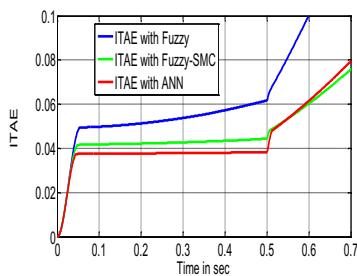


Fig. 8. ITAE of three intelligent controllers with stator resistance 150%

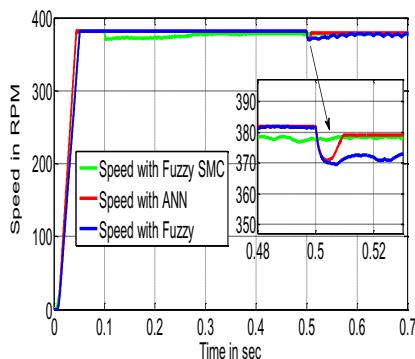


Fig. 9. Performance of three intelligent controllers with disturbance of $0.001\sin(100t)$

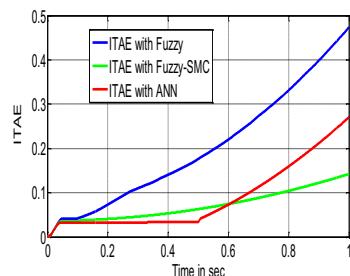


Fig. 10. ITAE of three intelligent controllers with disturbance of $0.001\sin(100t)$

Table 2. Performance Comparison based on Control Effort (Stator Current)

| | Stator current Norm | Total Harmonic Distortion for Stator Current |
|-----------|---------------------|--|
| Fuzzy | 282.97 | 120.44 |
| Fuzzy_SMC | 280.14 | 63.22 |
| ANN | 279.85 | 54.8 |

6 Conclusion

In this paper, a neural network-based SVM_DTC for induction motor is proposed. The neural network method adopted in the DTC system simplifies the control system and improves the system reliability. It is proved that Resilient Back Propagation training method can accurately model an induction motor. It offers superior performance as compared to other two intelligent controllers, Fuzzy and Fuzzy_SMC. Performance comparison between these three algorithms clearly indicate that the transient response with ANN based Control is comparable with other counterparts. Further, smooth steady state behavior is achieved with ANN based Control. Less control efforts, better current harmonics, robustness against parameter variations and disturbances are added advantages with the proposed ANN Control algorithm. Simulation results demonstrate that the ANN based Intelligent IM drives will gain wider acceptance in the future.

References

1. Takahashi, Noguchi, T.: A new quick response and high efficiency control strategy of an induction motor. *IEEE Transactions on Industrial Applications* IA-22(5), 820–827 (1986)
2. Zhu, P., Kang, Y., Chen, J.: Improved Direct Torque Control Performance of Induction Motor with Duty Ratio Modulation. In: *IEEE International Electric Machines and Drives Conference*, vol. 2, pp. 994–998 (2003)
3. Lee, S.-B., Song, J.-H., Choy, L., Yoo, I.-Y.: Torque ripple reduction in DTC of induction motor driven by three-level inverter with low switching frequency. *IEEE Transactions on Power Electronics* 17(2), 255–264 (2002)
4. Beerten, J., Verveckken, J., Driesen, J.: Predictive Direct Torque Control for Flux and Torque Ripple Reduction. *IEEE Transactions on Industrial Electronics* 57(1), 404–412 (2010)
5. Habetler, T.G., Profumo, F., Pastorelli, M., Tolbert, L.M.: Direct torque control of induction motor using space vector modulation. *IEEE Transactions on Industrial Electronics* 28(5), 1045–1053 (1992)
6. Harashima, F.: Power Electronics and Motion Control - A Future Perspective. *Proceedings of the IEEE* 82(8), 1107–1111 (1994)
7. Awwad, A., Abu-Rub, H., Toliyat, H.: Nonlinear Autoregressive Moving Average (Narma- L2) Controller for Advanced AC Motor Control. In: *34rd Annual Conference of the IEEE Industrial Electronics Society, IECON 2008*, Orlando, Florida (2008)
8. Bose, B.K.: Artificial Neural Network Applications in Power Electronics. *IEEE Transactions*, 1631–1638 (2001)
9. Demuth, H., Beale, M., Hagan, M.: Neural network toolbox user's guide for use with MATLAB. The Math Works, Inc., Natick (2006)
10. Abbou, M., Akherraz, A.: Real-time DSP implementation of DTC neural network-based control for induction motor drive. In: *5th IET International Conference on Power Electronics, Machines and Drives*, vol. 6, pp. 1–5 (2010)
11. Narendra, K.S., Mukhopadhyay, S.: Adaptive Control Using Neural Networks and Approximate Models. *IEEE Transactions on Neural Networks* 8, 475–485 (1997)

12. Riedmiller, M., Braun, H.: A direct adaptive method for faster back propagation learning: the RPROP algorithm. In: IEEE International Conference on Neural Networks, vol. 1, pp. 586–591 (1993)
13. Lascu, C., Boldea, I., Blaabjerg, F.: A Modified Direct Torque Control for Induction Motor Sensorless Drive. *IEEE Tran. on Ind. Appl.* 36(1), 122–130 (2000)
14. Vas, P.: Artificial-Intelligent-Based Electrical Machines and Drives. In: Application of Fuzzy, Neural, Fuzzy-Neural and Genetic-Algorithm-Based Techniques. Oxford Univ. Press, Oxford (1999)
15. Fodor, D., Ionescu, F., Floricau, D., Six, J.P., Delarue, P., Diana, D.: Neural Networks Applied for Induction Motor Speed Sensorless Estimation, vol. 1, pp. 181–186 (1995)
16. Luis, I.H., Cabrera, A., Elbuluk, M.E.: Tuning the stator resistance of induction motors using artificial neural network. *IEEE International on Power Electronics* 12(5), 779–787 (1997)
17. Kola, S., Varatharas, L.: Identifying 3 phase IM faults using neural networks. *ISA Transactions* (39), 433–439 (2000)
18. Tag Eldin, E.M., Emara, H.R., Aboul-Zahab, E.M., Refaat, S.S.: Monitoring and Diagnosis of External faults in Three Phase Induction Motor using Artificial Neural Network. In: IEEE Power Engineering Society General Meeting (2007)
19. Lascu, C., Boldea, Blaabjerg, F.: A modified direct torque control for induction motor sensorless drive. *IEEE Transactions on Industry Applications* 36, 122–130 (2000)
20. Ha, Q.P., Nguyen, Q.H., Rye, D.C., Durrant-Whyte, H.F.: Fuzzy sliding mode controllers with applications. *IEEE Trans. on Ind. Electron.* 48(1), 38–46 (2001)
21. Agamy, M.S., Yousef, H.A., Sebakhy, O.A.: Adaptive Fuzzy Variable Structure Control of Induction Motors. In: IEEE Canadian Conference on Electrical and Computer Engineering, pp. 89–94 (2004)
22. Jadhav, S.V., Chaudhari, B.N., Kumar, K.: Direct Torque Control of Induction Motor using Artificial Neural Network. In: PEDES 2012 (2012)

Thinking Machines versus Thinking Organisms

Petro Gopych

Universal Power Systems USA-Ukraine LLC, 3 Kotsarskaya Street, Kharkiv 61012 Ukraine
pmgopych@gmail.com, pmg@kharkov.com

Abstract. The recent hypothesis of concurrent infinity and a phenomenology formalization based on it allow us to mathematically define meaning, subjectivity and super-Turing machines that implement them. Using these results, in this paper, at mathematical level of rigor, the phenomena of life and thinking are defined and perspectives of their artificial reproduction are considered. It is demonstrated that machines built of inorganic raw materials can never think. Natural and artificial living organisms built of organic raw materials are the physically implemented partial universal super-Turing machines (tied collections of many super-Turing machines) that can think. The necessary and sufficient “Mowgli’s test” for thinking, solving simultaneously the problem of other minds, is proposed and discussed. It is concluded that cyborgs, the products of evolutionary symbiosis of humans and human artifacts, seem to be our evolutionary perspective.

Keywords: Turing’s test, concurrent infinity, subjectivity, artificial life, human-level thinking, super-Turing machines.

1 Introduction: The Turing’s Test for “Thinking”

Allan Turing answers the question, “Can machines think?” assuming the word “think” is “too meaningless to deserve discussion”. To evade inevitable considering the effects of subjectivity, he replaces this question by another one, “Are there imaginable digital computers which would do well in the imitation game?” [1]. Turing also presupposes the machine’s unemotional (typewritten) communication channels and the ban on practical demonstrations. As a result, the *informal* subjective notion of “thinking machines” is replaced by the restricted *formal* (independent on meaning/feeling/subjectivity) notion of “digital computers which would do well in the imitation game”. This is probably the reason why Turing calls the game he invented “our criterion for ‘thinking’”, with “thinking” in quotes. Hence, Turing’s test is the criterion for “thinking” (with quotes) but not for genuine thinking (without quotes).

Turing machines (TMs) computer everything that is computable but simulate a fraction of the mind only. Consequently, to simulate the whole mind, something *super-Turing* is required. To cope with paradoxes of mimicking the mind by TMs, we will appeal to the hypothesis of concurrent infinity and mathematics of meaningful computations [2] arising from this hypothesis and spreading beyond Zermelo-Fraenkel (ZF) mathematics. In the present paper, within this new framework, the notions of life and thinking will mathematically be defined and their complete implementations by super-Turing machines (STMs) will be considered. It will be

demonstrated that machines built of inorganic raw materials cannot think, whereas natural and artificial organisms built of organic raw materials can. The necessary and sufficient “Mowgli’s test” for thinking will be proposed and discussed.

2 Concurrent Infinity, Phenomenology, and the BSDT PL

The hypothesis of concurrent infinity adds to the ZF axiomatic the idea of evolution by postulating the infinity of the common “in the past” and open-ended “in the future” co-evolution of the universe, life, mind, language, and society. It is implied the *meaning* of a symbolic (binary for certainty) i -bit message/string/vector/name x_j^i is given by an *infinite on a semi-axis* binary string $c_{xi}x_j^i$ where the c_{xi} is an infinite on a semi-axis context in which the x_j^i appears. All one-way infinite strings $c_{xi}x_j^i$ have a *common fundamentally unspecified* one-way infinite beginning; any x_j^i taken in isolation is meaningless. Vectors x_j^i , $x_j^i \in S_{xi}$, are assumed to have the values of ± 1 of their spin-like components; S_{xi} is an i -dimensional binary space. If $x_j^i = u_r^p v_s^q$, then $c_{xi}x_j^i = c_{xi}u_r^p v_s^q$; the *focal*, v_s^q , and *fringe*, u_r^p , constituents of the x_j^i have *definite* and *conditional* meanings, respectively. If a composite vector x_j^i consist of more than two constituents then the right-most (focal) one takes the definite meaning; other its constituents (fringes) have conditional meanings only. The totality of infinite, with an infinite common beginning, strings $c_{xi}x_j^i$ constitutes an ultimate/proper class S_{cx0} – the set that cannot be a subset of any other set. Strings $c_{xi}x_j^i \in S_{cx0}$ and their common infinite beginning have the lengths of \aleph_0 bits; the number of these strings is infinite but countable, \aleph_0 ; their affixes x_j^i name/enumerate all the things of the world, known as well as unknown but conceivable. \aleph_0 is Cantor’s aleph naught. Resulting mathematics of one-way infinite meaningful strings arranged to have the common one-way infinite beginning is called *meaningful/semantic mathematics* or *primary language* (PL). If this special arrangement of meaningful strings is destroyed (meanings are ignored) then the PL transforms immediately into ZF mathematics.

Accepting the concurrent infinity leads to a *phenomenology formalization*. It means the postulated *equivalence* between such usually incommensurable things as a *one-way infinite binary string* $c_{xi}x_j^i$, *meaning* of the x_j^i , the *real-world physical device* devoted to recognize it and *feeling* (“quale” or primary thought or psychological state) it causes. The x_j^i is also the name of the device devoted to recognize it and the name of the thing of the world in which this name/message originates [2].

The best technique for PL computations gives the binary signal detection theory, BSDT [3], dealing with finite i -bit vectors x_j^i damaged by replacing binary noise [4]. The best rules for decoding these vectors can be presented as BSDT abstract selectional machines or ASMs [5] that are the BSDT’s universal neural network computational units [6]. For this reason, we refer to the PL as BSDT PL. Practical neural network BSDT PL computations are discussed in ref. 2.

3 Meaningful Communication between the Mirror STMs

For any *finite* set of meaningful infinite messages, their common one-way infinite beginning could be fixed and, then, subtracted to reduce them to a set of meaningful *finite* messages, amenable to be processed by TMs. Thus, we can safely claim that

genuine super-Turing computations can be reduced, without any reservations, to regular Turing computations. It is only needed to be sure the common infinite beginning of the considered one-way infinite messages is correctly subtracted.

Let us suppose in a communication process a transmitter and a receiver are the *mirror replicas* of each other, i.e., they were designed, implemented in a physical form and learned beforehand to perform the same meaningful function – selecting the same x_j^i given its same context, c_{xi} . If it is, then the transmitter and the receiver are completely described by the same string $c_{xi}x_j^i$ and encode/decode/reproduce or *understand* the meaning of the x_j^i in the same way. Infinite part of the $c_{xi}x_j^i$, c_{xi} , describes the real-world implementation (“hardware”) of identical devices devoted to recognize the x_j^i while the x_j^i itself specifies their identical “software”. It means the $c_{xi}x_j^i$ could *physically* be split into its finite, x_j^i , and infinite, c_{xi} , parts to ensure the symbolic processing of the x_j^i by a physical implementation of the c_{xi} learned to recognize the x_j^i . Consequently, to correctly process/understand the infinite $c_{xi}x_j^i$, it is enough to use the receiver and the transmitter that are the mirror copies of each other and to correctly transmit, receive, and decode/reproduce the finite x_j^i only [2].

The scheme presented is supported by the study in neuroscience of *mirror neurons* – the ones that are active when an animal/human behaves or only *observes* behaviors of others, e.g., [7]. The mirror-ASM computational system just described and the mirror-neuron circuitries observed in animals/humans could respectively be treated as theoretical and real-brain implementations of until now hypothetical STMs [8] dealing with one-way infinite binary inputs or real-valued numbers in other words.

4 Life and Its Instruments for Thinking, STMs and PUSTMs

4.1 Living Organisms, Their Genomic and Genetic Codes

Let us define a living *organism* as a thing of the world that is distinct from its permanently changing *environment* (all the world’s other things) and changes itself to self-sustain and self-reproduce itself in this environment (in the best way). The phenomenon of *life* is produced by all such organisms taken together.

The animate or inanimate real-world thing Y_l^k is described by an infinite on a semi-axis meaningful string $c_{yk}y_l^k$ where its affix y_l^k is the finite k -bits-in-length name of this string/thing and c_{yk} is its one-way infinite context:

$$Y_l^k = c_{yk}y_l^k = E_{ij} c_{xi}(y_l^k)x_j^i = E_{ij} X_j^i(y_l^k) \quad (1)$$

where strings $c_{yk}y_l^k \in S_{cx0}$ and $c_{xi}(y_l^k)x_j^i \in S_{cx0}$ have common fundamentally unspecified one-way infinite beginning and the same infinite length, $l(c_{yk}y_l^k) = l(c_{xi}(y_l^k)x_j^i)$, or *meaning (evolutionary) complexity* [2]. The inherently tied collection of things $X_j^i(y_l^k)$ or their infinite descriptions $c_{xi}(y_l^k)x_j^i$ are *embedded* into the thing Y_l^k or its infinite description $c_{yk}y_l^k$ in a way that cannot finitely be formalized. Such an *embedding* is designated by the sign E_{ij} whose indices i and j are from their infinitely wide in theory and finitely wide in practice ranges. The meaning of the E_{ij} , introduced to refer to the BSDT-PL-specific *operation of embedding*, is given by the string $c_{yk}y_l^k$ or the thing Y_l^k (see also Section 4.2). The possibility to combine in (1) a real-world thing Y_l^k with its

infinite description $c_{yk}y_l^k$, its real-world parts/properties $X_j^i(y_l^k)$ and their infinite descriptions $c_{xi}(y_l^k)x_j^i$ follows from our phenomenology formalization. In (1) and throughout this paper, upper-case-letter variables designate the real-world things, whereas same lower-case-letter variables designate their binary names/descriptions.

Given its context c_{yk} , the name y_l^k of the Y_l^k gives the formalized symbolic information needed to distinguish this thing from its environment. Consequently, the $c_{yk}y_l^k$ describes the Y_l^k and, indirectly, its place in its environment. If, given the c_{yk} , the information the y_l^k contains suffices to change the Y_l^k to self-sustain and self-reproduce itself then we call the Y_l^k a *living organism* and refer to its name y_l^k as its *genomic code*. The Y_l^k consists of its parts $X_j^i(y_l^k)$ corresponding to parts $AX_j^i(y_l^k)$ of its *ancestor*, AY_l^k . Genomic codes of things $X_j^i(y_l^k) = c_{xi}(y_l^k)x_j^i$, x_j^i , are *major genetic codes* of the genomic code of the Y_l^k , y_l^k . These codes are “*major*” because they are of the same meaning complexity as the y_l^k , have definite meanings simultaneously with the y_l^k and specify the properties of the Y_l^k that distinguish it from its *nearest ancestor* AY_l^k whose major properties are $AX_j^i(y_l^k)$; genetic codes x_j^i are combined to form the genomic code y_l^k in a way (1) that cannot finitely be specified. In (1), an organism or its *phenotype* Y_l^k , its given the context c_{yk} genomic code y_l^k , its *genotype* $c_{yk}y_l^k$ or its real-world *genome*, its real-world parts/properties or *phenotypic traits* $X_j^i(y_l^k)$, its given the *klith context* $c_{xi}(y_l^k)$ genetic codes x_j^i and its *genotypic traits* $c_{xi}(y_l^k)x_j^i$ or its *ijth real-world genes* are combined. Resulting *phenotype-genotype unity and disunity* are the consequence of the BSDT PL phenomenology formalization.

We call the AY_l^k – an ancestor of the Y_l^k – a *suborganism* of the organism Y_l^k . The Y_l^k is in turn a *superorganism* of the organism AY_l^k . In a way that remains unspecified until the organism’s physical entity is out of the consideration, evolution transforms the major part/property of the AY_l^k , $AX_j^i(y_l^k)$, into the major part/property of the Y_l^k , $X_j^i(y_l^k)$; this formally means the infinite description of the $AX_j^i(y_l^k)$, $c_{xi}(a, y_l^k)x_j^i$, becomes at least 1 bit longer and becomes the infinite description of the $X_j^i(y_l^k)$, $c_{xi}(y_l^k)x_j^i$. Evolution makes $l(c_{xi}(y_l^k)x_j^i) > l(c_{xi}(a, y_l^k)x_j^i)$ but *keeps intact* one-way infinite beginning of the $c_{xi}(y_l^k)x_j^i$, $c_{xi}(a, y_l^k)x_j^i$, defining the major property $AX_j^i(y_l^k)$ of the AY_l^k (in $c_{xi}(y_l^k)x_j^i$ and $c_{xi}(a, y_l^k)x_j^i$, affixes x_j^i may not coincide). As a result, the $X_j^i(y_l^k) = c_{xi}(y_l^k)x_j^i$ becomes the *ijth major property* of the Y_l^k keeping the *ijth major property* of the AY_l^k , $AX_j^i(y_l^k) = c_{xi}(a, y_l^k)x_j^i$, as a *minor part/property* of the $X_j^i(y_l^k)$ and, consequently, of the Y_l^k . Meaningful strings giving an organism’s minor properties describe the major properties of the organism’s ancestors. Minor properties’ meanings are conditioned and resulting meaning uncertainty is a manifestation of the famous in mathematics Burali-Forti antinomy. Names of major and minor properties of the Y_l^k are its focal and fringe names, respectively [2].

4.2 Living Organisms as PUSTMs with Embedded STMs

Infinite strings giving the meanings to an organism’s properties are, simultaneously, the descriptions of STMs devoted to process these strings [2]. As any organism has finitely many major properties (remember the finite sizes of known genomic codes), it should be thought of as a *finite*, inherently tied collection of STMs serving these properties, heritable or hereditarily predisposed to acquire them in the course of the organism’s learning or/and development (one STM per a property or meaningful name of this property). We refer to such an STM with a set of *embedded* STMs a

partial universal STM or a PUSTM. The *true universal STM* that would be able to process *all* possible meaningful strings is *impossible* (remember the total number of such strings is infinite and each of them is processed by a separate physically implemented STM). Relationships between an organism $Y_l^k = c_{yk}y_l^k$, the PUSTM(y_l^k) that represents/implements it, and $\text{STM}(y_l^k, x_j^i)$ representing/implementing the major parts $X_j^i(y_l^k) = c_{xi}(y_l^k)x_j^i$ of the Y_l^k and constituting the PUSTM(y_l^k) are described as

$$Y_l^k = \text{PUSTM}(y_l^k) = E_{ij} \text{STM}(y_l^k, x_j^i). \quad (2)$$

Equations 1 and 2 are equivalent if they concern living organisms.

For any S_{cx0} , there exists an *embedded* proper class S_{cxl} comprising meaningful strings that are I bits shorter ($I > 0$) than those constituting the S_{cx0} ; the I th S_{cxl} is, simultaneously, not a part of the S_{cx0} because their members are conditionally related. It means the PUSTM serving the members of the S_{cx0} includes an *in-theory-infinite hierarchy* of embedded PUSTMs serving I -bit shorter meaningful strings. In other words, any organism $Y_l^k = Y_l^k(I)$ having the major properties described by members of the S_{cxl} with $I = 0$ also has the minor properties defined by the members of classes S_{cxl} with $I = 1, 2, 3, \dots$. Consequently, the number of any organism's properties (major and minor ones taken together) should be *infinite*.

For the organism Y_l^k defined by members of an S_{cx0} , the I th S_{cxl} embedded in the S_{cx0} defines the I th *suborganism* embedded in the Y_l^k or the I th *ancestor* of the Y_l^k , $AY_l^k(I)$ with $I > 0$. The sequence of this organism's embedded suborganisms/ancestors could only be restricted if there exists such an I_{\max} that specifies the deepest or the *last* embedded ancestor $AY_l^k(I_{\max})$ that could be understood as a living organism or a PUSTM. If such a restriction exists, it could only be substantiated by empirical reasons. Studies of the terrestrial life's LUCA (last universal common ancestor, $AY_l^k(I_{\max})$ in our designations), e.g., [9] provide such reasons because, in the BSDT PL, the I_{\max} defines the maximal *evolutionary distance* (maximally possible number of evolutionary transformations and, consequently, ancestors) between the LUCA, $AY_l^k(I_{\max})$, and the evolutionary most developed (specified by $I = 0$) organism, Y_l^k .

As is now known [10, 11], LUCA was most probably an enormous global "mega-organism" of the scope of earth oceans consisting of different in space and time populations of primitive evolving pre-cells with intense lateral gene transfers. LUCA is unable to self-reproduce itself as a whole, is not an individual organism, and values $I \sim I_{\max}$ it defines give a blurred *boundary* between the cellular life and its organic precursors. Consequently, minor properties of an organism Y_l^k described by members of classes S_{cxl} with $0 < I < I_{\max}$ are, simultaneously, major properties of its extant and extinct cellular ancestors (these S_{cxl} describe respective PUSTMs), whereas the members of classes S_{cxl} with $I > I_{\max}$ describe minor properties of the Y_l^k concerning its cells' organic precursors and inorganic things these precursors built of (these S_{cxl} do not produce PUSTMs). The more the I and the further it from the I_{\max} the smaller the fraction of organic properties among the minor properties of the organism is, though the size of this fraction is unspecified. Strings with $I >> I_{\max}$ describe, as minor properties of an organism, even deep physical properties of the universe, from atoms and subatomic particles to stars, galaxies, etc.

5 Life Forms, the Milieu for Thinking

Similar to (1) BSDT PL semantic equations

$$EU_l^k = c_{yk}(eu)y_l^k = E_{ij}c_{xi}(eu_l^k)x_j^i = E_{ij}X_j^i(eu_l^k), \quad (3)$$

$$PL_l^k = c_{yk}(pl)y_l^k = E_{ij}c_{xi}(eu_l^k)c_{xi}(de_l^k)x_j^i = E_{ij}c_{xi}(pl_l^k)x_j^i = E_{ij}X_j^i(pl_l^k), \quad (4)$$

$$AN_l^k = c_{yk}(an)y_l^k = E_{ij}c_{xi}(pl_l^k)c_{xi}(so_l^k)x_j^i = E_{ij}c_{xi}(an_l^k)x_j^i = E_{ij}X_j^i(an_l^k), \quad (5)$$

$$GR_l^k = c_{yk}(gr)y_l^k = E_{ij}c_{xi}(an_l^k)c_{xi}(gr_l^k)x_j^i = E_{ij}c_{xi}(gr_l^k)x_j^i = E_{ij}X_j^i(gr_l^k) \quad (6)$$

formalize the definitions of organisms representing some forms of life, namely the unicellular organism EU_l^k (e.g., a eukaryote), multicellular organism PL_l^k (e.g., a plant built of the cells EU_l^k), social multicellular organism AN_l^k (e.g., a social animal having the body PL_l^k) and social group GR_l^k (e.g., a pack of animals AN_l^k). In (3) the concatenation $c_{xi}(eu_l^k)x_j^i$ describes (the meaning of) the ij th major unicellular property of the EU_l^k , e.g., the ij th cellular organelle $X_j^i(eu_l^k)$; the i -bit string x_j^i gives this property's name while the string $c_{xi}(eu_l^k)$ is an infinite evolutionary story of its design (eu after “eukaryote”). Equation 3 gives the description of the EU_l^k as a collection of its major unicellular properties $X_j^i(eu_l^k)$ and, simultaneously, its real-world implementation (remember BSDT PL phenomenology formalization). In (4) the concatenation $c_{xi}(eu_l^k)c_{xi}(de_l^k)x_j^i$ describes the ij th major multicellular property of the PL_l^k , e.g., a plant’s part $X_j^i(pl_l^k)$ named x_j^i ; the string $c_{xi}(eu_l^k)$ is an infinite evolutionary story of designing the cells EU_l^k constituting the $X_j^i(pl_l^k)$ while a string $c_{xi}(de_l^k)$ gives the finite story of evolutionary design of this part or an “instruction” for building the $X_j^i(pl_l^k)$ from cells EU_l^k in the course of the development of the PL_l^k (de after “development”). Equation 4 gives the description of the PL_l^k as a collection of its major multicellular or body properties $X_j^i(pl_l^k)$ and, simultaneously, its real-world implementation. In (5) the concatenation $c_{xi}(pl_l^k)c_{xi}(so_l^k)x_j^i$ describes the ij th major social property of the AN_l^k , e.g., the ij th courtship behavior $X_j^i(pl_l^k)$ named x_j^i ; the string $c_{xi}(pl_l^k)$ is an infinite evolutionary story of designing the body PL_l^k of the AN_l^k needed to implement its social property $X_j^i(an_l^k)$ while a string $c_{xi}(so_l^k)$ describes the finite evolutionary story of implementing the $X_j^i(an_l^k)$ or an instruction for executing it by this animal’s body (so after “social”). Equation 5 gives the description of the AN_l^k having the body PL_l^k as a collection of its major social properties $X_j^i(an_l^k)$ and, simultaneously, its real-world implementation. In (6) the concatenation $c_{xi}(an_l^k)c_{xi}(gr_l^k)x_j^i$ describes the ij th major social group property of the GR_l^k , e.g., the ij th social group aggressive behavior $X_j^i(gr_l^k)$ named x_j^i ; the string $c_{xi}(an_l^k)$ is an infinite evolutionary story of designing the social animals AN_l^k constituting the social group GR_l^k while a string $c_{xi}(gr_l^k)$ is the finite story of evolutionary design of the social group property $X_j^i(gr_l^k)$ or an instruction for executing this property by this social group (gr , after “group”). Equation 6 gives the description of the social group GR_l^k as a collection of its major social group properties $X_j^i(gr_l^k)$ and, simultaneously, its real-world implementation. In (3) to (6), strings $c_{yk}(eu)y_l^k$, $c_{yk}(pl)y_l^k$, $c_{yk}(an)y_l^k$, and $c_{yk}(gr)y_l^k$ describe genomes of unicellular eukaryotes, multicellular organisms, animals and animal groups, respectively (substrings y_l^k are their genomic codes).

Life forms' meaningful descriptions (3) to (6) have above been obtained assuming *two empirical restrictions*: the finiteness of any organism's genomic and genetic codes and the existence of LUCA. Resulting phenotype/genotype relationships rigidly fix but do *not* specify the physical entities of respective organisms. It gives the advantage of describing them regardless of their possible real-world implementations.

6 Thinking, the Faculty of Life

We differ perceptual/primary and abstract/secondary thinking [12].

Perceptual thinking is understood as meaningful (super-Turing) computations performed by a PUSTM/organism over a one-way infinite string or some such strings. Any PUSTM could in turn be presented as a hierarchy of PUSTMs/organisms serving meaningful strings related to proper classes $S_{c,I}$ with $I = 0, 1, 2, \dots, I_{\max}$. In theory, each perceptual thought could include deeper-level thoughts related to the embedded PUSTMs processing meaningful strings with $I \leq I_{\max}$ [12]. In practice, the number of levels of thinking is restricted by a PUSTM/organism's *logical or reasoning deepness* $I_d \ll I_{\max}$ defined by the organism's "anatomical" properties; in humans, $I_d \leq 3$ to 5 [13]. At the same time, any PUSTM/organism contains as its part *traditional TMs* devoted to process binary names of meaningful strings involved in its perceptual thinking [2]. Algorithms of these TM computations are *fixed* and rigidly predisposed. Hence, TMs participating in perceptual thinking of any organism (such thinking is its intrinsic property) are preprogrammed and run always in the same way.

Abstract thinking is understood as multiple perceptual thoughts served by an organism/PUSTM and coordinated by a *flexible* algorithm of TM computations over binary names of current perceptual thoughts *and* binary names of memories for previous perceptual/abstract thoughts. This gives abstractly thinking organisms (e.g., humans) the benefit of being able to generate the delay, socially relevant and flexible behavioral responses. The use of neural network BSDT PL computations for solving such problems is described in [2, 6, 14]. The symbolic fraction of abstract thoughts (it is that is "thinking" in terms of Turing) could be modeled by TMs and estimated by the test of Turing. Simultaneously, flexible TM computations ignoring the thought's perceptual components cannot completely mimic the thinking.

7 Artifacts, Machines, and Natural/Artificial Living Organisms

All real-world *inorganic* things $TH_j^i = c_{xi}(th)x_j^i$ (1) are at the same level of evolutionary development and, for this reason, of the same meaning (evolutionary) complexity. Things TH_j^i (that is present in the world) are *raw materials* for designing a something new (that is absent from the world). The products of such an intentional human activity are called *artifacts*; the activity itself is called the *artifact technology*. Artifacts designed to enhance human body functions are called *machines*. An axe enhancing the arm/hand or a computer enhancing the brain/nerve system are machines. A technology needed to design machines is called the *machine technology*. Following (3) to (6), the machine MA_i^k built of the inorganic things $TH_j^i(ma_i^k)$ can be presented as

$$MA_l^k = c_{yk}(ma)y_l^k = c_{yk}(rm.ma)c_{yk}(kh.ma)c_{yk}(in.ma)y_l^k = E_{ij} TH_j^i(ma_l^k) \quad (7)$$

where y_l^k is the machine's *specification* or its k -bits-in-length name comprising, given the context $c_{yk}(ma)$, the knowledge of it; $c_{yk}(ma)$ is the technology of building the MA_l^k with properties y_l^k written as a concatenation $c_{yk}(rm.ma)c_{yk}(kh.ma)c_{yk}(in.ma)$. In it $c_{yk}(rm.ma)$ gives a one-way infinite evolutionary story of designing the inorganic raw materials $TH_j^i(ma_l^k)$; $c_{yk}(kh.ma)$ is a finite description of *know-hows* needed to design the machine (actions humans *learn by mimicking the actions of the skilled teaches*) and $c_{yk}(in.ma)$ is a finite formal *instruction* how to manufacture the MA_l^k knowing the know-hows. In (7), the meaning of the embedding E_{ij} is the technology of constructing the MA_l^k from things $TH_j^i(ma_l^k)$ or, in other words, the MA_l^k itself.

As (7) demonstrates, any machine has the meaning complexity of its raw materials. For example, the axe $AXE_l^k = c_{yk}(rm.axe)c_{yk}(kh.axe)c_{yk}(in.axe)y_l^k$ and the computer $COM_l^k = c_{yk}(rm.com)c_{yk}(kh.com)c_{yk}(in.com)y_l^k$ have equal meaning complexities that are equivalent to equal meaning complexities of inorganic things $TH_j^i(axe_l^k)$ and $TH_j^i(com_l^k)$. These machines' equally long infinite descriptions differ only in lengths and contents of their *finite fractions* that are explicitly specified, namely $c_{yk}(kh.axe)c_{yk}(in.axe)y_l^k$ and $c_{yk}(kh.com)c_{yk}(in.com)y_l^k$. Consequently, the simplest axe and the most powerful of contemporary computers – Watson, publicly defeated best human players of TV quiz show Jeopardy! [15] – are of the meaning complexity of inorganic raw materials and are equally incapable of mirroring and understanding humans [2]. In sum, *any machine built of inorganic raw materials applying a machine technology can never completely mimic/reproduce even the simplest living organism; in particular, it can never think as an animal/human thinks*.

The living organism $O(I)_l^k$ of the I th evolutionary deepness or of a given meaning complexity, e.g., a human ($I = 0$) or an evolutionary less developed cellular organism ($0 < I < I_{\max}$) could similar be presented,

$$O(I)_l^k = c_{yk}(o_l)y_l^k = c_{yk}(rm.o_l)c_{yk}(kh.o_l)c_{yk}(in.o_l)y_l^k = E_{ij} OT_j^i(o_l^k), \quad (8)$$

if the size of genomic code y_l^k , $l(y_l^k) = k$ bits, suffices to change the $O(I)_l^k$ to self-sustain and self-reproduce itself under conditions defined by one-way infinite string $c_{yk}(o_l)$. The string $c_{yk}(rm.o_l)$ describes *organic things/raw materials* with the genomic code y_l^k needed to build the $O(I)_l^k$; $c_{yk}(kh.o_l)$ gives the finite description of evolutionary know-hows comprising previous evolutionary achievements in making the organisms (this description is a specification of a finite collection of one-way infinite substrings of one-way infinite string $c_{yk}(rm.o_l)$); $c_{yk}(in.o_l)$ is the finite part of the organism's description that could completely be specified. As (8) demonstrates, to design an organism of a given meaning complexity (evolutionary deepness), the raw materials of meaning complexity of the organism itself are to be used. The meaning of the sign E_{ij} is given by left-hand parts of (8) defining the I th *biotechnology* – the procedure/history of production/self-reproduction of the organism $O(I)_l^k$ from organic things $OT_j^i(o_l^k)$ of the meaning complexity of the $O(I)_l^k$ or from the $O(I)_l^k$ itself.

Virus engineering [16] or its counterpart used to synthesize bacterial cells is a natural evolutionary technique adopted by humans to modify the biotechnologies (8) designed by evolution. Symbiosis is such another technique invented by evolution

(e.g., insect-plant relationships [17]) and used by humans (e.g., human-plant relationships in human agriculture). Consequently, by a modification of existing natural biotechnologies, humans could engineer new organisms but those only that are evolutionary close to raw organisms that are of use. In sum, *an artificial “wet” designing/engineering of human-like organisms of human-level raw materials is possible (we, humans, are the examples) but not required (we, humans, already exist)*.

8 The Mowgli’s Test for Thinking/Perception in Others

Machines cannot think, but they could perform particular brain functions *described by humans* at the stage of designing the machines. For Watson the computer [15] and for the Turing’s test [1], this function is answering questions. By enlarging the size and improving the quality of this description, such a machine could eventually be built that could outperform humans doing a *particular* brain function [15]. But it is impossible for *all* brain functions simultaneously because, to achieve this aim, an *infinitely long* machine algorithm is required; see (1), (2), (7), and (8). An estimation of the degree to which a machine mimicking a separate intelligent brain function (human face recognition) could *outperform* humans is given in ref. 6.

It is important to have a test for thinking/perceiving in other living organisms or solving the problem of other minds [18]. The BSDT PL implies the organisms/agents could *completely* understand each other if they are the mirror replicas of each other, i.e., if they have common evolutionary history, are at the same level of evolutionary development and matured in typical for them environment [2]. This means they should be of the same lineage and meaning complexity and should have the same major properties. If, for any two individuals, these conditions are fulfilled then one of them can be sure the other has thinking/mind faculties as he/she/it has. Two individuals are of the same lineage and meaning complexity if they are of the same species (it could be confirmed by comparing their genomes); two individuals of the same species have the same major properties if they were developed/matured in typical for them environment (it could be confirmed by comparing their species-specific behavioral performance). We call these *necessary and sufficient* conditions that ensure the existence of equal thinking/perceiving capabilities in others the *Mowgli’s test*, though this Rudyard Kipling’s fictional character only partly meets new test’s demands because his master words, “We be of one blood, ye and I” reflect the test’s necessary part (the need to be of the same species) and ignore its part making it sufficient (the need to be matured in typical for this species environment).

9 Conclusion: Cyborgs Are Our Evolutionary Perspective

Hypothesis of concurrent infinity adds to the ZF foundations of mathematics the idea of evolution, thanks to which the notion of subjectivity was formalized and the things having this property were represented as real-world STMs [2]. Using this advance, in this paper, the notion of life has been formalized in a way that is consistent with available knowledge and demonstrated that thinking is the property of living organisms implemented as real-world PUSTMs. Machines/computers built of

inorganic raw materials cannot think but they could outperform humans doing a particular task, regardless of whether it is intelligent or not. All living organisms to an extent think. The necessary and sufficient Mowgli's test for thinking in others that gives also a solution to the problem of other minds has been proposed.

Since machines can never think, robots have no evolutionary perspective. The real challenge is the managing of permanently evolving societies of "humans-becoming-cyborgs" – the products of evolutionary symbiosis of humans and human artifacts. To the degree we depend on our artifacts we are cyborgs, possibly, so far rudimentary ones. Since this dependence inevitably grows, true cyborgs are most probably our evolutionary perspective. Numerous examples support this view.

References

1. Turing, A.M.: Computing Machinery and Intelligence. *Mind* 59, 433–460 (1950)
2. Gopych, P.: Beyond the Zermelo-Fraenkel Axiomatic System: BSDT Primary Language and its Perspective Applications. *Int. J. Advances Intelligent Systems* 5, 493–517 (2012)
3. Gopych, P.M.: Elements of the Binary Signal Detection Theory, BSDT. In: Yoshida, M., Sato, H. (eds.) *New Research in Neural Networks*, pp. 55–63. Nova Science, NY (2008)
4. Gopych, P.: BSDT Multi-valued Coding in Discrete Spaces. In: Corchado, E., Zunino, R., Gastaldo, P., Herrero, Á. (eds.) *CISIS 2008. ASC*, vol. 53, pp. 258–265. Springer, Heidelberg (2009)
5. Gopych, P.: Minimal BSDT Abstract Selectional Machines and Their Selectional and Computational Performance. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) *IDEAL 2007. LNCS*, vol. 4881, pp. 198–208. Springer, Heidelberg (2007)
6. Gopych, P.: Biologically Plausible BSDT Recognition of Complex Images: The Case of Human Faces. *Int. J. Neural Systems* 18, 527–545 (2008)
7. Rizzolatti, G., Craighero, L.: The Mirror-neuron System. *Ann. Rev. Neurosci.* 27, 169–192 (2004)
8. Copeland, B.J.: Hypercomputation. *Minds and Machines* 12, 461–502 (2002)
9. Penny, D., Poole, A.: The Nature of the Last Common Ancestor. *Curr. Opin. Genet. Dev.* 9, 672–677 (1999)
10. Woese, C.R.: On the Evolution of Cells. *Proc. Natl. Acad. Sci. USA* 99, 8742–8747 (2002)
11. Glansdorff, N., Xu, Y., Labeda, B.: The Last Universal Common Ancestor: Emergence, Constitution and Genetic Legacy of an Elusive Forerunner. *Biology Direct* 3, 29 (2008)
12. Gopych, P.: BSDT Atom of Consciousness Model, AOCM: The Unity and Modularity of Consciousness. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) *ICANN 2009, Part II. LNCS*, vol. 5769, pp. 54–64. Springer, Heidelberg (2009)
13. Ullman, S.: Object Recognition and Segmentation by a Fragment-based Hierarchy. *Trends Cogn. Sci.* 11, 58–64 (2007)
14. Gopych, P., Gopych, I.: BSDT ROC and Cognitive Learning Hypothesis. In: Herrero, Á., Corchado, E., Redondo, C., Alonso, Á. (eds.) *CISIS 2010. AISC*, vol. 85, pp. 13–23. Springer, Heidelberg (2010)
15. Ferrucci, D., Brown, E., Chu-Carroll, J., et al.: Building Watson: an Overview of DeepQA Project. *AI Magazin* 31, 59–79 (2010)
16. Mateu, M.G.: Virus Engineering: Functionalization and Stabilization. *Protein Engineering, Design, and Selection* 24, 53–63 (2011)
17. Joy, J.B.: Symbiosis Catalyses Niche Expansion and Diversification. *Proc. R. Soc. B* 280, 20122820 (2013)
18. Hyslop, A.: *Other Minds*. Kluwer, Dordrecht (1995)

Study of Influence of Parameter Grouping on the Error of Neural Network Solution of the Inverse Problem of Electrical Prospecting

Sergey Dolenko¹, Igor Isaev², Eugeny Obornev³,
Igor Persiantsev¹, and Mikhail Shimelevich³

¹ D.V. Skobeltsyn Institute of Nuclear Physics, M.V. Lomonosov Moscow State University,
Leninskie Gory, 119991, Moscow, Russia

² Physical Department, M.V. Lomonosov Moscow State University, Leninskie Gory, 119991,
Moscow, Russia

³ S. Orjonikidze Russian State Geological Prospecting University, 23 Miklukho-Maklaya st.,
117997, Moscow, Russia
Dolenko@srd.sinp.msu.ru

Abstract. In the electrical prospecting inverse problem, the sought-for distribution of electrical conductivity in Earth stratum is described by dividing the studied section into blocks arranged in layers, with determination of electrical conductivity in the center of each block. This inverse problem can be solved separately for each block, or simultaneously for a group of blocks. In this study, the dependence of solution error on the number of blocks for simultaneous solution of the problem with a single neural network, and on the method of their choice, was investigated.

Keywords: neural networks, inverse problems of high dimensionality, group determination of parameters, electrical prospecting.

1 Introduction

Solution of the inverse problem (IP) of electrical prospecting in geophysics is the process of construction of an operator mapping the vector of data on the values of electromagnetic field characteristics observed on the Earth's surface to the vector of sought-for geophysical parameters describing the distribution of electrical conductivity in the studied underground area. Actual distributions are very complex, so they require a very large number of parameters to describe them, thus leading to well-known instability of the ill-posed IP of electrical prospecting [1].

Artificial Neural Networks (ANN) are one of instruments applied to solve IP [2] by its modeling, including IP of electrical prospecting [3]. ANN are applied to solve IP in all the domains where ill-posed IP are encountered, including geophysics [4], spectroscopy [5], and materials science [6].

Since the beginning of 90s, ANN methods begin to be applied for the solution of IP in mathematical physics more and more widely, including IP in geophysics – in seismic prospecting [7], electrical prospecting and other areas. In [8], one can find

detailed review and analysis of possible ways for application of NN technologies in various types of geophysical research. In [9, 10], algorithms of electromagnetic data interpretation using ANN approaches were suggested. Books [11, 12] are devoted to detailed consideration of the opportunities of using ANN for processing geophysical data (mainly those of seismic prospecting).

In [13] and other papers, a number of examples of using ANN to solve the IP of magnetotelluric sounding (MTS) have been presented. It has been shown that ANN methods were efficient when the design of the geoelectric section was determined by a small number of parameters (10-20), i.e. the solution was sought within a relatively narrow class of models, and the dimensionality of the IP was relatively low. At the same time, the ANN method has a number of indisputable advantages over traditional (optimization-based) approaches to the solution of MTS IP: high speed of inversion, possibility of multiple uses, good interpolation properties etc.

At the same time, attempts to extend the applicability area for the case of multi-parameter sections and to work in wider classes encounters significant difficulties connected mainly with very high dimensionality of the problem, as input dimensionality, as output one (the number of parameters of the section). The number N_O of determined parameters, describing the distribution of electrical conductivity, may equal several hundred even for the considered 2D case, and the dimensionality N_I of the input vector of electromagnetic fields may reach tens of thousands.

Computational cost of ANN solution of the IP can be reduced by compression of the input vector of fields, e.g. by selection of the most significant features. If such selection is performed correctly, the IP solution error is also reduced [14].

As for the components of the output vector of parameters, the problem is usually solved for each of them separately, i.e. for complete description of electrical conductivity distribution it is necessary to solve N_O single-output problems, training N_O single-output neural networks. Acceleration of such computations (building many ANNs with identical architectures, with the same input data and different output data) can be achieved with hardware support by graphics processing units of a serially produced video display adapter based on NVIDIA CUDA technology. So, the authors of the present study succeeded in reducing the effective time of computations (for ANN solution of the considered 2D MTS IP) by two orders of magnitude [15].

Meanwhile, there are several possible approaches to ANN solution of multi-output problems, including multi-parametric inverse problems:

1) Solving a separate single-output IP with training a separate ANN for each of the determined parameters, as described above (*autonomous determination*). This approach is the most versatile one, and it is used most often.

2) Solving a single IP with *simultaneous determination* of all the sought-for parameters, by building a single ANN with N_O outputs. Efficiency of such approach degrades quite rapidly with increasing number of the determined parameters. For $N_O > 20$, it becomes practically unusable. However, for IP with a small number of parameters, it sometimes allows decreasing the error of parameters determination.

3) Association of parameters into groups with simultaneous determination of parameters (and single ANN training) within each group (*group determination*). The method of group forming is determined by the physical meaning of the determined parameters and by known relations among them. This approach is actually an interjacent one. Its study in more detail is the main goal of the present paper.

4) *Sequential determination* of parameters. Within this approach, at the first stage one determines (by autonomous or group determination) those parameters for which the problem can be solved with acceptable accuracy disregarding the values of all the remaining parameters. At the subsequent stages, the values of the already determined parameters, obtained by applying the ANNs of the first stage, are fed to the ANN input together with the values of the input features. Sometimes this approach allows reducing the error of ANN solution of the IP for those parameters, for which the quality of the IP solution within other approaches is unacceptable.

The present study is devoted to comparative investigation of the methods of parameters grouping in the solution of the IP of electrical prospecting by ANN methods with *group determination* of parameters. The ANN architecture used in this study was a multi-layer perceptron trained by error backpropagation method.

2 Given Data and Problem Statement

To perform this study, the results of numerical solution of the direct 2D problem of magnetotellurics [3] were used as the given data, as follows. In the solution of this problem, the determined values are those of various components of induced electromagnetic fields of diverse polarization at various frequencies and in diverse points on Earth surface during scattering of plane electromagnetic waves excited by perturbation of geomagnetic field by flows of charged particles from the Sun (solar wind). The calculation depends to a considerable degree on the so called medium parameterization scheme, i.e. on the method of description of the electrical conductivity of the underground area. In the present study, we used the data obtained for the most general G_0 parameterization scheme (Fig. 1).

| Layer | H | Z | Column | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 35 km | | | | | | | | | | |
|-------|-----|-----|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | | Y= | 35 | 25 | 17 | 12 | 6 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 6 | 12 | 17 | 25 | 32 | 33 | | | | | | | | | | | |
| 1 | 0.5 | 0.5 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | | | | | | | | | | |
| 2 | 0.5 | 1 | | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | | | | | | | | | | |
| 3 | 0.5 | 1.5 | | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | | | | | | | | | | |
| 4 | 1 | 2.5 | | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 | 132 | | | | | | | | | | |
| 5 | 1 | 3.5 | | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 | 161 | 162 | 163 | 164 | 165 | | | | | | | | | | |
| 6 | 1.5 | 5 | | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 335 |
| 7 | 2.5 | 7.5 | 334 | 187 | 188 | 189 | 190 | 191 | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 | 224 | 225 | 226 | 227 | 228 | |
| 8 | 2.5 | 10 | | 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 | 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 | 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | |
| 9 | 3 | 13 | | 250 | 251 | 252 | 253 | 254 | 255 | 256 | 257 | 258 | 259 | 260 | 261 | 262 | 263 | 264 | 265 | 266 | 267 | 268 | 269 | 270 | 271 | 272 | 273 | 274 | 275 | 276 | 277 | 278 | 279 | 280 | 281 | 282 | 283 | 284 | 285 | 286 | 287 | 288 | 289 | 290 | 291 | |
| 10 | 4 | 17 | | 292 | 293 | 294 | 295 | 296 | 297 | 298 | 299 | 300 | 301 | 302 | 303 | 304 | 305 | 306 | 307 | 308 | 309 | 310 | 311 | 312 | 313 | 314 | 315 | 316 | 317 | 318 | 319 | 320 | 321 | 322 | 323 | 324 | 325 | 326 | 327 | 328 | 329 | 330 | 331 | 332 | 333 | |

Fig. 1. Medium parameterization scheme G_0

Calculation is performed within the area bounded by the outer frame. Each block is labeled by its number; the first column shows layer numbers, the first row shows column numbers. Most interesting is the central area of the section, marked by bold frames: layers 1-13, columns 5-28. Parameters 334-336 define boundary conditions for calculations. H – layer thickness, Z – depth of occurrence of the lower border of a layer, Y – horizontal size of a block; all parameters in km. The blocks marked gray are those for which the investigations were performed in the present study.

The data array consisted of 30,000 samples obtained for random combinations of conductivity of different blocks in the range from 10^{-4} to 1 S/m. It was used for solution of the IP – determination of the values of electrical conductivity (336 parameters, Fig. 1) by the field values (4 field components \times 13 frequencies \times 126 pickets = 6552 input features). It has been discovered [14] that for the most interesting central area of the section, the solution error for this problem with autonomous determination of value of each parameter strongly depends on layer number (generally increasing with depth) and is practically independent on the block number within the same horizon.

In the present study, the following issues are investigated:

- 1) How will the error of solution of the considered IP change if *group* determination of parameters is used instead of autonomous determination?
- 2) What is the optimal method of parameter grouping?

3 Results

Solution of the problem was studied for parameters marked grey in Fig. 1. Parameter grouping was performed in two main ways: horizontally (the associated parameters belonged to the same layer no.2, for which the errors of autonomous determination were smaller than for all other layers) and vertically (the associated parameters belonged to the central column no.16).

As the initial input dimensionality of the problem is very high ($N_i=6552$), prior two-stage selection of significant input features using ANN weight analysis [14] was performed for each parameter. In group determination of the parameters, the inputs of each ANN were fed with all the input features detected as significant for at least one of the output parameters determined. Depending on the size of the group and on the size of intersection of sets of significant input features, total number of input features for group determination of parameters was from 32 to 940.

The ANN architecture used was perceptron with three hidden layers of 24, 16, and 8 neurons. When the number of parameters determined simultaneously exceeded 8, the number of neurons in the hidden layers was respectively increased, up to 48-36-24 for 21 outputs. All perceptrons were trained by error backpropagation. The transfer functions were logistic for all the hidden layers and linear for the output layer. To prevent overtraining, training was stopped when 1000 epochs passed after the best ANN corresponding to minimum MSE on an independent test data set was obtained.

All the results presented were obtained on an out-of-sample examination data set. 70% of data samples were used for training, 20% for test and 10% for examination.

3.1 Horizontal Grouping

In this series of numerical experiments, networks with the following numbers of outputs were used: $S_g = 1, 2, 3, 5, 7, 11, 15$, and 21. The desired outputs were the values of electrical conductivity of the blocks with numbers from 40 to 60 from layer No.2 (Fig. 1). For each network, the target data were taken for the corresponding number of horizontally adjacent blocks, with a shift. For example, the results of the IP solution for central block No.50 have been obtained with the help of networks that

simultaneously solved the problem for the following combinations of blocks: only 50, 49-51, 48-52, 47-53, 40-50, 45-55, 50-60, 40-54, 43-57, 46-60, and 40-60. Thus, there was an opportunity to study the dependence of solution quality as on group size, as on its horizontal position within the section.

The number of elements common for the sets of significant input features for associated blocks was not large. To reduce the dependence of the results on random factors, each ANN was trained five times with different sets of initial weights.

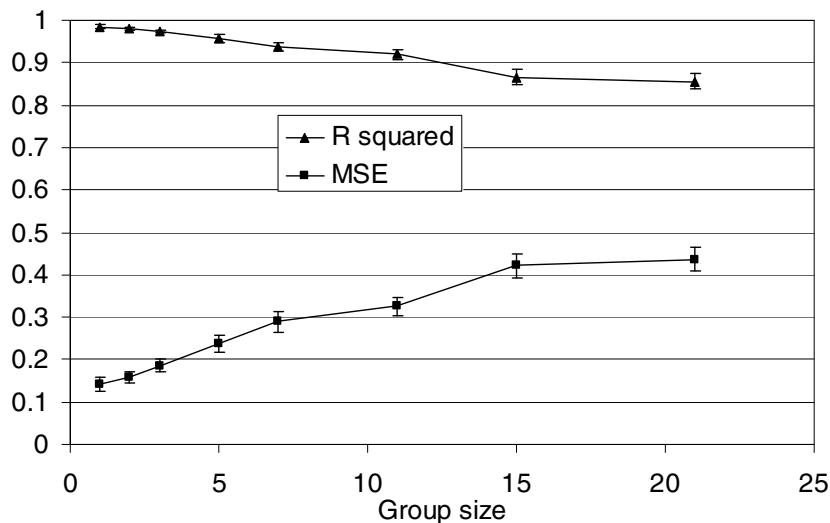


Fig. 2. Dependence of the IP solution quality indicators on group size S_g for horizontal grouping of parameters

Fig. 2 displays the dependences of the multiple determination coefficient R^2 (R squared) and of root mean squared error (RMSE) on group size S_g , i.e. on the number of ANN outputs. The values have been averaged as over realizations of identical ANNs with different initial weights, as over different positions of the horizontal window of selection of adjacent outputs with given size S_g within blocks 40-60 of layer No.2 (Fig. 1).

One can see that for horizontal grouping of blocks, the indicators of problem solution quality worsen monotonously with increasing group size. In this case, autonomous determination is preferable. Small dispersion of the average values (see error bars in Fig. 2) is an indirect evidence of weak dependence of the indicators on the position of the group of determined parameters within single layer of a section, what looks justified from geophysical point of view.

The latter fact is also confirmed by the dependence shown in Fig. 3. Relatively large error values are due to the fact that for each block the averaging has been performed not only over realizations of identical ANNs with different initial weights, but also over different values of the group size S_g , which has a noticeable influence on the IP solution quality (Fig. 2).

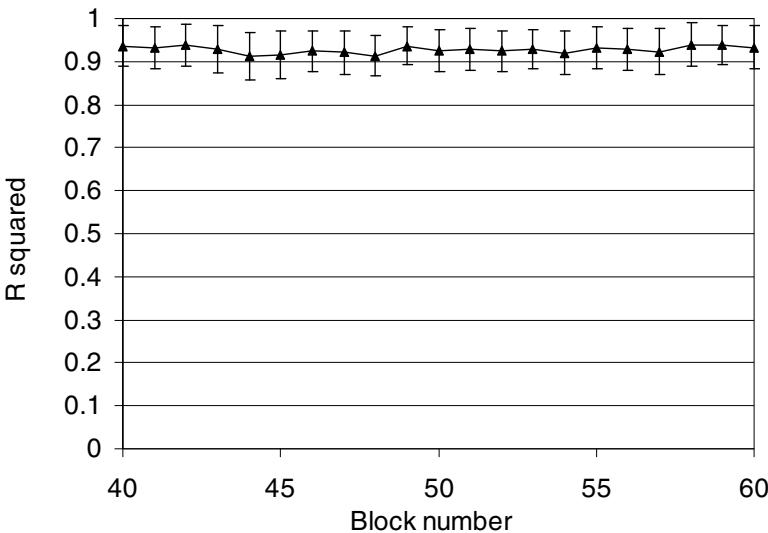


Fig. 3. Dependence of the coefficient of multiple determination R^2 on block number for horizontal grouping of parameters

3.2 Vertical Grouping

In this series of numerical experiments, networks with the following numbers of outputs were used: $S_g = 1, 2, 3, 4, 5, 7, 9, 11$, and 13 . The desired outputs were the values of electrical conductivity of the blocks with numbers $17, 50, \dots, 302, 323$ from column no.16 (Fig. 1). For each network, the target data were taken for the corresponding number of vertically adjacent blocks, with a shift. For example, the results of the IP solution for central block no.50 have been obtained with the help of networks that simultaneously solved the problem for the following combinations of blocks: only 50; (17, 50); (50, 83); (17, 50, 83); (50, 83, 116); (17, 50, 83, 116); (50, 83, 116, 149) etc. Thus, there was an opportunity to study the dependence of solution quality as on group size, as on its vertical position within the section. Note that in this case, the associated blocks had a substantial number of common significant input features. To reduce the dependence of the results on random factors, each ANN was trained five times with different sets of initial weights.

Fig. 4 shows the dependence of R^2 coefficient on block number. One can see that the quality of IP solution substantially degrades with increasing depth of block occurrence, what can easily be explained by geophysical considerations. Indeed, blocks occurring deeper are shaded to a considerable extent by those occurring higher, thus making observed field values low-sensitive to changes in conductivity of deep-occurring blocks. This effect manifests itself by IP solution error increasing with depth. Deviations from the described dependence occur near borders of the considered area (upper layer and two lower layers), and also near the border separating blocks of single and double size (layers 5 and 6, Fig. 1).

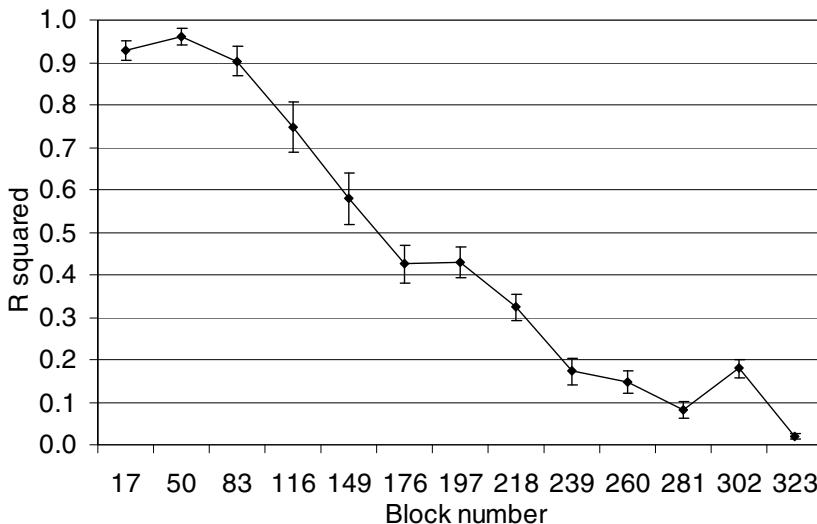


Fig. 4. Dependence of the coefficient of multiple determination R^2 on block number for vertical grouping of parameters

Such a strong dependence on block number makes it impossible to study the dependence of IP solution quality on group size, performing averaging over block numbers. For this reason, in Fig. 5 such dependences are presented separately for the most interesting blocks – from six upper layers.

For all blocks except the end ones, there are several ways to aggregate vertically adjacent blocks including some given block. In each of the diagrams in Fig. 5, two variants are displayed, with maximum possible shift of the parameter aggregation window up or down.

The following facts attract attention.

1) For the upper block no.17, the quality of the group solution of the IP is increased as compared with that of autonomous determination, if the group includes up to 4 output features. However, the best result is achieved for group size $S_g=2$.

2) For block no.50, for which the problem solution quality in autonomous mode is the best one (Fig. 4), group determination always gives worse results.

3) For deeper occurring blocks, the situation depends on the position of parameter grouping window. If the window is shifted up (towards parameters that are better determined in autonomous mode), group determination improves the results; if the window is shifted down, the group determination usually makes the results worse.

In the whole, the following conclusion can be made. Improvement in solution quality for some parameter with group determination (Fig. 5) can be achieved most often in cases when this parameter is determined together with such parameters for which quality of their determination in autonomous mode is better than that for this one (Fig. 4). The same factor influences the optimal number of grouped parameters.

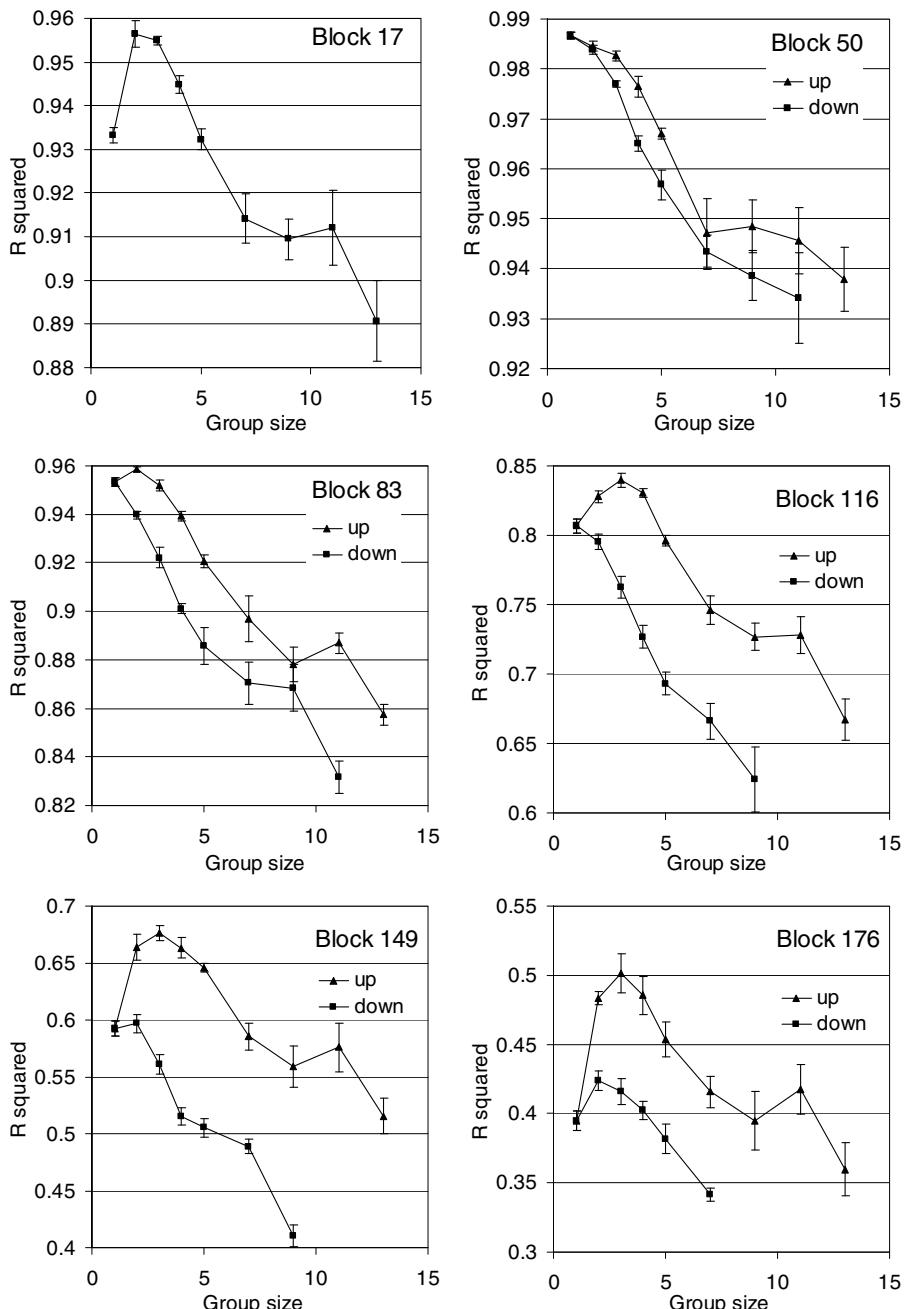


Fig. 5. Dependence of the coefficient of multiple determination R^2 on group size S_g for vertical grouping of parameters, for different blocks

4 Discussion

The result of parameter aggregation in ANN solution of multi-parameter IP with group determination of parameters is determined by the following.

Consider a perceptron with a single hidden layer (HL) and with a single output neuron. During its training, some composite high-level features describing input data and useful for solution of the required problem, are extracted in the HL. The weights of the neuron of the output layer determine relative significance of each of the features extracted in the HL for solution of the problem.

Now, let the ANN have at least two output neurons. Here we may consider two extreme cases.

In the first one, the sought-for parameters are strongly interconnected (in the limit, they are identical). In this case, the optimal composite features for all parameters will be the same, and these features will be extracted in the HL. General simplification of the shape of error functional will cause reduction in the probability of sticking in a local extremum for the procedure of its minimization. Addition of the error vectors from the outputs at the HL and their collinearity (within the differences caused by weight initialization) will cause effective increase in HL learning rate. Thus, when the aggregated parameters have similar dependences on the input variables, one could expect (except reduction in the number of nets required) decrease in the total training time and possible error reduction.

In the opposite case, when the sought-for parameters are completely independent of each other and have different dependences on the input variables, they will need different composite features for their optimal determination. Error vectors at the HL will be different and they will “pull” the optimization procedure asunder. It is clear that such a ANN will learn each output variable worse than a network with the same number of neurons in the HL and single output.

Certainly, usually neither of the described extreme cases, but something intermediate is realized. However, the dependence of problem solution error on the number and method of parameter grouping is observed in practice, and it has been demonstrated in this study.

5 Conclusion

Group determination of parameters may provide an efficient method of error reduction in ANN solution of multi-parameter inverse problems (and in general, in ANN solution of multi-output problems).

In order to reduce the IP solution error with group determination of parameters in comparison to that with autonomous determination of parameters, the following conditions need to be met:

1) Parameters aggregated into the group should have similar dependences on the input features. At least, significant input features for such parameters should coincide to a substantial degree.

2) It is preferable to aggregate the determined parameter with those having lower error of autonomous determination than this one.

In the present study, these statements have been proved by computational experiment, and their explanation has been suggested.

Acknowledgments. This study was supported by the Russian Foundation for Basic Research grants No. 11-07-00662-a and No.13-05-01135-a. Authors thank A. Guzhva for developing the software with which this study has been performed.

References

1. Berdichevsky, M.N., Dmitriev, V.I.: Models and Methods of Magnetotellurics. Springer (2008)
2. Gerdova, I.V., Churina, I.V., Dolenko, S.A., Dolenko, T.A., Fadeev, V.V., Persiantsev, I.G.: New Opportunities in Solution of Inverse Problems in Laser Spectroscopy Due to Application of Artificial Neural Networks. In: Proc. SPIE, vol. 4749, pp. 157–166 (2002)
3. Shimelevich, M.I., Obornev, E.A., Gavryushov, S.: Rapid Neuronet Inversion of 2D Magnetotelluric Data for Monitoring of Geoelectrical Section Parameters. Annals of Geophysics 50(1), 105–109 (2007)
4. Xu, H.-L., Wu, X.-P.: 2-D Resistivity Inversion Using the Neural Network Method. Chinese J. of Geophysics 29(2), 507–514 (2006)
5. Li, M., Verma, B., Fan, X., Tickle, K.: RBF neural networks for solving the inverse problem of backscattering spectra. Neural Computing & Applications 17(4), 391–397 (2008)
6. Yang, H., Xu, M.: Solving inverse bimodular problems via artificial neural network. Inverse Problems in Science and Engineering 17(8), 999–1017 (2009)
7. Devilee, R.J.R., Curtis, A., Roy-Chowdhury, K.: An efficient, probabilistic neural network approach to solving inverse problems: Inverting surface wave velocities for Eurasian crustal thickness. J. Geophys. Research 104(B12), 28841–28857 (1999)
8. Raiche, A.: A pattern recognition approach to geophysical inversion using neural nets. Geophysics J. Int. 105(3), 629–648 (1991)
9. Poulton, M., Sternberg, B., Glass, C.: Neural network pattern recognition of subsurface EM images. Journal of Applied Geophysics 29(1), 1534–1544 (1992)
10. Hidalgo, H.: Neural Network Approximation of an Inverse Functional. In: IEEE World Congress on Computational Intelligence, p. 5 (1994)
11. Poulton, M.M. (ed.): Computational Neural Networks for Geophysical Data Processing. Elsevier Science Ltd., Kidlington (2001)
12. Sandham, W., Leggett, M. (eds.): Geophysical Applications of Artificial Neural Networks and Fuzzy Logic. Kluwer Academic Publishers, Dordrecht (2003)
13. Spichak, V., Fukuoka, K., Kabayashi, T., Mogi, T., Popova, I., Shima, H.: ANN reconstruction of geoelectrical parameters of the Mionou fault zone by scalar CSAMT data. J. App. Geophys. 49, 75–90 (2002)
14. Dolenko, S., Guzhva, A., Obornev, E., Persiantsev, I., Shimelevich, M.: Comparison of Adaptive Algorithms for Significant Feature Selection in Neural Network Based Solution of the Inverse Problem of Electrical Prospecting. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009, Part II. LNCS, vol. 5769, pp. 397–405. Springer, Heidelberg (2009)
15. Guzhva, A., Dolenko, S., Persiantsev, I.: Multifold Acceleration of Neural Network Computations Using GPU. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009, Part I. LNCS, vol. 5768, pp. 373–380. Springer, Heidelberg (2009)

Prediction of Foreign Currency Exchange Rates Using CGPANN

Durre Nayab¹, Gul Muhammad Khan², and Sahibzada Ali Mahmud³

¹ Department of Computer Systems Engineering,
University of Engineering and Technology, Peshawar, Pakistan
nayaab_khan@nwfpuet.edu.pk

^{2,3} Department of Electrical Engineering, University of Engineering and Technology,
Peshawar, Pakistan
{gk502, sahibzada.mehmud}@nwfpuet.edu.pk

Abstract. This paper contributes an application of Cartesian Genetic Programming Evolved Artificial Neural Network (CGPANN) for forecasting the foreign currency exchange rates. The end product of our work is an efficient Artificial Neural Network (ANN) based prediction model that forecasts the foreign currency exchange rates, making use of the trends in historical data. These trends in the historical currency data serve as significant prognostic factor to train the prediction model. The algorithm exploited for the evolution of the prediction model is Cartesian Genetic Programming (CGP). CGP evolved ANNs have great potential in prediction models for forecasting systems. Historical daily prices of 500 days data of US dollars are monitored to train the prediction model. Once the model is trained, it is tested on 1000 days data of ten different currencies to predict these currency rates and the results are monitored to analyze the efficiency of the system. The results show that prediction model achieved with CGPANN is computationally cost effective and accurate (98.85%) that is unique as it is dependent on least amount of previous data for future data prediction.

Keywords: Foreign Currency Exchange, Prediction Model, Cartesian Genetic Programming Evolved Artificial Neural Network (CGPANN), ANN and Neuro Evolution.

1 Introduction

With the ongoing computerization, the financial time series has become one of the most challenging applications for the forecasting of currency exchange rates. The financial time series have noisy, unstable and abruptly changing setup [1][2][3][4]. The present statistical models used for forecasting of currency exchange rates are not flexible and efficient enough to effectively deal with the uncertainty and volatility in the nature of foreign exchange data. Hence some efficient techniques are required to manage the complex financial time series. Studies have shown that application of the Neural Networks (ANNs) to time series forecasting tasks has better results as compared to other techniques [5, 6]. As ANNs have efficient set up because of their learning ability, they can efficiently deal with such unpredictable and disputed data.

Artificial Neural Networks are analogous to naturally occurring Neural Networks in their composition [7]. Like naturally occurring Neural Network, the ANN evolves to the fittest situation by the process of learning. The functional units (Neurons) of the ANN take inputs, process them and produce outputs. The beneficial aspect of ANN is that it adds weights to the connections in the network accordingly and during the process of evolution different parameters of the network are modified and monitored unless the network is evolved into the fittest model [8][9].

In this paper a Cartesian Genetic Programming based architecture is exploited to evolve the ANN for currency exchange forecasting and a prediction model is proposed for it. The prediction model is basically the neural network generated with the CGP algorithm. CGPANN has the fast learning capability that is exploited for dealing with the volatility in the nature of foreign currency exchange data. The model is trained with the historical daily prices of the currency and is tested over other data sets to analyze the performance of the system. CGPANN proves to be efficient and suitable for the unstable data of financial time series as it has optimized network architecture. Previous work on CGPANN has shown that CGPANN is noise tolerant and can efficiently withstand abruptly changing data [9]. The network evolved with CGPANN has an optimized number of nodes that means all the nodes are not used for evolving the model. This is how it achieves its fast learning ability. Whereas other ANNs such as hidden Markov model (HMM), multilayer perceptron (MLP), neuro-evolution of augmenting topologies (NEAT), conventional neural evolution (CNE), and symbiotic adaptive neural evolution (SANE), use all the inputs to evolve their network. This makes the network huge and complex in terms of architecture and in turn computations [14].

2 Review of Related Work

Several implementations have been done in the field of time series forecasting. According to the survey carried out in [18] almost 127 neural networks models for business applications are published in international journals up to September, 1994. In next year 86 more manuscripts were published [18]. The aforementioned depicting the popularity of such methods. These methods have different approaches towards prediction and are distinguished in terms of their potential to efficiently forecast the future exchange rates and their ease in terms of implementation. The statistical models have shown better performance on time series forecasting but these traditional statistical techniques have their limitations with non linear data sets (such as stock indices) and changing environment [5, 6]. According to [1] the prediction model of Hidden Markov (HMM) is unstable that is influenced by many factors. Artificial neural network foreign exchange rate forecasting model (AFERFM) designed for foreign exchange rate forecasting to correct some of these problems. HMMs were applied to forecasting of Euro/Dollar (EUR/USD) using the European Central Bank (ECB) fixing series with only auto regressive terms as inputs. The system was tested using mean squared error and standard deviation with learning rate of 0.1%, single input layer, three hidden layers and a single output layer.

Time series forecasting carried out in [4] used two ANN techniques such as Multilayer Perceptron (MLP) and Volterra. The moving average tool that calculated the mean prices in a given time interval was used for the technical analysis of the

data. Hence the tool smoothed the price factors over several time periods. The experimental results were monitored for MLP ANN with multiple layers and Volterra ANN with four layers and twelve inputs. A multi-neural network model consisting of three sub-networks and one master network is developed in [17] for the forecasting of TWD/USD exchange rate. The tendencies of the exchange rate are forecasted by different sub-networks on five macro economics factors such as interest rate, import/export, productivity, money supply and price level. The results are monitored under seven technical indicators of fifteen-days and one-day intervals. An MLP prediction model is proposed in [19] that predicts the exchange rate Euro/ USD up to three days ahead of last available data. The MLP model predicts the exchange rate efficiently but the model does not show when the external factors affect the system. The output of the designed ANN was the daily exchange rate Euro/Dollar. The CGPANN model that we propose has an optimized architecture and performance on large datasets (see sections 5, 6 and 7).

3 Neuro Evolution

Neuro evolution is a concept of using genetic algorithms to evolve various attributes of Artificial Neural Networks (ANNs). The basic idea is to search for the optimal ANN attributes in accordance with the application being developed [9]. These attributes include the topology of the network, weights assigned to the connections, node functions [4, 10]. In traditional training methods the optimum network is searched using a single agent incrementally, having a possibility of getting trapped into local minima, whereas the evolutionary algorithms use the whole population for its search, thus avoiding getting trapped in local minima [6, 11].

There are several encoding approaches towards the search of feature space of neural networks depending upon the number and type of parameters considered for the search including: Conventional Neural Evolution (CNE), Neuroevolution of Augmenting Topologies (NEAT), Symbiotic Adaptive Neural Evolution (SANE), Enforced Sub Population (ESP), Cooperative Co Evolution (CCE), Continuous Time Recurrent Neural Network (CTRNN), Cooperative Synapse Neuroevolution (CoSyNE) [12, 14, 15, 20]. The main focus of all these Neuro Evolutionary techniques is to obtain an accurate and computationally cost effective network.

4 Cartesian Genetic Programming (CGP)

CGP is an efficient and highly flexible method of genetic programming that generates a two dimensional graphical representation of digital circuits or computer programs, introduced by J.F. Miller to evolve electronic circuits in 1997 [5, 13]. In CGP, programs are represented in directed acyclic graph formats that operate in the feed forward direction. These two-dimensional graphs of CGP are represented as a grid of programmable nodes. The CGP genotype is a combination of fixed number of array of integers that represent the interconnection of network in terms of its inputs, outputs and functions. Furthermore, CGP has myriad advantages including efficient reuse of non-coding genes and representation of arbitrary number of outputs that makes CGP preferable over tree-based programming representations.

5 CGP Evolved Artificial Neural Network

The strategy employed for ANN evolution is the major concern while obtaining the optimal neural networks for the specified application. Several approaches are introduced in this area as highlighted in section 2. The evolution of ANN model for the trends in currency exchange in this paper is conducted by CGP. As compared to other techniques CGP has proved to be efficient, computationally cost effective and has shown better results. Hence for traversing the Neural Network developed for currency exchange prediction efficiently and swiftly CGP is exploited.

CGPANN represents the network in two dimensional arrays of nodes. The functional unit of the CGPANN is a Neuron [7]. The network may exhibit as much of neurons as required for the representation of the network. During the process of evolution some neurons (Junk Nodes) may not perform any role in the ultimate output of the network while others (Active Nodes) may actively take part in producing the system output.

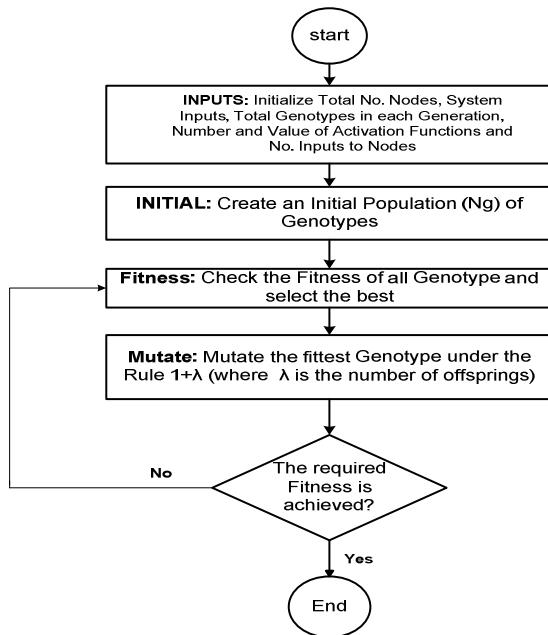


Fig. 1. A generalized CGPANN networks approach

Initially a population of genotype is generated to represent the network. This initial genotype is often called parent genotype and is mutated to produce the offspring. The offspring are produced under the evolutionary strategy of $(1+\lambda)$, where λ is equal to 9 and is the number of offspring produced (as $\lambda = 9$ has shown better results in such experiments [7]). Among these nine offspring, the fittest offspring is again chosen to become the parent and is mutated again to produce further generation of offspring. This process of producing new generations is repeated unless the desired fitness level is achieved and finally the fittest genotype is utilized for the application. *Fig. 1* depicts the flow chart of the generalized CGPANN approach.

During this evolution phase the network is trained. Once the network is trained it is tested for its performance during the testing phase. The actual values are compared with those estimated by the network which shows the performance of the network in terms of percentage error or accuracy.

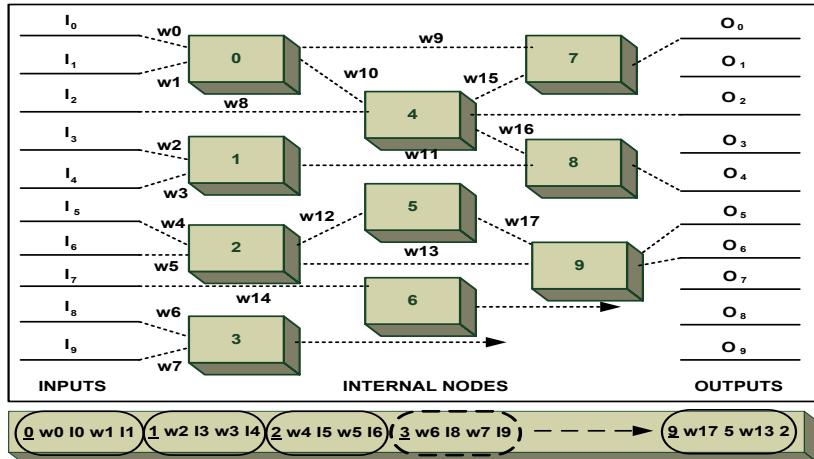


Fig. 2. A typical CGPANN phenotype and genotype

A typical CGPANN phenotype and genotype are shown in Fig. 2. It shows CGPANN inputs ($I_0, I_1, I_2, \dots, I_9$), outputs ($O_0, O_1, O_2, \dots, O_9$), weights ($w_0, w_1, w_2, \dots, w_{17}$), arity 2, active nodes (0, 1, 2, 4, 5, 7, 8 & 9) and inactive nodes (3 & 6). The genotype is also shown in the figure that is typically a string of numbers, each string representing a node in terms of its function, inputs to that function and the weights associated with those inputs respectively. The dotted string show the inactive nodes and the output is represented in a separate string.

6 Experimental Set Up

The proposed currency exchange rates forecasting model is trained on the historical daily prices of foreign exchange obtained from Australian Reserved Bank. Ten different networks are trained on 500 days data of US currency for five independent seeds each. The data is obtained in the form of average of 24 hours currency prices per day; taken from 1st Feb, 2001 to 500 consecutive days. A random population of CGPANN is generated at the beginning of the process. The activation function used is log-sigmoid. The number of inputs per node is 5. Mutation rate (μ_r) used in this case is 10% [9][19].The number of CGPANN rows in this case is one since an infinite number of graphs can be generated and thus the numbers of columns are equal to the number of nodes. The number of input(s) (ten in this case) and outputs (also ten) are initially defined for the Network. The initial randomly generated genotype is mutated and nine more networks are produced using $1+\lambda$ evolutionary strategy with λ set to 9 in this case. The fitness of these offspring is evaluated from the Mean Absolute

Percentage Error (MAPE) value and compared to select the fittest network for promotion to the next generation. This very network is then used to produce nine more networks by mutation and this process continues until the desired fitness is achieved. In this case, all experiments are run for one million generations during the training phase. The mathematical expression for MAPE value and fitness is given below:

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left(\frac{|L_{Fi} - L_{Ai}|}{L_{Ai}} \right) \times 100$$

$$Fitness = 100 - MAPE$$

Where L_F is the forecasted value, L_A is the actual value and N is the number of days. MAPE is generally considered as an international standard for determining the performance of a time series prediction based algorithm and the fitness or accuracy is the mathematical measure of performance of the system.

7 Results and Analysis

Historical daily prices of 1000 days of ten different currencies are monitored for testing the performance of the trained networks. For testing phase the data is taken from 1st Feb, 2003 to 1000 consecutive days. These currencies include Japanese Yen, Taiwanese Dollars, Euros, Great Britain Pounds, Swiss Franc, New Zealand Dollars, Canadian Dollars, Singapore Dollars, Malaysian Ringgits and Indonesian Rupiah. During the testing phase the data is estimated over the historical values of these currencies and compared with the actual values of these currencies to evaluate the performance of the network.

The training phase results are shown in Table 1, illustrating the performance in terms of average accuracy for various network sizes. These results show that the fittest network is achieved for 50 nodes with accuracy 98.42%. Since the maximum

Table 1. Average performance of various networks in the training phase for the best cases of 5 independent evolutionary runs

| No. Of Nodes | Accuracy | No. of Active Nodes |
|--------------|----------|---------------------|
| 50 | 98.42282 | 5 |
| 100 | 98.25054 | 4 |
| 150 | 98.37483 | 5 |
| 200 | 98.35625 | 6 |
| 250 | 98.33472 | 5 |
| 300 | 98.37091 | 1 |
| 350 | 98.31956 | 6 |
| 400 | 98.05929 | 4 |
| 450 | 98.34318 | 4 |
| 500 | 98.39374 | 5 |

number of generations is fixed so it is obvious to find the optimal solution quickly in a limited search space. It is myriad result as the network is showing the best accuracy for the least number of nodes used. The best and worst result for the training experiments is highlighted in Table 1.

Table 2 show the testing phase results of the networks for all currencies in terms of accuracy. All of these results are the average of results from the 5 independent evolutionary runs for each network size with tables representing the average performance. The highest value of accuracy is achieved for IDR that is **98.852%**. The results of the testing phase demonstrate that the networks generated for currency forecasting are efficient and accurate.

Table 2. Testing results of various networks for different currencies in terms of accuracy

| Nodes | Yen | TWI | Euro | GBP | CHF | NZD | CAD | SGD | MYR | IDR |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 50 | 98.64 | 98.387 | 97.851 | 98.229 | 98.192 | 98.251 | 98.315 | 98.517 | 98.401 | 98.844 |
| 100 | 98.279 | 97.808 | 97.64 | 97.878 | 97.742 | 98.159 | 98.191 | 98.064 | 97.91 | 98.514 |
| 150 | 98.469 | 98.037 | 97.767 | 98.061 | 97.934 | 98.229 | 98.277 | 98.275 | 98.119 | 98.717 |
| 200 | 98.438 | 97.993 | 97.749 | 98.03 | 97.899 | 98.22 | 98.265 | 98.238 | 98.081 | 98.685 |
| 250 | 98.401 | 97.941 | 97.724 | 97.99 | 97.856 | 98.211 | 98.25 | 98.192 | 98.035 | 98.645 |
| 300 | 98.653 | 98.441 | 97.798 | 98.193 | 98.182 | 98.196 | 98.286 | 98.545 | 98.435 | 98.852 |
| 350 | 98.385 | 97.927 | 97.714 | 97.978 | 97.844 | 98.199 | 98.24 | 98.178 | 98.022 | 98.628 |
| 400 | 97.817 | 96.871 | 97.249 | 97.218 | 96.997 | 98.06 | 98.032 | 97.284 | 97.065 | 98.044 |
| 450 | 98.424 | 97.981 | 97.739 | 98.018 | 97.887 | 98.213 | 98.256 | 98.225 | 98.069 | 98.671 |
| 500 | 98.491 | 98.059 | 97.782 | 98.08 | 97.953 | 98.239 | 98.288 | 98.294 | 98.139 | 98.739 |

Table 3 illustrates the volatility measure of the currencies in terms of standard deviation, depicting the frequency component of the data used for experiments. It can be observed that the data is highly fluctuating as their standard deviation is much high. Despite the abrupt fluctuations in the data, the given range of percentage error is negligible. These networks require only the latest data to predict the future values that is the eleventh day's currency price is predicted on the basis of previous ten days data only. This is a very efficient performance of the network as it is dependent on least and latest values of data for predicting the future values.

Table 3. Standard deviation of different currencies data

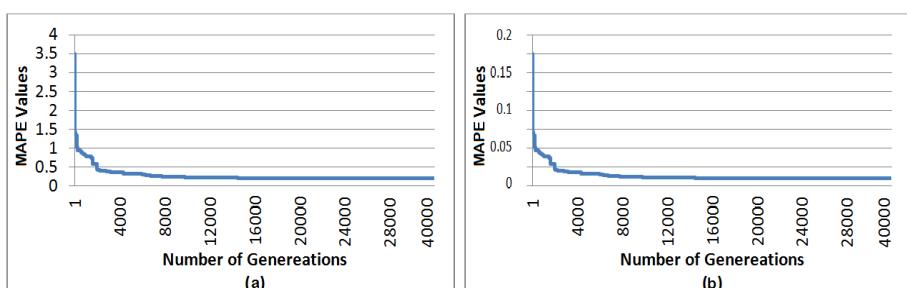
| Data | Standard Deviation |
|------|--------------------|
| Yen | 17.9401 |
| TWI | 22.7135 |
| Euro | 18.4509 |
| GBP | 20.8965 |
| CHF | 21.6349 |
| NZD | 13.2731 |
| CAD | 14.3408 |
| SGD | 21.2949 |
| MYR | 21.9656 |
| IDR | 21.2347 |

Table 4. Comparison between CGPANN and the accuracy of other prediction models

| Network | Accuracy |
|--|----------|
| Volterra Network ^[4] | 76 |
| MultiLayer Perceptron ^[4] | 72 |
| Back Propagation Network ^[17] | 62.27 |
| Multi Neural Network ^[17] | 66.82 |
| HFERFM ^[1] | 69.9 |
| AFERFM ^[1] | 81.2 |
| Regression Model ^[5] | 98.1 |
| Markov Model ^[5] | 98.072 |
| ARIMA ^[10] | 98.39 |
| CGPANN (Proposed) | 98.85 |

A comparison of various recent prediction models' accuracy is shown in Table 4. The proposed network is not compared with statistical time series methods as their results are not comparable with recent efficient neuro-evolutionary methods. The accuracies of these models for time series forecasting are compared and the best accuracy is observed for CGPANN. This means that the forecasting model of CGPANN is the most accurate of all. The network is also checked for more than one day prediction in order to evaluate its performance for multiple numbers of days. To predict more than one day exchange rate in advance we have conducted experiments to predict ten days daily prices in advance instead of a single day.

The network produces these accurate and efficient results so swiftly that it reaches the highest level of accuracy in few evolutionary runs. This proves the fast learning ability of CGPANN, making it much efficient and advanced in comparison to other contemporary techniques. The graphs shown in Fig. 3 illustrate the learning phase of the proposed network. It can be seen from the graphs that the network learns to the fittest level in little initial number of generations. This means that the network learns swiftly to the fittest level and reaches the optimum results steadily.

**Fig. 3.** Learning phase graphs of the network for (a) 250 and (b) 300 nodes

The networks generated by the CGPANN (shown in Fig. 4) for almost all the number of nodes show that the network output depends on the single input in most of the cases. The networks also show that their output is dependant only on the most recent preceding input. This means that only the recent data inputs effect the output and few inputs are enough for deciding the outputs. The trend is affected by international currency exchange market continuously but only the previous ten days data is used as the input to the model. This data is used to predict the currency exchange trend for the eleventh day and so on so forth. The network is almost independent of older inputs. In most of the cases the tenth input is used which is the most recent input; and in fact the most important input as it decides the outputs of the network. It can also be seen from the networks that all of them has been evolved from least number of nodes. This makes the system proficient as it makes use of only one or two recent data for deciding the output. Hence there is least need of all the input data and older data can easily be neglected in process of generating outputs. Fig.4 shows the phenotypes evolved by the CGPANN model for the prediction of currency exchange rate.

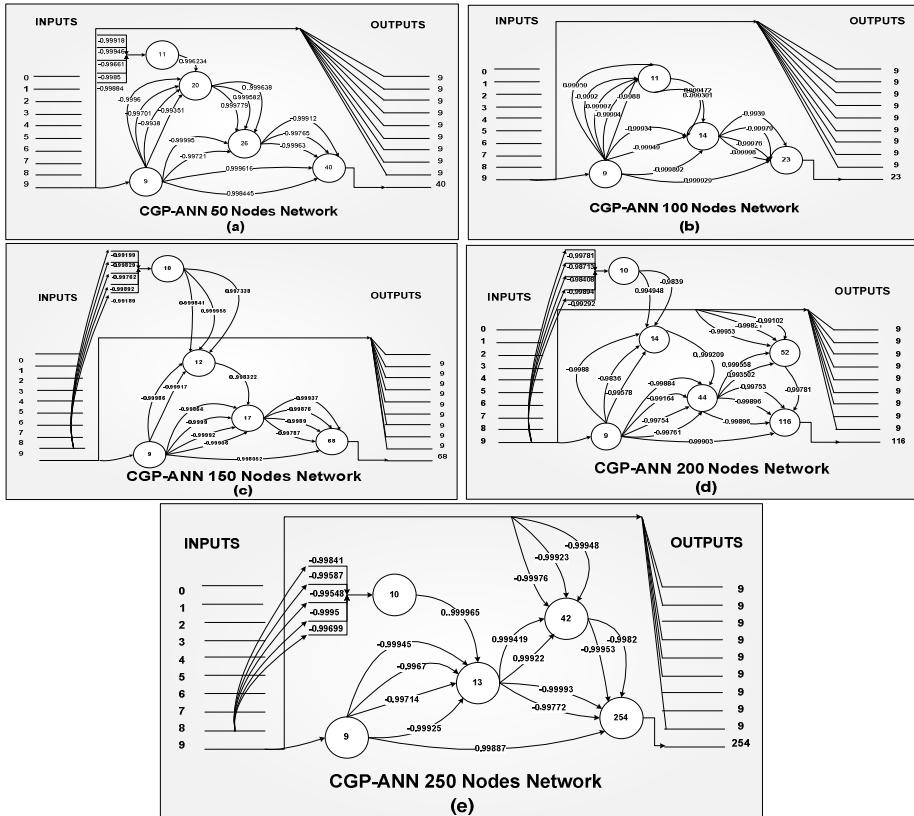


Fig. 4. Networks generated by CGPANN for (a) 50, (b) 100, (c) 150, (d) 200 and (e) 250 nodes

8 Conclusion and Future Work

In this paper a recently introduced Cartesian Genetic Programming evolved Artificial Neural Network (CGPANN) is explored for implementation of prediction model of foreign exchange rates. The experiments conducted in this paper for the prediction of trends in currency exchange via CGPANN have shown better and accurate results. As the network is evolved, it is observed that in almost all the cases the output is dependent on the recent inputs and least number of inputs is required to decide the output. This makes the network simple as prediction can be done with few and recent input data. The older and large amount of inputs is used during the training process. Hence once the network is trained, only one or two recent inputs are sufficient to predict the output (trends for the next day).

Future work in this area can be carried out in various fields, where forecasting of data can be done on the basis of historical records. These areas may include weather forecasting system, sports forecasting system, river flow forecasting system [16], elections forecasting system, chemical and physical laboratory results forecasting, disaster forecasting and many other fields where historical data is available to be utilized to forecast the future data. CGPANN produces an efficient prediction model for such systems that will forecast the output swiftly and accurately on the basis of least amount of previous data.

Acknowledgments. The authors acknowledge the support of the **National ICT R & D Fund Islamabad, PAKISTAN** for funding this research.

References

- [1] Philip, A.A., Tofiki, A.A., Bidemi, A.A.: Artificial Neural Network Model for Forecasting Foreign Exchange Rate. *World of Computer Science and Information Technology Journal* 1(3), 110–118 (2011)
- [2] Gould, J.H.: Forex Prediction Using An Artificial Intelligent System. Diss., Oklahoma State University (2004)
- [3] Zhang, G., Hu, M.Y.: Neural Network Forecasting of the British Pound/ US Dollar Exchange Rate. *Omega, Int. J. Mgmt. Sci.* 26(4), 495–506 (1998)
- [4] Kryuchin, O.V., Arzamastsev, A.A., Troitzsch, K.G.: The prediction of currency exchange rates using artificial neural networks. *Exchange Organizational Behavior Teaching Journal* (4) (2011)
- [5] Refenes, A.N., Azema-Barac, M., Chen, L., Karoussos, S.A.: Currency Exchange Rate Prediction and Neural Network Design Strategies. *Neural Computing & Applications* 1(1), 46–58 (1993)
- [6] Kadilar, C., Alada, H.: Forecasting the Exchange Rate Series with ANN: The case of Turkey. *Economics and Statistics Changes* 9, 17–29 (2011)
- [7] Khan, G.M., Khan, S., Ullah, F.: Short-term daily peak load forecasting using fast learning neural network. In: IEEE Int. 11th International Conference on Intelligent Systems Design and Applications, ISDA (2011)
- [8] Kamruzzaman, J., Sarker, R.A.: Forecasting of Currency Exchange Rates Using ANN: A Case Study. In: Proceedings of the IEEE International Conference on Neural Networks & Signal Processing, vol. 1, pp. 793–797 (2003)

- [9] Poli, R.: Parallel Distributed Genetic Programming Applied to the Evolution of Natural Language Recognisers. *Evolutionary Computing*, 163–177 (1997)
- [10] Bidlo, M.: Evolutionary Design of Generic Combinational Multipliers Using Development. In: Kang, L., Liu, Y., Zeng, S. (eds.) ICES 2007. LNCS, vol. 4684, pp. 77–88. Springer, Heidelberg (2007)
- [11] Haider, A., Hanif, M.N.: Inflation Forecasting in Pakistan using Artificial Neural Networks. *Pakistan Economic and Social Review* 47, 123–138 (2009)
- [12] Jeng, J.T., Tain, L.T.: An approximate equivalence neural network to conventional neural network for the worst-case identification and control of nonlinear system. In: IEEE International Joint Conference on Neural Networks, IJCNN, vol. 3, pp. 2104–2108 (1999)
- [13] Miller, J.F., Thomson, P.: Cartesian Genetic Programming. In: Poli, R., Banzhaf, W., Langdon, W.B., Miller, J., Nordin, P., Fogarty, T.C. (eds.) EuroGP 2000. LNCS, vol. 1802, pp. 121–132. Springer, Heidelberg (2000)
- [14] Floreano, D., Durr, P., Mattiussi, C.: Neuroevolution: from architectures to learning. *Evolutionary Intelligence* 01, 47–62 (2008)
- [15] Gomes, F., Schmidhuber, J.: Accelerated Neural Evolution through Cooperatively Coevolved Synapses. *Journal of Machine Learning Research* 9, 937–965 (2008)
- [16] Atiya, A.F., El-shoura, S.M., Shaheen, S.I., El-Sherif, M.S.: A Comparison between Neural-Network Forecasting Techniques- Case Study: River Flow Forecasting. *IEEE Transactions on Neural Networks* 10(2), 402–409 (1999)
- [17] Chen, A.P., Hsu, Y.C., Hu, K.F.: A Hybrid Forecasting Model for Foreign Exchange Rate Based on a Multi-neural Network. In: IEEE Fourth International Conference on. Natural Computation, ICNC 2008, pp. 293–298 (2008)
- [18] Azoff, E.M.: Neural network time series forecasting of financial markets. John Wiley and Sons Inc., Chichester (1994)
- [19] Pacelli, V., Bavelacqua, V., Azzolini, M.: An Artificial Neural Network Model to Forecast Exchange Rates. *Journal of Intelligent Learning Systems and Applications* 3(2), 57–69 (2011)
- [20] Igel, C.: Neuroevolution for reinforcement learning using evolution strategies. In: The 2003 Congress on Evolutionary Computation, CEC 2003, vol. 4. IEEE (2003)

Coastal Hurricane Inundation Prediction for Emergency Response Using Artificial Neural Networks

Bernard Hsieh and Jay Ratcliff

US Army Engineer Research and Development Center, Vicksburg, Mississippi, 39180, USA
hsiehb@wes.army.mil

Abstract. Emergency managers require both fast and accurate estimates of hurricane inundation to make critical decisions about evacuations, structure closures, and other emergency response activities before, during, and after events. Probability analyses require multiple simulations which, generally, cannot be performed with the physics-based models under the time constraints during emergency conditions. To obtain highly accurate results with a fast turnaround computation time a “surrogate” modeling approach is employed. This surrogate modeling approach uses an extensive database of storms and storm responses and applies “smart” pattern recognition tools such as Artificial Neural Networks (ANN) as well as interpolation techniques. The goal is to provide forecasts of hurricane inundation and waves with the accuracy of high-resolution, high-fidelity models but with very short execution time (minutes). The city of New Orleans as well as surrounding municipalities along the Gulf of Mexico coastal area encompasses the region used to demonstrate this approach. The results indicate that the developed surge prediction tool could be used to forecast both magnitude and duration to peak surge for multiple selected points in a few minutes of computational time once the storm parameters are provided. In this paper, only results of surge magnitude are presented.

Keywords: Storm Surge Prediction, surrogate modeling, neural networks, multilayer perceptron.

1 Introduction

The most severe loss of life and property damage occurred in New Orleans, Louisiana, USA which flooded as the levee system catastrophically failed; in many cases hours after the storm had moved inland. At least 1,836 people lost their lives in Hurricane Katrina and in the subsequent floods, making it the deadliest U.S. hurricane since the 1928 Okeechobee Hurricane. The storm is estimated to have been responsible for \$81.2 billion (2005 U.S. dollars) in damage, making it the costliest natural disaster in U.S. history. The city of New Orleans as well as surrounding municipalities is located well within hurricane striking distances along the Gulf of Mexico and within the north central region of the Gulf that has the highest probability of a major hurricane strike.

The Corps of Engineers of US (USACE) is committed to protection of these areas and has created a Hurricane and Storm Damage Risk Reduction System (HSDRRS)

consisting of hundreds of miles of levees, integrated with flood walls, locks, flood gates, and other water control structures. The exact timing of the storm surge to determine the gate operations during extreme weather conditions is critical to ensure the safety of populated areas and a minimal amount of flooding. The forecasting capability for surge behavior during the initial (critical) hours of approaching storms is particularly important to the decision making to support storm preparations and emergency operations. An operational storm surge water level forecast for the greater New Orleans area and coastal Louisiana has come to be realized as essential to provide critical data that can be used to potentially reduce damages, risk, and save lives.

2 A Surrogate Neural Network Storm Surge Model

Several storm surge forecasting systems have been studied by using Computational Intelligence/statistics methods ([1], [2], [3], [4], [5], [6], [7], and [8]) but this approach uses “surrogate modeling approach to simulate multiple selected surge points. Recently, a very successful interagency team effort in the United States has been formed to perform operational storm surge numerical modeling (Figure 1) in response to hurricane events in the coast of Gulf of Mexico. Application of the ASGS (ADCIRC Surge Guidance System) has demonstrated that due to the computational needs of numerical models, and real time operational requirements, mandatory compromises limit implementation aspects such as the model geometry size and region domain.

The Advanced CIRCulation model (ADCIRC) is a two-dimensional, depth-integrated, barotropic time-dependent long wave, hydrodynamic circulation model. Although the current warning system has been greatly improved through the use of a region specific efficient model geometry (SL15 Light), additional tools have the potential to more quickly and more accurately provide wind, wave height and surge levels that affect all critical coastal structures and low lying flood prone areas. The potential value of promising alternative tools should be explored. A number of significant compromises had to be made relative to the very detailed modeling and coupled wave-surge models that have been set up and validated for the region.

To avoid these significant compromises, a complimentary approach is proposed which combines the strength of both physical-driven (coupled surge and wave numerical modeling done at highly detailed, fine resolution) and data-driven methods (artificial neural networks – ANNs as well as other computational intelligence components) methods to form a predictive knowledge base to estimate the water levels including magnitude and duration to peak surge for selected locations. This approach is called surrogate modeling approach. A surrogate model is an engineering method or alternative model used when an outcome of interest cannot be easily directly measured nor quickly computed with physics-based models, so a model of the outcome is used instead. The technical tools include the well-validated full SL15 ADCIRC numerical storm surge model application (simulator), unsupervised (clustering) and supervised (prediction) ANNs. In this approach new situations (or events) can be performed with additional storm surge simulations, done offline, which can be added (retrained) to the existing knowledge base.

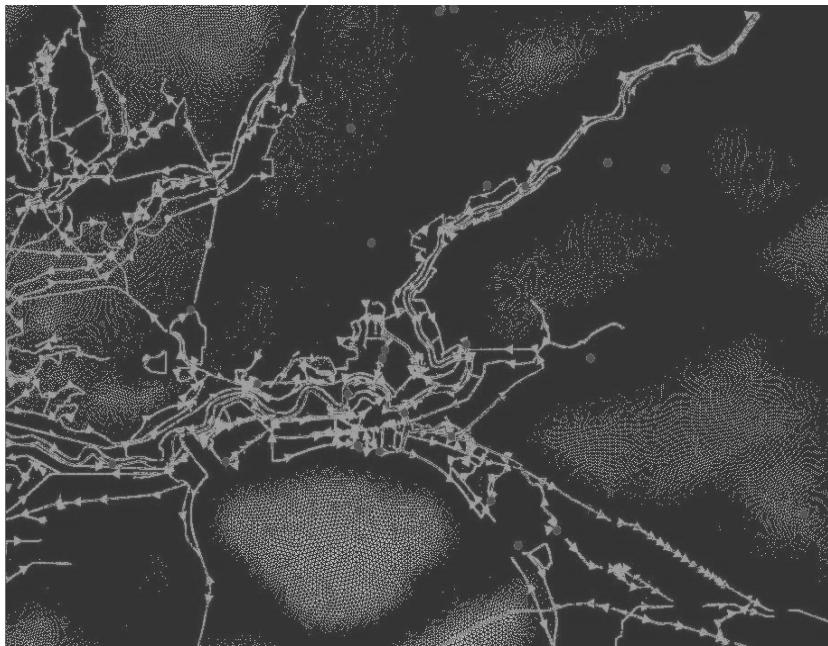


Fig. 1. A numerical storm surge model with computational mesh, levee stream (white), and selected key observed locations (gray)

3 Knowledge Base Development for a Surrogate Neural Networks Model and Basic Data Analysis

Peak surge prediction is the first goal of the ANN model. An initial form of a knowledge database is created to form a linkage between the ANNs and ADCIRC model peak surge results. The goal of this phase is primarily to estimate model peak surges at selected point locations. The main effort is to quantify the relationship between input parameters (such as track or maximum wind speed) and peak surge outputs for all selected interest points from the physics-based coupled wave-surge modeling system (ADCIRC / STWAVE), and convert to a data-driven system.

3.1 Selection of Forecasting Points

The ADCIRC model is executed with a high resolution triangular mesh containing million of nodes and elements. A small subset of nodes is selected to implement the ANN model. It is important that these nodes (points) are selected at key locations which can provide emergency decision makers with appropriate information needed to make critical time constrained decisions. Peak surge water levels, wind speeds, and wave heights are needed to know which flood gates to close/open, when to open/close gates and other structures such as which pumps (and times) to operate. Forecast point locations are selected near critical flood protection system components as well as at

key gages (measured observation stations) to enable model comparison and validation. In the New Orleans area a total of 30 point locations have been selected in the south eastern Louisiana (Figure 2) some of which are used during operational forecasting efforts. It is important to select points that are spatially well distributed throughout the area of vulnerability.

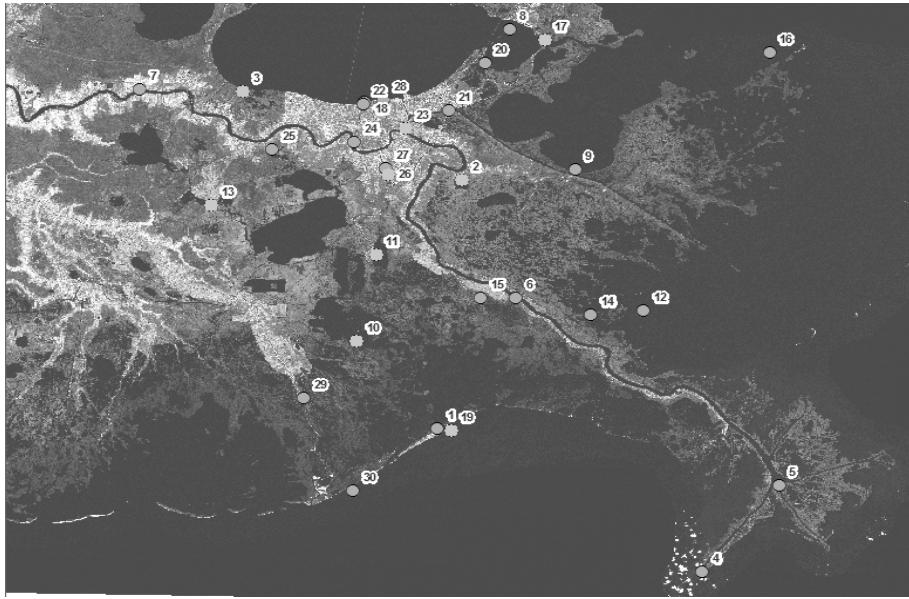


Fig. 2. 30 selected locations (gray color) as forecast points from surrogate model

3.2 Determination of Input/Output Parameters for ANNs Model

The ANN model is “trained” using selected components (parameters) of the physics-based system (in this case ADCIRC/STWAVE). The goal is to use the ANN model to predict storm response surges. In order to achieve this goal a strategy must be employed to select the key parameters from the physics-based system. An understanding of the numeric physics based model is important. This enables selection of the key “force” elements that most affect the final results. In this case the storms are the key force element. Quantitative characteristics of the key force elements must be defined to provide ANN input. An understanding and knowledge of how the physics-based system changes in force and their affects on results both spatially and temporally will enable selection of relevant ANN parameters. Key to the ANN (or any pattern recognition) method is that all parameters vary either spatially and/or temporally over the domain.

The final parameters selected for this effort are shown in Table 1. Sensitivity analyses were performed to determine these most significant parameters and the reliability for multiple point ANNs simulations. The force or input parameters can be grouped into geometric and storm related components. The geometric components are the distance between the forecast point and the storm land fall location, and the

angle of storm approach relevant to the forecast point location at time of landfall. The storm force components include the central pressure at the time of landfall, the average forward velocity, the radius to maximum winds, and the maximum wind speed achieved at the forecast point location over the entire storm event. The output parameters selected for ANN model are the peak storm surge. Usually, the distance from storm landfall location, angle of storm approach, and local maximum wind speed are considered as local forcing parameters while remaining forcing parameters are regarded as global forcing parameters.

Table 1. ANNs model input and response output parameters for storm surge model

| Forcing (Input) Parameters | Result (Output) Parameters |
|---|----------------------------|
| Geometry | |
| Distance from storm landfall location (local) | |
| Angle of storm approach (very minor impact – local) | Peak Storm Surge |
| Storm Force | |
| Central Pressure (global) | |
| Average Forward Speed (global) | |
| Radius to Maximum Winds (global) | |
| Local Maximum Wind Speed (local) | |

It is noted that due to some data error involved in the system 4 simulation runs as well as 2 selected points (point 25 and 29) are eliminated. This results in 442 sets of storms and associated parameters at 28 saved points as the knowledge base for the ANNs model. The typical central pressure and over 442 storm events are shown in Figure 3. The corresponding output functions, surge height and duration, are plotted as Figure 4.

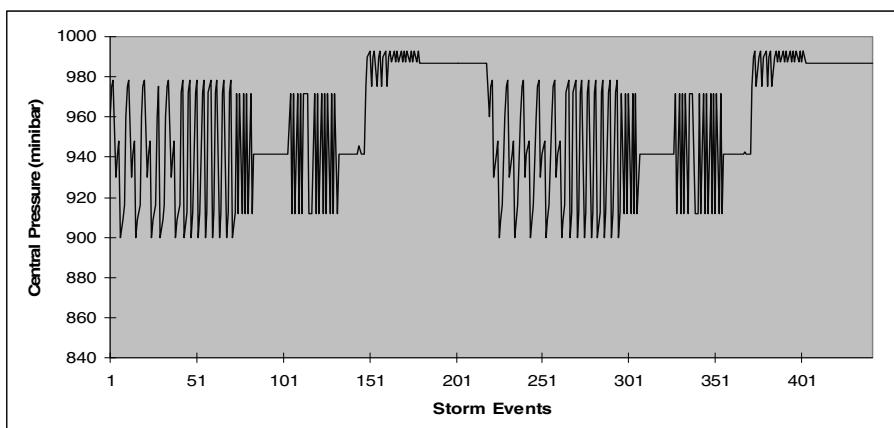


Fig. 3. Global storm parameter (central pressure- mb) for 442 ADCIRC physical model runs

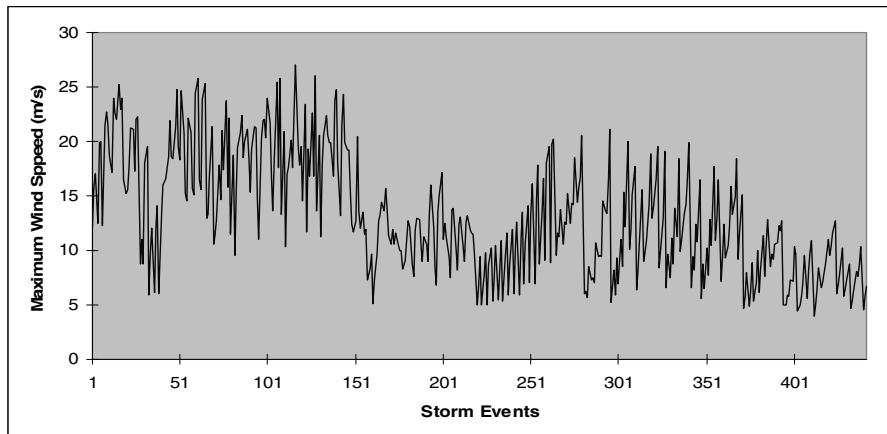


Fig. 4. Peak storm surge (ft) at gauge 3 for 442 ADCIRC model runs

3.3 Preliminary Multiple Linear System Identification and ANNs Model Design

Before dealing with a complex nonlinear system to quantify the relationship among parameters, a linear system, such as correlation coefficient analysis can be used as a preliminary analysis tool to determine the approximate functionality. Therefore, one set of correlation coefficients (maximum surge height) with six inputs and one output system are computed (Figure 5). Due to very low correlation coefficients associated with angle of storm approach for both magnitude and duration of surge; the system is reduced to five inputs/one output structure. Figure 5 shows two parameters (central pressure and distance from storm landfall location) that are negative related to corresponding surge height. This indicates an inverse physical relationship to surge for these two parameters. The local maximum wind speed is a dominate surge producing parameter.

Based on the above analysis, a nonlinear neural network model with feed forward architecture can be assumed as Figure 6. This architecture represents a system with 3 global inputs, N locations, 3 hidden nodes (one hidden layer and if 3 hidden nodes are selected), and n corresponding maximum surge magnitude and duration. The maximum hidden nodes which depend on the prevention from over-training under a set of optimal weights are adjustable. For example, if this system involves 28 prediction points, the size of the neural network is $59 \times 3 \times 28$ with a total number of 261 weights .It should be noted that the input arrays between surge height and surge duration are somewhat different sign although the values are the same. Therefore, the system is considered as two individual response structures – one for surge height and the other for surge duration. This paper only presents the results for surge height prediction.

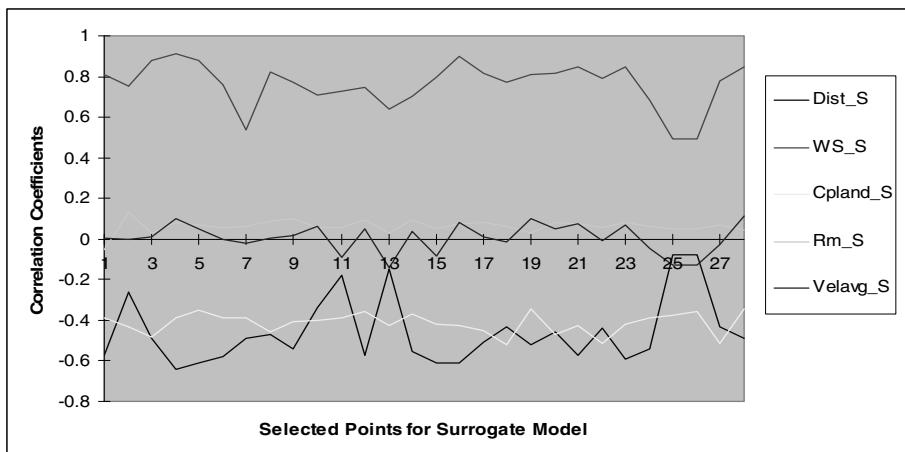


Fig. 5. 28 points correlation coefficients for storm parameters response to peak surge

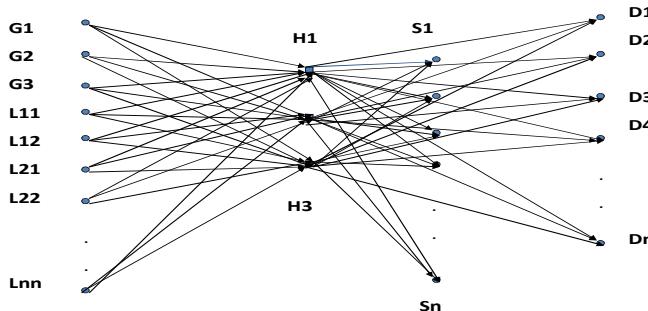


Fig. 6. A neural networks model for N selected points with 3 hidden nodes, N peak surge and N duration outputs architecture

4 Identification and Test for Surrogate Neural Networks Model

After initial model test based on general over-trained prevention, training algorithms, and training strategies using two most offshore points (point 4 and point 5), it found this surrogate model could get a satisfactory agreement with the multilayer perceptron procedure. The proposed ANNs model is further examined by its accuracy up to all 28 points. NeuroSolutions [9] is used to perform this analysis.

4.1 Optimal Point Selection for Surge Prediction

Due to the fact that strength of response from a set of locations is not equally distributed in a given domain, the accuracy of storm surge prediction using the surrogate approach will not readily provide the same results across the domain. It is important to determine the point(s) with the highest prediction accuracy. Table 2 summaries 5 different tests which extend the response points to multiple point bases.

Table 2. Optimal point selection for surge prediction

| Selection Reason | Total Point Number | Selected Points | Number of Hidden Nodes | Average CC |
|-----------------------------|--------------------|---|------------------------|------------|
| Priority | 9 | 2, 3, 10, 11, 13, 17, 19, 23, 26 | 5 | 0.932 |
| A Statistical Similar Group | 11 | 1, 4, 10, 11, 13, 18, 19, 22, 26, 27, 28 | 4 | 0.913 |
| Better Individual Response | 15 | 3, 4, 6, 7, 8, 12, 14, 17, 18, 20, 21, 22, 23, 24, 28 | 3 | 0.934 |
| Full Scale | 28 | Every points | 2 | 0.770 |

Based on Table 2, nine priority points are chosen as the most critical. The considerations are dependent on key locations which can provide emergency decision makers with water levels and time series data needed for critical decisions (time to close flood gates, start pump stations, etc.), near critical flood protection system components, at key gauge locations for comparison and validation, and spatially distributed through area of vulnerability. An unsupervised ANNs (SOFM) is used to cluster four different response groups with similar statistical parameters (mean, standard deviation, skewness, maximum, and minimum). The first group involves 11 response points. The average CC is computed from each test case. The results show the optimal maximum number of points for creating a surge prediction system is between 9 and 15. The lower CC from a similar group could be those points that are widely spatial distributed. Figure 8 shows the comparison between ADCIRC model simulation and ANNs results for point 23 (9 points priority case).

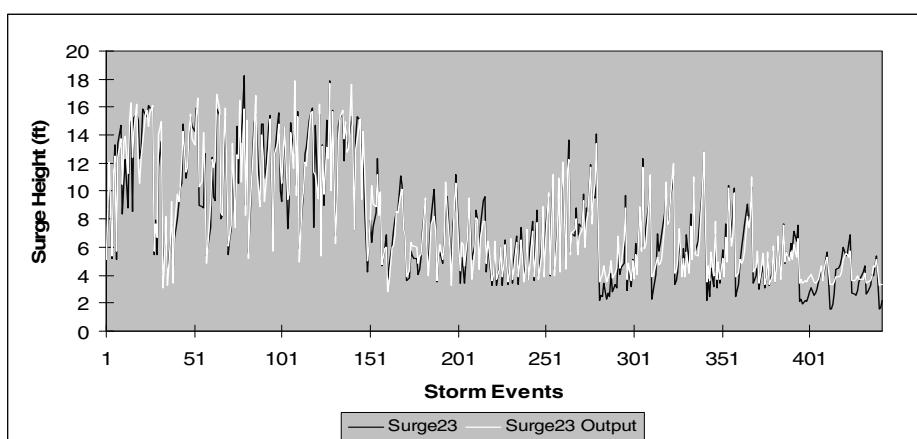


Fig. 7. Training results of station 23 for 9 priority points surge (ft) model (black color represents ANNs simulation and white color shows ADCIRC model simulation)

4.2 Surrogate Model Simulation for 9 Priority Point's Case

This 9 priority points surrogate ANNs model can be used to demonstrate the surge response (magnitude and duration) under very short period (less than few minutes) once new storm parameters are provided. Figure 9 displays the surge height of simulated 50 storm events for 9 priority points. It is note that the maximum local wind speeds as a major forcing parameter for these 50 storm events are presented in this figure as the impact contribution.

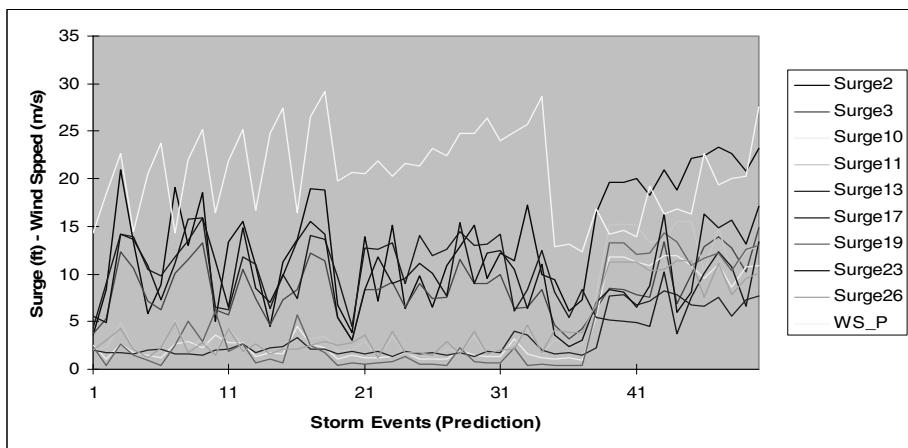


Fig. 8. 50 storm surge events prediction for 9 priority point's model with 442 events as knowledge base

5 Conclusions

This work demonstrates successfully use of ANNs to quantify the relationship between storm forcing as well geometry and response parameters (maximum surge magnitude and duration) from a knowledge base of 442 storm surge numerical model simulations. The city of New Orleans as well as surrounding municipalities along the Gulf of Mexico coastal area is used as the demonstration site. The developed “static” mode surrogate surge prediction tool can be used to predict surge response and duration to peak surge with multiple selected points within minute’s turnaround time once the storm parameters are provided. This effort investigates the most significant procedures for developing an ANN model from training strategies, algorithm selection, and prevention from overtraining consideration approaches. The results indicate that the surge is the most influenced by local maximum wind. The “dynamic” operational surrogate ANN model is being developed for further practical application.

Acknowledgement. The U.S. Army Corps Engineers Coastal Flood Reduction and Protection Program, Engineering Research and Development Center (ERDC) funded this work. Permission was granted by the Chief of Engineers to publish this information.

References

1. Walton, T.L.: Short Term Storm Surge Prediction. *Journal of Coastal Research* 21(3), 421–429 (2005)
2. Sztobryn, M.: Forecast of Storm Surge by means of Artificial Neural Network. *Journal of Sea Research* 49, 317–322 (2003)
3. Prouty, D., Tissot, P., Anwar, A.: Using Ensembles of ANNs for Storm Surge Predictions in the North Sea (Abstract). In: 6th Conference of AI Applications to Environmental Science (2008)
4. Prouty, D., Tissot, P., Anwar, A.: Using ANNs for Predicting Storm Surge Propagation in North Sea and the Thames Estuary, (Abstract). In: 4th Conference of AI Applications to Environmental Science (2006)
5. Lee, T.: Neural Network for Prediction of the Short-Short-Term Storm Surge in Taichung Harbor, Taiwan. *Engineering Applications of Artificial Intelligence* 21(1), 63–72 (2008)
6. You, S.H.: Storm Surge Prediction Using an Artificial Neural Networks. *Nat Hazards* 51, 97–114 (2008)
7. Gautam, D., Holz, K., Meyer, Z.: Forecasting Storm Surges Using Adaptive Neuro-Fuzzy Systems. In: Proceeding of 4th International Conference on Hydroinformatics, pp. 38–43 (2000)
8. Siek, M., Solomatine, D.: Real-Time Data Assimilation for Chaotic Storm Surge Model Using Neural Network. *Journal of Coastal Research SI* 64, 1189–1194 (2011)
9. Neurosolutions v5.0: Developers level for window, NeuroDimensions, Inc., Gainesville, Florida (2003)

Crossroad Detection Using Artificial Neural Networks

Alberto Hata, Danilo Habermann, Denis Wolf, and Fernando Osório*

University of São Paulo, Mobile Robotics Lab – LRM/ICMC,

São Carlos, São Paulo, Brazil

{hata,habermann,denis,fosorio}@icmc.usp.br

Abstract. An autonomous ground vehicle has to be able to execute several tasks such as: environment perception, obstacle detection, and safe navigation. The road shape provides essential information to localization and navigation. It can also be used to identify reference points in the scenario. Crossroads are usual road shapes in urban environments. The detection of these structures is the main focus of this paper. Whereas cameras are sensible to illumination changes, we developed methods that handle LIDAR (Light Detection And Ranging) sensor data to accomplish this task. In the literature, neural networks have not been widely adopted to crossroad detection. One advantage of neural networks is its capability to deal with noisy data, so the detection can be performed even in the presence of other obstacles as cars and pedestrians. Our approach takes advantage of a road detector system that produces curb data and road surface data. Thus we propose a crossroad detector that is performed by an artificial neural network and LIDAR data. We propose two methods (curb detection and road surface detection) for this task. Classification results obtained by different network topologies have been evaluated and the performance compared with ROC graphs. Experimental tests have been carried out to validate the approaches proposed, obtaining good results when compared to other methods in the literature.

Keywords: Crossroad detection, 3D LIDAR, road surface detection, curb detection and artificial neural networks.

1 Introduction

Autonomous ground vehicles may be a solution to increase highway capacity and traffic flow due to their capability to optimize the navigation of cars in the streets. Another benefit of them is the possibility of reducing accidents through collision avoidance, adaptive cruise control, and lane departure warning systems [1].

In order to avoid collisions and navigate safely, autonomous vehicles must rely on a robust environment perception system. The robot uses its sensors to identify obstacles as well as other vehicles, pedestrians, poles and trees to determine traversable spaces [2]. Once the vehicle is able to identify the elements

* The authors gratefully acknowledge the financial assistance from CAPES and would like to thank the Mobile Robots Laboratory for its support.

of an urban scenario, this information can be used to assist the vehicle navigation or match this information against a map to determine its localization. Road shapes are an example of structures that provide such possibilities. While different shapes are present in urban environments, they offer important localization information, in special when using topological maps [2]. Considering the importance of identifying road shapes, this paper focuses on the detection of crossroads.

In previous work [3], the use of road shapes for topological localization has already been explored using camera images. Whereas these sensors are sensible to illumination changing, LIDAR (Light Detection And Ranging) sensors are adopted in this paper, as they are not affected by this issue. We used a Velodyne HDL-32E 3D LIDAR sensor mounted on an experimental vehicle for data gathering and algorithms performance evaluation.

Currently there are many crossroad detection algorithms available in the literature [4–7]. In general, these methods use statistical analysis to detect crossroads. Consequently neural networks have not been widely explored to perform such task. In addition, these methods [5, 7] require preprocessing step to remove undesired obstacles from laser data (e.g. pedestrians and cars). Neural networks are known by its capability to deal with noisy data. Therefore the sensor data can be directly processed using these networks.

In our approach we take advantage of perception components of our autonomous car to identify crossroads. Specifically the road information obtained by the perception system is evaluated by a neural network to detect crossroads. Thus, all detection process is performed by a single neural network. Two approaches for crossroad detection have been developed base on that: curb detection-based and road surface detection-based. Both are evaluated with different neural network topologies.

The paper is organized as follows: Section 2 presents recent works in crossroad detection. Section 3 describes the crossing detection method by using road navigable area data and using lane curb data. Section 4 presents and compares the neural network classification results obtained by both approaches. Finally Section 5 discusses the conclusions of this work and the possible future works.

2 Related Works

Previous studies have used video cameras to detect road intersections [3], however in this article our concern regards only to techniques for detecting intersections based on LIDAR point clouds.

A crossroad detection method using 3D LIDAR data was proposed in [4]. The captured laser data is stored in a binary occupancy grid and the distance transform operator is applied. The presence of maximum value in the center of this grid indicates a crossing. Once the crossing detection is confirmed the branches are extracted by iteratively exploring the grid cells with values next to the maximum. In [5] a elevation map was used to represent the obstacles. To detect the crossing as far as possible, an admissible space is extracted by

searching the free region ahead the sensor. A virtual 2D LIDAR is used to trace beams in the elevation map, inside the admissible space. The obtained laser distance function is analyzed and its peaks are extracted through a peak-finding algorithm. The number and position of peaks are statistically processed to detect the crossroads. The main drawback of this method is that scans disturbed by obstacles are not treated, making it difficult to adopt in real environments.

Similarly as the previous work, [6] also uses virtual LIDAR for crossroad detection. However it preprocesses the elevation map to remove cars and pedestrians in order to suppress occlusions. Then the normalized lengths of the 360 beams are used as features to train a support vector machine model. An improvement of this work was presented in [7]. In this a 3D laser registration method that can match different scans is used to deal with dynamic objects. This makes the crossing detection more robust when dealing with occluded scenes.

The crossing detector method proposed in this paper takes advantage of outputs generated by road detector that composes the perception system of our autonomous vehicle. Thus the crossroad detection is performed basically by a neural network that can also handle noisy data as scans with occlusion. In this way any additional preprocessing in the sensor data is unnecessary for the detection.

3 Crossroad Detection

The detector consists in a neural network that classifies urban road data into crossing or non-crossing. We can extract two data from urban road points captured by a 3D laser: curb and road surface. Each one is obtained by two different methods that are described in Subsection 3.1 and 3.2. Later the integration of neural networks to road detector is detailed in Subsection 3.3.

3.1 Curb Detection

The curb detection method extends the obstacle detector method presented by [2]. In this approach obstacles are detected by analyzing the scan ring compression. This analysis is explained by the fact that the presence of obstacles causes the reduction of the distance between consecutive rings. Different obstacles create different compression patterns. Curb obstacles makes the sensor return distances between rings smaller than generated by flat terrains and larger than generated by walls. When the 360° laser sensor sweeps a flat terrain, the distance between successive rings are calculated by the following expression:

$$\Delta r_i^{plane} = r_i^{plane} - r_{i-1}^{plane} \quad (1)$$

$$= \frac{h}{\tan \theta_i} - \frac{h}{\tan \theta_{i-1}} \quad (2)$$

$$= h((\tan \theta_i)^{-1} - (\tan \theta_{i-1})^{-1}) \quad (3)$$

where i corresponds to the ring index, Δr_i^{plane} represents the distance, r_i^{plane} the radius of the i -th ring, h denotes the sensor height and θ_i is the angle formed

by the sensor beam and the ground. After the beam intercepts an obstacle we check if the compression Δr_i is inside an interval to classify as curb:

$$\mathcal{I}_i = [\gamma \Delta r_i^{plane}, \delta \Delta r_i^{plane}]$$

where \mathcal{I}_i is the valid range for the i -th ring, γ is a constant that defines the lower bound, δ the upper bound with $\gamma > \delta$.

In order to prevent exhaustive comparison between points placed in adjacent rings, each ring data is stored in a circular grid with fixed size. The grid cell stores the mean value of the point (x, y, z) position and the distance d to the sensor. Points with height (z) higher than a threshold are automatically discarded, so only points situated next to the ground are stored. The distance Δr_i is computed by subtracting the d value of the subsequent cell. Cells with Δr_i inside \mathcal{I}_i are considered as curb candidates. Classifying a laser data captured in an urban environment shown in Figure 1(a), the compression analysis could discard from the candidates set obstacles as walls, lamp posts and trees.

Whereas curb structures have height variation relative to road plane, this assertion can be checked in curb candidates data to remove false positives. As proposed in [8], a differential filter was applied. In this method a one-dimensional convolution mask is used. This mask comprises in a vector with values $(-2, 0, +2)$. For each ring cell, its adjacent cells are taken in account for this operation. Thus, convolution results higher than threshold t_s are discarded. With this filter, points situated in planar regions (e.g. sidewalk, road) are removed from curb candidates set as can be seen in Figure 1(a). The remaining points of curb candidates are classified as curb. In this work we adopted respectively the following values for γ , δ and t_s : 1.37543, 0.012462 and 0.123592. Occlusion situations are not filtered by the curb detector with the intent to train

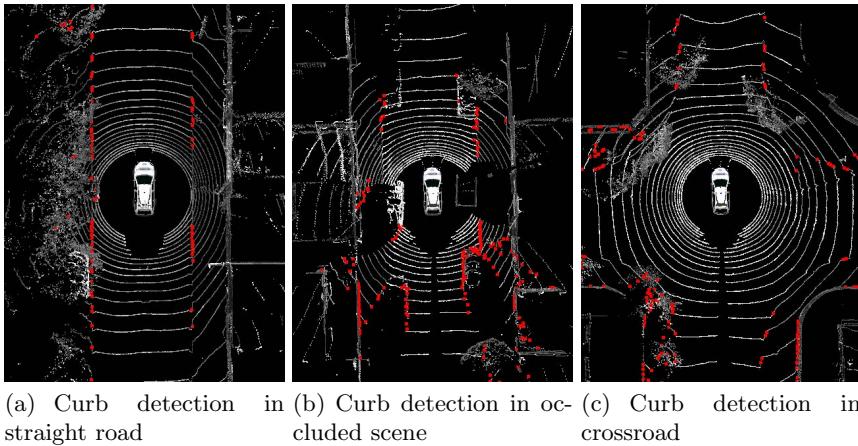


Fig. 1. Curb detection in different road scenes. Red points represents the detection results.

the neural network for these situations (Figure 1(b)). A sample of curb detection in crossroads is illustrated in Figure 1(c).

3.2 Road Surface Detection

The method used to detect free road surface is based on the same concept presented in [9], which tries to generate the navigable area without first detecting obstacles. Let $P_i = (x_i, y_i, z_i)$ be the rectangular coordinate representation of the points coming from a scan of the 3D laser sensor which are within range $5m < x < 25m$ and $-20m < y < 20m$ (Figure 2). These points are then distributed in a grid semi-circular divided into $n = \frac{2\pi}{\Delta\alpha}$ circular sections (S_1, \dots, S_n), where $\Delta\alpha$ is the angular resolution of the semicircular grid. Each circular section is divided into m bins as in [10].

However, in this case, the maximum value of m is 32, which is the same number of rings from Velodyne point cloud. Let P'_{jk} be the set of points belonging to S_j ($j = 1, \dots, n$) and to b_k ($k = 1, \dots, m$). Let P''_{jk} be the set of lowest elevation points chosen from each P'_{jk} set. In this manner the P''_{jk} set is formed by the points of lower elevation of each ring of a point cloud (b_k) contained in each circular section S_j . Similar to [10] we convert 3D points from P''_{jk} into 2D points. Then each point from P''_{jk} is formed by the 2D points $(xy_i = \sqrt{x_i^2 + y_i^2}, z_i)$. The goal now is to determine whether or not these points belong to the ground.

Initially, we take the first point of the set (the point belonging to the P''_{j1} bin) and we analyze if its elevation is inferior to a threshold t_h . If so, we say that this point belongs to the ground. Otherwise, we label this point as non-ground and take the next point (P''_{j2}) to do the same procedure and so on. By identifying the first point of P''_{jk} belonging to the ground, we start to analyze the elevation angle (β), according to equation 4, where the initial value of k represents the index of the first point labeled as ground point. If β is inferior to a threshold t_β we label $k + 1$ point as belonging to ground. All ground points will serve as a basis for crossroad detection. Red points in Figure 3 represent the ground found

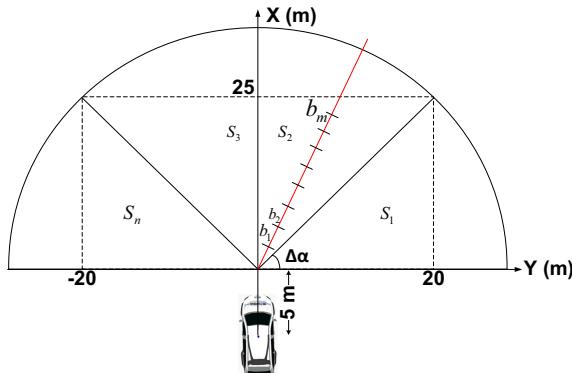


Fig. 2. Partitioning the 3D space in n circle section and m bins

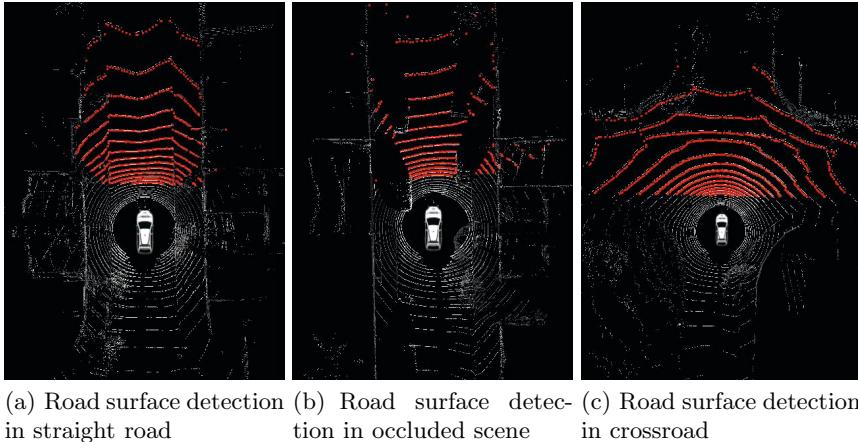


Fig. 3. Road surface detection. Red points represents the detected road surface.

by the method. Figure 3(b), shows the result of the method in occluded scenes. Even in this situation the method has proved robust. We adopted respectively values for $\Delta\alpha$, t_h and t_β : 0.1° , $0.2m$ and 0.2 .

$$\beta = \left| \frac{z_{b_{k+1}} - z_{b_k}}{xy_{b_{k+1}} - xy_{b_k}} \right| \quad (4)$$

3.3 Neural Networks for Crossroad Detection

The output of curb detector and road surface detector were adapted to feed the neural network. Thus the obtained curb points and road surface points were placed in a grid with the purpose of fixing the number of points that will be inputted in the network. All grid cells have same dimensions. The first step of building the grid is store in each cell the number of points that falls in. Later this number is divided by the maximum number of points observed in the cells of the grid. This results in a grid carrying the proportion of points in each cell and limits the cell value into $[0, 1]$ range. Thereby cells with few points will receive low values and cells with large amount of points will receive higher values. In case of road curb data, standard deviation of the left and right curbs points in x and y directions were also computed to feed the network. Therefore in crossroads the y deviation will receive larger values than in straight roads.

4 Experimental Results

The network training for crossroad classification was performed in two ways: using road curb examples and using road surface examples. Each data set was trained separately in tree network topologies. (Despite of single layer perceptron

being the most appropriate network for two class classification problems, we also verified results for other configurations). The following topologies were tested: x-2, x-5-2 and x-10-2, where the input layer size x was 210 in case of feeding road surface data and x was 404 in case of feeding curb data. Each output neuron was designated to crossroad or non-crossroad classes. All adopted networks were fully-connected with feed-forward processing.

In the learning process we used holdout validation method, so the manually labeled data set was split in $\frac{2}{3}$ for training and the remaining $\frac{1}{3}$ for validation. We also considered obstructions (presence of cars and pedestrians) in these databases. The transfer and learning function used was respectively, logistic and Resilient Back-propagation. The number of cycles (epochs) was limited by 1000. Obtained results for different topologies were evaluated through accuracy and Mean Squared Error. Receiver Operation Characteristic (ROC) was later used to compare crossing classification performance for curb and road surface data.

4.1 Crossroad Detection through Curb Data

In the network learning process, we used the curb database comprised by 1700 examples for training and 850 for validation. As can be seen in the learning graph (Figure 4), the 404-5-2 and 404-10-2 topologies obtained the best learning results with errors lower than 0.001. Validation error evolved similarly in all cases with

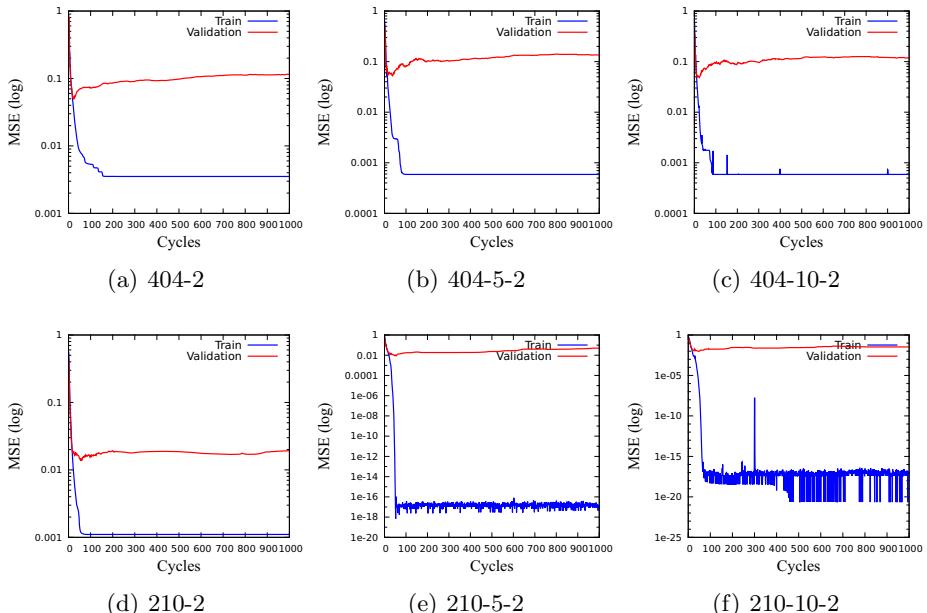


Fig. 4. Learning error of different topologies trained with curb data ((a), (b) and (c)) and road surface data ((d), (e) and (f))

errors near 0.10. However, analyzing the confusion matrix (Table 1) of the single layer network, we note that the accuracy was the highest and the MSE (Mean Square Error) were the lowest. This shows that the 404-2 topology results were superior in relation to the others.

4.2 Crossroad Detection through Road Surface Data

For road surface data training, we used 906 training examples and 452 validation examples. The graph in Figure 4 shows that 210-10-2 topology achieved training error near 10^{-20} , setting as the best result among other configurations. In respect to validation result, all topologies obtained MSE about 0.01. The confusion matrix of Table 1 shows that the 210-2 topology obtained the best MSE and accuracy despite of achieving the highest training error.

4.3 Results Comparison

Analyzing the previous results we note that the single-layer perceptron performed better both with road curb data and road surface data. Considering this configuration, the accuracy for road surface data was 4.43% superior and

Table 1. Confusion matrix for network trained with curb data ((a), (b) and (c)) and road surface data ((d), (e) and (f)). CR stands for crossing class and NCR stands for non-crossing class.

| (a) 404-2 | | (b) 404-5-2 | | (c) 404-10-2 | | |
|-----------|---------|-------------|---------|--------------|---------|-----|
| Predicted | | Predicted | | Predicted | | |
| | CR | NCR | | CR | NCR | |
| Actual | CR | 340 | 22 | CR | 327 | 35 |
| | NCR | 27 | 461 | NCR | 25 | 463 |
| Accuracy | 94.24% | | 92.94% | | 94.12% | |
| MSE | 0.11517 | | 0.13592 | | 0.11853 | |

| (d) 210-2 | | (e) 210-5-2 | | (f) 210-10-2 | | |
|-----------|---------|-------------|---------|--------------|---------|-----|
| Predicted | | Predicted | | Predicted | | |
| | CR | NCR | | CR | NCR | |
| Actual | CR | 334 | 4 | CR | 328 | 10 |
| | NCR | 1 | 112 | NCR | 2 | 112 |
| Accuracy | 98.67% | | 97.35% | | 98.45% | |
| MSE | 0.01941 | | 0.05046 | | 0.03331 | |

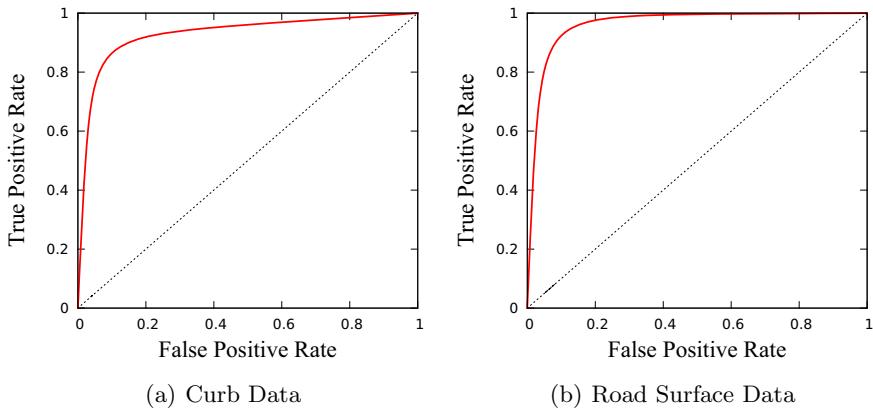


Fig. 5. ROC graphs obtained for the classifier using curb (a) and road surface data (b). Road surface data provides larger AUC.

the MSE was 83.15% lower. These results induce that using road surface data provides better results for road crossing detection.

In order to compare the classification performance of these data sets, the ROC method was adopted. It consists in a graph that takes in account the true positives rate versus the false positive rate of the classifier. The area under the curve (AUC) gives the performance of the network.

ROC graph of classifier that receives curb data as input was generated using the confusion matrix of Tables 1(a), 1(b) and 1(c) and the ROC graph for the road surface data was used the matrix of Tables 1(d), 1(e) and 1(f). These graphs are illustrated in the Figure 5. The AUC for the first case is 0.940589 and for the second is 0.981209. This means that road surface data leads to better classification performance. The larger area also can be visually confirmed in the Figure 5. Moreover the AUC next to 1.0 indicates that these classifiers conduct good results for crossroad detection. Comparing to ROC results obtained by [6] and recall rate of [7], our approach obtained similar results. Regarding to the network configuration, the road surface data set uses 210 inputs against 404 of the second one. This implies in the computation reduction and also memory usage, making it feasible to critical systems as autonomous vehicles.

5 Conclusion

The crossroad classification performance comparison between curb and road surface data was conducted through ROC graph. The AUC obtained was respectively 0.940589 and 0.981209 for curb and road surface data. In this way the use of data obtained from road surface is more appropriate to verify the presence of crossroad in the street. The AUC near 1.0 demonstrates that this classifier provides adequate true positives and false positives proportion. Analyzing distinct

neural network topologies, we confirmed that the single-layer perceptron brings higher accuracy and lower MSE to classify unseen patterns.

In general both curb and road surface data are feasible to crossroad detection, but aiming the real-time response, as the second one uses almost half number of inputs, it demands less memory storage and processing time. Thus with this results we have decided to integrate the road surface detector to the neural network to identify crossroads. The results also show that neural networks are viable to road structure recognition and can be more explored to perform this task. Hereafter we will conduct experiments to detect different road structures than crossroads as left, right turns and other intersections.

References

1. Luettel, T., Himmelsbach, M., Wuensche, H.J.: Autonomous ground vehicles – concepts and a path to the future. *Proceedings of the IEEE* 100(Special Centennial Issue), 1831–1839 (2012)
2. Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D., Hilden, T., Hoffmann, G., Huhnke, B., Johnston, D., Klumpp, S., Langer, D., Levandowski, A., Levinson, J., Marcil, J., Orenstein, D., Paefgen, J., Penny, I., Petrovskaya, A., Pflueger, M., Stanek, G., Stavens, D., Vogt, A., Thrun, S.: Junior: The stanford entry in the urban challenge. *Journal of Field Robotics* 25(9), 569–597 (2008)
3. Sales, D.O., Fernandes, L.C., Osorio, F.S., Wolf, D.F.: Fsm-based visual navigation for autonomous vehicles. In: *Workshop on Visual Control of Mobile Robots 2012, IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal*, pp. 37–42 (October 2012)
4. Mueller, A., Himmelsbach, M., Luettel, T., von Hundelshausen, F., Wuensche, H.J.: Gis-based topological robot localization through lidar crossroad detection. In: *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 2001–2008 (2011)
5. Tongtong, C., Bin, D., Daxue, L., Zhao, L.: Lidar-based long range road intersection detection. In: *2011 Sixth International Conference on Image and Graphics (ICIG)*, pp. 754–759 (2011)
6. Zhu, Q., Chen, L., Li, Q., Li, M., Nüchter, A., Wang, J.: 3d lidar point cloud based intersection recognition for autonomous driving. In: *IEEE Intelligent Vehicles Symposium*, pp. 456–461 (2012)
7. Zhu, Q., Mao, Q., Chen, L., Li, M., Li, Q.: Veloregistration based intersection detection for autonomous driving in challenging urban scenarios. In: *2012 15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1191–1196 (2012)
8. Zhang, W.: Lidar-based road and road-edge detection. In: *Intelligent Vehicles Symposium*, pp. 845–848 (2010)
9. Kuthirummal, S., Das, A., Samarasekera, S.: A graph traversal based algorithm for obstacle detection using lidar or stereo. In: *IROS*, pp. 3874–3880. IEEE (2011)
10. Himmelsbach, M., von Hundelshausen, F., Wünsche, H.J.: Fast segmentation of 3d point clouds for ground vehicles. In: *IEEE Intelligent Vehicles Symposium*, pp. 560–565 (2010)

Application of Particle Swarm Optimization Algorithm to Neural Network Training Process in the Localization of the Mobile Terminal

Jan Karwowski¹, Michał Okulewicz¹, and Jarosław Legierski²

¹ Warsaw University of Technology, Faculty of Mathematics and Information Science,
Koszykowa 75, 00-662 Warsaw, Poland

M.Okulewicz@mini.pw.edu.pl

² Orange Labs Poland,
Obrzeżna 7, 02-691 Warszawa, Poland
Jaroslaw.Legierski@orange.com

Abstract. In this paper we apply Particle Swarm Optimization (PSO) algorithm to the training process of a Multilayer Perceptron (MLP) on the problem of localizing a mobile GSM network terminal inside a building.

The localization data includes the information about the average GSM and WiFi signals in each of the given (x,y,floor) coordinates from more than two thousand points inside a five story building.

We show that the PSO algorithm could be with success applied as an initial training algorithm for the MLP for both classification and regression problems.

Keywords: Particle Swarm Optimization, Neural Network training, Mobile terminal localization.

1 Introduction

In recent years, biologically inspired computational intelligence algorithms have grown a lot of popularity. Examples of such algorithms, based on the idea of swarm intelligence, are PSO and bird flocking algorithm. Those algorithms have been used in the real world applications in the area of computer graphics and animation [17], solving hard optimization tasks [14] or document clustering [7].

PSO has not been widely tested in a high dimensional optimization problems but has already been applied in the process of training a neural network [11]. Error backpropagation (BP) is a well known algorithm for learning multilayer perceptron. In our paper we use stochastic gradient descent backpropagation algorithm proposed in [5], Algorithm (4). In experiments we used implementation of BP from [1].

In recent years on telecommunication market we can observe a large number of mobile application and services based on mobile terminal location. Unfortunately very popular location method used by most of them, is based on the Global Positioning System (GPS). GPS does not work inside the buildings, because the

GPS signal is too weak to be propagated indoors. Another mobile terminal location method is based on the Location Based Services (LBS) in communication service providers networks. LBS are characterized by significant location error e.g. in Poland location error in mobile networks reaches values from 170 to 400 meters in urban area [18]. Therefore an easy for implementation, low cost calculation point of view and fast algorithms to locate the mobile phones inside the buildings are an urgent business need in order to create new and innovative applications and services.

In this article the authors show that PSO algorithm could be with success applied as an initial algorithm for training MLP with two and more hidden layers and that the success of the algorithm does not depend much on the choice of the parameters for PSO or the MLP architecture (which is not true for BP as shown in [10] [22] [23]).

The algorithms were tested in the application of fingerprinting technique on localizing mobile terminal in the building. The localization is based on the GSM and WiFi signals.

The remainder of the paper is organized as follows: First, in section 2 we briefly summarize the PSO algorithm. In section 3 the problem of localizing the mobile terminal is presented. Application of the PSO algorithm and the experimental setup and results are presented in sections 4. The last section summarizes the experimental findings and concludes the paper.

2 Particle Swarm Optimization Algorithm

PSO algorithm is an iterative optimization method proposed in 1995 by Kennedy and Eberhart [12] and further studied and developed by many other researchers, e.g., [20], [19], [6]. In short, PSO implements the idea of swarm intelligence to solving hard optimization tasks.

In the PSO algorithm, the optimization is performed by the set of particles which are communicating with each other. Each particle has its location and velocity. In every step t a location of particle i , x_t^i is updated based on particle's velocity v_t^i :

$$x_{t+1}^i = x_t^i + v_t^i. \quad (1)$$

In our implementation of PSO (based on [2] and [20]) in $t + 1$ iteration i th particle's velocity v_{t+1}^i is calculated according to the following rules:

1. a weighted center c_t^i of $x_{best}^{neighbours_i}$, x_{best}^i and x_t^i points is computed:

$$c_t^i = \frac{gx_{best}^{neighbours_i} + lx_{best}^i + x_t^i}{3} \quad (2)$$

2. a new velocity is computed on the basis of current particle location x_t^i , a weighted center c_t^i and current particle's velocity v_t^i ,

$$v_{t+1}^i = u^{(u-ball)} \|c_t^i - x_t^i\| + (c_t^i - x_t^i) + av_t^i, \quad (3)$$

where

- $x_{best}^{neighbours}$ represents the best location in terms of optimization, found hitherto by the neighbourhood of the i th particle,
- x_{best}^i represents the best location in terms of optimization, found hitherto by the particle i ,
- g is a neighbourhood attraction factor,
- l is a local attraction factor,
- a is an inertia coefficient,
- $u^{(u-ball)}$ is a random vector with uniform distribution over a unit size n-dimensional ball.

In our case the value of the fitness function for the PSO algorithm is the sum of values of the errors on whole training set and the vector in a search space is a vector of weights of the neural network.

As already mentioned, Jing et al. [11] applied PSO to training MLP. It was used to train a much smaller MLP than we are using. Jing et al. network has layers consisting of 3, 6 and 1 neurons for input, hidden and output layer, respectively. Our networks have several neurons in each of the hidden layers. Moreover, in our approach PSO algorithm is used to find initial solution which is later used as starting point for BP algorithm.

3 Localization of a Mobile Terminal in a Building

The problem of localizing a terminal of a mobile network in a building with a usage of fingerprinting technique has been already presented in the literature [3,4,13,15].

The task is to predict location of mobile terminal – triple of a floor and x, y coordinates in the floor plane. In the fingerprinting technique the model on which the predictions are done is constructed on the basis of previously recorded WiFi or GSM signals in the known locations in a building.

It was also shown that the WiFi signals based localization methods are more precise than the GSM signals based, while on the other hand they might not be always available (f.e. during the loss of electricity in the building or simply lack of enough WiFi access points). We will show the difference in that precision found in our research.

The dataset and the problem discussed in this article are the same one as presented in the article showing the basis for predicting credibility of floor predictions [9]. In this article we present a comparison of predictions based on WiFi and GSM signals and also a comparison between MLP initialized with random weights and initially trained with PSO algorithm.

The dataset consists of a 1199 training and validation points gathered in a 1.5 x 1.5m or 3.0 x 3.0m grid at different dates (two series of measurements) and 1092 test points gathered at another day in the grid shifted by half of the resolution of the original grid (one series of measurements). The data comes from all the floors (including ground floor) of a five story building of Faculty of Mathematics and Information Science of Warsaw University of Technology. The data was gathered in halls, corridors, laboratories and lecture rooms.

Each vector of the data consists of the average Received Signal Strength (RSS) from the Base Transceiver Stations (BTS) of the GSM system and RSS from the Access Points (AP) of the WiFi network. Each vector is labeled with x, y and floor coordinates defined for each of the points in which the data was gathered.

Therefore the task of localizing a mobile terminal in a building may be looked upon as a regression task for x and y coordinates and classification task for floor coordinate, making a neural network a universal tool for both of the tasks.

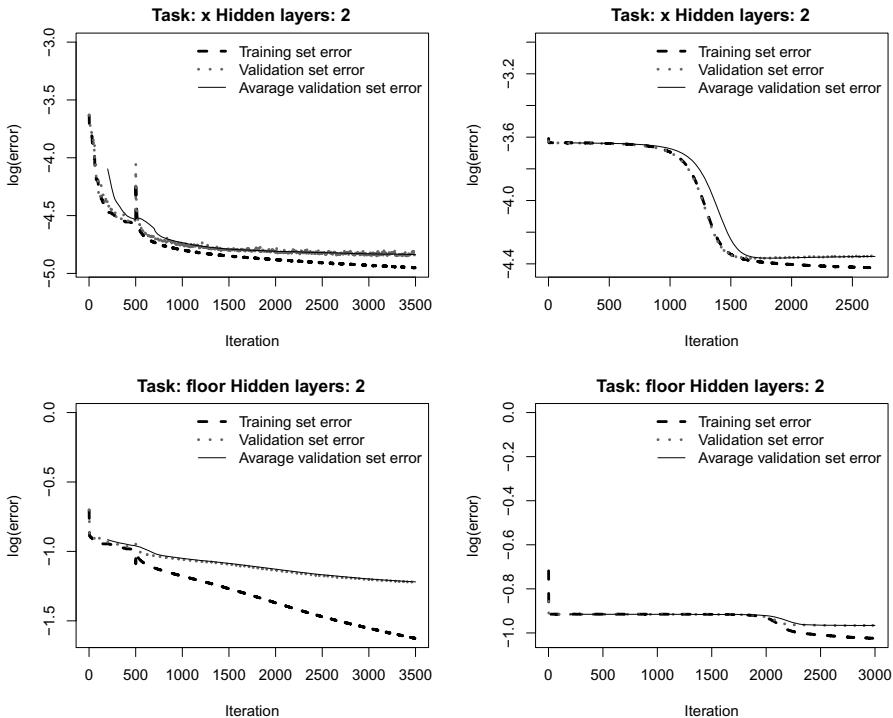


Fig. 1. The example training process of a neural network with 2 hidden layers for floor classification and regression in one of the directions. The left column presents the runtime where first 500 iterations were done by the PSO algorithm and the rest by the BP algorithm. The right column presents runtime of the BP algorithm.

4 Tests and Results

Datasets for all the tests were composed in a following way:

- one series of measurements was chosen as a training set,
- another series of measurements (in the same grid as training set, but gathered on a different day) was used as a validation set,
- a series of measurements gathered on a different day in a shifted grid was used as a test set.

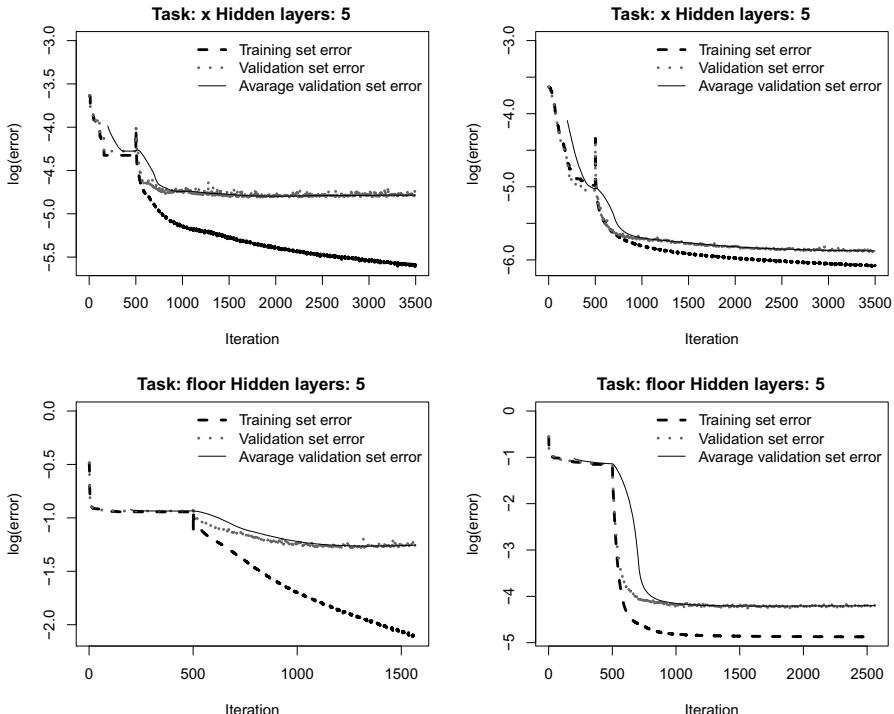


Fig. 2. The example training process of a neural network with 5 hidden layers for floor classification and regression in one of the directions. First 500 iterations were done by the PSO algorithm and the rest by the BP algorithm. Left column shows the runtime for the GSM data and the right one for the WiFi data.

All the tests were run in the following scenario:

1. A neural network was trained in a batch mode by the PSO algorithm.
2. An initially trained network was used as a starting point and trained by the online BP algorithm.
3. Separately a network was trained for comparison from a random point by the online BP with the same parameters.
4. In both cases the training process by online BP was stopped when the error on validation set began to rise.

The baseline of all experiments was performance of the network initialized by PSO with 500 iterations and 40 particles and further trained by the BP for at most 3000 iterations. The parameters for the baseline PSO were set as follows $g = 1.4$, $l = 1.4$, $a = 0.63$, $P(\text{Particle}_i \text{ is neighbor of } \text{Particle}_j) = 0.5$. For the baseline BP the learning rate for each example was set to 0.0008, momentum was set to 0.3 and value of average error on validation set was counted over 200 iteration. The comparison of the baseline experiment against experiments with

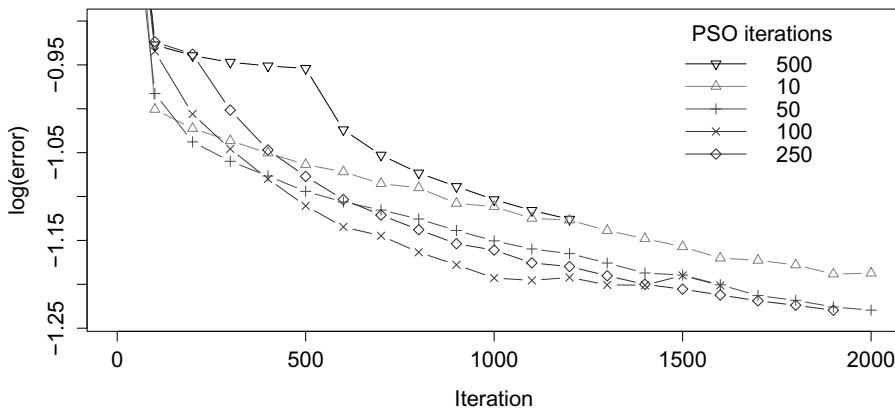


Fig. 3. Comparison of the average network convergence on the validation set for different number of initial PSO iterations

different parameters are shown in the Tables 1-4. In each comparison only the mentioned parameters are changed while the rest remains the same as in baseline experiment. The comparison has been done on the basis of classification accuracy (i.e. the ratio of the properly classified records from the test set) for the floor, and on the basis of 0.9 quantile of absolute error given in meters for prediction of the X and Y coordinates. The result of the baseline is boldfaced in each of the tables.

The PSO was additionally run with the following parameters:

- 5, 10, 25, 50, 100, 150, 200, 250 iterations,
- $g = 2.2$ and $l = 0.6$, $g = 0.6$ and $l = 2.2$.

The results of those tests are presented in the Tables 1 and 2

The BP was additionally run with the following parameters:

- learning rate = 0.0002, 0.0004, 0.0006, 0.0010.

The results of those tests are presented in the Table 3.

The following neural networks with following number of neurons in hidden layers were tested (all with full connections between subsequent layers):

- 60 and 60,
- 60, 40 and 20,
- 60, 40, 40 and 20,
- 60, 40, 40, 30 and 20.

The results of those tests are presented in the Table 4.

The networks had 26 inputs for GSM data and 107 inputs when used for WiFi data. Each input represented a strength of the given BTS or AP signal present anywhere in the building. For regression tasks the networks had one output

Table 1. Comparison for different number of PSO iterations for GSM and WiFi data. The results of baseline experiment are boldfaced.

| PSO iterations | Classification accuracy | | 0.9 quantile of $ ex $ | | 0.9 quantile of $ ey $ | |
|----------------|-------------------------|------------|------------------------|--------------|------------------------|--------------|
| | GSM | WiFi | GSM | WiFi | GSM | WiFi |
| 5 | 54% | 98% | 8.66m | 5.93m | 11.98m | 6.55m |
| 10 | 53% | 98% | 8.91m | 5.31m | 11.08m | 6.03m |
| 25 | 53% | 98% | 8.37m | 5.35m | 11.40m | 6.29m |
| 50 | 53% | 98% | 8.80m | 4.96m | 12.87m | 6.14m |
| 100 | 52% | 98% | 8.61m | 5.33m | 11.76m | 6.94m |
| 150 | 56% | 97% | 9.13m | 5.45m | 12.47m | 6.47m |
| 200 | 54% | 97% | 8.90m | 5.01m | 11.92m | 6.48m |
| 250 | 53% | 98% | 8.77m | 5.46m | 11.55m | 6.42m |
| 500 | 54% | 98% | 9.11m | 5.18m | 11.66m | 6.48m |

Table 2. Comparison for different PSO local and global attraction factor parameters for GSM and WiFi data. The results of baseline experiment are boldfaced.

| PSO parameters | Classification accuracy | | 0.9 quantile of $ ex $ | | 0.9 quantile of $ ey $ | |
|----------------|-------------------------|------------|------------------------|--------------|------------------------|--------------|
| | GSM | WiFi | GSM | WiFi | GSM | WiFi |
| 0.6 2.2 | 53% | 98% | 8.86m | 5.47m | 11.86m | 6.74m |
| 1.4 1.4 | 54% | 98% | 9.11m | 5.18m | 11.66m | 6.48m |
| 2.2 0.6 | 54% | 98% | 11.38m | 5.54m | 12.12m | 6.40m |

Table 3. Comparison for different values of a BP learning rate for GSM and WiFi data. The results of baseline experiment are boldfaced.

| Learning rate | Classification accuracy | | 0.9 quantile of $ ex $ | | 0.9 quantile of $ ey $ | |
|---------------|-------------------------|------------|------------------------|--------------|------------------------|--------------|
| | GSM | WiFi | GSM | WiFi | GSM | WiFi |
| 0.0002 | 50% | 98% | 9.53m | 6.21m | 12.29m | 7.35m |
| 0.0004 | 52% | 98% | 9.00m | 5.75m | 11.96m | 6.66m |
| 0.0006 | 53% | 98% | 8.80m | 5.48m | 11.89m | 6.68m |
| 0.0008 | 54% | 98% | 9.11m | 5.18m | 11.66m | 6.48m |
| 0.0010 | 54% | 98% | 8.77m | 5.56m | 11.73m | 6.10m |

Table 4. Comparison for different number of neural network hidden layers for GSM and WiFi data. The results of baseline experiment are boldfaced.

| Hidden layers | Classification accuracy | | 0.9 quantile of $ ex $ | | 0.9 quantile of $ ey $ | |
|---------------|-------------------------|------------|------------------------|--------------|------------------------|--------------|
| | GSM | WiFi | GSM | WiFi | GSM | WiFi |
| 2 | 52% | 98% | 9.06m | 5.54m | 11.93m | 6.79m |
| 3 | 54% | 98% | 9.11m | 5.18m | 11.66m | 6.48m |
| 4 | 52% | 98% | 8.67m | 5.37m | 11.73m | 5.82m |
| 5 | 53% | 98% | 8.92m | 5.27m | 12.65m | 5.87m |

representing the predicted location for X or Y coordinate in the building. For a classification the networks had six outputs, for each of the six classes representing the floor of the building.

The example training processes of BP and PSO+BP algorithm for a network with two hidden layers are presented on the Fig. 1. For the larger number of hidden layer BP starting from random point was not able to achieve in 3000 iterations better results then classifying all test data as the most frequently occurring floor and regression models predicted the weighted average in both directions.

Figure 2 shows the training processes of PSO+BP algorithm of the neural network with 5 hidden layers for GSM and WiFi data.

5 Discussion and Conclusions

As can be seen from results a good starting point for online BP algorithm is very important. When choosing random weights for BP error on the training set was not decreasing at all or was constant for a large number of iterations and network started to converge only after several hundreds of iterations. Popular approach to this problem is selecting learning rate and network topology individually for every problem or using adaptive learning rate. [22,21]

Our approach is to start learning network with PSO for a small number of iterations and then set the network weights found by this method as a starting point for BP. The results show that convergence of BP is much faster and stable than in experiments where BP was started from random point.

Our results show that the PSO algorithm is useful in the process of training MLP to solve the problem of localizing a mobile terminal. Hybrid learning method has given better results than plain BP. The important advantage of the PSO (being a method of global optimization) is possibility for leaving local minima, while BP has been reported to stuck in them [8].

It is important to notice, that one iteration of the PSO algorithm is much slower than one iteration of BP algorithm (approximately by a factor of half a number of particles). On the other hand it is possible to efficiently implement a parallel version of the PSO and even using a very small number of PSO iterations (e.g. 10) is sufficient for finding a good starting point for BP (although the convergence of BP was faster for a larger number of iterations as presented in Fig. 3).

Our results also confirm that WiFi based localization gives a more accurate location of the mobile terminal, especially in the problem of floor classification. Floor classification based on GSM signals was accurate in about 55% of observations and based on WiFi signals in about 98%.

Further research on the hybrid PSO+BP method should include comparison with another methods (e.g. BP algorithm with adaptive learning rate, batch BP algorithm). Research on the problem of localizing a mobile terminal with the usage of neural networks should take into account learning and testing on not aggregated data and also observations about the credibility of predictions for the GSM data should be considered [9].

Acknowledgements. Study was partially supported by research fellowship within "Information technologies: Research and their interdisciplinary applications" agreement number POKL.04.01.01-00-051/10-00 and partially financed from the funds of National Science Centre granted on the basis of decision DEC-2012/07/B/ST6/01527.

The analyses of the results and the plots have been done with the usage of R[16].

References

1. Neuroph Java Neural Network Framework (2012), <http://neuroph.sourceforge.net/>
2. Standard PSO 2011 (2012), <http://www.particleswarm.info/>
3. Benikovsky, J., Brida, P., Machaj, J.: Localization in Real GSM Network with Fingerprinting Utilization. In: Chatzimisios, P., Verikoukis, C., Santamaría, I., Ladomada, M., Hoffmann, O. (eds.) Mobile Lightweight Wireless Systems. LNICST, vol. 45, pp. 699–709. Springer, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-16644-0_60
4. Bento, C., Soares, T., Veloso, M., Baptista, B.: A Study on the Suitability of GSM Signatures for Indoor Location. In: Schiele, B., Dey, A.K., Gellersen, H., de Ruyter, B., Tscheligi, M., Wichert, R., Aarts, E., Buchmann, A.P. (eds.) AmI 2007. LNCS, vol. 4794, pp. 108–123. Springer, Heidelberg (2007), http://dx.doi.org/10.1007/978-3-540-76652-0_7
5. Bottou, L.: Stochastic gradient learning in neural networks. In: Proceedings of Neuro-Nimes 1991, vol. 8 (1991)
6. Cristian, I.T.: The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters 85(6), 317–325 (2003)
7. Cui, X., Potok, T., Palathingal, P.: Document clustering using particle swarm optimization. In: Proceedings 2005 IEEE Swarm Intelligence Symposium, SIS 2005, pp. 185–191 (June 2005)
8. Gori, M., Tesi, A.: On the problem of local minima in backpropagation. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(1), 76–86 (1992)
9. Grzenda, M.: On the prediction of floor identification credibility in RSS-based positioning techniques. In: Ali, M., Bosse, T., Hindriks, K.V., Hoogendoorn, M., Jonker, C.M., Treur, J. (eds.) IEA/AIE 2013. LNCS, vol. 7906, pp. 610–619. Springer, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-38577-3_63
10. Jacobs, R.A.: Increased rates of convergence through learning rate adaptation. Neural Networks 1(4), 295–307 (1988)
11. Jing, Y.W., Ren, T., Zhou, Y.C.: Neural network training using pso algorithm in atm traffic control. In: Huang, D.S., Li, K., Irwin, G. (eds.) Intelligent Control and Automation. LNCIS, vol. 344, pp. 341–350. Springer, Heidelberg (2006), http://dx.doi.org/10.1007/978-3-540-37256-1_41
12. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks. IV, pp. 1942–1948 (1995)
13. Lakmali, B., Dias, D.: Database Correlation for GSM Location in Outdoor and Indoor Environments. In: 4th International Conference on Information and Automation for Sustainability, pp. 42–47 (2008)

14. Okulewicz, M., Mańdziuk, J.: Application of Particle Swarm Optimization Algorithm to Dynamic Vehicle Routing Problem. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013, Part II. LNCS, vol. 7895, pp. 547–558. Springer, Heidelberg (2013),
http://dx.doi.org/10.1007/978-3-642-38610-7_50
15. Otsason, V., Varshavsky, A., LaMarca, A., de Lara, E.: Accurate GSM Indoor Localization. In: Beigl, M., Intille, S.S., Rekimoto, J., Tokuda, H. (eds.) UbiComp 2005. LNCS, vol. 3660, pp. 141–158. Springer, Heidelberg (2005),
http://dx.doi.org/10.1007/11551201_9
16. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2012) ISBN 3-900051-07-0,
<http://www.R-project.org/>
17. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. SIGGRAPH Comput. Graph. 21(4), 25–34 (1987),
<http://doi.acm.org/10.1145/37402.37406>
18. Sabak, G.: api.orange.pl tutorial for building localization applications. Tech. rep., Orange (2012) (in polish), <http://telco21.pl/orange-cell/>
19. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of IEEE International Conference on Evolutionary Computation, pp. 69–73 (1998)
20. Shi, Y., Eberhart, R.: Parameter selection in particle swarm optimization. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 591–600. Springer, Heidelberg (1998)
21. Vogl, T.P., Mangis, J., Rigler, A., Zink, W., Alkon, D.: Accelerating the convergence of the back-propagation method. Biological Cybernetics 59(4-5), 257–263 (1988)
22. Wilson, D.R., Martinez, T.R.: The need for small learning rates on large problems. In: Proceedings of the International Joint Conference on Neural Networks, IJCNN 2001, vol. 1, pp. 115–119 (2001)
23. Yu, X.H., Chen, G.A.: Efficient backpropagation learning using optimal learning rate and momentum. Neural Networks 10(3), 517–527 (1997)

Modeling Spatiotemporal Wild Fire Data with Support Vector Machines and Artificial Neural Networks

Georgios Karapilafis¹, Lazaros Iliadis¹, Stefanos Spartalis², S. Katsavounis²,
and Elias Pimenidis³

¹ Democritus University of Thrace, Orestias, Greece

² Democritus University of Thrace, Xanthi, Greece

³ University of East London, UK

evelpil@gmail.com, liliadis@fmenr.duth.gr,
{sspart, skatsav}@pme.duth.gr, E.Pimenidis@uel.ac.uk

Abstract. Forest fires have not only devastating catastrophic environmental consequences, but they also have serious negative social impact. Exploitation of related spatiotemporal data could potentially lead to the development of reliable models, towards forecasting of the annual burned area. This paper takes advantage of the regression capabilities of modern Soft Computing approaches. More specifically numerous Artificial Neural Networks' and e-Regression Support Vector Machines' models were developed, each one assigned locally to a distinct Greek forest department (GFD). The whole research effort was related to Greek wild fires incidents for the period 1983-1997 and to all of the GFDs. The performance of both methods has proven to be quite reliable in the vast majority of the cases and a comparative analysis was also used to reveal potential advantages or weaknesses.

Keywords: Forest Fires, Artificial Neural Networks, e-Regression Support Vector Machines, Rough Sets.

1 Introduction

For several decades, wild fires have been a very serious threat for the forest areas globally. Their consequences in forest ecosystems, in the landscape, in social and financial prosperity are enormous. Unfortunately the efforts that have been made so far have not caused the reduction of the problem. The recent catastrophic case of Greece in 2007 (with 64 deaths and a cost of 5 billion Euros) and also the vast destructions in Southern Europe during the last few years that were enforced by climate change and extreme weather conditions are quite remarkable (Council conclusions on prevention of forest fires within the EU, 2010) [2]. It is indeed a challenge to perform a successful modeling approach that would offer potential aid to the civil protection authorities towards planning a proper prevention and protection policy.



Fig. 1. Satellite image of the devastating Greek forest fires in 2007

The magnitude of destruction in Greece for 2007 can be clearly seen in the above satellite picture 1.

The specific aim of this research is to use Soft Computing algorithms (SCA) (Kecman, 2001), [11] in order to develop Artificial Neural Networks' (ANNs) and Support Vector Machines' (SVMs) models capable of forecasting the burnt area of the main forest species in each single forest department of Greece.

1.1 Literature Review

There are SCA modeling efforts in the literature that estimate forest fire risk by using supervised or unsupervised classification or hybrid approaches. Zammit et al., 2007, [19] address the problem of burnt area discrimination using classification of remote sensing images with SVMs. Another approach that uses Support Vector Machines classifier combined with multispectral Landsat TM images for obtaining burnt area mapping in Greece, has been proposed by Petropoulos et al., 2011, [14] Yu et al., 2011, [16] have used Self Organizing Maps and Back Propagation ANNs in a hybrid manner, in order to perform pattern clustering of forest fires based on meteorological data. Iliadis et al., 2010, [9] have clustered GFDs based on their burnt area by employing fuzzy c-means clustering and Iliadis, 2005, [8] has developed a fuzzy inference system to estimate annual forest fire risk indices for the GFDs. Cheng and Wang 2008, [1] used ANNs towards spatiotemporal forest fire risk modeling. Cortez and Mainer, 2010, [3] used ANNs and SVMs with Gaussian Kernel towards burnt area estimation in Portugal.

The innovation of the research presented in this paper is justified by the fact that a similar research effort has not been carried out in Greece (a country in the top of the most severely burnt ones globally) and moreover a comparative analysis between two distinct SCA methods is performed for the first time. Also it is really important the fact that all GFDs have been modeled and considered as independent cases (one by one) with fire data vectors corresponding to a quite long period of 15 years.

2 Area and Data

2.1 Parameters Related to the Problem of Forest Fires

The ignition of a forest fire is related to various parameters like the vegetation type and density, topography, altitude, slope, and moreover the meteorological conditions acting in a catalytic way (relative humidity (RH), wind speed (WS) and air temperature (AT)) (Kailidis, 1990) [10]. The AT has a serious interference in the drying of the dead fuel moisture and has a positive effect in both the ignition and the spread of the wild fire, whereas the RH plays an inhibitory role (Kailidis, 1990) [10]. Most of the forest fire breakouts are observed in an altitude that ranges from 700-1000 meters. On the other hand over 1500 meters the events are too rare. This is due to the fact that the air is becoming colder and colder as the altitude increases. The higher the slope of an area the higher is the extension speed of the fire, due to the fact that the heat is emitted more efficiently and the ground is drier.

In this research effort the following nine independent parameters were considered as the ones determining the total burnt area, namely: *average altitude* at the spatial and temporal point of the forest fire breakout (STPFFB), *average relative humidity* STPFFB, *average air temperature* STPFFB, *average wind speed* STPFFB, *average slope* STPFFB, *average vegetation density* STPFFB, *average grassland density* STPFFB, *intervention time* (elapsed time from the moment of fire detection till the arrival of fighting forces) and *type of wild fire*.

Wild fire, meteorological and topographic data related to each incident from 1983 till 1997 for each GFD were selected and used.

The following map is a graphical display of the classification of the Greek territories that appear to have the highest burnt area for a period of 23 years. It has been obtained using ArcMap 9.3.1 with the “*natural breaks*” approach and it is clearly not a part of this project. However it is presented here just to give the reader a hint on the nature of the problem. Natural breaks also called the Jenks classification method is designed to determine the best arrangement of values into different classes. This is done by seeking to minimize each class’s average deviation from the class

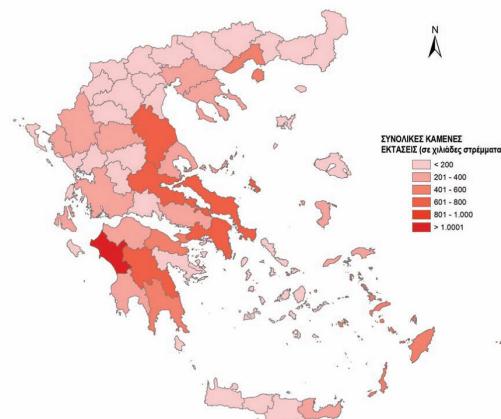


Fig. 2. Classification of the GFDs based on the average burnt area for 23 years

mean, while maximizing each class's deviation from the means of the other groups. In other words, the method seeks to reduce the variance within classes and maximize the variance between classes. It is quite clear that the most severely burnt areas are located in the southern and central parts of the country.

2.2 Supporting the Choice of the Parameters

In a previous research of our team (Tsataltzinos et al., 2011), [15] a detailed study towards the determination of the importance of the involved features has been performed, using Rough Sets Theory (RST). Thus, the choice of the independent parameters in this effort has been based on the results of Tsataltzinos et al., 2011 [15] and also on the availability of data.

One of the most promising methods, introduced by Pawlak in 1991, [12] (Peters and Skowron, 2007), [13] which can be used for attribute reduction, is the rough-set theory, which is still another approach to vagueness. Rough set theory is a good tool for attribute reduction and with the help of the *Rosetta software* <http://rosettasoftware.com/>, [7] the process of finding the reducts is much easier. Rough set theory can be viewed as a specific implementation of Frege's idea of vagueness, i.e., imprecision in this approach is expressed by a boundary region of a set, and not by a partial membership, like in fuzzy set theory. The basic idea is that given a dataset with discretized attribute values, a subset of the original attributes can be obtained (termed a *Reduct*) using RST that are the most informative, whereas all of the rest attributes can be removed from the dataset with very little information loss. Rough set concept can be defined quite generally by means of topological operations, interior and closure, called approximations. Given a set of objects U called the universe and an indiscernibility relation $R \subseteq UXU$ representing our lack of knowledge about elements of U we assume that R is an equivalence relation. If $X \subseteq U$ the target is to characterize the set X with respect to R . The basic concepts of RST are given below (Peters and Skowron 2007) [13].

The *lower approximation* of a set X with respect to R is $PX = \{X / [X]_R \subseteq X\}$ (1)

It includes the set of all objects, which can be safely and certainly classified as X with respect to R (are certainly X with respect to R).

The *upper approximation* of a set X with respect to R is $\bar{P}X = \{X / [X]_R \subseteq X \neq \emptyset\}$ (2)

It comprises of the set of all objects which can be possibly classified as X with respect to R (are possibly X in view of R). The *boundary region* of a set X with respect to R is the set of all objects, which can be classified neither as X nor as not- X with respect to R .

Set X is crisp (exact with respect to R), if the boundary region of X is empty.

Set X is rough (inexact with respect to R), if the boundary region of X is nonempty.

The whole dataset was inserted into the Rosetta software and the *Reducts* were calculated with the help of the dynamic reducer (Tsataltzinos et al., 2011) [15].

It is a fact that 5093 cases out of the 5100 can be described with average altitude and average forest cover as the only knowledge about the terrain however the combination of meteorological parameters is of exactly equal importance. So features like population density, expert's evaluation, touristic growth, and land value were removed from the input data set.

Table 1. The reducts that produce the best results for 5100 cases of forest fires

| | | |
|----|---|------|
| 1 | {Altitude, Forest Cover} | 5093 |
| 2 | {Humidity, Altitude} | 5093 |
| 3 | {Temperature, Altitude} | 5093 |
| 4 | {Humidity, Wind Speed, Population density} | 4497 |
| 5 | {Temperature, Forest Cover, Wind Speed, Other expert's evaluation)} | 3430 |
| 6 | {Temperature, Forest Cover, Wind Speed, Touristic growth} | 3264 |
| 7 | {Altitude, LandValue} | 2813 |
| 8 | {Altitude, Population} | 2813 |
| 9 | {Humidity, Wind Speed, Touristic growth } | 2728 |
| 10 | {Temperature, Wind speed, Land Value} | 2728 |

Another interesting point is that the attribute "Altitude" is found in 6 of those 10 Reducts. Most of the cases of major fire incidents were located in coastal areas that also had great forest cover and unprotected forests. Thus the altitude and the meteorological conditions at the moment of the wild fire breakout can be considered as some of the most basic factors that influence the risk indices. All the GFDs that are located far away from the sea, have greater altitude and they do not seem to be associated with major fire incidents.

3 Developing the Soft Computing Models

The whole model development process was carried out under the Matlab platform for 51 GFDs. Due to lack of proper data vectors for some years in specific cases, a small number of cases was not used in this research. Totally 49 optimal ANN models were obtained, a distinct one for each forest department, after making numerous experiments with different architectures, transfer and learning functions, whereas ANNs completely failed for two cases. On the other hand 51 SVM models were determined to be the optimal ones for an equal number of GFDs.

3.1 Developing the Optimal ANN Models

Totally 4955 data vectors were used for all of the GFDs cases. The computational power of Matlab with such a big number of data records was a key issue. As it was mentioned before all of the developed ANNs had nine neurons in the Input Layer

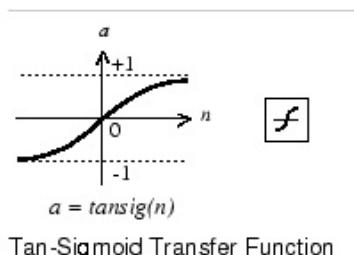
corresponding to an equal number of features, obtained at the exact spatial and temporal point of the wild fire breakout, namely: *average altitude*, *average relative humidity*, *average air temperature*, *average wind speed*, *average slope*, *average vegetation density*, *average grassland density*, *intervention time* and *type of wild fire*. The output layer had only one neuron relative to the one depended feature which was the total burnt area of the main forest species. The number of processing elements (PE) in the hidden layer was determined separately for each case after a tedious trial and error iterative process. Also the number of Epochs varied for each case. However it should be clarified that no ANN comprised of more than 10 hidden PEs in an effort to keep the networks as simple as possible and the number of iterations was kept quite low (determined by the validation process) in order to avoid overtraining.

Table 2. Description of the data vectors used for each GFD

| Forest Depart | Total | Train | Valid. | Test | Forest Depart | Total | Train | Valid. | Test |
|---------------|-------|-------|--------|------|---------------|-------|-------|--------|------|
| Agia | 38 | 26 | 6 | 6 | Kyklades | 203 | 143 | 30 | 30 |
| Aigio | 101 | 61 | 20 | 20 | Ioannina | 339 | 237 | 34 | 68 |
| Alex/polis | 65 | 39 | 13 | 13 | Edessa | 38 | 26 | 6 | 6 |
| Amfissa | 169 | 119 | 0 | 50 | Korinthos | 106 | 74 | 16 | 16 |
| Aliveri | 197 | 138 | 0 | 59 | Kozani | 86 | 60 | 13 | 13 |
| Argloida | 92 | 56 | 18 | 18 | Lefkada | 82 | 58 | 12 | 12 |
| Atalanti | 32 | 22 | 5 | 5 | Mesologgi | 69 | 49 | 10 | 10 |
| Aridaia | 68 | 48 | 10 | 10 | Metsovo | 54 | 38 | 8 | 8 |
| Almyros | 73 | 51 | 11 | 11 | Thiva | 169 | 119 | 25 | 25 |
| Amaliada | 109 | 77 | 16 | 16 | Thesprotia | 35 | 25 | 5 | 5 |
| Veroia | 44 | 30 | 7 | 7 | Thasos | 86 | 60 | 13 | 13 |
| Volos | 38 | 26 | 6 | 6 | Stavroup | 51 | 35 | 8 | 8 |
| Vytina | 58 | 40 | 9 | 9 | Stavros | 47 | 33 | 7 | 7 |
| Drama | 116 | 82 | 17 | 17 | Spercheias | 22 | 16 | 3 | 3 |
| Didimotixo | 153 | 107 | 23 | 23 | Skiathos | 30 | 20 | 5 | 5 |
| Tsotili | 125 | 87 | 19 | 19 | Serres | 51 | 35 | 8 | 8 |
| Dodekanisa | 206 | 144 | 31 | 31 | Samos | 40 | 28 | 6 | 6 |
| Leivadia | 169 | 119 | 25 | 25 | Preveza | 43 | 31 | 6 | 6 |
| Kinouria | 94 | 66 | 14 | 14 | Polygyros | 21 | 15 | 3 | 3 |
| Lidoriki | 178 | 124 | 27 | 27 | Pyrgos | 111 | 77 | 17 | 17 |
| Kastoria | 74 | 52 | 11 | 11 | Pieria | 302 | 212 | 45 | 45 |
| Kerkyra | 123 | 87 | 18 | 18 | Patra | 261 | 183 | 39 | 39 |
| Karpenisi | 46 | 32 | 7 | 7 | Nirgita | 72 | 58 | 12 | 2 |
| Karditsa | 175 | 123 | 26 | 26 | Mouzaki | 49 | 35 | 7 | 7 |
| Kalampaka | 45 | 31 | 7 | 7 | TOTAL CASES | 4955 | | | |

After several experiments in a trial and error mode, it was proven that the most reliable ANN models had the following attributes: The Matlab's *TRAINLM* (supporting training with validation and test vectors) was the ANNs training function used to update weights and bias values according to the *Levenberg-Marquardt* optimization (Hagan et al., 1996) [4]. It is generally considered the fastest backpropagation algorithm though memory consuming. The *LEARNGDM* was chosen to be the Adaption learning function, employed to calculate the weight change dW for a given neuron, from the neuron's input P and error E as it is shown in the following function 3. It should be clarified that W is the bias, LR stands for the learning rate, MC is the momentum constant according to gradient descent with momentum. The previous weight change dW_{prev} is always stored and it is read from the learning state LS (Haykin, 2009), [5] $dW = MC * dW_{prev} + (1 - MC) * LR * gW$ (3).

The Transfer function was the *TANSIG (Hyperbolic Tangent Sigmoid)* used to calculate a layer's output from its net input. Its formula is given by the following function 4 (Vogl et al., 1998) [17]. $Tansig(n) = \frac{2}{(1 + \exp(-2 * n))} - 1$ (4)



Tan-Sigmoid Transfer Function

Fig. 3. The Tangent Hyperbolic Sigmoid Transfer Function

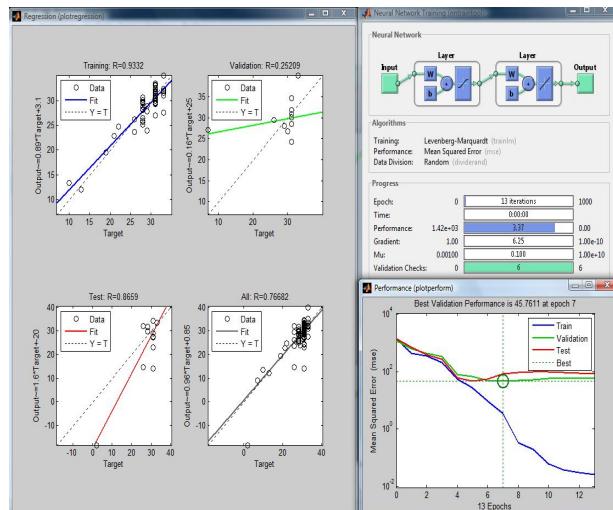
Table 3. Performance of the 49 optimal ANN

| Forest Depart | ANN Performance | | | Forest Depart | ANN Performance | | |
|----------------------|----------------------------|---------------------------|------------------|----------------------|----------------------------|---------------------------|------------------|
| | R² Train | R² Test | RMSE test | | R² Train | R² Test | RMSE test |
| Agia | 0.95 | 0.64 | 5.25 | Kyklades | 0.99 | 0.90 | 7.20 |
| Aigio | 0.84 | 0.85 | 2.36 | Ioannina | 0.52 | 0.51 | 8.18 |
| Alex/polis | 0.98 | 0.82 | 3.89 | Edessa | 0.85 | 0.75 | 6.35 |
| Amfissa | 0.91 | 0.70 | 2.75 | Korinthos | 0.65 | 0.62 | 3.01 |
| Aliveri | 0.99 | 0.69 | 3.01 | Kozani | 0.88 | 0.64 | 8.94 |
| Argloida | 0.72 | 0.54 | 5.70 | Lefkada | 0.90 | 0.95 | 1.81 |
| Atalanti | 0.87 | 0.82 | 2.64 | Mesologgi | 0.68 | 0.66 | 7.74 |
| Aridaia | 0.99 | 0.70 | 14.59 | Metsovo | 0.95 | 0.75 | 8.76 |
| Almyros | 0.95 | 0.61 | 7.98 | Thiva | 0.98 | 0.64 | 4.36 |
| Amaliada | 0.73 | 0.84 | 5.30 | Thesprotia | 0.78 | 0.97 | 5.10 |

Table 3. (continued)

| | | | | | | | |
|--------------|------|------|-------|--------------|------|------|------|
| Veroia | 0.83 | 0.85 | 2.24 | Thasos | 0.87 | 0.74 | 9.05 |
| Volos | 0.94 | 0.64 | 2.36 | Stavroupolis | 0.91 | 0.93 | 4.81 |
| Vytina | 0.82 | 0.80 | 6.65 | Stavros | 0.98 | 0.90 | 3.01 |
| Drama | 0.81 | 0.66 | 7.23 | Spercheiada | 0.99 | 0.90 | 9.05 |
| Didimoteicho | 0.84 | 0.54 | 2.42 | Skiathos | 0.99 | 0.79 | 7.20 |
| Tsotili | 0.84 | 0.59 | 5.00 | Serres | 0.83 | 0.70 | 7.47 |
| Dodekanisa | 0.65 | 0.34 | 9.35 | Samos | 0.77 | 0.94 | 7.34 |
| Leivadia | 0.79 | 0.52 | 8.41 | Preveza | 0.99 | 0.83 | 4.36 |
| Kinouria | 0.97 | 0.58 | 2.75 | Polygyros | 0.99 | 0.99 | 4.66 |
| Lidoriki | 0.37 | 0.71 | 5.56 | Pyrgos | 0.99 | 0.89 | 2.38 |
| Kastoria | 0.49 | 0.66 | 11.66 | Pieria | 0.41 | 0.40 | 9.42 |
| Kerkyra | 0.86 | 0.52 | 5.76 | Patra | 0.71 | 0.55 | 9.79 |
| Karpenisi | 0.98 | 0.55 | 4.07 | Nirgita | 0.99 | 0.77 | 3.80 |
| Karditsa | 0.68 | 0.22 | 3.39 | Mouzaki | 0.74 | 0.77 | 4.64 |
| Kalampaka | 0.86 | 0.54 | 17.91 | | | | |

The Root Mean Square Error (RMS) and the R^2 were used as the objective functions for all of the cases. The following figure 4 displays graphically, the evaluation of an ANN in Matlab. Due to space limitation it was not possible to present similar graphs for all of the GFDs. In figure 4 one can see the good level of convergence of the models.

**Fig. 4.** Sample graphs of the performance of an ANN

3.2 Developing the Optimal ANN Models

The e-Regression Support Vector Machines (ϵ -SVM) are keeping the training error fixed while at the same time they are minimizing the confidence interval. They are used for both regression and classification. Vapnik [16] introduced the following loss function that ignores errors less than a predefined value $\epsilon > 0$.

$|y - f(\vec{x})|_\epsilon = \max \{0, |y - f(\vec{x})| - \epsilon\}$ The ϵ -SVR algorithm offers in many cases the optimal function of the form: $f(\vec{x}) = k(\vec{w}, \vec{x}) + b \quad \vec{w}, \vec{x} \in R^N, b \in R$ (3)

(Kecman, 2001) [11]. The whole idea is based on the determination of the function with the minimum testing error. The actual solution can be reached by minimizing the following normalized risk function 4: $\frac{1}{2} \|\vec{w}\|^2 + C_{SVR} \cdot R_{emp}^\epsilon[f]$ (4)

$R_{emp}^\epsilon[f]$ is the function of empirical risk $R_{emp}[f] = \frac{1}{p} \sum_{i=1}^p L(f, \vec{x}_i, y_i)$ (5)

whereas the loss function is $L(f, \vec{x}_i, y_i) = |y_i - f(\vec{x}_i)|_\epsilon$ (6) and it ignores errors less than ϵ .

Code was developed in C in order to produce the SVM e-regression. The SVM models were developed in MATLAB, using the functions *svmreg* (*to perform the regression*), the *GenerateTrainingData* (to divide the initial data vectors randomly in Training and Testing ones) and the *svmval* (to evaluate the developed models) from the BioInformatics Toolbox (code developed in C). For more details refer to the MATLAB Bioinformatics ToolBox Product Documentation <http://www.mathworks.com/help/toolbox/bioinfo/> [6].

The *RMSE_calc* function was used to produce the RMSE of each model in training and in testing. The parameter C was set to the value of 0.1, the width of the tube ϵ (epsilon) was assigned the value of 0.1, the Kernel was chosen to be Gaussian whereas the value of σ varied from 0.1 to 0.4 (depending on the case see table 4) and λ was assigned the small value of 0.000001. The final values of the parameters were set after trial and error experiments.

Table 4. Performance of the SVM e-regression for the GFDs with epsilon=0.1

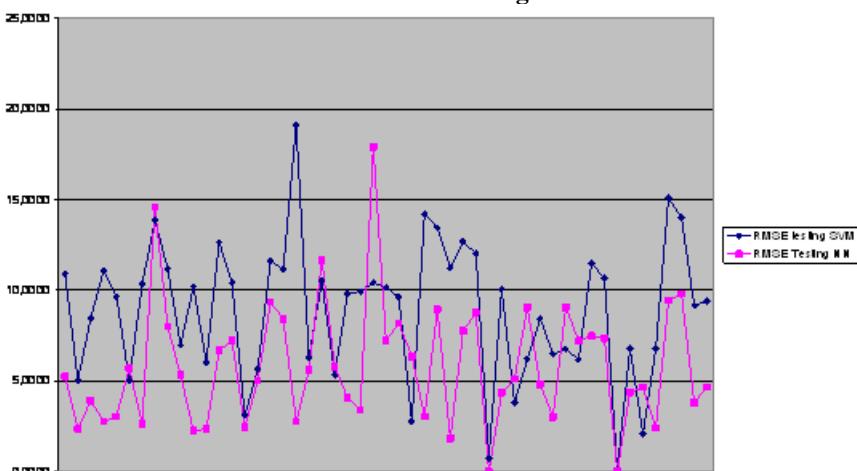
| Forest Depart. | RMSE test | RMSE train | R ² Test | C | σ | Forest Depart. | RMSE test | RMSE train | R ² Test | C | σ |
|----------------|-----------|------------|---------------------|-----|----------|----------------|-----------|------------|---------------------|-----|----------|
| Agia | 10.93 | 3.77 | 0.45 | 0.1 | 0.2 | Ioannina | 9.61 | 6.29 | 0.50 | 0.3 | 0.5 |
| Aigio | 5.03 | 2.82 | 0.53 | 0.1 | 0.5 | Edessa | 2.76 | 1.43 | 0.79 | 0.3 | 0.6 |
| Al/polis | 8.45 | 2.02 | 0.65 | 0.1 | 0.1 | Korinthos | 14.21 | 7.03 | 0.53 | 0.3 | 0.6 |
| Amfissa | 11.09 | 4.48 | 0.57 | 0.4 | 0.6 | Kozani | 13.43 | 4.73 | 0.52 | 0.3 | 0.7 |
| Aliveri | 9.66 | 4.25 | 0.47 | 0.1 | 0.1 | Lefkada | 11.24 | 1.51 | 0.56 | 0.3 | 0.6 |
| Argloida | 5.00 | 1.38 | 0.74 | 0.4 | 0.8 | Mesologgi | 12.69 | 3.85 | 0.57 | 0.2 | 0.4 |
| Atalanti | 10.31 | 4.77 | 0.51 | 0.1 | 0.4 | Metsovo | 12.00 | 5.58 | 0.70 | 0.2 | 0.4 |

Table 4. (continued)

| | | | | | | | | | | | |
|-----------|-------|------|------|-----|-----|-------------|-------|------|------|-----|-----|
| Aridaia | 13.84 | 4.26 | 0.73 | 0.4 | 0.8 | Xylokastro | 0.73 | 3.33 | 0.75 | 0.3 | 0.4 |
| Almyros | 11.19 | 3.59 | 0.51 | 0.1 | 0.1 | Thiva | 10.06 | 4.26 | 0.55 | 0.1 | 0.1 |
| Amalias | 6.96 | 3.58 | 0.79 | 0.1 | 0.2 | Thesprotia | 3.78 | 2.64 | 0.98 | 0.1 | 0.6 |
| Veroia | 10.20 | 1.39 | 0.37 | 0.1 | 0.1 | Thasos | 6.20 | 2.84 | 0.78 | 0.1 | 0.6 |
| Volos | 6.03 | 0.20 | 0.39 | 0.2 | 0.2 | Stavroupoli | 8.44 | 3.74 | 0.84 | 0.1 | 0.1 |
| Vytina | 12.64 | 4.27 | 0.46 | 0.2 | 0.4 | Stavros | 6.47 | 1.53 | 0.73 | 0.1 | 0.1 |
| Drama | 10.40 | 4.81 | 0.42 | 0.1 | 0.5 | Spercheiad | 6.73 | 0.60 | 0.94 | 0.1 | 0.1 |
| Didimotix | 3.09 | 0.94 | 0.49 | 0.1 | 0.2 | Skiathos | 6.18 | 0.20 | 0.81 | 0.2 | 0.4 |
| Tsotili | 5.62 | 4.29 | 0.55 | 0.1 | 0.1 | Serres | 11.47 | 3.30 | 0.66 | 0.2 | 0.4 |
| Dodekanis | 11.62 | 3.57 | 0.29 | 0.1 | 0.1 | Samos | 10.67 | 1.91 | 0.71 | 0.1 | 0.4 |
| Leivadia | 11.14 | 3.82 | 0.44 | 0.1 | 0.1 | Olympia | 0.11 | 0.10 | 0.80 | 0.1 | 0.2 |
| Kinouria | 19.09 | 3.81 | 0.33 | 0.1 | 0.4 | Preveza | 6.76 | 2.36 | 0.71 | 0.1 | 0.4 |
| Lidoriki | 6.28 | 2.45 | 0.67 | 0.1 | 0.4 | Polygyros | 2.08 | 0.41 | 0.99 | 0.1 | 0.4 |
| Kastoria | 10.52 | 3.37 | 0.69 | 0.2 | 0.3 | Pyrgos | 6.77 | 1.96 | 0.77 | 0.4 | 0.4 |
| Kerkyra | 5.32 | 2.27 | 0.54 | 0.1 | 0.3 | Pieria | 15.10 | 8.99 | 0.30 | 0.1 | 0.1 |
| Karpenisi | 9.81 | 2.15 | 0.43 | 0.1 | 0.2 | Patra | 14.01 | 8.96 | 0.41 | 0.2 | 0.2 |
| Karditsa | 9.92 | 4.70 | 0.18 | 0.1 | 0.2 | Nirgita | 9.16 | 2.87 | 0.51 | 0.4 | 0.4 |
| Kalampak | 10.42 | 5.51 | 0.69 | 0.1 | 0.2 | Mouzaki | 9.41 | 3.07 | 0.59 | 0.1 | 0.1 |
| Kyklaides | 10.15 | 3.85 | 0.86 | 0.1 | 0.4 | | | | | | |

The basic command line to obtain the e-regression SVM models was the following:

```
Data = xlsread('Primitive_Data1.xls') /* Read data vectors from an Excel
spreadsheet */
[Xsup, Ysup, w, wo]=svmreg(Data(:,1:9),Data(:,10),10,0.1,'gaussian',0.1,0.000001,
0)
/* Perform SVM e-regression */
```

ANN versus SVM Testing**Fig. 5.** Comparing RMSE of the ANN versus SVM in Testing for all GFDs

The figure 5 above is a graphical comparative representation of the RMSEs for the ANN versus SVM approaches. It is clearly shown that the SVMs in most of the cases appear to have bad local behavior.

4 Discussion and Conclusions

The ten GFDs with the most efficient ANN models are the following: *Lefkada, Veroia, Volos, Aigio, Pyrgos, Didymoteicho, Atalanti, Kynouria, Amfissa, Stavros*, 50% of them located in southern Greece and 50% in northern. It is not possible to find something in common between the places that correspond to the most successful ANN models. For example Lefkada (an island in the Ionian sea) has nothing in common in terms of climate conditions or vegetation with Veroia which is located in the northern main land.

On the other hand the ten cases corresponding to the ANNs with the worst performance are: Metsovo, Kozani, Thasos, Sperchiada, Dodekanisa, Pieria, Patra, Kastoria, Aridaia, Kalambaka (70% of them located in the northern part of the country). In the above figure 3, the curves of the ANN and SVM seem to move in a parallel mode but the ANN have a better performance with a smaller RMSE in most of the cases. However there are totally twelve (12) specific GFDs where the SVMs perform better than the ANNs namely: *Argolida, Aridaia, Amaliada, Kastoria, Kerkyra, Kalambaka, Edessa, Thesprotia, Sperchiada, Skiathos, Xylokastro, Olympia*. It must be clarified that for the last two cases of Xylokastro and Olympia the ANNs fail and only the SVMs offer reliable approaches. Totally the ANNs have very low R^2 value (less than 0.5) only for 2 out of the 49 cases (areas of *Dodekanisa and Karditsa*). On the other hand the validity of the SVMs judging with the same criterion is low for 13 cases out of the 51.

Generally we come to the conclusion that both SCA methods are reliable and very useful, because they can be used in a complementary manner (where the one does not succeed the other offers an enhanced solution). This paper has proven the potential use of ANNs and SVMs models as useful tools towards forecasting the extent of the burnt areas of the main forest species in various areas of Greece. Future research can be performed with data for longer periods of time and similar efforts can be motivated in other countries by this work.

Finally, trying to avoid bad local behavior phenomena of the SVMs, a future research effort could focus on the development of several local models related to distinct temporal windows. This research has shown the reliable applicability of both soft computing approaches in this case. More enhanced analysis can follow in the near future.

References

- Cheng, T., Wang, J.: Integrated Spatiotemporal Data Mining for Forest Fire Prediction. *Transactions in GIS* 12, 591–611 (2008)
- Council conclusions on prevention of forest fires within the EU Luxemburg (2010), http://www.consilium.europa.eu/uedocs/cms_data/docs/pressdata/en/jha/114026

3. Cortez, P.: Data Mining with Neural Networks and Support Vector Machines Using the R/rminer Tool. In: Perner, P. (ed.) ICDM 2010. LNCS, vol. 6171, pp. 572–583. Springer, Heidelberg (2010)
4. Hagan, M.T., Beale, H.B.: Neural Networks Design. PWS Publishing, Boston (1996)
5. Haykin, S.: Multilayer Perceptron. In: Neural Networks and Learning Machines, 3rd edn., ch. 4, pp. 122–139. Publishing as Pearson Prentice Hall (2009)
6. <http://www.mathworks.com/help/toolbox/bioinfo>
7. <http://rosettasoftware.com/>
8. Iliadis, L.: A decision support system applying an integrated fuzzy model for long-term forest fire risk estimation. *Environmental Modelling & Software* 20, 613–621 (2005)
9. Iliadis, L., Vangeloudh, M., Spartalis, S.: An intelligent system employing an enhanced fuzzy c-means clustering model: Application in the case of forest fires. *Computers and Electronics in Agriculture* 70(2), 276–284 (2010)
10. Kailidis, D.: Forest fires. Giapouli Publishers, Giahoudi (1990) (in Greek)
11. Kecman, V.: Learning and Soft Computing. MIT Press, USA (2001)
12. Pawlak, Z.: Theoretical Aspects of Reasoning About Data. Springer (1991)
13. Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orlowska, E., Polkowski, L. (eds.): Transactions on Rough Sets VI. LNCS, vol. 4374. Springer, Heidelberg (2007)
14. Petropoulos, G., Kontoes, C., Keramitsoglou, I.: Burnt area delineation from a unitemporal perspective based on Landsat TM imagery classification using Support Vector Machines. *International Journal of Applied Earth Observation and Geoinformation* 13, 70–80 (2011)
15. Tsataltzinos, T., Iliadis, L., Spartalis, S.: A Generalized Fuzzy-Rough Set Application for Forest Fire Risk Estimation Feature Reduction. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H. (eds.) EANN/AIAI 2011, Part II. IFIP AICT, vol. 364, pp. 332–341. Springer, Heidelberg (2011)
16. Vapnik, V.: Statistical Learning Theory. The Support Vector Method of Function Estimation (1998)
17. Vogl, T.P., Mangis, J.K., Rigler, A.K., Zink, W.T., Alkon, D.L.: Accelerating the convergence of the backpropagation method. *Biological Cybernetics* 59, 257–263 (1998)
18. Yu, Y.P., Omar, R., Harrison, R., Sammاثuria, M.K., Nik, A.R.: Pattern clustering of forest fires based on meteorological variables and its classification using hybrid data mining methods. *Journal of Computational Biology and Bioinformatics Research* 3(4), 47–52 (2011) ISSN: 21412227
19. Zammitt, O., Descombes, X., Zerubia, J.: Support Vector Machines for burnt area discrimination. *Rapport de Recherche nr 6343 Unité de recherche INRIA (Institute National De Recherche En Informatique et en Automatique), ISRN INRIA/RR—6343*, Sophia Antipolis, France (2007) ISSN 0249-6399

Prediction of Surface Texture Characteristics in Turning of FRPs Using ANN

Stefanos Karagiannis¹, Vassilis Iakovakis², John Kechagias²
Nikos Fountas³, and Nikolaos Vaxevanidis³

¹ Dept. of Mechanical Engineering, Technological Education Institution of Western Macedonia, Greece

² Dept. of Mechanical Engineering, Technological Education Institution of Larissa, Greece

³ Dept. of Mechanical Engineering Educators, School of Pedagogical & Technological Education (ASPETE), GR-14121, Greece

Abstract. The objective of the present study is to develop an artificial neural network (ANN) in order to predict surface texture characteristics for the turning performance of a fiber reinforced polymer (FRP) composite. Full factorial design of experiments was designed and conducted. The process parameters considered in the experiments were cutting speed and feed rate, whilst the depth of cut has been held constant. The corresponding surface texture parameters that have been studied are the R_a and R_t . A feed forward back propagation neural network was fitted on the experimental results. It was found that accurate predictions of performance can be achieved through the feed forward back propagation (FFBP) neural network developed for the surface texture parameters.

Keywords: Fiber reinforced composite (FRC), Turning, surface texture parameters, Artificial neural networks (ANNs).

1 Introduction

Fiber reinforced polymer composites (FRC) constitute an important class of materials in advanced structural applications owing to their light weight, high stiffness and specific strength, and nowadays are competitive to metals. They are widely used for manufacture in diverse industrial fields: defence, car, aerospace and electronic industries.

The structure of FRPs is heterogeneous; it consists of two phases of materials possessing different mechanical and thermal properties, which, along with anisotropy caused by the orientation of the fibres, affect chip and surface formation [1-3] as well as machinability.

Interest in the machining of composite materials by conventional machining techniques has grown, and attempts have been made to predict machinability parameters [1-7].

Nevertheless, because of the aforementioned material peculiarities it has been difficult to predict machinability parameters reliably.

Texture and mostly roughness of engineering surfaces are among the most crucial machinability parameters that influence functionality and serviceability of machined parts. The characterisation and evaluation of surface roughness has been a demanding metrological task over several decades. Four surface parameters have been adopted by international standards and proposed in view of numerous research studies; indicative references [4-9]. The most widely used surface descriptions are arithmetic and statistical surface parameters, with the latter being more descriptive.

It is acceptable for a contemporary analysis of roughness to be multi-parametric, as different aspects of surface characteristics are needed for high machining demands. The association of the parameter values with cutting conditions and especially the modelling of possible correlation is a challenging perspective.

An extended multi-parameter analysis of surface roughness and morphology is still lacking, although necessary under current requirements for manufacturability and functionality. The modelling of such surface texture parameters is especially helpful to end users when predictions of performance measures are needed.

In the present study the application of ANN models for the prediction of the surface roughness in turning of Ertalon GF-30 composite is described and discussed. The arithmetic mean roughness (centre line average), R_a (CLA), and maximum peak to valley distance, R_t , were examined towards topographic and tribological (functional) characterisation.

These parameters serve as roughness measures and cutting conditions –i.e., feed-rate and cutting speed are employed as the controlling factors (independent variables) in the postulated models. The independent variables are considered at assigned levels, based on a full factorial design of experiments and varied over a range, from semi-roughing to finishing, typically applied in turning of this material.

2 Experimental Set-Up

The composite used for cutting is specified as ERTALON 66- GF30 (PA 66-GF 30) (black). In comparison with pure PA 66 this type of polyamide has been reinforced by the addition of 30% of glass fibres. It is characterised by better expansion and compression endurance. It also possesses better mechanical properties and rigidity. Insignificant hygroscopic behaviour ensures high creep resistance, dimensional stability and abrasion resistance.

Polyamide Ertalon finds nowadays, a wide range of applications; specific requirements of a certain application can be facilitated by subtle variations in physical properties caused by a different molecular chain structure in the various grades of the product (Table 1).

The test specimens were in the form of bars of 150 mm in diameter and 500 mm in length. They were carefully clamped on both the headstock and the tailstock.

The material of the cutting tool was a P-20 cemented carbide of throwing insert type and of square form. The clearance angle was $\alpha = 5^\circ$, the tip radius 0.8 mm, and rake angle $\gamma = +6^\circ$.

Table 1. Mechanical and thermal properties of PA 6 and PA 66-GF 30

| Mechanical and thermal properties | PA 6 | PA 66-GF30 | Units |
|---|----------------|------------|-------------------|
| Tensile modulus (E) | 1400 | 3200 | MPa |
| Rockwell hardness | M85 | M76 | - |
| Charpy impact resistance | Without fract. | 50 | KJ/m ² |
| Tensile strength | 76 | 100 | MPa |
| Melting temperature | 220 | 255 | °C |
| Density | 1.14 | 1.29 | g/cm ³ |
| Coefficient of thermal expansion (<150°C) | 90x10-6 | 50x10-6 | m(m·k) |
| Coefficient of thermal expansion (>150°C) | 105x10-6 | 60x10-6 | m(m·k) |

The measured surface roughness amplitude parameters were the following:

- R_a (CLA), arithmetic mean roughness (centre line average).
- R_t , maximum peak to valley distance.

The Ertalon composite rod was turned at cutting speeds 100, 200 and 400 [m/min]. The depth of cut was set to $a = 1$ [mm]. The following feed-rate values were applied: 0.05; 0.10; 0.16; 0.20 and 0.32 [mm/rev]. Note that the above values correspond to finish and semi-finish cutting conditions in order to broaden the evaluation. Range.

The resulting values of the roughness parameters together with the corresponding cutting conditions are tabulated below (Table 2).

Table 2. Measurements of the roughness parameters considered

| No Exp | Feed rate (f) mm/rev | Cut. speed (V) m/min | R_a μm | R_t μm |
|--------|-------------------------|-------------------------|-------------|-------------|
| 1 | 0.05 | 100 | 1.301 | 15.733 |
| 2 | 0.10 | 100 | 2.237 | 15.633 |
| 3 | 0.16 | 100 | 3.257 | 23.567 |
| 4 | 0.20 | 100 | 5.237 | 40.433 |
| 5 | 0.32 | 100 | 9.223 | 45.133 |
| 6 | 0.05 | 200 | 1.973 | 16.167 |
| 7 | 0.10 | 200 | 2.447 | 18.300 |
| 8 | 0.16 | 200 | 3.490 | 32.300 |
| 9 | 0.20 | 200 | 5.390 | 36.533 |
| 10 | 0.32 | 200 | 6.523 | 42.200 |
| 11 | 0.05 | 400 | 2.113 | 16.667 |
| 12 | 0.10 | 400 | 2.193 | 20.833 |
| 13 | 0.16 | 400 | 3.833 | 32.767 |
| 14 | 0.20 | 400 | 5.387 | 35.000 |
| 15 | 0.32 | 400 | 9.673 | 58.467 |

In general, the parameters considered become more pronounced at the lower and higher cutting speeds. This can be ascribed to the fact that increase in cutting speed is favourable for cutting but higher speeds lead to vibration due to the reduced elasticity of the polymer matrix.

3 Neural Network Set-Up

In general, artificial neural networks (ANNs) are parallel-distributed information processing systems that demonstrate the ability to learn, recall, and generalise from training patterns or data. Models of NN are specified by three basic entities: models of synaptic interconnections and structures, models of the neurons, and the training rules for updating the connecting weights [10-14].

An ANN consists of a set of highly interconnected neurons that each neuron output is connected through weights to other neurons or to itself. Hence, the structure that organises these neurons and the connection geometry among them should be specified for an ANN.

An artificial neural network consists of at least three layers, where input vectors (p_i) applied at the input layer and output vectors (a_i) are obtained at the output layer (Fig. 1).

Each layer consists of a number of neurons which are also called processing elements (PE). PEs can be viewed as consisting of two parts: input and output. Input of a PE is an integrated function which combines information coming from the net. Considering a feed forward NN, the net input to PE i in layer k+1 is:

$$n_i^{k+1} = \sum_j w_{i,j}^{k+1,k} a_j^k + b_i^{k+1} \quad (1)$$

where $w_{i,j}$ and b_i are the corresponding weights and biases respectively.

The output of PE i will be

$$a_i^{k+1} = f^{k+1}(n_i^{k+1}) \quad (2)$$

where f is the transfer function of neurons in the (k+1)th layer.

Training of the network uses training rules to adjust the weights associated with PE inputs. Unsupervised training uses input data alone, while supervised training works by showing the network a series of matching input and output examples $\{(p_1, t_1), (p_2, t_2), \dots, (p_Q, t_Q)\}$.

MATLAB® programme was used to create, train, and test the feed forward back propagation neural network (FFBP-NN) through a network data manager. Three layers are considered: The input layer which contains parameter settings (input variables), the output layer which contains the responses (dependant variables) and the hidden layers which facilitates prediction operations and are determined by a “trial and error” procedures.

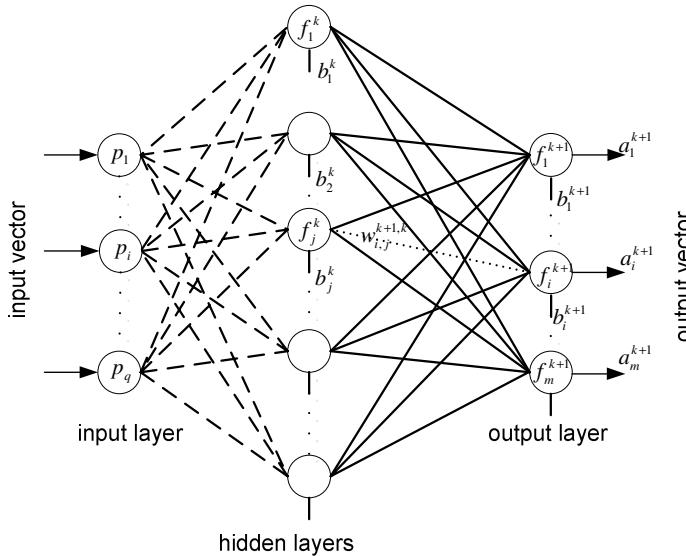


Fig. 1. A multilayer feed-forward ANN

A number of trials were executed to select the appropriate topology of the FFBP-ANN, which is shown in Fig. 2.

Feed rate and cutting speed were used as the input vector into the ANN. Surface roughness parameters were used as the output layer. One hidden layer was selected having 12 neurons.

The hyperbolic tangent sigmoid transfer function (tansig) was used as the transfer function for the hidden layer (Eq. 3). The transfer function for the output layer was the linear function (Eq. 4).

$$\begin{aligned}
 a^1 &= f^1(w^{1,1}p^1 + b^1) = \\
 \tan sig(w^{1,1}p^1 + b^1) &= \\
 \tan sig(n) &= \frac{2}{1+e^{-2n}} - 1
 \end{aligned} \tag{3}$$

$$\begin{aligned}
 a^2 &= f^2(w^{2,1}a^1 + b^2) = \\
 purelin(w^{2,1}a^1 + b^2) &= \\
 purelin(n) &= n
 \end{aligned} \tag{4}$$

Training functions repeatedly apply a set of input vectors to a network, updating the network each time, until some stopping criteria are met. Stopping criteria can consist of a maximum number of epochs, a minimum error gradient, and an error goal.

The Levenberg-Marquardt [15, 16] algorithm was selected for training the FFBP-NN –which is a variation of the classic backpropagation algorithm that, unlike other variations that use heuristics, relies on numerical optimisation techniques in order to minimise and accelerate the required calculations– resulting in much faster training; see also Ref [12] for details.

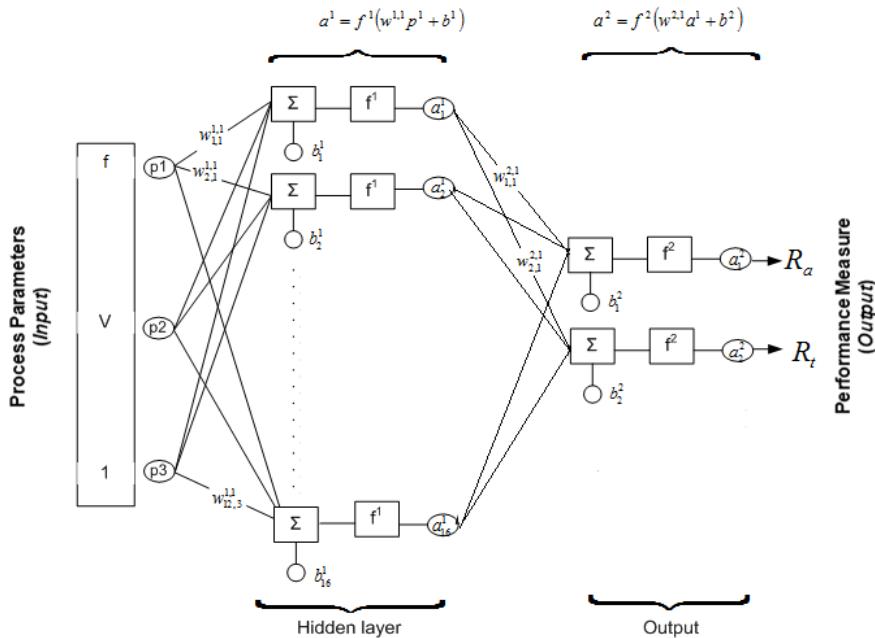


Fig. 2. FFBP-NN details (scheme was adopted by the MATLAB® programme)

The LEARNGDM was used as the ‘adaption learning function’, which is the gradient descent with momentum weight and bias learning function. Biases (b_j) are simply added to the product ($w_{j,i}+b_j$).

The performance of the FFBP-ANN was measured with the mean squared error (MSE) of the testing subset which was calculated by the form:

$$\begin{aligned}
 MSE &= \\
 \frac{1}{2} \sum_{q=1}^Q (t_q - a_q^M)^T (t_q - a_q^M) &= \\
 \frac{1}{2} \sum_{q=1}^Q e_q^T e_q
 \end{aligned} \tag{5}$$

where a_q^M is the output of the network, corresponding to the q^{th} input p_q , while $e_q = (t_q - a_q^M)$ is the error term. It must be noted that the outcome of the training greatly depends on the initialisation of the weights, which are randomly selected. The

performance, regression and training state plots can be seen in Figs. 3-5, respectively. The MSE of training of the created ANN was equal to $1.1925e^{-14}$ and its training took 5 epochs to complete (Fig. 5). The best validation performance is 0.40427 at epoch 5 (Fig. 3).

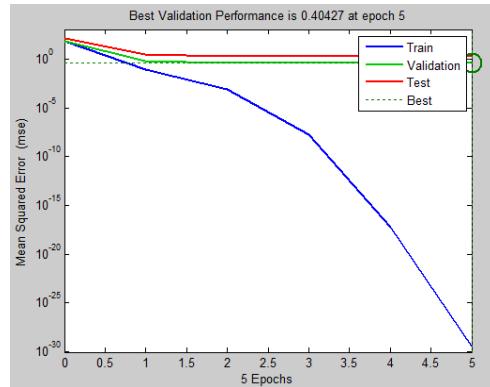


Fig. 3. Performance plot (scheme was modified from the MATLAB[®] programme)

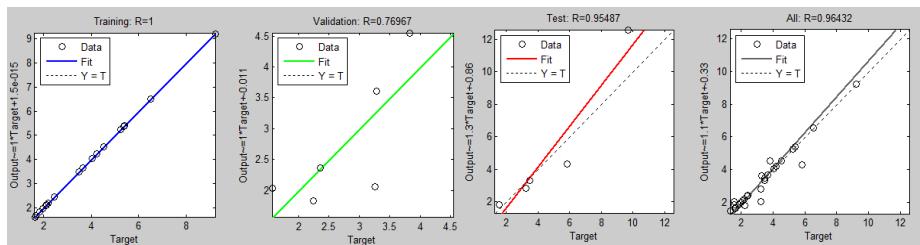


Fig. 4. Regression plots (scheme was modified from the MATLAB[®] programme)

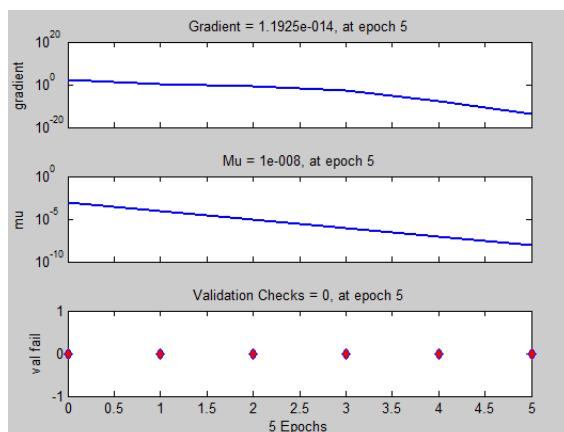


Fig. 5. Training state plots (scheme was modified from the MATLAB[®] programme)

Regression values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship. The regression analysis of the developed ANN model resulted to R values for training, validation and testing, which were very close to 1; a fact indicative of a very good agreement between the output (simulated values) and the target (experimental values). A good correlation between experimental data and simulated data (neural network output) is evident; all: R values are close to 0.964; Fig. 4).

To avoid the inherent bias owing to the different magnitudes of the parameters involved, proper normalization was applied; see also [17].

4 Evaluation of the FFBP-NN

Two evaluation experiments were conducted in the following manner:

In the first experiment the fourteenth out of fifteen experiments were used as training experiments; except for experiment No 4, which was used as the evaluation experiment (Table 3).

The same procedure was followed using experiment No 9 as the evaluation experiment (Table 4).

From the evaluation experiments realised, it is evident that even though only a few experiments were used for training the ANN, yet, efficient prediction results were obtained for R_a and R_t parameters. Better accuracy of the ANN is expected, if all the experiments were used as training experiments.

Table 3. First evaluation experiment

| Multiplied by | Exp. No 4 | | | | |
|---|---------------|---------------|----------------|--------|-----------|
| 1000 | Feed rate | 200 | | | |
| 1 | Cutting speed | 100 | | | |
| | | Actual values | Network output | Error | Error (%) |
| | R_a | 5.237 | 6.447 | 12.106 | 23.1% |
| 0.1 | R_t | 4.043 | 3.788 | -2.549 | -6.3% |
| (Training Experiments: 1,2,3,5,6,7,8,9,10,11,12,13,14,15) | | | | | |

Table 4. Second evaluation experiment

| Multiplied by | Exp. No 9 | | | | |
|---|---------------|---------------|----------------|--------|-----------|
| 1000 | Feed rate | 200 | | | |
| 1 | Cutting speed | 200 | | | |
| | | Actual values | Network output | Error | Error (%) |
| | R_a | 5.39 | 5.199 | -1.903 | -3.5% |
| 0.1 | R_t | 3.653 | 3.758 | 1.048 | 2.9% |
| (Training Experiments: 1,2,3,4,5,6,7,8,10,11,12,13,14,15) | | | | | |

5 Conclusions

From the results reported in the present work it was verified that accurate predictions of R_a and R_t surface characteristics during the process of turning of an FRP composite can be achieved through an FFBP neural network with one hidden layer of 12 neurons. Moreover, the following conclusions can be drawn:

- The machined surface of the composite under study, when intensifying the cutting conditions and especially feed rate, becomes rougher, wavier, steeper, more open (exposed), emptier but less complex.
- The surface texture parameters become more pronounced at the lower and higher cutting speed values. The meaning of this is that cutting speed implies a critical maximum speed to benefit from the softening of the thermoplastic material and subsequently to perform advantageous cutting.
- These findings may be a step towards a full functional and quality oriented topographic characterisation in the machining of polymer composites.
- Furthermore, based on NN modelling and the evaluation procedure, roughness parameters R_a and R_t can be predicted efficiently even if fewer experiments were used to train the FFBP artificial NN.
- The above analysis is useful for end users when predictions of performance measures are needed. This methodology could be easily applied to different materials and initial conditions for optimisation of machining performance of other composites.

Acknowledgements. This research is implemented through the Operational Program "Education and Lifelong Learning" and is co-financed by the European Union (European Social Fund) and Greek national funds. The research is based on preliminary work realised by the late Prof. G. Petropoulos at the Laboratory for Manufacturing Processes, Dept. of Mechanical Engineering, University of Thessaly, Volos, Greece, and this paper is dedicated to his memory.

References

1. Mata, F., Petropoulos, G., Ntziantzias, I., Davim, J.P.: A surface roughness analysis in turning of polyamide PA-6 using statistical techniques. *Int. J. Mater. Prod. Technol.* 37(1-2), 173–187 (2010)
2. Palanikumar, K., Mata, F., Davim, J.P.: Analysis of surface roughness parameters in turning of FRP tubes by PCD tool. *J. Mater. Process. Technol.* 204, 469–474 (2008)
3. Davim, J.P., Silva, L.R., Festas, A., Abrão, A.M.: Machinability study on precision turning of PA66 polyamide with and without glass fibre reinforcing. *Materials and Design* 30(2), 228–234 (2009)
4. Aravindan, S., Naveen, S.A., Noorul, H.A.: A machinability study of GFRP pipes using statistical techniques. *Int. J. Adv. Manuf. Technol.* 37, 1069–1081 (2008)

5. İşik, B.: Experimental investigations of surface roughness in orthogonal turning of unidirectional glass-fiber reinforced plastic composites. *Int. J. Adv. Manuf. Technol.* 37, 42–48 (2008)
6. Palanikumar, K., Karunamoorthy, L., Karthikeyan, R.: Parametric optimization to minimise the surface roughness on the machining of GFRP composites. *J. Mater. Sci. Technol.* 22(1), 66–72 (2006)
7. Kechagias, J., Petropoulos, G., Iakovakis, V., Maropoulos, S.: An investigation of surface texture parameters during turning of a reinforced polymer composite using design of experiments and analysis. *Int. J. of Experimental Design and Process Optimisation* 1(2-3), 164–177 (2009)
8. Gadeltawla, E.S., et al.: Roughness parameters. *J. Mat. Proces. Techn.* 56, 1–13 (2002)
9. Petropoulos, G.P., Pantazaras, C.N., Vaxevanidis, N.M., Ntziantzias, I., Korlos, A.: Selecting subsets of mutually unrelated ISO 13565-2:1997 surface roughness parameters in turning operations. *Int. J. Comp. Mat. Sci. & Surf. Eng.* 1(1), 114–128 (2007)
10. Lin, C.T., Lee, G.C.S.: Neural fuzzy systems-A neuro-fuzzy synergism to intelligent systems, pp. 205–211. Prentice Hall PTR (1996)
11. Kechagias, J., Iakovakis, V.: A neural network solution for LOM process performance. *Int. J. Adv. Manuf. Technol.* 43(11), 1214–1222 (2008)
12. Vaxevanidis, N.M., Markopoulos, A., Petropoulos, G.: Artificial neural network modelling of surface quality characteristics in abrasive water jet machining of trip steel sheet. In: Davim, J.P. (ed.) *Artificial Intelligence in Manufacturing Research*, ch. 5, pp. 79–99. Nova Publishers (2010)
13. Ozel, T., Karpat, Y.: Predictive modeling of surface roughness and tool wear in hard turning using regression and neural networks. *Int. J. Mach. Tools Manuf.* 45, 467–479 (2005)
14. Jiao, Y., Lei, S., Pei, Z.J., Lee, E.S.: Fuzzy adaptive networks in machining process modeling: surface roughness prediction for turning operations. *Int. J. Mach. Tools Manuf.* 44, 1643–1651 (2004)
15. Levenberg, K.: A method for the solution of certain problems in least squares. *Quart. Appl. Math.* 2, 164–168 (1944)
16. Marquardt, D.: An algorithm for least-squares estimation of nonlinear parameters. *SIAM. J. Appl. Math.* 11, 431–441 (1963)
17. El-Mounayri, H., Kishawy, H., Tandon, V.: Optimized CNC end-milling: A practical approach. *Int. J. CIM* 15, 453–470 (2002)

A State Space Approach and Hurst Exponent for Ensemble Predictors

Ryszard Szupiluk¹ and Tomasz Ząbkowski²

¹ Warsaw School of Economics, Al. Niepodleglosci 162, 02-554 Warsaw, Poland

² Warsaw University of Life Sciences, Nowoursynowska 159, 02-787 Warsaw, Poland
`rszupi@sgh.waw.pl, tomasz_zabkowski@sggw.pl`

Abstract. In this article we propose a concept of ensemble methods based on deconvolution with state space and MLP neural network approach. Having a few prediction models we treat their results as a multivariate variable with latent components having destructive or constructive impact on prediction. The latent component classification is performed using novel variability measure derived from Hurst exponent. The validity of our concept is presented on the real problem of load forecasting in the Polish power system.

Keywords: state space approach, Hurst exponent, Independent Component Analysis, ensemble methods.

1 Introduction

The ensemble methods integrate the information generated by many models what usually concern combining of the models results. The well-known procedures for regression and classification improvement are bagging, boosting or stacked regression [2,5]. In this paper, we develop an alternative concept based on multidimensional decompositions [11]. Having a few prediction models (primary results) we treat their outcomes as a multivariate variable with latent components. These components can have destructive or constructive impact on the outcome, i.e. prediction. The constructive ones are associated with the desired value. On the other hand the destructive components can be present due to many reasons like: missing data, lack of significant variables or not precise parameter estimation and distribution assumptions. The identification and elimination of destructive components should improve final prediction.

Identification of hidden components can be done in many ways but it seems reasonable to use blind source separation methods (BSS) [4]. In this methodology, application of blind signal separation methods can be treated as specific filtration processes resulting in elimination of noises from predicted signals. The term *aggregation* for this process of is appropriate because the effect of the filtration is a certain combination of the initial prediction results.

This aggregation concept requires solutions to two major issues. The first one is selection of appropriate separation method, and the second one is correct identification of destructive components. The choice of separation method is typical

for blind source separation in which we make a priori assumptions about the data characteristics and the mixing system, taking into account separation methods such as independent component analysis (ICA), principal component analysis (PCA), smooth component analysis (SmCA), or second order statistics algorithm like AMUSE [4,6,10]. In this paper we will present the problem in general form of which is the dynamic state space system. The main motivation for choosing dynamic state space system is its general frame for separation/deconvolution problems which, in special cases, can represent such popular data models as autoregressive moving average (ARMA) or autoregressive integrated moving average (ARIMA) [4,14].

The issue of determining the nature of the hidden components as destructive or constructive can be done in various ways [11]. One possibility, which is especially adequate for complex separating systems or a large number of aggregated models, is to compare the destructive components with the random noise. That is, to distinguish signals which are more or less random, since in case of components, we rarely deal with the pure white noise and pure deterministic signals. The assessment of the relative randomness (noise ratio) requires the adoption of specific measures or criteria. For this task we develop the Hurst exponent application [7].

2 Prediction Aggregation with State Space Model and Nonlinear Mixing

Let $x_i(k)$ denote the particular prediction results from the model $i=1,\dots,m$ for observation indexed by $k=1,\dots,N$. Let also $\mathbf{x}(k)=[x_1(k),\dots,x_m(k)]^T$ denote the collection of the models in a multivariate variable. We assume that each prediction result $x_i(k)$ is a mixture of the latent components $s_j(k)$, $j = 1, \dots, n$. The components can be constructive $s_l(k) = \hat{s}_l(k)$, associated with the predicted variable and destructive $s_p(k) = \tilde{s}_p(k)$, associated with the inaccurate and missing data, imprecise estimation, misspecified distributions etc. In the following discussion we assume, for simplicity, that $m=n$. Next, if we assume that $\mathbf{s}(k)=[\hat{s}_1(k),\dots,\hat{s}_r(k),\tilde{s}_{r+1}(k),\dots,\tilde{s}_m(k)]^T$ is vector of the latent components, the relation between observed prediction results and latent components for the dynamical linear mixing system can be represented by state space model as

$$\mathbf{u}(k+1) = \overline{\mathbf{A}}\mathbf{u}(k) + \overline{\mathbf{B}}\mathbf{s}(k), \quad (1)$$

$$\mathbf{x}(k) = \overline{\mathbf{C}}\mathbf{u}(k) + \overline{\mathbf{D}}\mathbf{s}(k), \quad (2)$$

where matrices $\overline{\mathbf{A}}, \overline{\mathbf{B}}, \overline{\mathbf{C}}, \overline{\mathbf{D}} \in R^{m \times m}$ represents the mixing system and $\mathbf{u}(k)$ is state vector. The problem is to find source signals $\mathbf{s}(k)$ and system parameters $\overline{\mathbf{A}}, \overline{\mathbf{B}}, \overline{\mathbf{C}}, \overline{\mathbf{D}}$ what allow to identify and eliminate destructive components (replace some signal values $s_j(k)$ with zero). As a result, we can obtain vector with pure

constructive components $\hat{\mathbf{s}}(k) = [\hat{s}_1(k), \dots, \hat{s}_n(k), 0_{n+1}, \dots, 0_m]^T$ forwarded to mixing system give us improved prediction results $\hat{\mathbf{x}}(k)$ from:

$$\mathbf{u}(k+1) = \bar{\mathbf{A}}\mathbf{u}(k) + \bar{\mathbf{B}}\hat{\mathbf{s}}(k), \quad (3)$$

$$\hat{\mathbf{x}}(k) = \bar{\mathbf{C}}\mathbf{u}(k) + \bar{\mathbf{D}}\hat{\mathbf{s}}(k). \quad (4)$$

This process can also be treated as signals $x_i(k)$ filtration, in which some internal components with particular characteristics are eliminated. Let's note that term $\bar{\mathbf{D}}\hat{\mathbf{s}}(k)$ is equivalent with $\hat{\mathbf{D}}\mathbf{s}(k)$ where $\hat{\mathbf{D}} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n, \mathbf{0}, \mathbf{0}, \dots, \mathbf{0}]$ is matrix created as a result of replacement a certain columns corresponding to destructive signals with zero vectors.

The most adequate methods to solve the above system identification problem seem to be the blind signal separation and deconvolution techniques [4]. The process of dynamic system space identification with BSS can be based on dynamic ICA method what provide to following separation system:

$$\mathbf{v}(k+1) = \mathbf{Ay}(k) + \mathbf{Bx}(k), \quad (5)$$

$$\mathbf{y}(k) = \mathbf{Cv}(k) + \mathbf{Dx}(k). \quad (6)$$

The output of the separation system $\mathbf{y}(k) \approx \mathbf{s}(k)$ can be represented by transfer function $\mathbf{H}(z) = \bar{\mathbf{C}}(z\mathbf{I} - \bar{\mathbf{A}})^{-1}\bar{\mathbf{B}} + \bar{\mathbf{D}}$ and $\mathbf{G}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$ as [14]

$$\mathbf{y}(k) = \mathbf{G}(z)\mathbf{H}(z)\mathbf{s}(k) = \mathbf{P}\Lambda(z)\mathbf{s}(k), \quad (7)$$

where z is time delay operator, \mathbf{P} is permutation matrix, Λ is filtration matrix. It means that separated signals can be permuted and filtered, what are typical ambiguities for dynamic multichannel bind deconvolution methods [4]. The matrices estimation for separation system can be performed via following rules [14]

$$\mathbf{C}(k+1) = \mathbf{C}(k) - \eta[\varphi(\mathbf{y}(k))\mathbf{v}^T(k)] \quad (8)$$

and

$$\mathbf{D}(k+1) = \mathbf{D}(k) + \eta[\mathbf{I} - \varphi(\mathbf{y}(k))]\mathbf{y}^T(k)\mathbf{D}(k) \quad (9)$$

where $\varphi_i(y_i) = -\frac{d \log p_i(y_i)}{dy_i} = -\frac{p'_i(y_i)}{p_i(y_i)}$. The estimation of the matrices \mathbf{A}

and \mathbf{B} is somewhat complex task. One of the possible solutions is to make some a priori assumptions about their values. The other approach can utilize information backpropagation approach [13]. To estimate the state vector the modified Kalman filtering with hidden innovations can be used [14]. In particular case, with null matrices $\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}$ separation process is reduced to standard ICA method and the filtering process is determined by the separation \mathbf{D} and recombination $\hat{\mathbf{D}}$ matrices as

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{D}} \mathbf{D} \mathbf{x}(k). \quad (10)$$

For such case, the algorithm (9) is an effective method to find a separation matrix. However, the adoption of filtration technique defined by the expression (10) instead of (3) - (4), means the adoption of static system in which we lose general properties of dynamic models. From the other hand, for the general dynamic case, we do not have simple methods for all parameters estimation. Moreover, the signals obtained from such a system can be filtered version of the input signals, while, taking into account (7), it makes uncertain to process the effective identification of the destructive components. Therefore, the question arises whether is it possible to find a solution that uses the information supplied by dynamic blind source separation methods.

In our model aggregation concept with blind signal separation methods we can utilize findings from Bussgang [3] algorithm where time delay linear system is modeled by nonlinear functions. The main advantage for such approach is prediction context, in which we have target values derived from historical data. It allows us to apply an adaptive nonlinear model like MLP neural network defined as

$$\hat{\mathbf{x}} = \mathbf{g}_2(\mathbf{W}_2[\mathbf{g}_1(\mathbf{W}_1\mathbf{y} + \mathbf{w}_1)] + \mathbf{w}_2), \quad (11)$$

where $\mathbf{g}_i(.)$ is non-linear function vector, \mathbf{W}_i is weight matrix, and \mathbf{w}_i is biased for i -th layer, $i=1,2$. It is very likely that if a neural network learning starts from a point which is the result of the decomposition stage with initial values $\mathbf{W}_1(0)=\hat{\mathbf{D}}$ and $\mathbf{W}_2(0)=\mathbf{C}$ then the final results will be better than these obtained from re-composition.

Additional motivation for the concept of non-linear re-recomposition is the fact that the matrix $\hat{\mathbf{D}}$ is the best matrix that can be obtained through the complete elimination of destructive components (treated as noise, interference) in the linear model. This means the assumption that obtained components represent a pure form of being destructive or constructive. In practice, however, the situation may look much different. Components can be noisy or can have different impact on the individual models. By introducing non-linear, multi-layered, adaptive re-composition and making its optimization (learning) we get the results which are weakening slightly the classification but are the more appropriate to approximate the actual importance of the component. Often, the best estimates can be obtained with appropriate suppression, enhancing or non-linear transformation of the underlying components. In equation (10) a two-layer neural network structure was adopted, as an example illustrating the concept of a generalized mixing selection. In practice, the issue of the proper selection of a neural network is in line with the general stream of neural network modeling.

3 Multivariate Hurst Analysis as Decision System

The key issue in the proposed concept is the classification of the components into destructive or constructive. In case of a small number of models, and thus a small number of basic components, classification of components can be done by the full

search. In fact, this means examining the impact of the elimination of basic components and their combination on the final prediction outcome. In case of a larger number of models, examining the impact of all possible combinations of the basic components on the forecasts can be computationally difficult. Therefore, assessment of the component impact must be done based on certain characteristics or derived criteria. One possible approach is to assume that the destructive components are random noises. However, it should be noted that most of the destructive components are not pure white noises, but they are rather a mixture of random and deterministic components. Therefore, it is rather impossible to determine in advance what level of randomness can be associated with each component. For this reason, this problem must be considered in terms of the mutual similarity of the signals and noise rather than their individual characteristics. Approach that can be applied here is adopted for multivariate case the R/S analysis and Hurst exponent interpretation.

Hurst exponent allows for the assessment of internal relations and the similarity of the signal itself [7-9]. It can be obtained by dividing signal into parts with n -observation each and calculate in each part variance σ and range $R = \max(y) - \min(y)$ what lead us to following equations

$$E\left\{\frac{R}{\sigma}\right\}_n = cn^H. \quad (12)$$

where c is a constant, n is the number of observation in each part, H is Hurst exponent, and the expectation is taken over the all parts. Taking the logarithm of (12) we have

$$\ln E(R/\sigma)_n = \ln c + H \ln n. \quad (13)$$

giving us H value from regression on n . The value of the H near 0.5 means pure white noise whereas values near the 1 mean the deterministic signal [8]. Hurst exponent defined as above can be used for the analysis of individual signals. To deal with a multidimensional relations it is necessary to use multi-fractal approach, in which the determination and analysis with interpretation of the Hurst exponent is relatively complex and ambiguous [1]. With the use of basic formulas related to the R / S analysis and at the same time to avoid inconvenience related to multi-fractal approach we propose the following concept based on Hurst exponent.

For the squared difference between Hurst exponents for the given signals we have

$$(H(x_1) - H(x_2))^2 = H^2(x_1) - 2H(x_2)H(x_1) + H^2(x_2) \quad (14)$$

and taking into account that for two different random white noises v_1, v_2 we have

$$H(v_1 + v_2) \approx H(v_1) \approx H(v_2), \quad (15)$$

then we can define a similarity measure of random signals as

$$d = H^2(x_1) - 2H(x_1 + x_2)H(x_1) + H^2(x_2) = H^2(x_1) - 2H^2(x_1 + x_2) + H^2(x_2), \quad (16)$$

or in a relative form as:

$$z_{ij} = \frac{H^2(x_i) + H^2(x_j)}{2H^2(x_i + x_j)}. \quad (17)$$

The interpretation of this measure is relatively simple and intuitive. Hurst exponent for the sum of the random components (white noises) is equal to the Hurst exponent for a single white noise. Similarly, the Hurst exponent for the sum of two deterministic signals is equal to the Hurst exponent of the single deterministic signal. As a result we get the measure which indicates the similarity of signals with the same level of randomness (determinism). Mutual similarity for many variables x_1, x_2, \dots, x_n can be presented as a matrix

$$\mathbf{Z} = [z_{ij}] \in \Re^{n \times n}. \quad (18)$$

The measure (17) can be generalized in the form of multi-dimensional case as

$$z_{i_1, i_2, \dots, i_n} = \frac{\sum_{j=1}^n H^2(x_j)}{nH^2(\sum_{i=1}^n x_i)}. \quad (19)$$

To illustrate the effect of measure (17) we will use a set of artificially generated signals y_1, y_2, \dots, y_6 , shown in Figure 1. These signals are obtained after ICA transformation, which means that they are independent of each other (and thus decorrelated) and also they have the same individual variances. In this case, based only on correlation characteristics it is difficult to assess their similarities. However, the visual inspection quite clearly shows the similarity between certain signals; please see the first and the second signal from the top in Fig. 1.

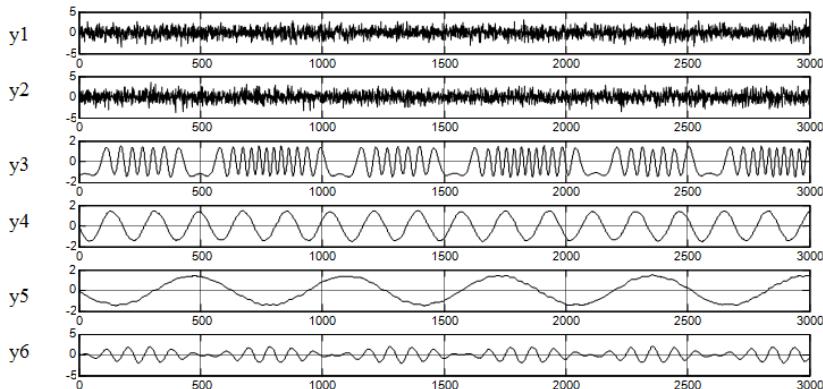


Fig. 1. Similarity between decorrelated signals

In this case, the application of measure (17) allows the assessment of the signals similarity that is, to a large extent, consistent to individual introspection. Elements of matrix Z to assess the signals similarity are presented in Table 1.

Table 1. Similarity measured with $z_{ij} = z(y_i, y_j)$ for the signals presented in Fig. 1

| z_{ij} | y_1 | y_2 | y_3 | y_4 | y_5 | y_6 |
|----------|--------|--------|--------|--------|--------|--------|
| y_1 | 1.0000 | 1.0972 | 0.6704 | 0.5842 | 0.5231 | 0.7891 |
| y_2 | 1.0972 | 1.0000 | 0.6601 | 0.5762 | 0.5246 | 0.6972 |
| y_3 | 0.6704 | 0.6601 | 1.0000 | 0.8980 | 1.1259 | 0.8863 |
| y_4 | 0.5842 | 0.5762 | 0.8980 | 1.0000 | 0.9852 | 0.9146 |
| y_5 | 0.5231 | 0.5246 | 1.1259 | 0.9852 | 1.0000 | 0.8893 |
| y_6 | 0.7891 | 0.6972 | 0.8863 | 0.9146 | 0.8893 | 1.0000 |

In order to present the similarity in a more transparent way, we assume, for the elements of the Z matrix, the following distance

$$d_{ij} = \text{abs}(1 - z_{ij}), \quad (20)$$

where abs is absolute value function. The d_{ij} results, for the bottom triangle part of Table 1, are presented in Table 2. It was observed that, as the most similar signals were considered signals y_4 and y_5 which are two sinusoidal signals with different frequencies. Similarly, high scores of similarity were attributed to noise signals.

Table 2. Similarity between the signals measured as distance from unity

| d_{ij} | y_1 | y_2 | y_3 | y_4 | y_5 | y_6 |
|----------|--------|--------|--------|--------|--------|-------|
| y_1 | 0 | | | | | |
| y_2 | 0.0972 | 0 | | | | |
| y_3 | 0.3296 | 0.3399 | 0 | | | |
| y_4 | 0.4158 | 0.4238 | 0.1020 | 0 | | |
| y_5 | 0.4769 | 0.4754 | 0.1259 | 0.0148 | 0 | |
| y_6 | 0.2109 | 0.3028 | 0.1137 | 0.0854 | 0.1107 | 0 |

It was noted that this measure corresponds to the human perception of similarity. It should be also be noted that the quantitative assessment of the similarity based on the autocorrelation function, in essence, need to adopt a quantitative measure of similarity what leads to the same analysis but performed on the data that are derivatives of the original signals.

4 Load Power Forecasting

Forecasting electricity demand is an important issue from an economic point of view, both taking into account the microeconomic and macroeconomic scale. Direct financial incentives are related to the fundamental characteristics of the energy market, on which the possibility to store the electricity is very limited. Any mismatch between the size of demand and supply results in tangible losses. Over estimation, due to the impossibility of storage, causes its irretrievable loss, while under estimating leads to urgent purchase on higher prices, existing on balancing market.

Characteristics of the energy market, is naturally reflected in the financial instruments related to trade conditions on this market. It is primarily futures market well developed and characterized by very high volatility, sometimes referred as extreme [12]. However, the futures market does not satisfy all the needs, especially taking into account the short-term changes in the energy demand site. As a result, for daily operations, forecasting the electricity demand plays an important role. In particular, a short-term forecast is crucial for the economic efficiency of power sector entities, since it is associated with costly transaction realized on balancing market.

To verify the validity of the models ensemble concept we used the data from Polish power system. The data set included 86400 observations (hourly data) covering time span of 1988-1998. The available variables to create the forecast included energy demand from the last 24 hours and calendar variables such as month, day of the month, day of the week, and holiday indicator. There were six neural networks models build with different learning methods (delta, quasi-Newton, Levenberg-Marquardt) and one hidden layer (with 12, 18, 24, 27, 30, 33 neurons respectively) to forecast hourly energy consumption in Poland in next 24 hours. The models were labeled as M1:MLP12, M2:MLP18, M3:MLP24, M4:MLP27, M5:MLP30, M6:MLP33. The predictive quality of the models was measured with mean absolute percentage error (MAPE).

Table 3. Prediction results for primary models and after aggregation measured with MAPE error

| MAPE | M1 | M2 | M3 | M4 | M5 | M6 |
|----------------|--------|--------|--------|--------|--------|--------|
| primary models | 0.0239 | 0.0236 | 0.0237 | 0.0240 | 0.0240 | 0.0236 |
| aggregated | 0.0243 | 0.0225 | 0.0242 | 0.0240 | 0.0228 | 0.0239 |

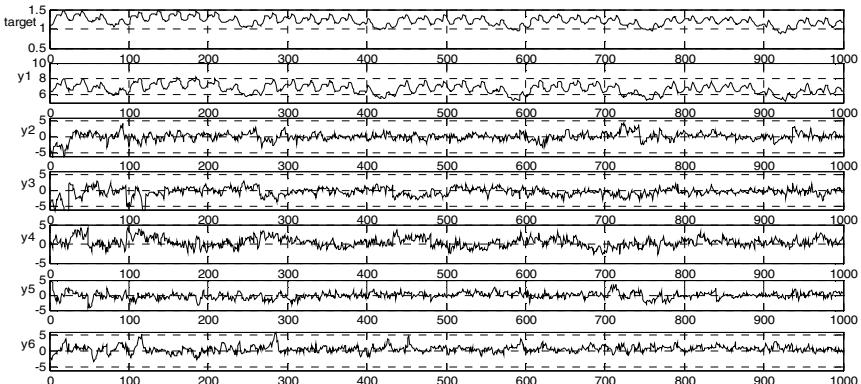


Fig. 2. The basic components and the target variable

The basic components along with the target variable (original time series of electricity demand) are shown in Figure 2. The greatest improvement was obtained after removal of the component y6. The results of the primary model and after aggregation (with filtered noise) are presented in Table 3. The identification was

based on the full numerical test. Our experiment on load data confirmed the validity of the proposed solutions. We could benefit of about 5-6% of MAPE reduction (best primary model vs. best model after aggregation).

While testing the correlation between the basic components and the target, the strong correlation between y_1 and target component can be found. From this, we can conclude that it is an important constructive component. However, in case of the other components the situation is not so clear. Based on the correlation analysis with the target and taking into account the individual characteristics of the autocorrelation function, it is not possible to distinguish, with great certainty, which components are purely destructive or purely constructive, as shown in Table 4 and Figure 3.

Table 4. Correlation coefficients for basic components and the target

| | target | y_1 | y_2 | y_3 | y_4 | y_5 | y_6 |
|--------|---------|---------|---------|---------|---------|---------|---------|
| target | 1.0000 | 0.9731 | -0.0007 | -0.0560 | 0.1064 | -0.1673 | 0.1252 |
| y_1 | 0.9731 | 1.0000 | -0.0442 | 0.0229 | 0.0575 | -0.1414 | 0.1004 |
| y_2 | -0.0007 | -0.0442 | 1.0000 | 0.1848 | -0.0948 | 0.0646 | -0.0115 |
| y_3 | -0.0560 | 0.0229 | 0.1848 | 1.0000 | -0.2031 | -0.0143 | -0.0820 |
| y_4 | 0.1064 | 0.0575 | -0.0948 | -0.2031 | 1.0000 | -0.1172 | 0.0816 |
| y_5 | -0.1673 | -0.1414 | 0.0646 | -0.0143 | -0.1172 | 1.0000 | 0.0445 |
| y_6 | 0.1252 | 0.1004 | -0.0115 | -0.0820 | 0.0816 | 0.0445 | 1.0000 |

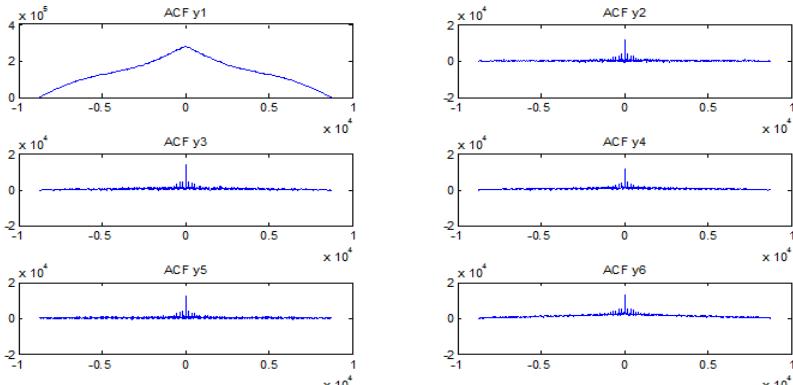


Fig. 3. Autocorrelation functions for basic components

Taking into account the z_{ij} measure and measured distance d_{ij} , what is presented in Table 5 and Table 6, it was observed that the greatest similarity to the target could be attributed to y_1 which is consistent with the correlation approach. However, in contrast to the correlation analysis, z_{ij} measure clearly shows the difference of the signal y_6 from the target – which, most likely, can be assumed as destructive component.

Table 5. Measure z_{ij} for target variable and the basic components

| z_{ij} | target | y ₁ | y ₂ | y ₃ | y ₄ | y ₅ | y ₆ |
|----------------|--------|----------------|----------------|----------------|----------------|----------------|----------------|
| target | 1.0000 | 1.0014 | 0.9778 | 0.9486 | 0.9828 | 0.9932 | 1.3171 |
| y ₁ | 1.0014 | 1.0000 | 0.9077 | 1.0280 | 0.9921 | 1.0611 | 1.3725 |
| y ₂ | 0.9778 | 0.9077 | 1.0000 | 1.1264 | 1.0667 | 1.2328 | 0.9582 |
| y ₃ | 0.9486 | 1.0280 | 1.1264 | 1.0000 | 0.9611 | 1.0371 | 0.9910 |
| y ₄ | 0.9828 | 0.9921 | 1.0667 | 0.9611 | 1.0000 | 1.0263 | 0.8660 |
| y ₅ | 0.9932 | 1.0611 | 1.2328 | 1.0371 | 1.0263 | 1.0000 | 1.1517 |
| y ₆ | 1.3171 | 1.3725 | 0.9582 | 0.9910 | 0.8660 | 1.1517 | 1.0000 |

Table 6. Similarity between the signals measured as distance d_{ij}

| d_{ij} | target | y ₁ | y ₂ | y ₃ | y ₄ | y ₅ | y ₆ |
|----------------|--------|----------------|----------------|----------------|----------------|----------------|----------------|
| target | 0 | | | | | | |
| y ₁ | 0.0014 | 0 | | | | | |
| y ₂ | 0.0222 | 0.0923 | 0 | | | | |
| y ₃ | 0.0514 | 0.0280 | 0.1264 | 0 | | | |
| y ₄ | 0.0172 | 0.0079 | 0.0667 | 0.0389 | 0 | | |
| y ₅ | 0.0068 | 0.0611 | 0.2328 | 0.0371 | 0.0263 | 0 | |
| y ₆ | 0.3171 | 0.3725 | 0.0418 | 0.0090 | 0.1340 | 0.1517 | 0 |

5 Conclusions

Presented in this work aggregation method, using separation techniques in a dynamic mixing environment establishes a general framework for information aggregation. In practice, for simplicity, it is reasonable to adopt some standard methods of decomposition like ICA, PCA and SmCA. In this way, the results of the decomposition are considered as a starting point for the process of training a system, as it was done using MLP neural network in our case.

In particular, the article is focused on the model's results decomposition based on dynamic independent component analysis for electric load prediction. The concept was extended with the approach for components identification based on Hurst exponent. The experiment on load data resulted in 5-6% of MAPE reduction (best primary model vs. best model after decomposition) what can be perceived as substantial improvement, especially in comparison to the potential costs associated with purchasing load on the balancing market.

Acknowledgements. The work was funded by the National Science Center in Poland based on decision number DEC-2011/03/B/HS4/05092.

References

- [1] Barabasi, A.-L., Vicsek, T.: Multifractality of self-affine fractals. *Physical Review A*44, 2730–2733 (1991)
- [2] Breiman, L.: Bagging predictors. *Machine Learning* 24, 123–140 (1996)
- [3] Bussgang, J.J.: Cross-correlation function of amplitude-distorted Gaussian signals, MIT Research Laboratory of Electronics, Technical Report 216 (1952)
- [4] Cichocki, A., Amari, S.: *Adaptive Blind Signal and Image Processing*. John Wiley, Chichester (2002)
- [5] Hoeting, J., Madigan, D., Raftery, A., Volinsky, C.: Bayesian model averaging: a tutorial. *Statistical Science* 14, 382–417 (1999)
- [6] Hyvärinen, A., Karhunen, J., Oja, E.: *Independent Component Analysis*. John Wiley, Chichester (2001)
- [7] Hurst, H.E.: Long term storage capacity of reservoirs. *Trans. Am. Soc. Civil Engineers* 116, 770–799 (1951)
- [8] Peters, E.: *Fractal market analysis*. John Wiley, Chichester (1996)
- [9] Samorodnitskij, G., Taqqu, M.: Stable non-Gaussian random processes: stochastic models with infinitive variance. Chapman and Hall, New York (1994)
- [10] Szupiluk, R., Wojewnik, P., Ząbkowski, T.: Smooth Component Analysis as Ensemble Method for Prediction Improvement. In: Davies, M.E., James, C.J., Abdallah, S.A., Plumley, M.D. (eds.) *ICA 2007*. LNCS, vol. 4666, pp. 277–284. Springer, Heidelberg (2007)
- [11] Szupiluk, R., Wojewnik, P., Ząbkowski, T.: Noise detection for ensemble methods. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) *ICAISC 2010, Part I*. LNCS (LNAI), vol. 6113, pp. 471–478. Springer, Heidelberg (2010)
- [12] Weron, R., Bierbrauer, M., Trück, S.: Modeling electricity prices: jump diffusion and regime switching. *Physica A* 336, 39 (2004)
- [13] Zhang, L., Cichocki, A.: Blind Separation of Filtered Source Using State-Space Approach. In: *Advances in Neural Information Processing Systems*, vol. 11, pp. 648–654 (1999)
- [14] Zhang, L., Cichocki, A.: Blind Deconvolution of Dynamical Systems: A State Space Approach. *Journal of Signal Processing* 4(2), 111–113 (2000)

3D Molecular Modelling of the Helicase Enzyme of the Endemic, Zoonotic Greek Goat Encephalitis Virus

Dimitrios Vlachakis, Georgia Tsiliki, and Sophia Kossida

Bioinformatics & Medical Informatics Team, Biomedical Research Foundation,
Academy of Athens, Soranou Efessiou 4, Athens 11527, Greece
skossida@bioacademy.gr

Abstract. The Flaviviridae family of viruses infects vertebrates and is primarily spread through arthropod vectors. The Greek Goat Encephalitis flavivirus belongs to the Flaviviridae family and specifically to the genus Flavivirus. GGE virus, which is endemic in Greece, is the causative agent of tick-borne encephalitis, an infection of the central nervous system that can be transmitted from animals to humans by ticks. However, despite the severity of Flaviviridae causing diseases (e.g. Hepatitis C, Dengue fever, Yellow fever, Classical swine fever, Japanese encephalitis), currently there is not any available anti-flaviviridae therapy. Thus, there is a need for the development of effective anti-GGE viral pharmaceutical strategies. It has been shown that RNA helicases represent promising antiviral targets. Therefore, we suggest that establishing the 3D structure of the GGE viral helicase would be an effective approach of interrupting the life cycle of the GGE virus..

Keywords: Flaviviridae, Greek Goat Encephalitis virus, viral helicase, antiviral drug design.

1 Introduction

The Flaviviridae family of viruses infects vertebrates and it is primarily transmitted through arthropod vectors (mainly ticks and mosquitoes). Virus particles are enveloped and spherical, about 40-60 nm in diameter. The Flaviviridae family includes four genera: Genus Flavivirus (Yellow fever virus, West Nile virus, Dengue virus, Tick-borne encephalitis viruses), Genus Hepacivirus (Hepatitis C virus), Genus Pestivirus (Classical swine fever virus, Bovine viral diarrhea virus) and Genus Unclassified Flaviviridae (Hepatitis GB virus, GB viruses) Flaviviridae have monopartite, linear, single-stranded, positive sense RNA genomes, ranging from 10 to 12 kilobases (kb) in length. The 3'-termini of Flaviviridae are not polyadenylated. The 5'-termini of Flaviviruses have a methylated nucleotide cap, while other members of the Flaviviridae family are uncapped and have an internal ribosome entry site (IRES) instead [1].

The Greek Goat Encephalitis (GGE) flavivirus, which is endemic to Greece, is the causative agent of tick-borne encephalitis (TBE), a zoonotic infection of the central

nervous system that can be transmitted from animals to humans by ticks from the family Ixodidae. The GGE virus belongs to the genus Flavivirus and specifically the group of mammalian tick-borne encephalitis viruses (TBEV). The data regarding the GGE virus and its epidemiology in Greece are very limited. The first GGE virus strain was isolated in northern Greece from the brain of a newborn goat with encephalitis symptoms. Moreover, based on hemagglutination inhibition tests, it was found that 16.8% of goats in northern Greece had antibodies against the GGE virus [2].

Goat farming is a vital economic activity in Greece. In particular, Greece has the largest goatmeat production in EU followed by Spain and France. In 2011, Greece's sheepmeat and goatmeat production was 104,760 tones [3]. Therefore, the production of safe goatmeat, both for local consumption and export, is a major concern. Both, farming and agriculture have suffered seriously in the past because of livestock infected by Flaviviridae. Thus far, the European Union has dealt with diseases that affect livestock such as avian influenza, bovine brucellosis, bovine tuberculosis, swine vesicular disease, classical swine fever, enzootic bovine leucosis, ovine and caprine brucellosis. Nevertheless, according to the European Centre for Disease Prevention and Control TBE incidents are increasing in Europe.

RNA helicases are involved in duplex unwinding during viral RNA replication. It is suggested that viral helicases represent very promising antiviral targets. In particular, inactivation of the Gengue and Bovine diarrhea viral helicase led to reduced viral replication [4]. Therefore, we suggest that inhibition of the GGE viral helicase, which is encoded by the viral NS3B gene, would be an effective approach for the reduction of the replication rate of the GGE virus. Today there is not any available anti-flaviviral therapy. Thus, there is a need for the development of an efficient anti-GGE viral pharmaceutical strategy. Our proposed research will be directed towards the design and development of a series of drug-like low molecular weight compounds capable of inhibiting the helicase enzyme of GGE virus.

Computer-aided homology modeling techniques will be employed to predict the three-dimensional structure of the GGE viral helicase, since its 3D structure has not been resolved. In silico simulations of the predicted helicase structure and Flaviviridae helicase substrates will be also conducted in order to identify specific helicase enzyme-ssRNA interaction patterns, as well as key enzyme residues involved in dsRNA unwinding..

2 Methods

Homology Modelling and Model Evaluation

The homology modelling of the five algae enzymes was carried out using the Molecular Operating Environment suite (MOE, 2004.03) package and its built-in homology modelling application [5]. The overall homology modeling process was divided into the following steps: First, the initial spatial constraints for the target sequence were derived from a large number of template protein structures; the target sequence was aligned to the backbone of a template structure copying the geometric coordinates of the template to the target sequence. Second, target regions, where

geometric constraints could not be copied from the template easily, were modeled. These regions represented either deletions or insertions with respect to the template. The third step involved loop selection and side chain packing, where a collection of independent models was obtained. Fourth, the final models were scored and ranked, after they had been stereochemically tested and evaluated with a built-in module for protein geometry error-checking.

The produced models were initially evaluated within the MOE package by a residue packing quality function, which depends on the number of buried non-polar side chain groups and on hydrogen bonding. Furthermore the suite PROCHECK was employed to further evaluate the quality of each one of the five algae enzyme models.

Model Optimisation

Energy minimisation was done in MOE initially using the Amber99 forcefield implemented into the same package and up to a RMSd gradient of 0.0001 to remove the geometrical strain. The model was subsequently solvated with SPC water using the truncated octahedron box extending to 7 Å from the model and molecular dynamics were performed after that at 300K, 1 atm with 2 fsecond step size, using the NVT ensemble in a canonical environment. NVT stands for Number of atoms, Volume and Temperature that remain constant throughout the calculation. The results of the molecular dynamics simulation were collected into a database by MOE and can be further analysed.

3 Results and Discussion

Flaviviridae protease and the helicase genes are located in the NS3 region. Herein, the helicase enzyme of the Greek Goat Encephalitis (GGE) was modeled (GenBank: DQ235153.1), using the 3D structure of the Dengue virus helicase (PDB entry: 2VBC) as template structure (Figure 1). The alignment produced sequence identity and similarity of 47 and 62 % respectively. Notably, all characteristic helicase motifs, unique to the flaviviridae viral family have been properly aligned between the model and its template.

Secondary structure prediction for the model was quite accurate and led to a very satisfactory result, which was expected due to the high sequence similarity of the two sequences. The sequence alignment between the GGE and the Dengue viruses is very reliable as the conserved domains have been properly aligned. For purposes of statistical fidelity, since none of the existing methods of predicting secondary elements of a protein can be considered thoroughly accurate absolutely correct, herein, three different prediction programs were used. Namely Predict Protein, PSI-Pred and J-Pred. As expected, the results of the three programs were almost identical with some minor variance and local differences in their predictions. There is only one gap that has been introduced by the automated alignment, provided that the gaps at the end of the sequence can be ignored. The area of the gap has not been predicted to be structured and does not belong to any of the conserved helicase motifs. Therefore, this gap can be ignored, as it will not disrupt or affect the quality of the produced model.

| | | |
|------------|---|-----|
| GrGen 1 | MWVHVTGAAALSIDDAVAGPYWADVREDVVCYGGAWSLEEKW-KGEAVQIHAFFPPGRAHEV | 59 |
| Dengue 49 | MWVHVTGSVICHESGRLEPSTADVRNDMISYGGWRLGDKWIKDEEDQQLATEPGKKNPKH | 108 |
| GrGen 60 | HQCOPGELIILDTGKRIGAVPIDLAKGTSGSPILNAHGVVVGLYGNGLKTNE-TYVSSIAQ | 118 |
| Dengue 109 | Q+PG TG+ +GAV +D GTSGSPi+N G V+GLYGN+ T YVS+I Q | 167 |
| GrGen 119 | GEVEKSRPNLPOAVVGTGWMSKGQITVLDMHFCGKTHRVLPPELICDRLRLTIVIAP | 178 |
| Dengue 168 | E R P V K ++T++D+HPC+GKT R+IP ++R+ + RRLRLT+IAP | 223 |
| GrGen 179 | TRVVILKEMERAINGKRVRFHSPAVSDQQVGGAIWVDMCHATVNNRRLLPOGRONWEAIM | 238 |
| Dengue 224 | TRVV EME AL G +R+ +PAV G IVD+MCCHAT+ R I N+ + M | 283 |
| GrGen 239 | DEAHWTDPHSIAARGHLYTLAKENKCALVLMATTATPPGKSEPFPESENAGITSEERQIPEGE | 298 |
| Dengue 284 | DEAH+TDP S+AARG++ T + + A + M+ATPPG ++PFP+SN I ER+IPE | 343 |
| GrGen 299 | WRQGFDWITEYDGRTAWFVPSIAKGGVIARALRQKGKSVICLNSTKTFEKDYSRKVKDEKPD | 358 |
| Dengue 344 | V GFDWITD+Y G+T WFPSI G IA IR+ GK VI L+ KTF+ +Y + K D | 403 |
| GrGen 359 | FVVTIDISEMGANLDVSRVIDGRTNIKPPEEV-DG—KVELTGVRVITASAORRGRVGR | 415 |
| Dengue 404 | FVVTIDISEMGAN RVID R +KP + DG +V L G VT ASAORRGR+GR | 463 |
| GrGen 416 | OQGRRT-DEYIYSGQCDDDDSGLVQWKAEQIILIDNITLRLGPWATFYGPQQKMPPEVAGHF | 474 |
| Dengue 464 | + D+Y++SG +D V EA++LIDNI T G + T +GPE++K + G F | 523 |
| GrGen 475 | RITKEKRKBFRHLLTHCDFTPWLAWHVAANVSSVTDRSVTWCPEANAVDEANGELVIFR | 534 |
| Dengue 524 | RL E+RK F L+ D WL++ VA+ S DR V + G N + E N E V | 582 |
| GrGen 535 | SPNGAERTIRPVWRDARMREGRDIK 560 | |
| Dengue 583 | + G ++ IRP V DAR++ + +K TREGEKKKLRPKWILDARVYADPMALK 608 | |

Fig. 1. The sequence alignment between the Greek Goat Encephalitis viral helicase and the X-ray structure of the Dengue helicase template. All conserved motifs have been highlighted in yellow color.

The designed three dimensional model of the Greek Goat Encephalitis helicase, has acquired its template conformational structure. More importantly though, the 3D modeled GGE helicase has structurally conserved all key helicase motifs within a Ca RMSd less than 0.6 Å (Figure 2). Confirming and assessing the quality and reliability of the homology modelling experiment necessitated partial modification of some libraries of the MOE program. Without these changes, it would not be possible to identify the ribonucleotides of RNA oligomer and the Mn+2 ions which exist in the structure of the HCV helicase (PDB entry: 1A1V). Usage of this structure was considered important, since only the HCV helicase has been co-crystallized with an

oligonucleotide and Mn⁺² atoms in the ATP site. The interactions established between the GGE model and the oligonucleotide and Mn⁺² atoms, were identical to the interactions established between the enzyme and its substrates in the HCV crystal structure (data not shown).

More specifically, after the first base was defined as uracil ribonucleotide new files for the topology and configuration of residues these substrates were created. In addition the Mn⁺² ions had to be defined in their personal libraries too. The execution of the homology modelling experiment produced a set of a total of 200 candidates initial models. After screening and filtering with PROCHECK v. 3.5 only one (the top ranking) model was selected for further studies. The final score was obtained using an equation that is affected by both static and dynamic parameters, with the former being more related to statistical probabilities of spatial constraints of the model and the latter with the variable value of the total energy, as calculated by the AMBER forcefield. At each step the assessment criterion for the generated model was the achieved PROCHECK score.

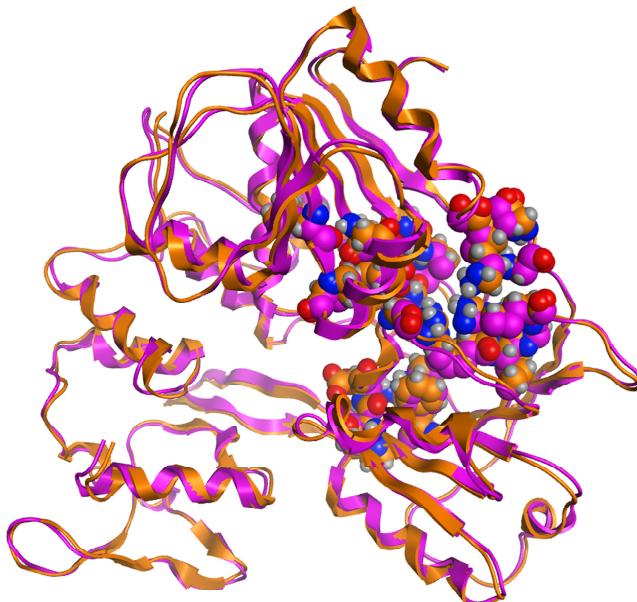


Fig. 2. The 3D model of the GGE helicase in orange color, superposed on its Dengue helicase template, in pink color. The conserved structural motifs of helicases of the Flaviviridae viral family have been completely conserved on the GGE model and are showing in spacefill atom representation.

Structure optimization was achieved by energy minimization using MOE. This way the overall quality of the model was improved by applying conjugate gradient methods and molecular dynamics. The first conjugate gradient method was used at the beginning (first stage) of a repetitive process and at the end (the third stage), in which

the minimum individual shift (gradient) applied is of the order of 0,010. During the main part (second stage) of optimization, a set of molecular dynamic cycles were conducted at the constant temperature of 300° K, using a time step in the order of 2 fempto seconds. This method is basically recurring solutions of Newtonian equations of motion. Figure 3 is the PROCHECK resulting Ramachandran plot of the final, optimized GGE model.

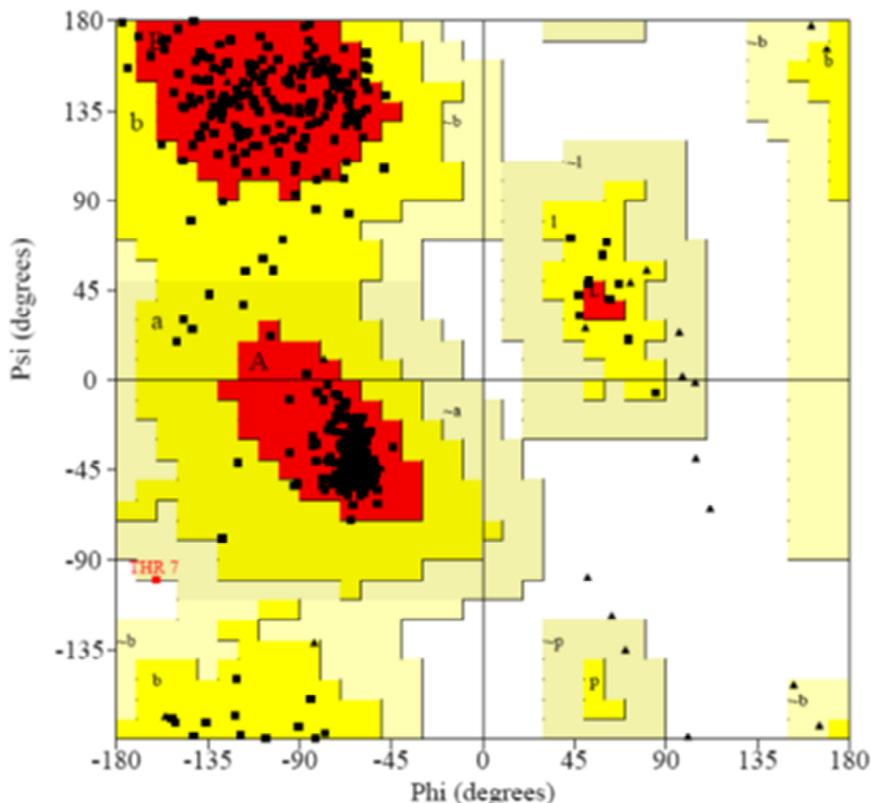


Fig. 3. The Ramachandran chart of the final model of viral GGE helicase

Overall, the 3D model of the GGE virus helicase was designed using the homologous X-ray crystal structure of the Dengue helicase as template. The model was successfully evaluated both in terms of its geometry, fold recognition and compliance to the criteria required as a member to the Flaviviridae viral family. It is therefore proposed that the Classical Swine Fever virus helicase model will be suitable for further *in silico* structure-based *de novo* drug design experiments. These computer-based methodologies are now becoming integral part of the drug discovery process that may eventually lead to the development of potential inhibitor structures against the GGE viral helicase in the future..

References

1. Verh, K.: Acad Geneesk Belg. Infections with Flaviviridae 61(6): 661-697; discussion 697-9 (1999)
2. Guzmán, M.G., Kourí, G.: Dengue diagnosis, advances and challenges. International Journal of Infectious Diseases 8(2), 69–80 (2004)
3. Shepard, C.W., Finelli, L., Alter, M.J.: Global epidemiology of hepatitis C virus infection. The Lancet Infectious Diseases 5(9), 558–567 (2005)
4. Thompson, B., Finch, R.: Hepatitis C Virus Infection. Clinical Microbiology and Infection 11(2), 86–94 (2005)
5. DeLano, W.L.: The PyMOL Molecular Graphics System. DeLano Scientific, San Carlos (2002), <http://www.pymol.org>
6. MOE CCG, 1010 Sherbrooke St. West, Suite 910, Montreal, Canada, H3A 2R

Feature Comparison and Feature Fusion for Traditional Dances Recognition

Ioannis Kapsouras, Stylianos Karanikolos, Nikolaos Nikolaidis,
and Anastasios Tefas

Department of Informatics,
Aristotle University of Thessaloniki,
54124 Thessaloniki, Greece

{jkapsouras,nikolaid,tefas}@aiai.csd.auth.gr

Abstract. Traditional dances constitute a significant part of the cultural heritage around the world. The great variety of traditional dances along with the complexity of some dances increases the difficulty of identifying such dances, thus making the traditional dance recognition a challenging subset within the general field of activity recognition. In this paper, three types of features are extracted to represent traditional dance video sequences and a bag of words approach is used to perform activity recognition in a dataset that consist of Greek traditional dances. Each type of features is compared in a stand alone manner in terms of recognition accuracy whereas a fusion approach is also investigated. Features extracted through the training of a neural network as well as fusion of all three types of features achieved the highest classification rate.

Keywords: Dance recognition, Dense Trajectories, Spatio-temporal interest points, Subspace Analysis, Neural networks.

1 Introduction

Activity recognition is an active research topic that deals with the identification of activities performed by a human subject as captured in video. Activity recognition deals mainly with the recognition of everyday actions such as walking, running, sitting etc. and is important for various application such as video surveillance and video annotation. Moreover, activity recognition can be used for creating intelligent environments, for computer-human interaction etc. A significant amount of research has been performed on activity recognition. Surveys of activity recognition approaches can be found in [1], [2], [3]. Many methods of activity recognition use global representations whilst others extract features from local areas. Classification can be performed by many ways, such as nearest neighbour, SVM, HMM, dynamic time wrapping etc. Methods for activity recognition can also work on multi view sequences.

Dancing is a very wide motion class that includes many different styles (e.g. tango, breakdance, waltz, traditional dances etc) and has many particularities. Thus recognition of dances can be considered as a different research field.

Although video based activity recognition is a very active research field, research on dance recognition both on video and motion capture data is very limited. Samanta et al. in [4] propose a method for classifying Indian Classic Dances. The authors propose a pose descriptor to represent each frame of a sequence. The descriptor is based on the histogram of oriented optical flow, in a hierarchical manner. The pose basis is learned using an on-line dictionary learning technique and each video is represented sparsely as a dance descriptor by pooling pose descriptors of all the frames. Finally, dance videos are classified using support vector machine (SVM) with intersection kernel. In [5] Raptis et al. introduce a method for real-time classification of dance gestures from skeletal animation. An angular skeleton representation that maps the motion data to a smaller set of features is used. The full torso is fitted with a single reference frame. This frame is used to parametrize the orientation estimates of both the first-degree limb joints (joints adjacent to torso) and second-degree limb joints (tips of the wireframe extremities such as the hands and the feet). Then a cascaded correlation-based maximum-likelihood multivariate classifier is used to build a statistical model for each gesture class. The trained classifier compares the input data with the gesture model of each class and outputs a maximum likelihood score. An input gesture is finally compared with a prototype one using a distance metric that involves dynamic time-warping. Deng et al. in [6] proposed a method that performs recognition of dance movements on skeletal animation data. The authors proposed a new scheme for motion representation, the segmental SVD. A motion pattern is represented in a hierarchical structure with multiple levels and SVDs generated on the corresponding levels are used to extract features across time. The authors also proposed a similarity measure to compare segmental SVDs representations. Two methods for dance pose recognition (which is a task related to dance recognition) are presented in [7], [8].

In this paper we use a bag of words approach to perform traditional dance recognition on video data. Features extracted from the training data are clustered using K-means to find discriminative representations of the features. Then each feature vector is mapped to the closest cluster center and a histogram over the cluster centers is created. An SVM classifier with χ^2 kernel is trained to classify an unknown sequence of a traditional dance. Three state of the art methods used in general activity recognition research were used for feature extraction. The first one, proposed by Le et. al [9] extends the Independent Subspace Analysis algorithm to learn spatio-temporal features from video data by training a neural network. The second, proposed by Laptev et. al [10] detects spatio-temporal interest points using an extension of the Harris detector (Harris3D). The third, proposed by Wang et al [11] represents a video sequence based on dense trajectories and motion boundary descriptors. The performance of each individual type of features in the recognition of 5 Greek traditional dances is experimentally evaluated. Furthermore, two fusion approaches are investigated and compared.

2 Problem Statement

Parts of history of the world and various traditional customs are reflected in traditional music and dances. There is a great variation of traditional dances and the preservation and dissemination of such dances to the younger generations is a very important issue for a specific country or region.

In Greece, there are many traditional dances due to its rich history and cultural diversity. There is a great variation between dances even within a specific region. There are fast and slow dances, dances performed only by women and dances that change tempo from slow to fast. The recordings of such dances are usually of low quality and with no annotation. Some dances are very rare and known only to some senior citizens. An annotated traditional dances database will be of great importance for educational, research and cultural heritage preservation purposes. Such a database will help the youngsters to stay in touch with their cultural heritage and increase their awareness for it.

Traditional dance recognition can be considered more challenging than generic activity recognition. A system that can recognize traditional dances needs a robust feature selection procedure and a reliable classifier, both of which are also parts of a general activity recognition algorithm. However traditional dance recognition has important particularities and difficulties. At first, there are traditional dances that have similar tempo and steps making the recognition between them more difficult. Moreover, the rhythm of some songs changes from slow to fast within the song thus affecting the tempo of the dance. These changes increase the inter-class variation and thus the difficulty of recognition. Furthermore, most of the Greek traditional dances are group, circular dances and it is highly unlikely that methods designed for activity recognition in one subject will achieve high recognition rates when dealing with many subjects (dancers). Finally, professional dancing groups often perform the same dance by traditional costumes that differ as shown in Fig. 1 making activity recognition methods that rely on appearance less effective.



Fig. 1. Stankena Greek folk dance performed by professional dancing groups with different costumes

3 Method Description

The aim of this paper is to test if a well-known framework applied in general activity recognition can be used with good results to recognize Greek traditional

dances. To test this framework three different state of the art feature extraction approaches are presented and compared. Moreover, the fusion of these features in two different ways has been considered.

The bag of words recognition framework [12] is summarized as follows. Let a number of feature vectors represent the video data. In order to recognize a number of dance classes, feature vectors of training data are clustered using the K-means algorithm. The centroids $\mathbf{v}_c, c = 1, \dots, C$ where C is the number of clusters of the K-means, form a discriminative representation of the feature vectors. Then the feature vectors of all the training data are mapped to the closest centroid using *Euclidean* distance. Next for each training sequence the frequency of appearance of every centroid is computed and thus, a histogram for each sequence, that characterizes it, is formed.

Feature vectors are also extracted for a testing sequence and the same procedure is used. Thus, the feature vectors are mapped to the closest centroid and the histogram that characterizes the testing sequence is formed. At last, the testing sequence is recognized using an SVM classifier trained by the histograms of the training sequences. We used a non-linear SVM with χ^2 kernel [10]:

$$K(\mathbf{s}_j, \mathbf{s}_k) = \exp\left(-\frac{1}{2A} \sum_{i=1}^C \frac{(s_{j,i} - s_{k,i})^2}{s_{j,i} + s_{k,i}}\right) \quad (1)$$

where A is the mean value of distances between all training samples, C is the number of centroids and $s_{j,i}$ and $s_{k,i}$ are the values of the i-th bin for the histograms \mathbf{s}_j and \mathbf{s}_k . The *one-against-rest* approach was used for the SVM. As already mentioned, we used this framework to test three types of features proposed in [9], [10] and [11] in traditional dances recognition. Fusion of these features is also considered. The three features are described below.

Le et al. use unsupervised feature learning by training a neural network as a way to extract features from video data. The authors extend the algorithm of Independent Subspace Analysis (ISA). An ISA network can be described as a two-layered neural network with square and square-root nonlinearities in the first and second layer respectively. In more detail, the activation of each second layer unit for an input pattern \mathbf{x}^t is given by:

$$p_i(\mathbf{x}^t; \mathbf{W}, \mathbf{V}) = \sqrt{\sum_{k=1}^m V_{ik} \left(\sum_{j=1}^m W_{kj} x_j^t \right)^2} \quad (2)$$

Parameters \mathbf{W} are learned through sparse representation in the second layer by solving:

$$\underset{\mathbf{W}}{\text{maximize}} \sum_{t=1}^T \sum_{i=1}^m p_i(\mathbf{x}^t; \mathbf{W}, \mathbf{V}) \text{ subject to } \mathbf{W}\mathbf{W}^T = \mathbf{I} \quad (3)$$

where $\mathbf{W} \in \Re^{k \times n}$ is the matrix that contains the weights connecting the input data to first layer units and $\mathbf{V} \in \Re^{m \times k}$ is the matrix that contains the weights connecting the units of the first layer to second layer units.

The authors use 3D video blocks (patches) as input to the first layer of the neural network. In order to reduce the computational cost of the algorithm, they use small patches and convolve the trained network by overlapping the first layer trained features to compute the input of the second layer of the network. PCA is used as a preprocessing step to reduce the dimension of the input data. Finally, they combine features from both layers and use them for classification. Their network is trained using a batch projected gradient descent. In what follows, the features generated by this approach will be denoted as ISA features.

Laptev et al. in [13] proposed a method for the determination of Space-Time Interest Points (STIPS) from each action video and their description by a set of histograms of oriented gradient (HOG) and histograms of optic flow (HOF) descriptors, which refer to local shape and motion. The authors employ the Harris3D detector, which was proposed by Laptev and Lindeberg in [10], in order to detect video locations where the image intensity values undergo significant spatio-temporal changes. Harris3D extends the Harris interest point detector and the basic idea is to extend the notion of interest points in the spatiotemporal domain, by requiring the image values in local spatio-temporal volumes to have large variations along both spatial and temporal directions. Points with such properties are named STIPS and they correspond to local spatio-temporal neighbourhoods with non-constant motion. The authors construct the linear scale-space representation of a spatio-temporal image sequence f , by convolution of f with an anisotropic Gaussian kernel, with independent spatial and temporal scale values σ_l^2 and τ_l^2 .

$$L = (\cdot; \sigma_l^2, \tau_l^2) = g(\cdot; \sigma_l^2, \tau_l^2) * f(\cdot) \quad (4)$$

Then, they consider a spatio-temporal second-moment matrix \mathbf{M} at each video point, which is a 3×3 matrix composed of first order spatial and temporal derivatives averaged with a Gaussian function $g(\cdot; \sigma_i^2, \tau_i^2)$, with $\sigma_i^2 = s\sigma_l^2$ and $\tau_i^2 = s\tau_l^2$. The final locations of space-time interest points are given by local maxima of $H = \det(\mathbf{M}) - k\text{trace}^3(\mathbf{M})$, $H > 0$. The HOG/HOF descriptors are used to compute histograms of oriented gradient and optical flow, accumulated in space-time volumes in the neighbourhood of detected interest points. The size of each volume is related to the detection scales by Δ_x , $\Delta_y = 2k\sigma$ and $\Delta_t = 2k\tau$. Each volume is subdivided into a $n_x \times n_y \times n_t$ grid of cuboids and for each cuboid, histograms of gradient orientations and histogram of optical flow are computed. Finally, normalized histograms are concatenated into HoG, HoF as well as HoG/HoF descriptor vectors. Sample STIPs detected in a folk dance video are shown in Fig. 2.

Wang et al. at [11] propose a method for activity recognition based on trajectories extracted by dense sampling. At first, dense sampling is performed on a grid spaced by W pixels. Sampling is performed in a number of spatial scales in order to track the sampled points through the video. In order to avoid samples in homogeneous image areas they use the criterion presented in [14] to remove points from these areas.

Feature points are tracked on each spatial scale separately by computing the optical flow field $\omega_t = (u_t, v_t)$ for each frame I_t , where u_t and v_t are the

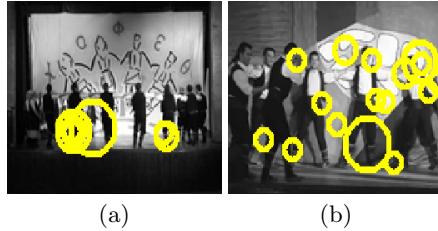


Fig. 2. Spatiotemporal interest points detected in folk dance videos: a) Stankena dance performed indoors, b) Zablitsena dance performed indoors

horizontal and vertical components of the optical flow. Given a point $\mathbf{P}_t = (x_t, y_t)$ in frame \mathbf{I}_t , its tracked position in frame \mathbf{I}_{t+1} is smoothed by applying a median filter on ω_t . Points of subsequent frames are concatenated to form trajectories: $(\mathbf{P}_t, \mathbf{P}_{t+1}, \mathbf{P}_{t+2}, \dots)$. The authors limit the length of trajectories to L frames. The shape of a trajectory is described by a sequence $(\Delta\mathbf{P}_t, \dots, \Delta\mathbf{P}_{t+L-1})$ of displacement vectors, where $\Delta\mathbf{P}_t = (\mathbf{P}_{t+1} - \mathbf{P}_t)$. The resulting vector is normalized by the sum of displacement vector magnitudes. Trajectories extracted for a random frame from Stankena dance are shown in Fig. 3.

The authors also use a space-time volume aligned with a trajectory to encode motion information. The size of the volume is $N \times N$ pixels and L frames long and is subdivided into a spatio-temporal grid. In each cell of the spatio-temporal grid of the volume various descriptors are computed. These include HOG and HOF descriptors and motion boundary histograms (MBH) descriptors [15] in order to deal with camera motion. The final descriptor (that will be denoted as TRAJ) is computed via the concatenation of these descriptors.

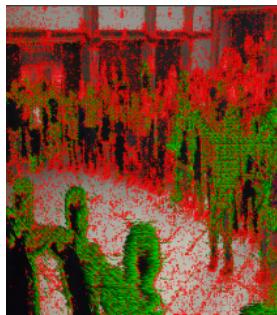


Fig. 3. Visualization of dense trajectories detected in a Stankena dance video

The above features were used in a bag of words manner, as described in Section 3, to train an SVM classifier with χ^2 kernel. Fusion of the above features was also considered. Fusion was performed either by adding the kernels of the SVM produced by the histograms of each type of features or by concatenating the histograms produced by different types of features.

4 Experimental Results and Discussion

The three methods for feature extraction described in the previous Section were used in order to verify if these features can successfully be used for the task of traditional dances recognition either individually or within a fusion framework. The features were tested within the framework presented in the Section 3 on a dataset of videos of Greek traditional dances. The dataset contains 10 videos of 5 Greek traditional dances, namely *Lotzia*, *Capetan Loukas*, *Ramna*, *Stankena* and *Zablitsena*. More precisely, 1 video from each dance performed by professional dancing groups indoors was used for training and the other one was used for testing. The training videos of the 5 dances were temporally segmented in a manual way into overlapping clips of duration 80 to 100 frames each resulting to 78, 113, 110, 95 and 101 clips respectively. The same procedure was used for the testing videos resulting to 102, 107, 110, 106 and 91 clips respectively. Thus the training and test set consisted of 496 and 516 short sequences respectively. The overall correct classification rate for the three types of features (STIP, ISA and TRAJ) and for various numbers of clusters of the K-means algorithm are shown in Table 1.

Table 1. Performance of STIP, TRAJ and ISA features on the folk dances dataset

| Number of clusters | STIP | TRAJ | ISA |
|--------------------|---------------|---------------|---------------|
| 10 | 49.61% | 32% | 54.84% |
| 100 | 37.60% | 35.85% | 78.68% |
| 1000 | 38.18% | 44.60% | 72.86% |

As can be seen in this table the features learned via deep learning techniques (ISA) clearly outperform the other two features. Considering the difficulties of traditional dance recognition the classification rate achieved with the use of ISA features (78.68%) is very satisfactory.

The histograms produced by the three features were fused to check if the fusion brings performance gains. As already mentioned, the histograms were fused in two ways: either by addition of the χ^2 kernels computed for the initial histograms or by the concatenation of the initial histograms before the kernel computation. The highest correct classification rates for all combinations of the three features are shown in Table 2.

As can be seen in Table 2 the fusion by adding the SVM kernels achieved better classification rate than the fusion by concatenation. The overall best classification rate (79.96%) is achieved by the combination of all three features through kernel addition. However, this classification rate is only slightly better than that achieved by using only ISA features (78.68%). It can also be seen that the classification rate is high whenever ISA features are involved in the fusion. These observations indicate that ISA features are the most suitable ones for the recognition of traditional Greek dances and that their individual classification rate is already high enough to be significantly improved by fusing them with the other

Table 2. Best classification rates for all features combinations

| Fused Features | Classification Rate | Fusion Method |
|----------------|---------------------|----------------|
| STIP/TRAJ | 52.71% | adding kernels |
| ISA/STIP | 78.69% | |
| ISA/TRAJ | 78.68% | |
| ISA/STIP/TRAJ | 79.26% | |
| STIP/TRAJ | 52.51% | concatenation |
| ISA/STIP | 78.69% | |
| ISA/TRAJ | 72.29% | |
| ISA/STIP/TRAJ | 73.04% | |

types of features. When features with lower performance are fused (STIP and TRAJ features) the performance increases by a larger margin (52.71% compared to 49.61% and 44.6% for STIP and TRAJ features respectively). However the performance remains low. This fact can also be seen in Table 3, where STIP and TRAJ feature histograms obtained with the best parameters are fused with ISA feature histograms with low classification rate. The fused classification rate increases to 61.24%. Thus, it is fair enough to assume that the fusion of features is desirable only when the initial features have similar performance.

Table 3. Comparison of the fused classification rate with the initial features when their classification rate is low

| | ISA | STIP | TRAJ | Fused (kernel addition) |
|-----------------------------|-----------|-----------|-------------|-------------------------|
| Class. Rate/Nr. of clusters | 54.84%/10 | 49.61%/10 | 44.60%/1000 | 61.24% |

The confusion matrix of the best classification rate can be seen in Table 4.

Table 4. Confusion matrix of fused features via adding kernels (ISA, STIP, TRAJ)

| | Lotzia | Capetan Loukas | Ramna | Stankena | Zablitsena |
|-----------------------|--------------|----------------|--------------|--------------|--------------|
| Lotzia | 95.10 | 4.90 | 0 | 0 | 0 |
| Capetan Loukas | 7.48 | 91.59 | 0 | 0.93 | 0 |
| Ramna | 10 | 19.10 | 70.91 | 0 | 0 |
| Stankena | 0 | 3.77 | 0 | 57.55 | 38.68 |
| Zablitsena | 2.20 | 15.38 | 0 | 0 | 82.42 |

As can be seen in this table *Capetan Loukas* and *Lotzia* are recognized with high recognition rates while *Stankena* achieves the worst recognition rate.

It should be noted that, STIP and TRAJ features have been tested on various databases in generic activity recognition with very good results [13], [11]. The low classification rates of these features in the presented experimental setup proves that traditional dance recognition bears significant difficulties making recognition, even between relatively few classes (5), difficult.

5 Conclusions

In this paper we deal with the recognition of Greek traditional dances. Three state of the art methods for feature extraction are used, fused and compared within a bag of words approach. The method is applied on five traditional dances from the Western Macedonia region. The results on these challenging videos are promising but also prove that traditional dance recognition is a very difficult task. STIP and TRAJ features fail to achieve high classification rates. On the other hand the high classification rate of ISA features indicates that features learned through neural networks can be successfully used to recognize videos from Greek traditional dances. Feature fusion provided no significant improvement to the already high performance of ISA features. In the future, we plan to test the presented approach in more rich datasets.

Acknowledgements. This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operation Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALIS-UOA-ERASITECHNIS MIS 375435.

References

- Poppe, R.: A survey on vision-based human action recognition. *Image Vision Comput.* 28, 976–990 (2010)
- Turaga, P., Chellappa, R., Subrahmanian, V.S., Udrea, O.: Machine Recognition of Human Activities: A Survey. *IEEE T. Circ. Syst. Vid.* 18, 1473–1488 (2008)
- Xiaofei, J., Honghai, L.: Advances in View-Invariant Human Motion Analysis: A Review. *IEEE T. Syst. Man. Cy. C* 40, 13–24 (2010)
- Samanta, S., Purkait, P., Chanda, B.: Indian Classical Dance classification by learning dance pose bases. In: 2012 IEEE Workshop on the Applications of Computer Vision, pp. 265–270. IEEE Press, Washington, DC (2012)
- Raptis, M., Kirovski, D., Hoppe, H.: Real-time classification of dance gestures from skeleton animation. In: 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 147–156. ACM, New York (2011)
- Deng, L., Leung, H., Gu, N., Yang, Y.: Recognizing Dance Motions with Segmental SVD. In: 20th International Conference on Pattern Recognition (ICPR), pp. 1537–1540. IEEE Press, Istanbul (2010)
- Bo, P., Gang, Q.: Binocular dance pose recognition and body orientation estimation via multilinear analysis. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–8. IEEE Press, Anchorage (2008)
- Feng, G., Gang, Q.: Dance posture recognition using wide-baseline orthogonal stereo cameras. In: 7th International Conference on Automatic Face and Gesture Recognition, pp. 481–486. IEEE Press, Southampton (2006)
- Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3361–3368. IEEE Press, Colorado (2011)

10. Laptev, I., Lindeberg, T.: Space-Time Interest Points. In: International Conference on Computer Vision (ICCV), Nice, pp. 432–439 (2003)
11. Wang, H., Kläser, A., Schmid, C., Liu, C.: Dense trajectories and motion boundary descriptors for action recognition. *Int. J. Comput. Vision* 103, 60–79 (2013)
12. Wang, H., Ullah, M.M., Kläser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: British Machine Vision Conference, London, p. 127 (2009)
13. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8. IEEE Press, Anchorage (2008)
14. Shi, J., Tomasi, C.: Good Features to Track. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 1994, pp. 593–600. IEEE Press, Seattle (1994)
15. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part II. LNCS, vol. 3952, pp. 428–441. Springer, Heidelberg (2006)

Intelligent Chair Sensor

Classification of Sitting Posture

Leonardo Martins¹, Rui Lucena¹, João Belo¹, Marcelo Santos¹, Cláudia Quaresma^{2,3},
Adelaide P. Jesus¹, and Pedro Vieira¹

¹ Departamento de Física, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

² CEFITEC, Departamento de Física, Faculdade de Ciências e Tecnologia,

Universidade Nova de Lisboa, Quinta da Torre P-2829-516, Caparica, Portugal

³ Departamento de Saúde, Instituto Politécnico de Beja, P-7800-111, Beja, Portugal

{l.martins,mg.santos}@campus.fct.unl.pt,
{rui.lucena,joao.belo}@ngns-is.com,
{q.claudia,ajesus,pmv}@fct.unl.pt

Abstract. In order to build an intelligent chair capable of posture detection and correction we developed a prototype that gathers the pressure map of the chair's seat pad and backrest and classifies the user posture and changes its conformation. We gathered the pressure maps for eleven standardized postures in order to perform the automatic posture classification, using neural networks. First we tried to find the best parameters for the neural network classification of our data, obtaining an overall classification of around 80% for eleven postures. Those neural networks were exported to a mobile application in order to do real-time classification of those postures. Results showed a real-time classification of around 70% for eleven standardized postures, but we improved the overall classification score to 93.4% when we reduced the posture identification to eight postures, even when this classification was done with unfamiliar users to the posture identification system.

Keywords: Sensing chair, Pressure-distribution sensors, Sitting posture, Posture Classification, Posture correction, Neural Networks.

1 Introduction

Changes in transportation, communications, workplace and entertainment in the last century led to a sedentary lifestyle, forcing the population to spend long periods of time in a sitting position [1, 2]. While seated, most of the bodyweight is transferred to the ischial tuberosities and to the thigh and the gluteal muscles. The rest of the weight is distributed to the ground through the feet and to the backrest and armrest when they are available [3]. Adopting a lumbar flexion position for long periods of time, leads to a decrease of the lumbar lordosis [4], which has been linked to back and neck pain due to the anatomical changes of the spine and the degeneration of the intervertebral discs and joints. Adopting a bad posture while seated can worsen these health problems [5].

The long term goal of this project is to build an intelligent chair that can detect the sitting posture and effectively correct an incorrect posture adoption in order to minimize the health issues that were previously described. In order to correct a bad posture we developed a first prototype with 8 pressure cells (4 in the seat pad and 4 in the backrest) which are able to change their conformation by inflation or deflation and can increase the user discomfort when a bad posture is adopted, encouraging the user to change to a correct position. We can also produce slight changes in the chair conformation over a period of time, which can help to evenly distribute the applied pressure on contact zones, reducing the fatigue and discomfort of the user due to the pressure relief on compressed tissues.

In order to do posture classification, we gather the pressure inside each bladder, which is then used as an input for the classification of eleven different postures using neural networks. Neural Networks were chosen as the classification method, since after creating and training the Neural Network we can easily export the weights and bias and apply to other applications. In our case we exported them to a mobile and portable application in order to build a system capable of real-time classification and correction of the user posture.

2 Related Work

The adoption of an incorrect posture in a sitting position over long periods of time can lead to neck and back pains [4, 5], which have a huge impact in the cost of work-related illness. Estimates show that, only in the USA, 50\$ billion dollars are spent every year for the treatment of back pain [6].

There are a wide number of clinical views of ‘correct’ or ‘incorrect’ postures, but until recent years there were little quantitative studies to define those postures. Recent studies have been trying to determine whether the so called ‘good’ postures actually provide a clinical advantage [7].

To solve the problem of incorrect posture adoption for long periods of time in a sitting position, several investigation groups have been working with pressure sensors placed in chairs. These pressure sensors were able to detect the user posture, using the acquired pressure maps and various Classification Algorithms.

Various studies equipped with the same sensor sheets (one for the seat pad and one for the backrest) were able to distinguish various postures [8, 9, 10]. Slivovsky et al. (2000) and Tan et al. (2001) used Principal Component Analyses (PCA) for posture detection for human-machine interactions obtaining an overall classification accuracy of 96% and 79% for familiar and unfamiliar users, respectively [8, 9]. Zhu et al. (2003) used the same data acquisition methods from the previous two studies to investigate which classification algorithms would be the best for static posture classification. The authors found that among k-Nearest Neighbor, PCA, Linear Discriminant Analysis and Sliced Inverse Regression (SIR), both PCA and SIR outperformed the other methods [10].

Mutlu et al (2007) and Zheng and Morrell (2010) reduced drastically the number of pressure sensors for posture identification. The first group determined the near

optimal placement of 19 FSR (Force Sensitive Resistors) sensors obtaining an overall classification accuracy of 78%, improving the classification to 87% when the number of sensors was increased to 31 [11]. The second group adapted a chair with just 7 FSR and 6 vibrotactile actuators, in order to direct the subject towards or away from a certain position through haptic feedback. They obtained an overall classification of 86.4% on the same 10 postures using the mean squared error between the pressure measurements and their reference for each posture, showing also the effectiveness of haptic feedback on posture guidance [12]. A smart chair equipped with 6 sensors (4 in the seat and 2 in the backrest) was used to study how feedback can influence the sitting behavior of office workers. They showed that there was an average increase in basic posture in groups that received feedback [13].

3 Materials and Methods

3.1 Equipment

We built this prototype with the aim of producing an office chair capable of detecting the user posture and also correct bad posture adoption over long periods of time.

Considering a low cost and commercially available solution we produced a low resolution matrix of pressure sensors, which are able to change their conformation by inflation and deflation. Strategically sensor placement was required in order to achieve good performance results. Previous literature identified two types of strategies: a pure mathematical and statistical approach [11] and an anatomical approach [12]. Based on the second method we placed the pressure sensors in order to cover the most important and distinguishable areas of the body for detecting a seated posture, such as the ischial tuberosities, the thigh region, the lumbar region of the spine and the scapula. These are also the areas where most of the bodyweight is distributed [3].

The distribution of pressure cells is illustrated in figure 1. Both the seat pad and backrest were divided into a matrix of 2-by-2 pressures cells.

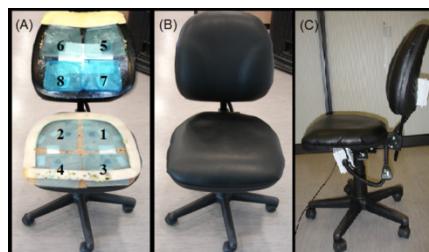


Fig. 1. (A) Distribution of the pressure cells in the chair. In the seat pad we accounted for the ischial tuberosities (Sensors 1 and 2) and the thigh region (Sensors 3 and 4). For the backrest we accounted for the scapular region (Sensors 5 and 6) and finally for the lumbar region (Sensors 7 and 8). Frontal (B) and lateral (C) view of the chair with the padding foam.

We used the original padding foam of the chair, placing it above the pressure cells to maintain the anatomical cut of the seat pad and backrest as shown in figures 1-B and 1-C. Cell size was chosen in order to minimize the gaps between cells (large gaps would be uncomfortable for the users), while also covering the areas described above. We used a Honeywell 24PC Series piezoelectric gauge pressure sensor to measure internal cell pressure. Cells in the seat pad were rated to 15 psi, with a sensitivity of 15mV/psi while the cells in the backrest were rated to 5 psi with a sensitivity of 21 mV/psi.

3.2 Experiments

Two experiments were done with different datasets. The first experiment (A) served for data acquisition in order to create the Seated Posture Classification Algorithms while the second experiment (B) was done to test the Classification in real-time using a mobile application. The dataset for both experiments is presented in table 1. Half of the subjects (15) participated in both experiments, so in experiment B we also tested with the classification to unfamiliar users, since the other half did not participate in A.

Table 1. The dataset for experiments A and B. Here (M/F) corresponds to (Male/Female).

| Dataset | No. of subjects (M/F) | Age (years) ^a | Weight (Kg) ^a | Height (cm) ^a |
|---------|--------------------------|--------------------------|--------------------------|--------------------------|
| A | 30 (15/15) | 20.9±2.4 | 67.8±13.3 | 172.0±8.1 |
| B | 30 (15/15) | 20.5±2.0 | 68.9±12.4 | 172.3±8.7 |

^a Values for Average±Standard Deviation

Before conducting the experiments, we needed to define the specific time of inflation for each pressure cell, in order for them to have enough air to sense the pressure of the subject in the sitting position, but not enough to cause discomfort to the users. After some tests (data not shown), we decided to use a value of 4 seconds for inflating pressure cells represented by 1, 2, 3, 4, 7, 8 and 5 seconds for the inflation of pressure cells number 5 and 6 for every subject during both experiments.

Before undergoing any experiment, subjects were asked to empty their pockets and to adjust the stool height so that the knee angle was at 90° (angle between the thigh and the leg) and to keep their hands on their thighs.

Experiment A was comprised of two tests, the first involved showing a presentation of the postures P1 to P11, each for a duration of 20 seconds, asking the subject to mimic those postures without leaving the chair. The second consisted in showing the same presentation, with every posture being repeated three times, but after every 20 seconds we asked the subject to walk out of the chair, take a few steps and sit back.

The eleven postures used in experiment A are represented in figure 2 and were based on previous works [8, 11, 12], since they include the most common posture found in office environments. We added the posture P5 - “Leaning back with no lumbar support” (also reported as a posture that some office workers might adopt [14]) to the previous 10 postures.

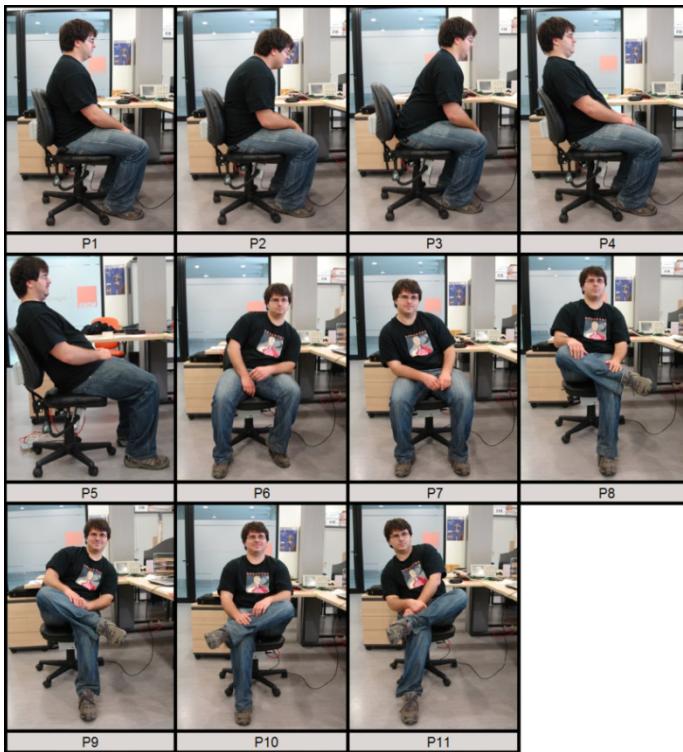


Fig. 2. Seated postures used in the experiments and their respective Class label: (P1) Seated upright, (P2) Slouching, (P3) Leaning forward, (P4) Leaning back, (P5) Leaning back with no lumbar support, (P6) Leaning left, (P7) Leaning right, (P8) Right leg crossed, (P9) Right leg crossed, leaning left, (P10) Left leg crossed, (P11) Left leg crossed, leaning right.

Not all of the data acquired was used for the classification, because when a user changes his posture, the pressure maps will oscillate (Transient zone) until they stabilize (Stable zone), as shown in figure 3. Here we focus our study on the Stable zone (figure 3) of the pressure maps and therefore, approximately 13 out of the 20 seconds were used. Since our sampling rate is 18.4 Hz, we were able to extract 240 data-points out of the 13 seconds, which were divided in groups of 40 points.

The average of those groups was used to create 6 pressure maps for posture classification, giving a total of 720 maps for each posture (30 subjects * 4 repetitions * 6 pressure maps) and a total of 7920 maps (720 * 11 postures). For each user, 12 seconds of data points from the Stable Zone (see figure 3) were previously acquired in posture P1 in order to define a baseline pressure. All the 7920 maps were normalized to an input interval of [-1, 1] for the Artificial Neural Networks (ANNs). For the creation of the ANNs we used the MATLAB® Neural Network Toolbox™ and then exported the ANN to a mobile application for experiment B.

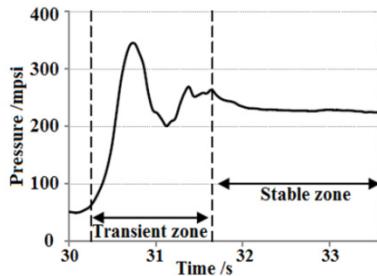


Fig. 3. Pressure measurement from pressure cell 1, when a subject went from posture P7 to P8, showing the Transient zone and the Stable zone

In experiment B we also showed a presentation with the postures in a specific order: (P1 → P3 → P4 → P1 → P6 → P7 → P1 → P8 → P10 → P1 → P5 → P1), where each posture lasted 15 seconds.

In this experiment, each posture was classified every 2 seconds, using the average of all the data-points acquired during that time. Before starting the classification 12 seconds of data-points were also gathered in order to set a baseline for each user. Half of the subjects from experiment B did not participate in experiment A, allowing us to test how the classification in real time acted for people that were first time visitors, and were not part of the training database.

4 Results

4.1 Results for Experiment A

Initially, we tested various parameter combinations such as number of neurons, number of layers, transfer function and network training function. In table 2 we present all considered combinations of parameters and the ANN returned the best overall classification. We also tried different combinations of transfer functions depending on the number of layers. For this test we used a “leave-1-out” program that would use 29 subjects to create the network, and then the last subject was used to test the network. Using the average of all the 30 “leave-1-out” processes, we were able to choose the best parameters for our ANNs. Here we only present the data from the best ANN.

Table 2. Parameter combination for the Neural Networks and the parameters that gave the best overall classification scores. Here, LM correspond to Levenberg-Marquardt algorithm, SCG to Scaled Conjugate Gradient algorithm and RP to Resilient Backpropagation algorithm.

| Parameters | Combination | Best |
|---------------------------|---------------------------|-------------|
| Nº of Neurons | 10, 15, 20, 25, 30, 35,40 | 15 |
| Nº of Layers | 1,2,3 | 1 |
| Transfer function | Tansig and Logsig | Only Tansig |
| Network training function | LM, SCG, RP | RP |

With the best parameterization, we used the best parameters obtained in Table 2 to train a new ANN to gather the weights and bias of the ANN in order to export them to a mobile application for real time posture classification. For this we divided the entire dataset (7920 pressure maps) in 60% for the ANN training, 20% for the validation and the rest for the ANN testing. A simple feedforward network with one-way connections from input to the output layers was able to fit our multidimensional mapping problem with a good overall classification score and can be very simply implemented in other systems without needing the MATLAB® Neural Network Toolbox™, since we can obtain the weights and bias of the ANN and export them to other systems.

| TRAIN Confusion Matrix | | | | | | | | | | | | TEST Confusion Matrix | | | | | | | | | | | | | |
|------------------------|------|------|------|------|------|------|------|------|------|------|------|-----------------------|----|------|------|------|------|------|------|------|------|------|------|------|------|
| 1 | 375 | 50 | 9 | 8 | 0 | 2 | 0 | 8 | 0 | 5 | 3 | 85.1 | 1 | 140 | 16 | 2 | 4 | 0 | 0 | 0 | 2 | 0 | 3 | 1 | 83.3 |
| 2 | 11 | 253 | 67 | 11 | 0 | 0 | 13 | 0 | 4 | 3 | 0 | 69.9 | 2 | 10 | 82 | 29 | 3 | 0 | 0 | 4 | 0 | 1 | 0 | 0 | 63.6 |
| 3 | 4 | 83 | 316 | 27 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 72.3 | 3 | 4 | 25 | 85 | 6 | 5 | 2 | 0 | 1 | 0 | 0 | 0 | 66.4 |
| 4 | 15 | 38 | 25 | 314 | 26 | 3 | 0 | 2 | 0 | 7 | 0 | 73.0 | 4 | 4 | 11 | 13 | 118 | 13 | 0 | 0 | 0 | 0 | 4 | 0 | 72.4 |
| 5 | 0 | 4 | 7 | 32 | 391 | 3 | 0 | 0 | 0 | 0 | 0 | 89.5 | 5 | 0 | 1 | 1 | 10 | 131 | 1 | 0 | 0 | 0 | 0 | 0 | 91.0 |
| 6 | 7 | 0 | 4 | 3 | 0 | 411 | 3 | 0 | 31 | 6 | 0 | 88.4 | 6 | 4 | 0 | 2 | 1 | 0 | 115 | 2 | 0 | 11 | 4 | 0 | 82.7 |
| 7 | 3 | 0 | 0 | 0 | 1 | 0 | 355 | 4 | 0 | 0 | 57 | 84.5 | 7 | 2 | 0 | 0 | 0 | 2 | 0 | 124 | 2 | 0 | 0 | 15 | 85.5 |
| 8 | 0 | 5 | 0 | 13 | 0 | 0 | 5 | 421 | 20 | 0 | 0 | 90.7 | 8 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 129 | 5 | 0 | 0 | 94.2 |
| 9 | 0 | 0 | 0 | 6 | 0 | 21 | 0 | 9 | 386 | 0 | 0 | 91.5 | 9 | 0 | 0 | 0 | 2 | 0 | 9 | 0 | 7 | 125 | 0 | 0 | 87.4 |
| 10 | 3 | 3 | 4 | 5 | 6 | 4 | 0 | 0 | 0 | 400 | 22 | 89.5 | 10 | 4 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 121 | 5 | 90.3 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 46 | 4 | 0 | 18 | 340 | 83.3 | 11 | 0 | 0 | 0 | 0 | 0 | 17 | 2 | 0 | 10 | 125 | 81.2 | |
| | 89.7 | 58.0 | 73.1 | 74.9 | 91.6 | 91.9 | 84.1 | 93.8 | 87.5 | 91.1 | 80.6 | 83.4 | | 83.3 | 60.3 | 63.9 | 80.3 | 86.2 | 90.6 | 83.8 | 90.2 | 88.0 | 85.2 | 85.6 | 81.8 |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | |

Fig. 4. Confusion Matrices for experiment A. Rows indicates the Output Class and Columns indicates the Target Class. The Class labels correspond to the respective postures from figure 1. The gray boxes give the percentages of correct classification in relation to the respective class and the blue box represents the overall classification score.

We obtained an overall classification of 83.4% and 81.8%, respectively for the training and testing of all postures. As confirmed in previous studies, postures P2, P3 and P4 had the lowest classification scores. [11]. In these postures, the torso shifts in the anteroposterior axis while the lower part of the body remains still. Therefore, the classification of these postures greatly rely on the backrest's pressure cells, opposed to postures such as P5 or P6, featuring lateral movement and affecting all pressure cells.

4.2 Results for Experiment B

Our next objective was to export the ANN to a mobile application in order to classify the user position in real-time. Our first test with the eleven postures had very low scores for real-time classification, especially for position P1. With this in mind, we decided to test again with just 4 postures (P1 to P4) using the data-points from experiment A, but creating an ANN with just four outputs instead of the original eleven. In

this test, we still observed that the classification of P1 was much worse than expected (beneath 25%), so in order to identify the problem, we observed the acquired pressure maps, during experiment A, for each posture. In figure 5, we present the normalized pressure data from each Sensor and how it varies for Posture P1, P2, P3 and P4. The normalized pressure data is calculated by subtracting the experimental data from each posture with the baseline that was obtained before doing the experiments.

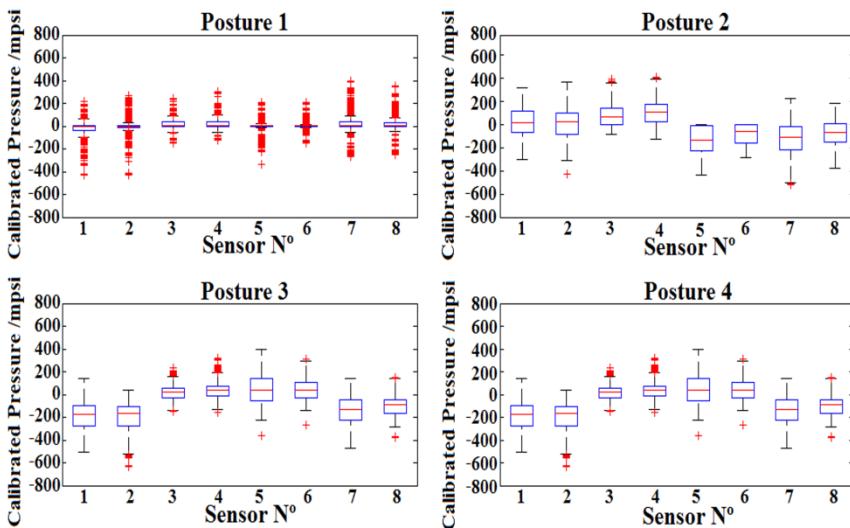


Fig. 5. Difference between the pressures measured in each sensor for postures P1, P2, P3 and P4 and the baseline pressure

As expected, the pressure maps for Posture 1 in Figure 5 tend to cluster around 0 mpsi because the data-points for the baseline were also acquired while the user maintained the P1 posture. This causes a problem to the real-time classification, because any deviations larger than 50 mpsi (which can happen during the transition from one posture to another) in any sensor makes the ANN to classify incorrectly the Posture P1.

To solve this problem we created 2 different ANNs, one specifically designed to classify P1, and one to target the other Postures. To choose between each ANN we defined a Threshold for each sensor. The goal of this Threshold was to include the maximum of P1 data-points (from experiment A), while excluding the other Postures. We tested several value combinations and were able to obtain 2 different Thresholds.

The first (Threshold 1) included 86% of P1, but also included 32 % of P4 (other postures were all below 5%), while the second (Threshold 2) included 84% of P1 and 24% of P4. Due to these values we also included P4 in the first ANN. The Threshold 1 and 2 values for sensors 1, 2, 7, 8, were 250mpsi and 225 mpsi, respectively. For sensors 3,4,5,6 the values were 180 and 150 mpsi, respectively for Threshold 1 and 2. The presentation order for this experiment was devised in order to test P1 after other postures. In figure 6 we present a flowchart of the real-time classification of those 8 postures.

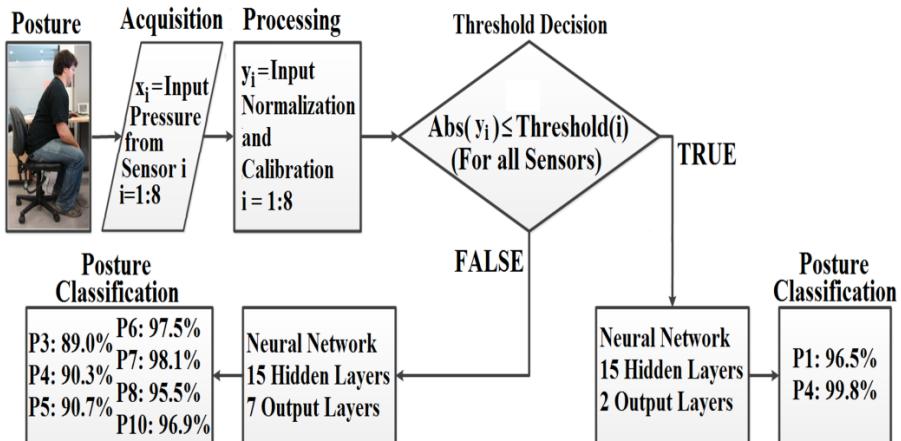


Fig. 6. Flowchart representing the real-time classification for experiment B. The Posture Classification boxes have the classification scores of the respective ANN.

The classification values using each Threshold are presented in table 3. We were able to achieve an overall score of 93.4% (increasing the previous real-time classification without the Thresholds) for those eight postures using Threshold 1, although P1 still has a lower classification score than the others.

Table 3. Classification for each posture of experiment B

| Position | Classification using Threshold 1 | Classification using Threshold 2 |
|----------|----------------------------------|----------------------------------|
| P1 | 74.0% | 62.0% |
| P3 | 93.3% | 91.7% |
| P4 | 88.3% | 91.7% |
| P5 | 100.0% | 100.0% |
| P6 | 98.3% | 98.3% |
| P7 | 98.3% | 98.3% |
| P8 | 95.0% | 98.3% |
| P10 | 100.0% | 100.0% |

5 Conclusions and Future Work

A chair prototype with pressure cells in the seat pad and backrest was developed to detect the posture and correct bad posture adoption over long periods of time. Pressure maps of eleven postures were gathered in order to classify each posture using ANNs. First we studied the best parameters of the ANNs for the classification of our data-points and then, using the best parameters, we created an ANN and exported it to mobile application and execute the postural classification in real-time.

Results showed that for the eleven postures, real-time classification the overall classification of each posture was around 70% but when we reduced the classification to eight postures, we were able to obtain an overall score of 93.4 %.

Our next aim is to continue studying classification algorithms in order to improve them (mainly for classification of P1 and the other 3 postures) and to start studying the posture correction algorithms. We will do clinical trials to evaluate those correction models but also to validate our classification algorithms, which we will use to build an intelligent chair capable of posture correction, which will help in the reduction of health problems related to back pain.

References

1. Chau, J.Y., der Ploeg, H.P., van Uffelen, J.G., Wong, J., Riphagen, I., Healy, G.N., Gilson, N.D., Dunstan, D.W., Bauman, A.E., Owen, N., Brown, W.J.: Are workplace interventions to reduce sitting effective? A systematic review. *Prev. Med.* 51(5), 352–356 (2010)
2. Hartvigsen, J., Leboeuf-Yde, C., Lings, S., Corder, E.: Is sitting-while-at-work associated with low back pain? A systematic, critical literature review. *Scand. J. Public Health* 28(3), 230–239 (2000)
3. Pynt, J., Higgs, J., Mackey, M.: Seeking the optimal posture of the seated lumbar spine. *Physiother. Theory Pract.* 17(1), 5–21 (2001)
4. Van Dieën, J.H., De Looze, M.P., Hermans, V.: Effects of dynamic office chairs on trunk kinematics, trunk extensor EMG and spinal shrinkage. *Ergonomics* 44(7), 739–750 (2001)
5. Lis, M., Black, K., Korn, H., Nordin, M.: Association between sitting and occupational LBP. *Eur. Spine J.* 16(2), 283–298 (2007)
6. DHHS (NIOSH): National occupational research agenda for musculoskeletal. Publication No. 2001-117 (2001)
7. Claus, A.P., Hides, J.A., Moseley, G.L., Hodges, P.W.: Is ‘ideal’ sitting posture real?: Measurement of spinal curves in four sitting postures. *Man. Ther.* 14(4), 404–408 (2009)
8. Slivovsky, L., Tan, H.: A Real-Time Static Posture Classification System. In: Proceedings of 9th Int. Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, ASME Dynamic Systems and Control Division, vol. 69(2), pp. 1049–1056 (2000)
9. Tan, H., Slivovsky, L., Pentland, A.: A sensing chair using pressure distribution sensors. *IEEE/ASME Transactions on Mechatronics* 6(3), 261–268 (2001)
10. Zhu, M., Martinez, A., Tan, H.: Template-Based Recognitions of Static Sitting Postures. In: Proceedings of IEEE Computer Vision and Pattern Recognition for Human Computer Interaction Workshop (2003)
11. Mutlu, B., Krause, A., Forlizzi, J., Guestrin, C., Hodgins, J.: Robust low-cost, non-intrusive sensing and recognition of seated postures. In: Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, pp. 149–158 (2007)
12. Zheng, Y., Morrell, J.: A vibrotactile feedback approach to posture guidance. In: IEEE Haptics Symposium, pp. 351–358 (2010)
13. Goossens, R., Netten, M., Van der Doelen, B.: An office chair to influence the sitting behavior of office workers. *Work* 41(suppl. 1), 2086–2088 (2012)
14. Vergara, M., Page, A.: System to measure the use of the backrest in sitting-posture office tasks. *Applied Ergonomics* 31(3), 247–254 (2000)

Hierarchical Object Recognition Model of Increased Invariance

Aristeidis Tsitiridis¹, Ben Mora¹, and Mark Richardson²

¹ Swansea University,
Department of Computer Science,
Singleton Park, Swansea, SA2 8PP, United Kingdom

² Cranfield University,
Department of Informatics and Systems Engineering,
Defence College of Management and Technology,
Shrivenham, Swindon, SN6 8LA, United Kingdom

Abstract. The object recognition model described in this paper enhances the performance of recent pioneering attempts that simulate the primary visual cortex operations. Images are transformed into the log-polar space in order to achieve rotation invariance, resembling the receptive fields (RF) of retinal cells. Via the L*a*b colour-opponent space and log-Gabor filters, colour and shape features are processed in a manner similar to V1 cortical cells. Visual attention is employed to isolate an object's regions of interest (ROI) and through hierarchically-layers visual information is reduced to vector sequences learned by a classifier. Template matching is performed with the normalised cross-correlation coefficient and results are obtained from the frequently used Support Vector Machine (SVM) and a Spectral Regression Discriminant Analysis (SRDA) classifier. Experiments on five different datasets demonstrate that the proposed model has an improved recognition rate and robust rotation invariance with low standard deviation values across the rotation angles examined.

Keywords: Rotation invariance, Human vision models, Generic Object recognition, Machine vision, Biological-like vision algorithms.

1 Introduction

Visual perception in primates has been evolving for millions of years. It has enabled the effortless detection and recognition of objects in milliseconds for a multitude of dynamic environments. The recent leaps in computer processing power have allowed early models of biologically-inspired vision to mimic processes of visual perception. In the last decade, research on neuroscience, artificial intelligence and biologically-inspired computer modelling has intensified with an increasing number of applications emerging. Past physiological research [1] proved the existence of the “two cortical streams” operation in the primates’ cortex, the dorsal visual cortical stream (parietal lobe) or “*where/how*” pathway responsible for visual attention and the ventral stream (temporal lobe) or “*what*” pathway for object recognition.

It has been shown experimentally through various studies that visual saliency together with feedforward recognition can improve the recognition rate [2]. For example, Walther's model [3] introduces the notion of "proto-objects". Walther's model concentrates on the recognition of areas in a visual scene, lacking a thorough experimental analysis and validation. Moreover, this model does not process any colour information in the recognition phase. A similar method was proposed in[4] using different models (Graph-Based Visual Saliency [5] and FHLib[6]), with the same notion of isolating multiple salient objects from images as the first step in the two visual streams cooperation. An approach from [7] unified saliency maps with Bayesian probability theory for top-down visual search tasks, i.e. predicting the location of salient objects and features given prior knowledge. This method is compared against the well-known model Scale-Invariant Feature Transform (SIFT)[8] and Hierarchical Model and X (HMAX)[9], portraying ambiguous and dataset-dependent behaviour. However, all of the above models lack a complete integration of biological-like colour and morphological features and are only limited to some of the object invariance properties.

The work here builds upon previous work [10] and it is different from past studies as it: a) incorporates biologically-like colour and morphological features for visual attention and recognition, b) provides an alternative structure that improves the recognition performance over previous efforts, c) introduces biologically-inspired rotation invariance and in addition allows for partial illumination invariance, d) uses the SRDA classifier in a biologically-inspired model for the first time and compares it directly against a SVM. All code utilised in this work has been implemented in the MATLAB environment and experimental results are compared with anFHLibvariant which is referred as Salient-FHLib (SFHLib)[10]. By *Hierarchical Object Recognition Model of Increased Invariance* (HORMII) we refer to the version of model designed here.

2 The HORMII Model

Input Layer: As shown in Fig. 1 the input layer is common to the two streams. Each input Red-Green-Blue (*RGB*) image is converted into the $CIEL^*a^*b$ where L is the luminance, a the red-green and b the blue-yellow opponent channels. The use of colour-opponent channels follows human colour perception characteristics [11]. Each input image is set to 240 pixels for the shortest edge to preserve the aspect ratio. A spatial pyramid of six scales is created, with each scale being 0.6 (chosen for computational efficiency) downsampled using the bicubic interpolation. The spatial pyramid approximates the successive RF in the retina.

For the dorsal-like operations the input image remains in the Cartesian coordinate system. Similarly to the parietal lobe, this stream only needs to extract the spatial coordinates of where eyes/attention should focus.

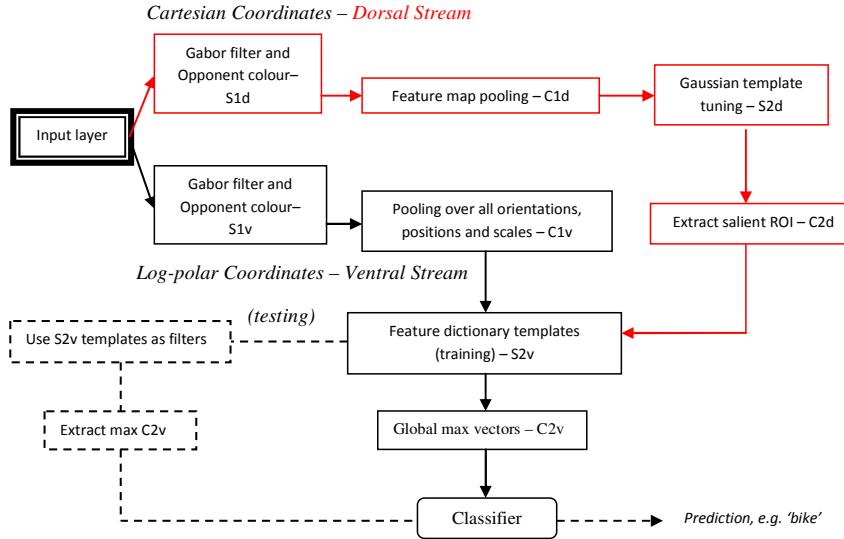


Fig. 1. The architecture of the hierarchical object recognition model proposed in this work. Red path indicates dorsal operations and dotted line the testing phase.

S1d Layer: The input image from the retina is processed from V1 simple cells [12] which are most accurately represented with log-Gabor filters in the frequency domain [13]. In many models, spatial domain Gabor filters have been used as means to represent the operations of V1 cells. However, they are particularly difficult to tune in multiresolution schema without an overlap in spatial frequencies. Furthermore, they are sensitive to low frequencies and introduce undesirable DC components higher than zero. Log-Gabor filters on the other hand, have a symmetric frequency response on a log axis which leads to an effective design of bandwidths in octaves in the frequency domain [13]. This is achieved in the frequency domain through the shift of Gaussians in log-polar (ρ, θ) coordinates where ρ is the logarithmic distance from the origin or eccentricity and θ is the angle of a point from the origin:

$$G(s, o) = \exp \left\{ -\frac{1}{2} \left(\frac{\rho - \rho_s}{\sigma_\rho} \right)^2 + \left(\frac{\theta - \theta_{(s,o)}}{\sigma_\theta} \right)^2 \right\} \quad (1)$$

In equation (1), G is the filter, $s \in \{1, \dots, n\}$ is the scale, $o \in \{1, \dots, m\}$ is the orientation:

$$\rho_s = \log_2(n) - s \quad (2)$$

$$\theta_{(s,o)} = \begin{cases} \frac{\pi}{m} & \text{if } s \text{ odd} \\ \frac{\pi}{m} \left(o + \frac{1}{2} \right) & \text{if } s \text{ even} \end{cases} \quad (3)$$

$$(\sigma_\rho, \sigma_\theta) = 0.996 \left(\sqrt{\frac{2}{3}}, \frac{1}{\sqrt{2}} \frac{\pi}{m} \right) \quad (4)$$

In equation (4), σ_ρ and σ_θ are the bandwidths for ρ and θ respectively. Two octaves can cover the extent of the RF from retina to V1 [14]. Here the chosen number of orientations is sixteen (to preserve spatial integrity minimum number is eight [15]), and for bandwidth two octaves. The responses from all Gabor orientations are pooled together using a maximum operator [9]. This operation significantly reduces subsequent operations since the 16 dimensions of the S1 layer reduce to one.

C1d Layer: The task of this layer is to form the conspicuity maps from orientation, intensity and colour. Symbol Θ refers to across scale differences, symbol \oplus to the across scale addition and N stands for normalisation. The centrec scales 1-3 are subtracted from scales 4-6 insurround s as seen in equation (5) and this leads to a total of three C1d feature maps which are normalised and summed as in equation (6). This yields to one conspicuity map for Gabor orientations.

$$O_{(c,s)} = O_{(c)} \Theta O_{(s)} \quad (5)$$

$$\bar{O} = \sum N \left(\begin{smallmatrix} 3 \\ 1 \end{smallmatrix} \oplus N(O_{(c,s)}) \right) \quad (6)$$

The double opponent channels a and b are processed using the same number for centre and surround scales. These channels are not filtered with Gabor filters as they already represent the spatial distribution of RG and BY differences. The across scale differences of these channels, equations (7), (8), lead to three double-opponent colour conspicuity maps:

$$RG_{(c,s)} = (a_{(c)} \Theta a_{(s)}) \quad (7)$$

$$BY_{(c,s)} = (b_{(c)} \Theta b_{(s)}) \quad (8)$$

$$\bar{C}_{RG} = \sum N \left(\begin{smallmatrix} 3 \\ 1 \end{smallmatrix} \oplus N(RG_{(c,s)}) \right) \quad (9)$$

$$\bar{C}_{BY} = \sum N \left(\begin{smallmatrix} 3 \\ 1 \end{smallmatrix} \oplus N(BY_{(c,s)}) \right) \quad (10)$$

The intensity conspicuity map is simply the luminance channel L processed using the same number for centre and surround scales as before:

$$I_{(c,s)} = I_{(c)} \Theta I_{(s)} \quad (11)$$

$$\bar{I} = \sum N \left(\begin{smallmatrix} 3 \\ 1 \end{smallmatrix} \oplus N(I_{(c,s)}) \right) \quad (12)$$

S2d Layer: A Gaussian mask of 9×9 units and $\sigma=4$ is convolved with the conspicuity maps I , O , RG and BY in this layer. All conspicuity maps are normalised together and share the same size in three dimensions i.e. (i, j, f) where i is a location in the x axis, j in the y axis and f is I , O , RG or BY .

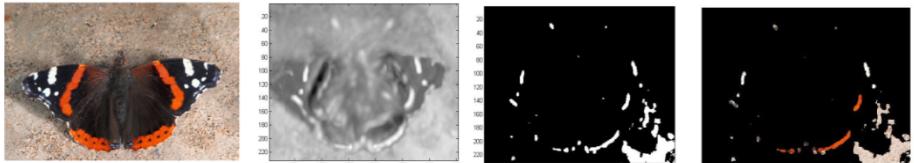


Fig. 2. A summarised example of the dorsal stream process. From left to right, an input image, its salience map, the highest areas of attraction, the areas of attraction overlayed on the original image.

C2d Layer and Beyond: The three dimensional feature maps from the S2d layer are processed with the max operator. This yields a saliency map for all features with many possible ROI as shown in Fig. 2. Neighbourhood relationships carry a particular impact on visual salience. By using k-means clustering on the saliency map data, we isolate the information on the cluster with the highest C2d values while their size is ranked from larger to smaller. Hence the highest ranking ROI are where the ventral stream extracts features similarly to the shift the eyes make in search of the next ROI in a given visual scene.

The salient ROI from C2d are transformed into the log-polar coordinate system (Fig. 3). Past physiological studies [16, 17] focused on striate cortex research, have shown that the organisation of the RF from retinal cells to striate cortex resemble the log-polar space and are quite different from the Cartesian coordinates. In support of this research, physiological experiments proved the existence of rotation invariance in humans [18, 19] which can be partly attributed to this coordinate system.

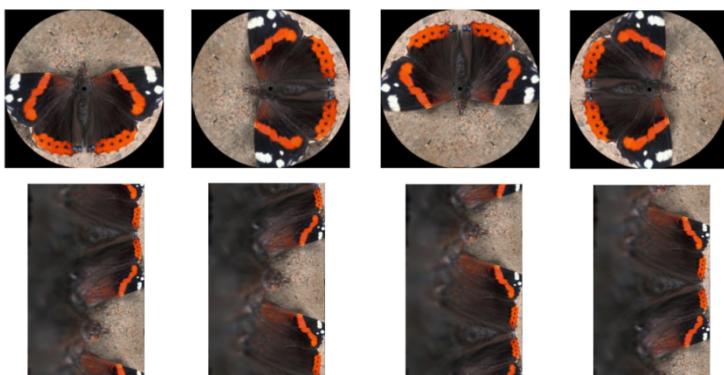


Fig. 3. Top row shows an example image in Cartesian coordinates being rotated successively by $\pi/2$ degrees in four steps from left to right. Bottom row shows each rotated image in the log-polar domain where the vertical shift is apparent.

Rotation invariance is an important property which ensures that irrespective of the rotation angle of an object, the recognition rate is unaffected or marginally reduced. Rotations in Cartesian coordinates (x, y) appear as a shift in the log-polar plane (ρ, θ) by converting with the equations below:

$$\rho = \log \sqrt{x^2 + y^2} \quad (13)$$

$$\theta = \arctan(y, x) \quad (14)$$

S1v Layer: Like S1d this ventral-like stream layer simulates V1 simple cells and while S1d applies Gabor filters on an image in (x, y) coordinates, S1v performs Gabor filtering on log-polar transform. Gabor filters are used on the luminance channel in order to obtain a spatial pyramid of 6 scales. The a and b colour opponent channels are scaled down to two pyramids without any further processing.

C1v Layer: Position and scale invariance is achieved by supplying the complex cell only with the strongest of the afferent simple cells.

$$r = \arg \max_j x_j \quad (15)$$

In the equation above, r is the response, i.e. the maximum amplitude of the S1v vectors (x_1, \dots, x_m) in a particular neighbourhood j . The max filter operates with a subsampling factor of 2 i.e. the max-pooling window is 10x10 units in size and moved in steps of 5 units along two adjacent scales in the spatial pyramid. This operation is repeated for all three pyramids, the luminance, RG and BY .

S2v Layer: During training, templates of different sizes (4x4 to 12x12) are extracted directly from the S2v layer and stored in a feature dictionary. Essentially all areas in the S2v layer are salient and templates can be extracted randomly in these ROIs. This layer only retains the uniquely extracted templates which become filters used on the C1v pyramids of all training images to obtain the maximum responses. For the template matching operation, Normalised Cross-Correlation (NCC) is used in the spatial or frequency domain, depending on the amount of computations. Advantages of the NCC are its tolerance to linear brightness and contrast changes. So similarity (s) of a patch of C1v units (f) and template (t) over (i, j) , the NCC is:

$$s(i, j) = \frac{\sum_{i,j} (f - \bar{f})(t - \bar{t})}{\sqrt{\sum_{i,j} (f - \bar{f})^2 \cdot \sum_{i,j} (t - \bar{t})^2}} \quad (16)$$

C2v Feature Vectors: If an image is rotated in Cartesian coordinates, the image in log-polar is shifted as shown in Fig. 3. Shifts in log polar coordinates may chop areas and distort efficient template matching. Regardless if rotation occurs, template matching is always performed in a given C1v layer which has been shifted three times by a third of the image. The maximum response over these three fixed shifts yields a C2v feature vector. C2v vectors during training are fed to a classifier for learning. During testing the entire process of feature extraction from the input layer for both

visual streams is repeated. The C2v vectors from testing images are collected by applying the stored S2v templates over the test images and passed on the pre-trained classifier for supervised classification.

3 Experiments

The datasets are first tested against aFHLib-like variant known as SFHLib. The second step of the experiments introduces the current model HORMII without any rotation invariance and so both the dorsal and ventral streams operate in Cartesian coordinates. The third version of the model includes rotation invariance and log-polar coordinates in the ventral stream. The chosen number of templates per training image is 50, a previously known “abundant” number picked to avoid underfitting. The number of training images is 15 and the number of testing images is a different sample of also 15. Classification accuracy is defined as the percentage of the test samples correctly classified over the total number of samples in each classification experiment. The methodologies are compared under two different classification schema, a linear kernel SVM [20] and a linear kernel SRDA [21]. Classifier parameters were set by employing the cross-validation technique. Overall, five different image datasets are used: 1) The “butterflies UIUC” dataset, from the University of Illinois at Urbana-Champaign [22]. 2) The “birds UIUC” dataset, again from the UIUC [23]. 3) The “Birds Caltech” dataset of the first 10 unique bird classes [24]. 4) A 10 class dataset assembled from the ImageNet database [25]. 5) A 25 class dataset of rigid shape objects from the ImageNet database [25].

All datasets are in RGB and in our experiments with the dataset “Butterflies”, classes “Monarch 1” and “Monarch 2” were merged into one class. Moreover, the class “black swallowtail” was enriched with more images from the internet since the initial number of images was low. The Caltech-UCSD Birds 200 is a particularly challenging object recognition dataset. In order to simplify it here, the first 10 unique classes were separated and converted into a dataset referred as “Birds Caltech” without further modifications.

Table 1. The average classification accuracies in percentage (%), over three independent runs for SFHLib and HORMII. The annotation Rotation Invariance (R.I.) indicates log-polar coordinates.

| <i>Dataset</i> | <i>Birds</i> | <i>Butterflies</i> | <i>Birds</i> | <i>10 class</i> | <i>Rigid-25</i> |
|--------------------|--------------|--------------------|----------------|-----------------|-----------------|
| <i>Algorithm</i> | <i>UIUC</i> | <i>UIUC</i> | <i>Caltech</i> | <i>ImageNet</i> | <i>ImageNet</i> |
| SFHLib-SVM | 48.25 | 46.53 | 21.87 | 41.03 | 40.75 |
| SFHLib-SRDA | 43.05 | 44.79 | 19.37 | 39.16 | 42.58 |
| HORMII-SVM | 50.69 | 53.12 | 36.66 | 63.95 | 62.5 |
| HORMII-SRDA | 57.29 | 61.8 | 44.37 | 64.37 | 65.33 |
| HORMII-SVM (R.I.) | 44.78 | 65.27 | 35.41 | 51.24 | 42 |
| HORMII-SRDA (R.I.) | 51.04 | 66.66 | 40.83 | 56.87 | 44.66 |

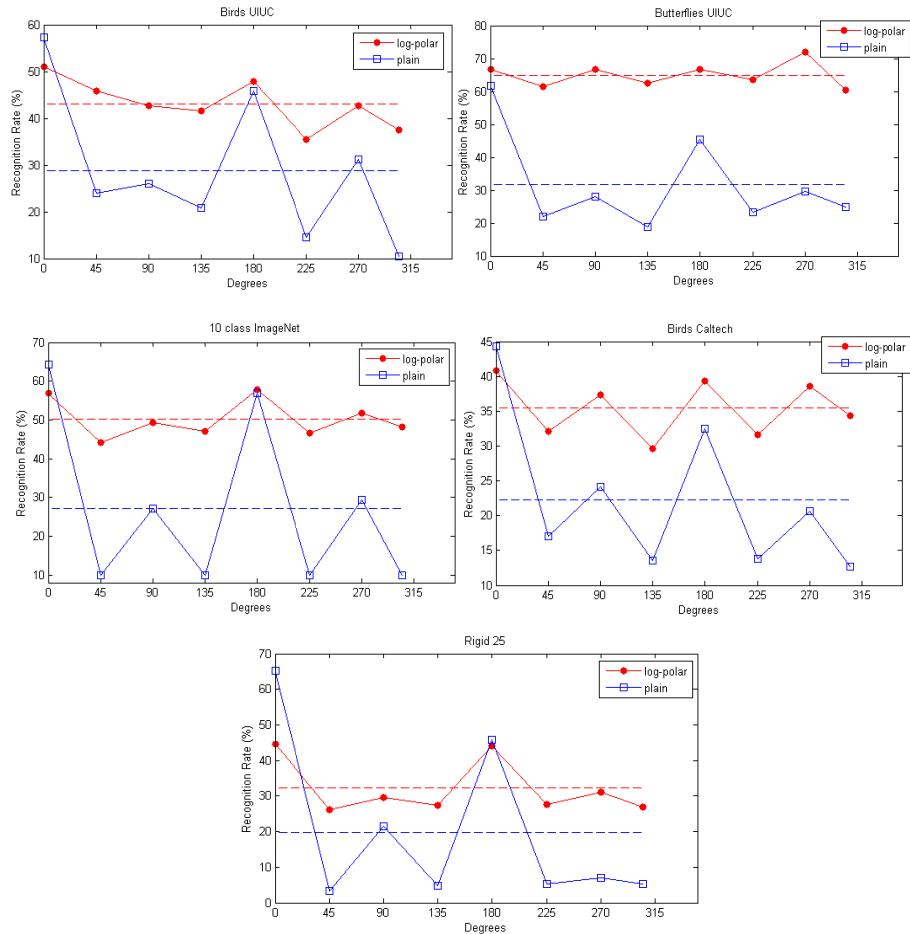


Fig. 4. Image rotation graph results, the blue plot shows the HORMII original and the red RI HORMII. Dashed lines illustrate mean values. All results were obtained over three independent runs using the SRDA classifier.

Table 1 summarises the recognition rates achieved through the two biologically-inspired models by using two different classifiers. At first it can be observed that the lowest recognition rates were produced from SFHLib. Its poor performance is highlighted even more in the “birds-Caltech” dataset. This behaviour is primarily caused by the lack of colour features and the sole use of shape features which limit the extent of information on the template matching comparisons being made. Another trend emerging from the particular SFHLib results is that although the two classifiers produced similar results, SRDA has shown a slightly inferior performance in all datasets.

In the original HORMII, there is a noticeable increase in classification accuracy. The highest increase for the SVM is seen in the “10 class” dataset with almost 23%.

On average across all datasets the difference between SFHLib and HORMII with an SVM is 13.7% higher for HORMII. The results obtained from the HORMII-SRDA combination, in contrast with SFHLib, show improved performance over SVM. This improvement amounts to an average of 5.25% better for SRDA across all datasets. For HORMII-SRDA, there is a significant discrepancy between the “Rigid-25” and the “Birds Caltech” datasets. The difference is approximately 21% better for the “Rigid-25”. As mentioned above the “Birds Caltech” is a challenging dataset due to the large number of similar bird species and for the significant variation in pose these animals exhibit. Perhaps less so compared to the other datasets, the model relies heavily on colour information. This also explains the substantial increase of 25% accuracy between SFHLib-SRDA and HORMII-SRDA for this dataset. On the contrary, even though the “Rigid-25” has a larger number of classes, these are distinct with each other and portray almost no variation in pose. The biologically-inspired models through the use of sparse local features, exhibit some pose invariance which largely depends on the number and variation of the training samples.

In the rotationally invariant HORMII, classification accuracy decreases in SVM by 5.64% (mean value at 47.74%) and SRDA by 6.61% (mean value at 52.01%) on average across all datasets lower compared to HORMII original (mean values at 53.38% and 58.63% for SVM and SRDA respectively). The reason for this reduction resides in the partial loss of spatial integrity of the objects when transformed to log-polar coordinates, especially when the actual information is denser in the outer regions of the log-polar image (Fig. 3). The set of results for HORMII RI shown in Table 1 are at 0° i.e. all images in the datasets during training and testing are used directly without any influence. The experiments in Table 1 are extended using the trained instances of HORMII original-SRDA and RI HORMII - SRDA, by rotating the test images in steps of 45° for all datasets as shown in Fig. 4. SRDA is presented instead of SVM as it performed better but the trend is similar in both classifiers.

Fig. 4 shows that RI HORMII produced better and consistent results as opposed to HORMII original. The only instance where HORMII original surpasses RI HORMII is for 0° and does not recover at any other rotation angle. It is equally noticeable in all datasets that the HORMII original version of the model produces consistently high results when the image is flipped vertically at 180°. This trend is also loosely followed by RI HORMII. The HORMII original version produced results which fail to achieve any rotation invariance characteristics and it becomes apparent that in real-world situations any camera movement with rotation beyond 0° would have detrimental effects in the recognition rate. The mean recognition rates of the RI HORMII (red) in all the plots of Fig. 4 remain well above the HORMII (blue). In RI HORMII, the average recognition rate is highest for the “butterflies dataset” at 64.96% as opposed to just 31.71% for HORMII original. Furthermore, as Fig. 4 shows the highest percentage for this particular dataset is obtained at 270°. This is not surprising, since as the image rotates in Cartesian coordinates and shifts in the log-polar domain occur, effective template matching follows the stored feature book templates as an average over the training samples and not of the object alignment itself. Increasing the number of shifts the model computes when performing template matching might resolve this difference in performance but at the same time will introduce further computations.

The standard deviation values σ , under the RI HORMII for all datasets, can be considered as ‘low’, the lowest value $\sigma=3.73$ is noticed in the “butterflies dataset” and the highest in the “Rigid-25”, $\sigma=7.65$. Conversely for the HORMII original, the lowest $\sigma = 11.12$ is at Birds Caltech and highest “Rigid-25”, $\sigma=23.41$. Overall, the HORMII original is dominated by larger standard deviation values that demonstrates its inefficiency with rotation problems. In contrast, the improved version that uses the log-polar coordinate system performs consistently better across all angles.

4 Conclusions

Following earlier cortex-like machine vision models of visual attention and object recognition, the contributions of this paper included the efficient extraction of colour and morphological saliency features in order to enhance the object recognition performance. Furthermore, rotation invariance is introduced by transforming input images from Cartesian coordinates into the log-polar domain which is consistent with recent neurological findings [16, 17]. Experiments across all of the five different datasets in this work have shown that the proposed model has a significantly increased recognition rate e.g. the highest improvement observed at nearly 25.5%. By applying the log-polar transformation, the same algorithm portrayed rotation invariance characteristics with consistently small standard deviation values across all of the rotation angles tested. These results were validated under two different classification schema and have shown similar trends. For the improved version of the model, SRDA proved more efficient than SVM. It is planned to use more datasets to verify the HORMII algorithm further. Future work will concentrate on further improving recognition rates and reducing computation times by investigating the use of GPUs. Moreover, it is planned to simulate the Lateral Geniculate Nucleus role in human vision so as to utilise the context (or gist) information for simplifying the classification process into a structured learning problem.

References

1. Mishkin, M., Ungerleider, G., Macko, A.K.: Object vision and spatial vision: Two cortical pathways. *Trends in Neuroscience* 6, 414–417 (1983)
2. Han, S., Vasconcelos, N.: Biologically plausible saliency mechanisms improve feedforward object recognition. *Vision Research* 50, 2295–2307 (2010)
3. Walther, D., Koch, C.: Modeling attention to salient proto-objects. *Neural Networks* 19, 1395–1407 (2006)
4. Tsitiridis, A., Yuen, P., Hong, K., Chen, T., Kam, F., Jackman, J., James, D., Richardson, M.: A biological cortex-like target recognition and tracking in cluttered background. In: SPIE, Optics and Photonics for Counterterrorism and Crime Fighting, Berlin, p. 74860G (2009)
5. Harel, J., Koch, C., Perona, P.: Graph-Based Visual Saliency. In: Advances in Neural Information Processing Systems, vol. 19 (2007)
6. Mutch, J., Lowe, D.: Object class recognition and localisation using sparse features with limited receptive fields. *International Journal of Computer Vision* 80, 45–57 (2008)

7. Elazary, L., Itti, L.: A Bayesian model for efficient visual search and recognition. *Vision Research* 50, 1338–1352 (2010)
8. Lowe, D.: Object recognition from local scale-invariant features. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, pp. 1150–1157 (1999)
9. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M.: Robust Object Recognition with Cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 411–426 (2007)
10. Tsitiridis, A., Richardson, M., Yuen, P.: Salient feature-based object recognition in cortex-like machine vision. *Engineering Intelligent Systems. Special Is.* (2012)
11. Engel, S., Zhang, X., Wandell, B.: Colour Tuning in Human Visual Cortex Measured with functional Magnetic Resonance Imaging. *Nature* 388, 68–71 (1997)
12. Daugman, J.G.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of Optical Society of America* 2, 1160–1169 (1985)
13. Field, D.J.: Relations between the statistics of natural images and the response properties of cortical cells. *Journal of Optical Society of America A* 4, 2379–2394 (1987)
14. DeValois, R., Albrecht, D., Thorell, L.: Spatial Frequency Selectivity of Cells in Macaque Visual Cortex. *Vision Research* 22, 545–559 (1982)
15. Fischer, S., Sroubek, F., Perrinet, L., Redondo, R., Cristobal, G.: Self-Invertible 2D Log-Gabor Wavelets. *International Journal of Computer Vision* 75, 231–246 (2007)
16. Van Essen, D.C., Newsome, T.W., Maunsell, J.H.: The visual field representation in striate cortex of the macaque monkey: Asymmetries, anisotropies and individual variability. *Vision Research* 24, 429–448 (1984)
17. Johnston, A.: A spatial property of the retino-cortical mapping. *Spatial Vision* 1, 319–331 (1986)
18. Hollard, V.D., Delius, J.D.: Rotational Invariance in Visual Pattern Recognition. *Science* 218, 804–806 (1982)
19. Harris, I.M., Dux, P.E.: Orientation-invariant object recognition: evidence from repetition blindness. *Cognition* 95, 73–93 (2005)
20. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. *Machine Learning Research* 9, 1871–1874 (2008)
21. Cai, D., He, X., Han, J.: SRDA: An Efficient Algorithm for Large-Scale Discriminant Analysis. *IEEE Transactions on Knowledge and Data Engineering* 20, 1–12 (2008)
22. Lazebnik, S., Schmid, C., Ponce, J.: Semi-Local Affine Parts for Object Recognition. In: *Proceedings of the British Machine Vision Conference*, London, pp. 959–968 (2004)
23. Lazebnik, S., Schmid, C., Ponce, J.: A Sparse Texture Representation Using Local Affine Regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 1265–1278 (2005)
24. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-UCSD Birds 200 (2010)
25. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: *Computer Vision and Pattern Recognition* (2009)

Detection of Damage in Composite Materials Using Classification and Novelty Detection Methods

Ramin Amali and Bradley J. Hughes

University of the West of England, Frenchay Campus,
Coldharbour Lane, Bristol, BS16 1QY, UK

Abstract. The increased use of composite materials in engineering applications, and their susceptibility to damage means that it's imperative that robust testing technique are developed to help in the detection of damage. Many of the detection techniques currently available are highly complex, difficult to conduct and rely on human interpretation of data. Simple testing methods are available but are too unreliable to be used effectively. This investigation explores the development of simple testing methods which use classification and novelty detection methods to detect the presence of damage in composite materials, making the process of damage detection much quicker, simpler and more versatile.

Keywords: Composite, NDT, NDI, Damage Detection, Delamination, Fourier Transform, Digital Signal Processing, Neural Network, Novelty Detection.

1 Introduction

Composite materials such as carbon fibre, are continually finding new and ever more interesting applications in the world of engineering, they are now widely used in the aerospace industry and are increasing being used in the automotive sector, particularly in motorsport where every effort to save weight is made.

These new composite materials can be designed to be stronger and lighter than traditional materials, giving composite materials a significant performance advantage. Whilst superior in many areas composites are however very susceptible to damage which is an obvious major drawback. With traditional materials like steel, impact damage will typically form a dent in the surface of the material and not significantly affect the performance of the system. In a composite material however the effect is more severe; because composite materials are made from a combination of fibres and resin, when it is impacted the fibres become damaged causing a significant weakening of the material and hence the engineering system built from it [1].

The method by which composite materials are manufactured can also cause performance issues. Composite parts are typically made by layering sheets of composite material, called plies, on top of one another. If this process is not properly executed then multiple issues may arise. Plies could be laid and an overlap formed, there could be gaps between plies or there could be ply drop off. One of the most critical flaws that can be formed is known as a *delamination* this is where the bonding

between two plies has broken down and the two plies have come slightly apart causing an air gap to form. The occurrence of delaminations is a problem because it reduces the strength of the material, and is a possible source of further damage through the life of the part. Delaminations often occur as a result of impact damage.

With such a dramatic uptake of composite usage, particularly in the aerospace industry, it is important that any defects in servicing material are quickly found. The detection of these flaws is often carried out through Non-Destructive Testing (NDT). The use of NDT aims to identify the presence of defects within a material, determining its location, and in many cases the size and type of defect found. Many different NDT methods have been developed for composite materials, including radiography, ultrasound and thermography. The issue here is that despite the developments made in the use and development of non-destructive testing for composite materials and structures, few simple, cheap and effective methods exist for the testing of composite materials. A number of methods have been developed but these are often too complex to be quickly applied in the engineering workplace [2] [3].

One method that does exist however is the tap test method, which is a commonly used in-service inspection technique used by many as the first testing method for composite materials. The tap test can be used to find the presence of many of the defects present in composite material, namely delaminations. Similarly to finding a wall stud, the material is tapped using a tool, commonly a coin or small hammer, and the acoustic responses observed. By noting the changes in acoustic tone it is possible to use tap testing to find the location of delamination in composite materials [4].

Whilst tap testing is somewhat successful, it typically suffers from a number of problems, most noticeably the subjective interpretation of acoustic results.

This paper details the research undertaken in the design of a new NDT method that aims to remove the problem of subjective human interpretation of acoustic tones, instead using Digital Signal Processing (DSP) combined with Artificial Neural Networks (ANNs) to perform the interpretation of the obtained test data.

2 Impact Damaged Detection Using Classification Methods

As previously stated a wide range of NDT methods have been developed for use with composite materials. Because of the unique nature of composite materials these testing methods are often very complex and expensive to perform, and for many applications they are completely impractical.

Tap testing has been around for many years but until relatively recently few attempts have been made to transition the method to a more formal and reliable method. Those that have developed the method have developed it using a dynamics based approach, monitoring the time of the impact and using that to determine the state of the material, no attention was paid to the sounds that were being made.

To address this, an investigation was conducted that would seek to determine whether the acoustic response to a tap could be used to detect the presence of damage in a composite material. Logic dictates that it should be possible; humans can detect the change in tone by ear, so it should be possible for a computer program to do the same [5].

To facilitate this investigation a single composite laminate panel was manufactured, and was used throughout the investigation. The panel would be excited using a range of different sound inputs; tap testing would be used as well as 7 other sound input methods. 4 of these input types made use of a contact speaker to impart different noise types into the sample with the other 4 being impacts performed with different tools or items.

The testing was conducted by placing the composite panel into a test rig and impacting or exciting the panel in 9 separate areas, a condenser microphone above the panel would record the resultant acoustic responses. Each of the 8 input types was used 10 times in each area to get a total of 90 recordings for each input type. With the recordings from the undamaged sample obtained the panel was then removed from the test ring and damaged using a drop weight impact machine. The central area of the panel was impacted creating a noticeable damage including broken fibers and delamination.

With the panel damaged it was placed back into the test rig and more acoustic data collected. In this case only the damaged area of the panel was tested, with 20 recording made for each sound input method.

The next phase of the investigation required the processing of the sound data collected. Simply by looking at the sound signal obtained it is often very difficult to determine anything about the signal characteristics. Instead the signal is transformed from the time domain into the frequency domain using a Fourier Transform algorithm. Further processing of the results was conducted by truncating the frequency range, banding and averaging the results as well as normalizing them [6].

A visual inspection of the data processed showed that there was significant difference in the frequency spectra for some of the sound input types. (Fig. 1).

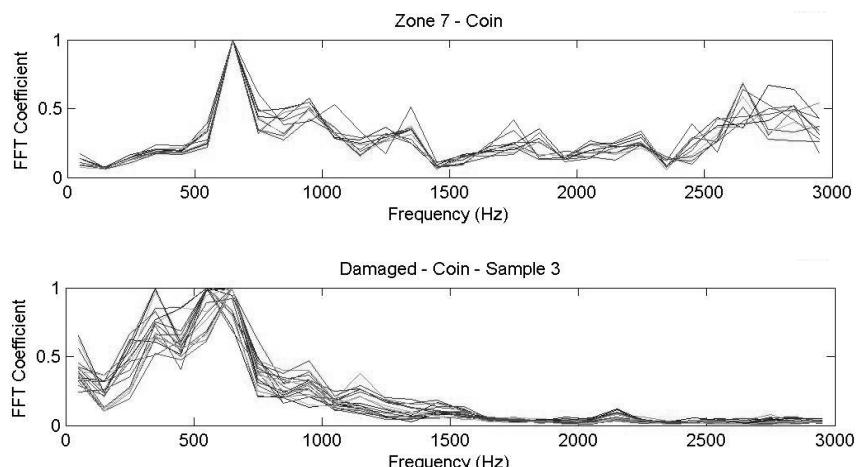


Fig. 1. A comparison of the frequency spectra obtained from the coin tap testing data. The top graph shows the data from the sample in the undamaged state, with the bottom graph showing the damaged state results. There is a clear difference between the results obtained.

The final phase of the investigation involved the development of classifier that would be able to correctly determine whether the acoustic data came from the sample in either the damaged or undamaged state. To do this a 3 layer feed-forward neural network was created and programmed using the frequency data obtained. The input to the network would be the 30 average frequency band values for each of the data sets, with the training target being the corresponding state of the sample, either damaged or undamaged. For the hidden layer it was decided that 10 neurons would initially be used. For each of the 8 sound input methods, 5 classifiers were trained to allow for a more complete evaluation of their performance.

With the classifiers trained it would then be possible to validate their performance. Throughout the experimental process additional recording had been made in preparation for this. The new validation data was processed in the same way as before to get the average frequency band values. The validation data contained 20 sets of results from the undamaged sample, and 20 from after it was damaged, 40 in total for each of the 8 input types.

The unseen data sets were fed into the networks and the networks outputs were compared against the actual classification of the data. The success of the ANN was measured by the percentage of correct classifications that were made.

Table 1. Results of Network Classification Validation

| Input Type | Network 1 | Network 2 | Network 3 | Network 4 | Network 5 |
|------------|-----------|-----------|-----------|-----------|-----------|
| Coin | 100% | 100% | 100% | 100% | 100% |
| Ball | 85% | 100% | 100% | 85% | 85% |
| Scissors | 100% | 100% | 100% | 100% | 100% |
| Tack | 95% | 100% | 95% | 95% | 95% |
| White | 80% | 50% | 50% | 50% | 50% |
| Pink | 100% | 85% | 100% | 65% | 100% |
| Brown | 100% | 100% | 100% | 100% | 100% |
| Chirp | 85% | 100% | 95% | 80% | 95% |

The results of the classifier testing proved to be extremely encouraging. It can be seen that with the exception of the white noise testing method, the classifiers were able to correctly classify the data with very good accuracy, exceeding 80% for each sound input apart from white noise.

The investigation had proved to be successful; the results show that it is possible for a classifier to be trained to correctly interpret the acoustic data with up to 100% accuracy, hence removing the possibility of selective human interpretation.

Whilst this testing and classification method proved successful there are a number of issues associated with it, the most noticeable of which being its application in real engineering systems. A classifier relies on data being known for a system not just in the as-is state, but also when it is damaged. For many engineering systems it is not practical or feasible to get the damaged state data for the system and hence this testing method is deemed useless. Further issues arise when using a condenser microphone to record the acoustic data, these microphones pick up a lot of environmental noise and could be difficult to use when conducting testing. The prevalence of these issues

makes it difficult for this testing method to be used effectively for real engineering systems. A different approach however using different microphones and making use of unsupervised learning could make it a more relevant testing method.

3 Delamination Detection Using a Novelty Measure

For a NDT method to work using a classification method it requires data from an example system in both its undamaged and damaged forms. This is often impractical to acquire and hence makes classification based NDT difficult to accomplish. One way around this issue is to make use of an unsupervised learning method in the design of the NDT system.

It is possible to design a network and train it using only the known data from an undamaged system. The network learns that this is the expected type of data that it should receive when the system is undamaged. If the network is shown damaged state data it is able to identify that it is different to what it should be, and hence it signals this difference, showing to the user that damage has been identified by the network.

The use of unsupervised learning in this manner for engineering systems was termed by Worden as *novelty detection* [7]. In his work he sought to use novelty detection to detect the presence of damage on aircraft wing panels. His work proved to be successful which was encouraging since it was analogous to the investigation being conducted here [8].

With that in mind the investigation moved on to determine whether delaminations could be identified in composite laminates using a novelty detection method. The investigation focused only on the presence of delaminations resulting from the manufacturing process of the composite laminates, as these are more difficult to identify compared to delamination due to impact damage. It was also decided that a different type of microphone would be used. Instead of condenser microphones contact microphones were used, these microphones can be stuck to a surface and record the vibrations caused by sound waves in the material. These microphones are much more suitable for the engineering applications of this detection system.

Because the microphones are stick on microphones they only pick up vibrations at one point on the composite laminates, as such their potential use is slightly more varied. This investigation looks at two possible cases where the microphones could be used. Firstly it considers if the system can detect damage directly under the positions of the microphone, and secondly whether it can detect damage when the delamination is directly beneath the sound input location [9].

3.1 Methodology

The methodology for this investigation is broadly similar to what was conducted previously. It is broken down into a 3 phase process, manufacture and data acquisition, data processing and network design and validation.

In total 3 composite plates were manufactured for this investigation, made from aircraft grade unidirectional carbon-fibre pre-preg material. Each plate was made using four 250mm square plies/layers and cured using a heated press. The first two plates were made to be undamaged so as to give good data for which to train and

validate the networks. The other plate was manufactured so that it contained an artificial delamination; made using a 7mm diameter disc placed in the center of the plate. This would generate damaged state data that could be used to further validate performance of the networks.

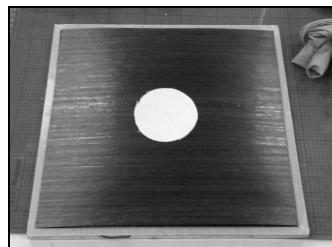


Fig. 2. The manufacture of the new plates required one of them to contain an artificial delamination. Above shows how the delamination was formed, by placing a piece of backing material between the top two plies stopping them from bonding during the curing process.

With the composite plates manufactured the next step dealt with the acquisition of contact microphones with which to conduct the tests. Two types of contact microphone were sourced, the first type were the kind that are used to tune musical instruments, with the second type of being manufactured using piezoelectric discs.

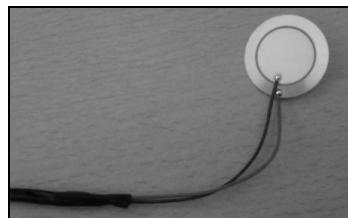


Fig. 3. The second kind of contact microphone was made by soldering together a piezoelectric disc and 3.5mm audio jack cable. This set up allows the vibration picked up by the disc to be recorded by plugging the jack into a relevant device.

With the materials acquired it would then be possible to acquire the test data. The first undamaged panel was placed into the test rig and secured in position, ensuring a uniform boundary condition was applied to each edge. Each panel had been split into 9 areas in a 3x3 grid. The central area would be used as the sound input location; with the microphone attached to the plate in one of the other areas. The sample would then be tested and the acoustic response recorded, after each test the microphone was moved to a different area, ensuring that a range of readings was recorded.

The sound inputs used to excite the sample would be some of the inputs used in the previous investigation. White noise imparted into the material using a vibration speaker had proved to be successful and so that was again used, as was the use of a chirping sound. Sound impacted made by tapping the sample with a coin and also a pair of scissors was also used.

The first composite sample was tested using each of the 4 input types detailed. In total 20 recordings were made for each input type, with the microphone being moved

after each individual test. With testing on this sample complete the composite was replaced with the second undamaged sample and the experiment repeated to gather yet more acoustic data.

The data obtained from the first undamaged sample would be used to train the novelty detector and is termed the *clean* data. The acoustic data obtained from the second panel termed the *undamaged* data, is used to help validate the performance of the novelty detector.

With the undamaged samples tested the experiment moved on to test the damaged composite sample. The damaged plate was placed in the test rig and tested in two different ways. Firstly the microphones were placed in the center of the plate directly over the delamination; the sound was then imparted into the material in one of the adjacent areas. Theoretically because the bonding of the plies beneath the microphone is compromised by the delamination, different sounds would be picked up by the microphone, compared to before. Each sound input type was again used, obtaining 20 readings for each; the process was repeated for the second variant of microphone.

The second method of testing the panel swapped the position of the microphone and sound input location. The sound would now be generated directly above the delamination, with the microphone placed in an adjacent area. Again different sounds should be picked up by the speaker because of the lack of bonding underneath the sound input location. Both microphones were used with the four different sound input types, 20 recording were again made. This concluded the testing on the two undamaged panels.

With the testing on this panel finished the process of data acquisition was complete. For each microphone and sound input type, acoustic data had been obtained from the undamaged panels, as well as from the damaged panel, tested in two different ways.

3.2 Data Processing

With the required data now obtained it would be possible to process the data to make it more useful for training the networks. As stated previously it is very difficult to use time based readings to distinguish between acoustic results, the sound signals are much too complex, to counter this the sound signals are converted into the frequency domain using the Fourier transform. Simply converting the data is not enough however, there is still too much information for it to be useful and as such it must be condensed. One method of simplifying the data came from truncating the range of frequencies investigated to include only those that a human could hear. Tap testing had relied on human interpretation for many years so this was a logical step. Research suggested that truncating the series to only look at frequencies up to 3000Hz would yield good results.

Even with the truncation of the frequency spectrum there were too many data points to make training a network suitable, the spectrum would need to be simplified further. To do this the frequency spectrum, 0-3000Hz, was split into 30 bands each being 100Hz in size. Bands were 0-100Hz, 100-200Hz and so on. Within each band the frequency magnitudes were then averaged, this produced 30 averaged frequency bands which could be used to describe the form of the frequency spectrum. Finally the bands were normalized so the effect of variable impact intensities could be

minimized. The data processing here followed the same methods as used in the previous experiment, as this had proven to be successful.

The data processing was applied to all of the data that had been collected throughout the experimental process to generate for each data set, 30 discrete frequency band magnitudes, these values could be used to describe the profile of the frequency spectra for each data set.

3.3 Training the Novelty Detectors

With acquired data processed it would then be possible to train the novelty detectors. The detectors work by training an auto associative network with the seen data. The network is asked to reproduce at the output layer the values which are presented at the input; a bottleneck in the hidden layer is used. For this investigation a three layer feed-forward network was created with 30 neurons in the input and output layers, and 10 neurons in the second layer representing the bottleneck. Because of this architecture the network learns the characteristics that describe the input signals. The *novelty* of the data sets can be assessed by subtracting the output value of the network from the input. Taking the magnitude for each node and summing gives the *Novelty Index* of the data set. Worden et al. had previously used this concept of novelty detection for the successful detection of damage on aircraft wing panels [10].

For both experimental methods, the one with the microphone on the delamination and the one with sound input over the delamination, a range of novelty detectors were programmed; one detector was made for each microphone and sound input type. The detectors were trained using the data obtained from the *clean* composite sample. Doing so allowed the detectors to learn the patterns that correspond to the undamaged data. The resultant detectors returned novelty values of essentially zero for each of the data sets shown at this stage, which is to be expected.

Having trained the detectors it would then be possible to find out their performance by asking them to find the novelty of the unseen data.

3.4 Validating the Novelty Detectors' Performance

To validate the performance of each detector, it would be asked to find the novelty index value of data that it had not seen before. In each case the detectors would be shown a vector of 200 inputs, the first 100 inputs would be the data obtained from the *undamaged* composite sample, replicated 5 times. The second 100 inputs would be the data obtained from the tests on the *damaged* sample, again replicated 5 times. Theoretically the novelty index values for the first 100 results should be approximately zero, since this data should be similar to the *clean* data used to train the network. The index values for the second 100 sample should be higher than zero, indicating that there is a difference in frequency spectra and hence the presence of a delamination.

3.5 Results of the Novelty Detection

The validation data was shown to the novelty detectors and the novelty index value for each sample calculated. The results are shown as a novelty trace over the sample

range. The difficulty with detecting damage using this method is defining what level of novelty indicates the presence of damage in the system. A system was proposed where the cumulative mean and standard deviation of the novelty values is used to set up a pair of limits, if a novelty value exceeds these limits then it is signaled as being novel and that there is damage in the system. These limits were applied to each set of results and the overall results analyzed.

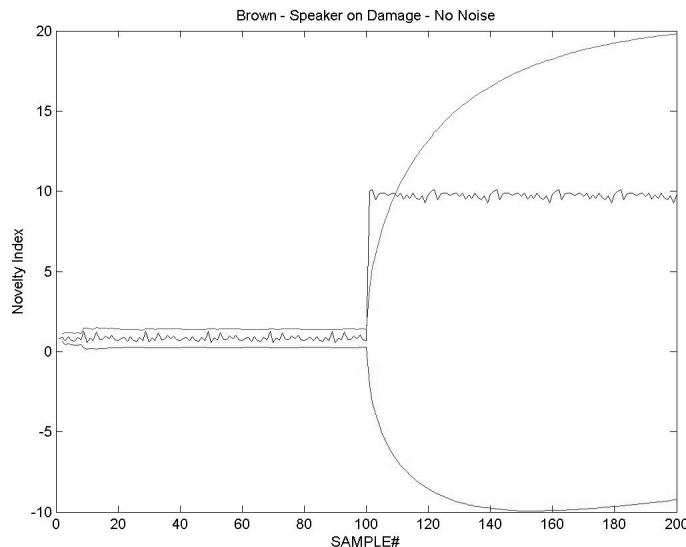


Fig. 4. The results from the detector are shown as a novelty trace. It can be seen that for the undamaged data sets (0-100#) the novelty value is low and within the warning limits, signaling no damage is present. Once the detector is presented with damaged state data (101-200), the novelty value increase, crossing the warning limits hence signaling the presence of a delamination.

The majority of the results obtained showed a noticeable difference between undamaged and the damaged state novelty values. Both microphones performed as well as each other with neither one significantly outperforming the other, although the results from the custom made piezoelectric microphone were noisier in appearance. What was also noticeable in the results was that both experiments worked as well as each other, neither was more successful than the other. Overall each sound input type, microphone type and testing method was able to produce a detector that functioned as required. Some adjustment of the warning limits was required in some places but every detector did work, showing a huge increase in novelty value when shown the damaged state data.

In general the results for the test using brown noise as an input sound proved to be the most consistent, with exceptionally consistent results being observed across all samples (Fig. 4.). The results for the two impact based tests were also consistent with the chirp test results being not as good as the others.

For each microphone type, test method and sound input it was possible to train a novelty detector and use it to detect the presence of damage in the material.

4 Concluding Remarks

The research conducted in this investigation is at best rudimentary. Many possible complications to the testing method are not considered but none the less as a proof of concept the investigation has been successful, it has been shown that it is possible to use the idea of novelty detection to identify the presence of delaminations in composite materials. A range of possible sound input methods have been evaluated and proved to be successful, two variations of using this technique were testing with both producing good quality novelty detectors.

This early research shows that there is potential for this simple testing method to be further adapted and developed to turn it into a fully fledge NDT method. Further work can be done to investigate other possible methods of imparting sound into the test pieces, or other possible methods of testing for damage. The potential for detecting different damage types could also be explored.

Very little attention in the investigation was paid to the way in which the frequency spectra were processed. It is highly likely that there is a better way of extracting relevant frequency features that could be used to detect novelty. It is also noted that the network architecture was also very simple; because the novelty detectors worked well first time no effort was made to change the network architecture to possibly improve the results.

Ultimately though what has been done here has been proven successful, it is possible to use the concept of novelty detection, as well as classification methods, to detect the presence of damage in composite material. Further work could be carried out to improve the detections methods making them more accurate and more versatile.

References

1. Hsu, D.K.: Nondestructive Testing of Composite Structures. In: 17th World Conference on Nondestructive Testing
2. Bray, D.E., Stanley, R.K.: Nondestructive Evaluation – A Tool inDesign, Manufacturing, and Service. CRC Press (1997)
3. Halsshaw, R.: Non-Destructive Testing. Butterworth-Heinemann (1991)
4. Falk, J.P., Steck, J.E., Smith, B.L.: A Nondestructive Testing Technique for Composite Panels Using Tap Test Acoustic Signals and Artificial Neural Networks. *Journal of Smart Engineering System Design* 5(4), 491–506 (2003)
5. Amali, R., Hughes, B.J.: Development of a novel NDT technique for damage classifications of composites. In: FRP Bridges Conference 2012, pp. 89–100. NetComposites, London (2012)
6. Smith, S.W.: Digital Signal Processing. Newnes (2003)
7. Worden, K.: Structural Fault Detection Using a Novelty Measure. *Journal of Sound and Vibration* 201, 85–101 (1997)
8. Farrar, C.R., Worden, K.: An Introduction to Structural Health Monitoring. *Phil. Trans. R. Soc.* 303, 303–315 (2007)
9. Giurgiutiu, V., Soutis, C.: Enhanced Composites Integrity Through Structural Health Monitoring. *Applied Composite Materials* 19, 813–829 (2012)
10. Worden, K., Manson, G.: The Application of Machine Learning to Structural Health Monitoring. *Phil. Trans. R. Soc.* 365, 515–537 (2007)

Impact of Sampling on Neural Network Classification Performance in the Context of Repeat Movie Viewing

Elena Fitkov-Norris¹ and Sakinat Oluwabukonla Folorunso²

¹ Kingston University, Kingston Hill, Kingston-upon-Thames, KT2 7LB, UK

E.Fitkov-Norris@kingston.ac.uk

² Mathematical Sciences Department, Olabisi Onabanjo University (OOU) Ago – Iwoye,

Ogun State, Nigeria

sakinatfolorunso@yahoo.co.uk

Abstract. This paper assesses the impact of different sampling approaches on neural network classification performance in the context of repeat movie going. The results showed that synthetic oversampling of the minority class, either on its own or combined with under-sampling and removal of noisy examples from the majority class offered the best overall performance. The identification of the best sampling approach for this data set is not trivial since the alternatives would be highly dependent on the metrics used, as the accuracy ranks of the approaches did not agree across the different accuracy measures used. In addition, the findings suggest that including examples generated as part of the oversampling procedure in the holdout sample, leads to a significant overestimation of the accuracy of the neural network. Further research is necessary to understand the relationship between degree of synthetic over-sampling and the efficacy of the holdout sample as a neural network accuracy estimator.

Keywords: Over-sampling, under-sampling, neural network, classification.

1 Introduction

Building accurate classifiers for imbalanced data sets is challenging due to the high probability of misclassification of the underrepresented data class. The class imbalance problem corresponds to a problem domain for which one class is represented by a large number of examples, while the other is represented by just a few, the ratio of the small to large classes can be as high as 1:100, 1:1000, or 1:10,000 [1], overwhelming standard classifiers such as decision trees, neural networks and support vector machines, which exhibit a strong bias towards the majority class and ignore the smaller class [2]. This imbalance causes suboptimal classification performance since typical learning algorithms tend to maximize the overall prediction accuracy at the expense of the minority class [1], [3].

Class imbalance occurs naturally in a wide range of domains including medicine, e.g. diagnosing rare diseases, gene mutations or DNA sequencing [4], in engineering, when identifying oil spills in satellite radar images, document retrieval and classification, spam or speech patterns detection [5], [6], or banking and finance, when detecting

fraudulent transactions or assessing risk [7]. Imbalanced data sets also occur in areas where data for the minority class are rare e.g. space shuttle failure or in cases when cost, privacy and the effort required to obtain a representative data set create ‘artificial’ imbalances [1], [7].

The extent to which class imbalance affects classifier learning varies depending on the characteristic of the problem, the degree of class imbalance, the training sample size and the type of classifier used [2], [4], [8]. A large class imbalance has a significant impact on classifier performance only if a classifier is tackling a complex problem or is presented with a small training sample [2]. The type of classifier also has an impact, some classifiers, such as decision trees and support vector machines being very sensitive to class imbalance [2].

Numerous solutions to the class imbalance problem have been proposed both at data and algorithmic levels. The majority are designed for a two-class or binary problem where one class is highly under-represented but associated with a higher identification importance. Solutions at data level attempt to re-balance the class distribution by resampling the data space, while at the algorithm level solutions try to adapt existing classifier learning algorithm to strengthen learning with regards to the minority class [2]. The main advantage of data level techniques is that they are independent of the underlying classifier [3].

A number of data resampling techniques have been suggested to deal with the problem of class imbalance by balancing the distribution of the training data [9]. The most intuitive approach is either to add examples to the minority class (over-sampling) or remove examples from the majority class (under-sampling) [2]. The selection of cases for under- and over-sampling could be performed at random or in a systematic manner, following a predefined rule or objective [8], [10].

A number of studies have evaluated the impact of sampling on decision tree classifiers with mixed findings and empirical evidence is emerging that the best approach could be domain specific [10]. Fewer studies have concentrated on the performance of neural network classifiers in conditions of class imbalance, possibly because they are believed to be more flexible and less likely to be affected by class imbalance problems [2]. In addition, a standard recommendation in neural network training is that duplicate observations are removed from a data set, to minimise the probability of the neural network over-fitting the data, and loosing its ability to generalise [10], [11], thus rendering the random oversampling technique irrelevant. Empirical studies have shown that a large class imbalance in the training dataset has a detrimental effect on neural network performance, in particular when the training sample size is small and random over-sampling has an advantage over under-sampling approaches [12], [13]. Furthermore, the impact of class imbalance on performance of neural networks is less pronounced compared to other types of classifier, although the classification results show significant variance [2].

This article investigates the impact of various sampling techniques on neural network classifier performance in the context of predicting repeat movie going. The problem domain of repeat movie going is naturally imbalanced as only a small proportion of movie goers are likely to see the same movie twice [14]. Identifying repeat movie going accurately is of particular interest to practitioners in the field as movie

revenues are notoriously difficult to predict [15]. Neural networks have been used successfully to build accurate predictor models for movie box office success [16] although the optimal classifier type seems to be domain specific [17]. The use of neural network to predict repeat viewing has been attempted before, and it was found that, surprisingly, neural networks did not offer a significant advantage over parametric approaches such as logistic regression [18]. However, the study did not take into account the imbalanced nature of the data set and the adverse effect it may have on classifier performance. The purpose of this empirical study is to evaluate the impact of different sampling techniques on the predictive accuracy of back-propagation neural network in the context of repeat movie going.

The paper starts with a brief overview of the different over- and under-sampling techniques, followed by an introduction to the data set and research methodology. The experimental results are reported and discussed in section 4 and followed by conclusions.

2 Sampling Techniques

2.1 Under-Sampling Techniques

Random Under-Sampling (RUS). This technique removes instances from the majority class at random, until a desired class distribution is achieved. As it makes no attempt to remove examples “intelligently”, it can discard potentially useful data that could be important for the learning process and make the decision boundary between minority and majority class harder to learn [13].

Condensed Nearest Neighbour Rule (CNN). This technique finds a consistent subset of the original data set which, when used as a reference for the nearest neighbour rule can classify correctly all instances in the original dataset [19]. The main problem with the CNN rule is that it is likely to include a large proportion of noisy examples which are hard to classify and are, therefore, more likely to be included in the training set [20].

Tomek Links (TL). This technique modifies the condensed nearest neighbour technique by retaining only borderline examples within the condensed subset and so reduces computational load. Let E_i, E_j , belong to different classes, and $d(E_i, E_j)$ is the distance between them. A (E_i, E_j) pair is called a Tomek link if there is no example E_l , such that $d(E_i, E_l) < d(E_i, E_j)$ or $d(E_j, E_l) < d(E_i, E_j)$. Examples qualifying as Tomek links are observations that are either borderline or noisy and their removal could improve the decision boundary of the problem [21]. Combinations of Tomek Link and CNN have been suggested, with the aim of utilising the benefits of each approach [5], [8].

Wilson’s Edited Nearest Neighbour Rule (ENN). Wilson proposed an edited k nearest neighbours (k-NN) rule, which consists of two steps. Firstly the k-NN rule is used to edit the set of pre-classified samples by deleting all examples whose class

differs from the majority class of its k-NNs. Afterward, new examples are classified using a 1-NN rule and the reduced reference set derived in step one [22].

Neighbourhood Cleaning Rule (NCL). In this technique, the ENN rule is used to identify and remove majority class noisy examples. For each example (E_i) in the training set, if E_i belong to the majority class and is misclassified by its three Nearest Neighbours (3-NNs), then E_i is removed. If E_i belongs to the minority class and it is misclassified by its 3-NNs from the majority class, then the 3 nearest neighbours are removed. To avoid excessive reduction of small classes, the rule is modified to remove examples misclassified by 2-NN instead of 3-NN [20].

2.2 Over-Sampling Techniques

Random Over-Sampling (ROS). This is the continuous replication of the minority class at random until a more balanced or desired distribution is reached. As mentioned, this approach can increase the likelihood of classifier over-fitting and higher computational load for the classifier [10], [11].

Synthetic Minority Oversampling Technique (SMOTE). This technique generates synthetic examples by operating in feature space rather than data space. The minority class is oversampled by introducing synthetic examples along the line segments joining any/all of k minority class nearest neighbours. This technique overcomes the over-fitting problem and broadens the decision region of the minority class examples, dealing with both relative an absolute imbalance [23].

Advanced Sampling Techniques. A number of approaches combine over-sampling of the minority class, using SMOTE with under-sampling of the majority class by using ENN or TL in order to balance the training dataset and optimise classifier performance [23], [24].

3 Experimental Design

3.1 Data Set Background and Description

The data set used to test the impact of under- and over-sampling consists of the 2002 iteration of the Cinema And Video Industry Audience Research (CAVIAR) survey which identifies the demographic characteristics of cinema-goers and if they had seen a film in the cinema more than once [14]. After removing duplicate observations, the original data set consisted of 786 observations depicting whether an individual visited the cinema to see the same movie twice, their age category, social class, and preference for visiting the cinema. 33% of the entries in the data set were repeat viewers, showing a moderate imbalance ratio of 2.06 [25]. Further details of the data set can be found in [14].

3.2 Neural Networks Overview

Neural networks were developed to simulate the function of the human brain and in particular its ability to handle complex, non-linear pattern recognition tasks efficiently. Neural networks are built from simple processing units or neurons, which enable the network to learn sets of input-output mappings and thus solve classification problems. Each processing unit or neuron consists of three elements: a set of synapses or connecting links which take the input signals, an adder for summing the input signals and an activation function which limits the level of a neuron's output. In addition, each input is allocated a weight of its own, which is adjusted during training and represents the relative contribution of that input (positive or negative) to the overall neuron output. The output function of neuron k can be depicted in mathematical terms as:

$$y_k = \varphi\left(\sum_{j=0}^m w_{kj}x_j\right) \quad (1)$$

where y_k is the output of neuron k , x_j denotes neural network inputs (from 0 to m), w_{kj} denotes the synaptic weight for input j on neuron k and $\varphi(\circ)$ is the neuron activation function. The input for neuron 0 is always +1 and it acts as an overall bias, increasing or decreasing the net output of the activation function.

Multilayer feed-forward neural networks are a subtype of neural network distinguished by the presence of hidden layers of neurons and are particularly well suited to solving complex problems by enabling the network to extract and model non-linear relationships between the input and output layers. Typically, the outputs from each layer in the network act as input signals into the subsequent layer, so the final output layer presents the response of the network to different input patterns. The optimal number of hidden layers is problem specific, and previous research has shown that a feed forward network with one hidden layer is most suited for predicting repeat viewing [18]. Back propagation, essentially a gradient-descent technique and one of the most widely used algorithms will be used to train the network and minimise the error between the target and actual classifications. The network will be simulated using Matlab (Release 2013a) with a tansig activation function and between 6 and 16 neurons in the hidden layer.

3.3 Sampled Data Sets

The under and over-sampling was carried out using the original data set and a subset was created for each of the different sampling approaches outlined above. The class distribution in each data set is shown in Table 1. Random over-sampling was not carried out to avoid neural network over-fitting and an artificially inflated accuracy. The data set consists of binary indicator variables and the HVDM rule, which uses agreement between the values of the nominal/binary variables to determine similarity between observations [11].

Table 1. Sampling Approaches and Resulting Data Sets

| | Sampling Approach | Number of cases (non-repeat viewers : repeat viewers) |
|----------------|-------------------|--|
| Under-sampling | CNN | 298:229 |
| | ENN | 415:79 |
| | NCL | 205:257 |
| | RUS | 257:257 |
| | TL | 269:31 |
| Over-sampling | SMOTE | 529:514 |
| | SMOTE + ENN | 400:346 |
| | SMOTE + TL | 268:235 |
| | SMOTE (300%) | 529:1028 |
| | Original Data | 529:257 |

3.4 Experimental Set Up

To evaluate the performance of the neural network models across a range of different inputs, including new objects that the network has not seen before, it is common practice to use a holdout sample. The data set is split into three subsets: a training sample, a testing sample and a holdout sample [17]. The network learns pattern mappings by minimising the errors on the training set. The testing set is used as a benchmark to prevent over-fitting, while the holdout sample is used as an independent means to test the classification accuracy of the network on a sample of data that it has not seen before (out of sample accuracy). Choosing the holdout sample randomly could lead to a bias in the accuracy estimation due to random sample fluctuations but K-fold cross validation provides an alternative for testing the ability of a neural network to generalise [18]. 10-fold cross validation is well established as a reliable estimate of neural network performance [16]. The 10 subsets are derived at random for each data set and tested using 5 different random seeds. As the number of instances in each data set is different, the original data set is used as a baseline for comparison and, at the end of each training cycle, the network is also tested with the original data file. The classification performance is calculated as the average accuracy across the 10-folds for the holdout samples and the benchmark original data file.

3.5 Performance Measures

One of the most common metrics for measuring classification accuracy for categorical classifiers such as neural networks is the confusion matrix. Various measures, derived from the confusion matrix, including overall classification accuracy, sensitivity and specificity are widely used to assess classifier performance [17].

Overall classification accuracy for a particular categorical classifier is defined as the percentage of correct predictions by the classifier. A common criticism of the overall classification accuracy measure is that it does not take into account class imbalance between different categories/classes and as a result could lead to misleading results since the impact of underrepresented groups would be small [25].

Minimum sensitivity is an alternative measure, which overcomes the problem of imbalanced representation. It is the lower of the sensitivities from the different classes in the problem, in effect the worst performance of the classifier defined as:

$$MS = \min\{P(i); i=1, \dots, J\} \quad \text{where } P(i) = n_{ii} / \sum_{j=1}^J n_{ij} \quad (2)$$

The problem with minimum sensitivity is that it could make direct comparison of results difficult, as the worst performing class is problem specific.

An alternative measure of classification accuracy which overcomes some of the problems of the overall and minimum sensitivity accuracy is the geometric mean (GM) which uses the concept of true positive and true negative classifier accuracy [5], [25]. It is defined as:

$$g = \sqrt{a^+ \times a^-} \quad (3)$$

where a^+ denotes the accuracy in positive examples (or true positive rate and defined as the proportion of correctly classified majority class examples), and a^- is the accuracy in negative examples (or true negative rate and defined as the proportion of correctly classified minority class examples). This study will use the minimum sensitivity and the geometric mean as measures of neural network performance.

4 Results and Discussion

The ranked average classification results of the neural networks using the minimum sensitivity and geometric mean measures on the holdout and benchmark datasets under different under- and over-sampling conditions are shown in Table 2. The performance of all sampling approaches were compared using ANOVA means comparison test, and the same rank was allocated to approaches without statistically significant difference in their mean accuracy performance.

SMOTE oversampling, either on its own, or combined with Tomek Link (TL) or Edited Nearest Neighbour (ENN) techniques, led to higher classification accuracy in predicting both repeat and non-repeat viewers compared to the original data and the under-sampled data. This finding is in line with other empirical studies that concluded that synthetic over-sampling has an advantage over under-sampling approaches [12], [13], in particular studies which identified SMOTE + ENN and SMOTE + TL as robust and reliable over-sampling approaches [8]. Therefore neural network performance can benefit by expanding the problem space (with SMOTE) and the removal of noisy observations using under-sampling of the majority class with ENN or TL.

Two under-sampling approaches performed better than the original data: NCL and ENN. NCL, the best performing under-sampling approach, provided the best prediction accuracy for the repeat viewer minority class (91% accuracy), but this was at the expense of shifting the error to the majority class of non-repeat viewers (26% accuracy). ENN offered one of the best accuracies in predicting the majority class in the benchmark data file (86% accuracy), although this was at the cost of predicting the

minority repeat viewer class (31% accuracy). Both NCL and ENN retained the largest proportion of observations in the data sets they predicted the best (repeat viewer and non repeat viewers, respectively) in comparison to other under-sampling approaches and this suggest that observations that were identified as noisy were in fact essential dimensions of the problem space. This data set may have both absolute and relative imbalance [26] and this explains the superior performance of over-sampling.

Table 2. Neural Network Classification Results

| Sampling Approach | % Non-Repeaters Correct | | % Repeaters Correct | | % Overall Correct | | % Geometric Mean | | Classifier Performance Ranking | | | | Rank Average | |
|-------------------|-------------------------|-------------|---------------------|-------------|-------------------|-------------|------------------|-------------|--------------------------------|-----------|----------------|-----------|--------------|--|
| | | | | | | | | | Minimum Sensitivity | | Geometric Mean | | | |
| | Holdout | Benchmark | Holdout | Benchmark | Holdout | Benchmark | Holdout | Benchmark | Holdout | Benchmark | Holdout | Benchmark | | |
| SMOTE + ENN | 92.2 | 84.2 | 85.7 | 36.9 | 89.1 | 68.8 | 88.7 | 55.5 | 2 | 3.5 | 2 | 2.5 | 2.5 | |
| SMOTE | 81.3 | 63.3 | 64.0 | 60.0 | 72.9 | 62.2 | 71.9 | 61.3 | 5.5 | 1.5 | 4.5 | 1 | 3.13 | |
| SMOTE + TL | 95.0 | 90.2 | 90.4 | 24.7 | 93.0 | 68.8 | 92.5 | 46.6 | 1* | 7.5 | 1 | 5.5 | 3.75 | |
| NCL | 64.4 | 26.1 | 67.8 | 90.5 | 66.1 | 47.2 | 65.4 | 47.7 | 5.5* | 3.5 | 4.5 | 5.5 | 4.75 | |
| ENN | 83.6 | 86.6 | 67.2 | 31.4 | 80.6 | 68.5 | 70.6 | 48.3 | 5.5 | 5 | 4.5 | 5.5 | 5.13 | |
| ORIGINAL DATA | 58.5 | 62.3 | 51.2 | 56.6 | 55.8 | 60.5 | 51.5 | 56.0 | 9* | 1.5 | 8 | 2.5 | 5.25 | |
| SMOTE (300%) | 90.7 | 88.0 | 78.4 | 29.9 | 82.6 | 69.0 | 83.6 | 50.5 | 3 | 9 | 4.5 | 5.5 | 5.5 | |
| TL | 83.7 | 93.9 | 54.7 | 17.8 | 80.4 | 69.0 | 56.9 | 37.0 | 5.5 | 10 | 8 | 10 | 8.38 | |
| RUS | 55.7 | 24.8 | 56.1 | 88.6 | 55.8 | 45.7 | 54.8 | 44.4 | 9* | 7.5* | 8 | 8.5 | 8.5 | |
| CNN | 51.4 | 26.7 | 46.0 | 83.4 | 48.7 | 45.2 | 46.0 | 43.5 | 10 | 7.5* | 10 | 8.5 | 9.5 | |
| Overall Average | 78.6 | 71.5 | 66.9 | 43.2 | 75.7 | 62.2 | 67.0 | 45.6 | | | | | | |

* Denotes minimum sensitivity on the majority class.

The classification accuracy derived using the holdout sample was consistently higher than the accuracy on the benchmark data set. This is particularly true for the predictive accuracy of the minority class. This suggests that even synthetically over-sampled data should be removed from the holdout sample to ensure that it is representative. This recommendation is generally given for random over-sampling which introduces identical examples in the data set and can lead to neural network over-fitting problems [2], but this finding is somewhat counter-intuitive for synthetic oversampling which introduces interleaved copies of the minority class. However, as the representation of the sample data is binary (0 or 1), the small data range and discrete data values give a greater likelihood of introducing examples that are identical to existing observations. This experiment suggests that, it is advisable to remove synthetically interleaved observations from the holdout sample used to test neural network in the case of discrete data with small data range.

The best sampling approach was determined using average rank of its performance; however, there is no significant agreement between the ranking of the models across the two different accuracy measures in the holdout and benchmark data sets and averaging could be masking some very poor performances for some classifiers. For example, the second best ranked sampling approach SMOTE + TL has a relatively weak overall performance and one could argue that it is not suited this data set. Although the optimal way for choosing a classifier is beyond the scope of this work, the findings suggest that the best sampling approach, in the context of repeat viewing data is dependent on the objectives of the classification (overall or minority class).

5 Conclusions and Future Work

This paper assessed the impact of different sampling approaches on neural network classification performance in the context of repeat movie going. The results showed that synthetic oversampling of the minority class, either on its own or combined with under-sampling and removal of noisy examples from the majority class using SMOTE + ENN or SMOTE + TL offered the best performance. The identification of the optimal approach for this data set is not trivial as the recommendations would be highly dependent on the accuracy measure used. The findings also suggest that including examples that were generated by the oversampling procedure in the holdout sample, leads to a significant overestimation of the accuracy of the neural network. It is hypothesised that this is a context specific problem as the data set consisted of indicator variables and so further research would be necessary to understand the relationship between synthetic oversampling and the efficacy of the holdout sample as an estimator of neural network.

References

- Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: Special Issue on Learning from Imbalanced Data Sets. *SIGKDD Explor. Newsl.* 6(1), 1–6 (2004)
- Japkowicz, N., Stephen, S.: The Class Imbalance Problem: A Systematic Study. *Intell. Data. Anal.* 6(5), 429–449 (2002)
- Fernández, A., García, S., Herrera, F.: Addressing the Classification with Imbalanced Data: Open Problems and New Challenges on Class Distribution. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) *HAIS 2011, Part I. LNCS*, vol. 6678, pp. 1–10. Springer, Heidelberg (2011)
- Pearson, R., Goney, G., Shwaber, J.: Imbalanced Clustering of Microarray Time-Series. In: Fawcett, T., Mishra, S. (eds.) *12th International Conference on Machine Learning Workshop on Learning from Imbalanced Datasets II*, Washington DC, vol. 3 (2003)
- Kubat, M., Matwin, S.: Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. In: *14th International Conference on Machine Learning*, Nashville, Tennessee, USA, pp. 179–186 (1997)
- Manevitz, L.M., Yousef, M.: One-Class SVMs for Document Classification. *JMLR* 2, 139–154 (2002)

7. Thai-Nghe, N., Busche, A., Schmidt-Thieme, L.: Improving Academic Performance Prediction by Dealing with Class Imbalance. In: 9th IEEE International Conference on Intelligent Systems Design and Applications, Pisa, Italy, pp. 878–883 (2009)
8. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. SIGKDD Explor. Newsl. 6(1), 20–29 (2004)
9. Folorunso, S.O., Adeyemo, A.B.: Theoretical Comparison of Undersampling Techniques Against Their Underlying Data Reduction Techniques. In: EIE 2nd International Conference Computing, Energy, Networking, Robotics and Telecommunications (EIECON 2012), Lagos, Nigeria, pp. 92–97 (2012)
10. Kotsiantis, S., Kanellopoulos, D., Pintelas, P.: Handling Imbalanced Datasets: A Review. GESTS International Transactions on Computer Science and Engineering 30(1), 25–36 (2006)
11. Zhou, Z.-H., Liu, X.-Y.: Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem. IEEE T. Knowl. Data. En. 18(1), 63–77 (2006)
12. Mazurowski, M.A., Habas, P.A., Zurada, J.M., Lo, J.Y., Baker, J.A., Tourassi, G.D.: Training Neural Network Classifiers for Medical Decision Making: The Effects of Imbalanced Datasets on Classification Performance. Neural Networks 21(2), 427–436 (2008)
13. Crone, S.F., Finlay, S.: Instance Sampling in Credit Scoring: an Empirical Study of Sample Size and Balancing. Int. J. Forecasting 28(1), 224–238 (2011)
14. Collins, A., Hand, C., Linnell, M.: Analyzing Repeat Consumption of Identical Cultural Goods: Some Exploratory Evidence from Moviegoing. J. Cult. Econ. 32(3), 187–199 (2008)
15. Sawhney, M., Eliashberg, J.: A Parsimonious Model for Forecasting Gross Box-Office Revenues of Motion Pictures. Market. Sci., 113–131 (2001)
16. Sharda, R., Delen, D.: Predicting Box-Office Success of Motion Pictures with Neural Networks. Expert Syst. Appl. 30(2), 243–254 (2006)
17. Paliwal, M., Kumar, U.A.: Neural Networks and Statistical Techniques: A Review of Applications. Expert Syst. Appl. 36(1), 2–17 (2009)
18. Fitkov-Norris, E., Vahid, S., Hand, C.: Evaluating the Impact of Categorical Data Encoding and Scaling on Neural Network Classification Performance: The Case of Repeat Consumption of Identical Cultural Goods. In: Jayne, C., Yue, S., Iliadis, L. (eds.) EANN 2012. CCIS, vol. 311, pp. 343–352. Springer, Heidelberg (2012)
19. Hart, P.E.: The Condensed Nearest Neighbor Rule. IEEE T. Inform. Theory 14(3), 515–516 (1968)
20. Laurikkala, J.: Improving Identification of Difficult Small Classes by Balancing Class Distribution. In: Quaglini, S., Barahona, P., Andreassen, S. (eds.) AIME 2001. LNCS (LNAI), vol. 2101, pp. 63–66. Springer, Heidelberg (2001)
21. Tomek, I.: Two Modifications of CNN. IEEE T. Syst. Man. Cyb. 11(6), 769–772 (1976)
22. Wilson, D.L.: Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE T. Syst. Man. Cyb. SMC-2(3), 408–421 (1972)
23. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-Sampling Technique. J. Artif. Intell. Res. 16, 321–357 (2002)
24. Ramentol, E., Caballero, Y., Bello, R., Herrera, F.: SMOTE-RSB*: a Hybrid Preprocessing Approach Based on Oversampling and Undersampling for High Imbalanced Data-Sets Using SMOTE and Rough Sets Theory. Knowl. Inf. Syst. 33(2), 245–265 (2011)
25. García, S., Herrera, F.: Evolutionary Undersampling for Classification with Imbalanced Datasets: Proposals and Taxonomy. Evol. Comput. 17(3), 275–306 (2009)
26. Chen, S., He, H., Garcia, E.A.: RAMOBoost: Ranked Minority Oversampling in Boosting. IEEE T. Neural Netw. 21(10), 1624–1642 (2010)

Discovery of Weather Forecast Web Resources Based on Ontology and Content-Driven Hierarchical Classification

Anastasia Moumtzidou, Stefanos Vrochidis, and Ioannis Kompatsiaris

Centre for Research and Technology Hellas, Information Technologies Institute
`{moumtzid,stefanos,ikom}@iti.gr`

Abstract. Monitoring of environmental information is critical both for the evolution of important environmental events, as well as for everyday life activities. In this work, we focus on the discovery of web resources that provide weather forecasts. To this end we submit domain-specific queries to a general purpose search engine and post process the results by introducing a hierarchical two layer classification scheme. The top layer includes two classification models: a) the first is trained using ontology concepts as textual features; b) the second is trained using textual features that are learned from a training corpus. The bottom layer includes a hybrid classifier that combines the results of the top layer. We evaluate the proposed technique by discovering weather forecast websites for cities of Finland and compare the results with previous works.

Keywords: Environmental, weather forecast, classification, ontology.

1 Introduction

The state of the environment is considered one of the main factors that directly affect the quality of life. For instance, the environmental conditions such as the weather, the air quality and the pollen concentration are strongly related to health issues, as well as to a variety of important human activities (e.g. agriculture). In everyday life, forecasts and observations of environmental information are also of particular interest for outdoor activities such as sports and trip planning. Environmental conditions are usually measured by dedicated stations and the data are, in some cases, made available through web services. However, such services are not many and usually not publicly accessible. Therefore, the main sources of such information are environmental web sites and portals (i.e. environmental nodes). In order to foresee upcoming environmental phenomena (e.g. tsunami, hurricanes) and support people in everyday action planning considering the environmental conditions, we need to provide them with services that combine environmental information from several providers, with a view to generating reliable environmental measurements [1]. The first step towards this direction is the discovery of environmental nodes. In this context, this paper addresses the discovery of weather forecast nodes (Figures 1 and 2), which provide predictions for weather aspects such as temperature, wind speed, humidity and sky condition.



Fig. 1. Screenshot of typical example of weather site (Intermeteo)

| 30.04 Tuesday weather for today | Weather conditions | t°C | Pressure | Rel. Hum. | Wind |
|---------------------------------|-------------------------------------|--------|----------|-----------|------------------|
| Morning | Partly cloudy No Precipitation. | +4..+6 | 747 | 82% | [S-W] 5-7 m/s |
| Day | Mostly Cloudy, Chance of Rain. | +4..+6 | 748 | 82% | [S-W] 6-8 m/s |
| Evening | Mostly Cloudy, Chance of Rain. | +2..+4 | 749 | 94% | [S-W] 3-5 m/s |
| Night | Mostly Cloudy, No Precipitation. | +1..+3 | 751 | 94% | [W] 2-4 m/s |

Fig. 2. Part of the “Intermeteo” site containing interesting information

The discovery of environmental nodes can be considered as a domain-specific search problem. To this end we generate domain-specific queries using an environmental ontology and then we submit them to a general purpose search engine. Subsequently, we employ a hierarchical two layer post-processing module based on supervised classification. The top layer includes two classification models: the first is trained using ontology concepts as textual features, while the second is based on textual features that derive from a training corpus. The bottom layer includes a hybrid classifier that fuses the results of the models in the top layer. We evaluate the proposed technique by discovering weather nodes for several cities of north Europe focusing on Finland and compare with the results of our previous work [2].

The contribution of this paper is the development of a weather forecast-specific search engine that draws upon an environmental ontology, as well as the methodology for domain-specific search, which builds upon a hierarchical scheme that combines ontology and content-driven supervised classification, based on textual features.

This paper is structured as follows: section 2 presents the relevant work, while section 3 describes the environmental ontology. Section 4 introduces the discovery architecture. The results and the evaluation are presented in section 5. Finally, section 6 concludes the paper.

2 Related Work

This work is relevant both to domain-specific search and website classification.

2.1 Domain-Specific Search

In general, domain-specific search refers to the discovery of information relevant to a given domain or topic. The main domain-specific search methodologies contain techniques such as web searching and web crawling and are divided into two categories: a) based on existing web search engines to retrieve a first set of results, which are subsequently filtered and b) based on focused crawling techniques.

In the approaches of the first category, existing general-purpose search engines (e.g. Google Search, Yahoo! Search) are adopted and two query generation approaches are used. In the first approach, a domain oriented query is generated by

applying machine learning techniques to extract terms (called keyword spices). This query is then forwarded to a general-purpose search engine [3]. An alternative methodology to generate such terms [4] employs the notion of context (i.e. domain), in order to automatically construct queries. The query creation mechanism is based upon entity-typed patterns and the expansion is performed using co-occurring terms that are produced from the analysis of previous user queries. In the second approach, the results obtained from the general-purpose search engines by submitting queries consisting of empirical terms that describe the domain, are filtered with post-analysis of the retrieved information [5], [6]. More recently, a work that focuses on discovery of environmental nodes is proposed [2], which combines keyword spices [3], content extraction with KX tool and supervised classification [6].

The second category of methods is based on domain focused crawlers, which are able of retrieving webpages in a directed fashion using machine learning techniques. The proposed work is inspired by [2] and [6], however it substantially differentiates, since it proposes a novel set of features based on an environmental ontology to perform classification and considers a hierarchical classification scheme to filter the results of the general purpose search engine.

2.2 Webpage Classification

Webpage classification is viewed as a supervised learning problem. The classification features can be divided into two classes: a) on-page features, which are directly located on the page, and b) features of neighbours, which are found on the pages linked to the page to be classified [7]. The algorithms using on-page features can be further divided into two categories. The first make use of the textual content, tags and URL, while the second use the visual webpage representation. We focus on works that use on-page textual features, because these features are fairly well developed and relatively inexpensive to use compared to the ones using features of neighbours.

The use of textual content, tags and URL is the most common approach in webpage classification. The procedure includes the following steps: a) feature selection, b) bag-of-words representation of the webpage and c) a learning algorithm. In [8], an approach that is based on the Yahoo! Hierarchy is proposed. The authors in [9] take into account the HTML structure and proposed a tuned linear combination of the text found in the page's title, heading, metadata and body. In [10], webpage classification based on k-Nearest Neighbours is proposed, in which terms in different tags are given different weights. Finally, [11] proposes webpage classification using summarisation.

Our approach applies website classification for domain-specific search by employing a hierarchical scheme, which fuses results from ontology and content-driven classification models, contrary to the proposed approaches, which have used concept hierarchies [8] and only content-driven features (e.g. [10]).

3 Environmental Ontology

In the proposed approach, both the query formulation and the classification make use of an environmental oriented ontology. An ontology is an explicit formal specification

of the concepts in the domain and relations among them. In this work, the environmental ontology developed within the context of the European Project PESCaDO was used¹. The PESCaDO ontology covers environmental content such as meteorological conditions and phenomena, air quality, and pollen, as well as other environment-related content (e.g. human diseases). The ontology contains 215 classes, 157 attributes and properties, 649 individuals and it has been obtained by (i) including customised versions of existing ontologies (e.g., parts of the SWEET ontology [12]), (ii) automatically extracting key concepts from domain relevant sources, (iii) manually adding additional properties [13]. In this work, we focus on the ontology part that describes weather related phenomena such as temperature and wind (Figure 3).

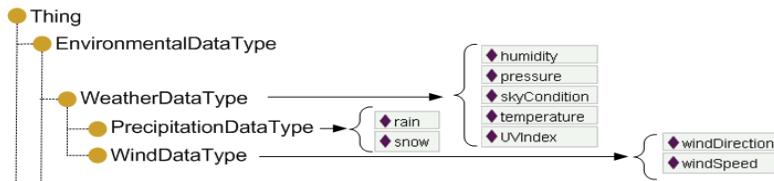


Fig. 3. Part of PESCaDO ontology containing the weather aspects

4 General Framework

The architecture (Figure 4) is based on the first category of methodologies that have been described in section 2.1 (i.e. based on general-purpose search engines) and consists of two parts: The first part (*Web Search*) includes the formulation of weather related queries by combining the ontology concepts with geographical information, which are then fed to a general-purpose search engine. The results are filtered through the second part (*Hierarchical Webpage Classification*), which includes a hierarchical two-layer classification scheme to remove the irrelevant sites. Specifically, in the top layer two models are developed: the first draws upon the concepts of the ontology, while the second uses a corpus derived from learning procedure and takes into account the structure of the webpage. In the bottom layer, the results of both models are used as features to drive a hybrid classification model to provide the final results.

4.1 Web Search

The first step (i.e. web search) comprises the formulation of domain-specific queries. The query (qb_i) is produced by combining one or more members w_i of the set W , which contains weather-specific concepts, with the members g_i of the geographical location set G , which includes city names [2]. Therefore the query format is:

$$qb_i = w_i + g_i \quad (1)$$

¹ <http://www.pescado-project.eu/ontology.php>

An example of such a query is: “weather+Helsinki”. Based on the ontology, we consider the following set of weather-specific concepts:

$$W = \{temperature, sky\ condition, pressure, rain, humidity, wind\}$$

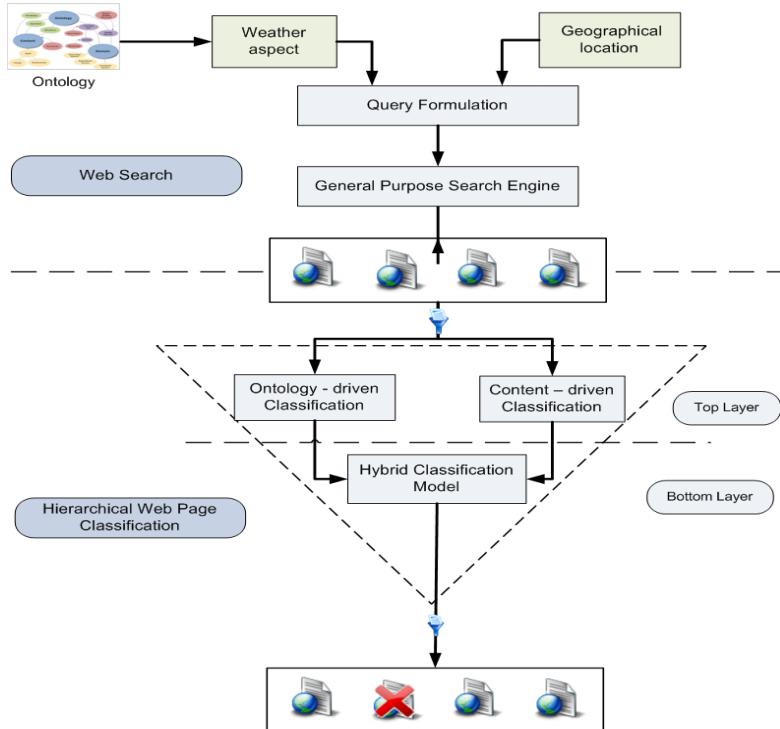


Fig. 4. Weather site discovery architecture showing the web search and the hierarchical web page classification scheme parts

4.2 Hierarchical Webpage Classification

In this section, we describe the filtering procedure applied to the weather sites retrieved in the first step. We introduce a hierarchical two layer post processing scheme that functions as a filter and removes the irrelevant sites to improve the precision performance. Specifically, the top layer includes two classifiers, which are trained with content and ontology-driven textual features, while the bottom layer includes a hybrid classifier that is trained with both the prediction scores (i.e. a vector with two features) of the previous classifiers. In the following we present the textual features used in the top layer and the classification algorithms employed in both layers.

4.3 Textual Features Extraction

The procedure used for extracting textual features information is based on the bag-of-words approach and the *term frequency-inverse document frequency (tf-idf)*. In this work the *tf-idf* is calculated as follows:

$$tf - idf = tf(t, d) * idf(t, d) = f(t, d) * \log\left(\frac{|D|}{|d \in D: t \in d|}\right) \quad (2)$$

where t is a term, d a webpage and D is the total number of webpages. The term frequency $tf(t, d)$ is simplified to the number of occurrences of term t in this webpage d and is denoted by $f(t, d)$. $idf(t, d)$ is the logarithm of the fraction of the total number of webpages D to the number of webpages containing the term t .

In the top layer, two classification models are developed by considering two different types of features (i.e. lexicons):

1. Ontology-driven model: the feature vector of this model consists of 55 terms obtained from the PESCaDO ontology. The terms include several weather aspects and their synonyms (e.g. temperature), words characterizing the rate of the aspects (e.g. cool, hot) and measurement units (e.g. mm/H, °C). Then, each document is represented by a feature vector with the *tf-idf* score of the lexicon terms.
2. Content driven model: the feature vector of this model consists of the terms that occur more frequently in weather forecast websites. These are extracted from a corpus of weather forecast sites. We also take into account the structure of the webpage by multiplying with an experimentally tuned weighting factor the frequency of the keywords embedded in “important” tags of the webpage (i.e. title, headings) [9] and calculate the “weighted *tf-idf* score. Then, each document is represented by a vector with the weighted *tf-idf* score of the lexicon terms.

WebDownload [14] and *Lemur*² were used to perform text processing.

4.4 Classification Algorithm

Support Vector Machines (SVM) are employed for classification due to the fact that they have been applied successfully on several webpage classification problems (e.g. [15]). We employed the LIBSVM [16] library used as kernel the radial basis function:

$$K(f_i, f_j) = e^{-g|f_i - f_j|^2} \quad (3)$$

where f_i are the features of data i , f_j are the support vectors and g is a parameter.

5 Experimental Study

A set of experiments was conducted to evaluate the effectiveness of the proposed framework. Since there is no need to perform an evaluation of an external component

² <http://www.lemurproject.org/>

(i.e. the general purpose search engine), we first evaluate the webpage classification component and then the discovery framework by presenting quantitative results.

5.1 Evaluation of Webpage Classification

The classification step includes model building with manually annotated training sets and testing with the results from “web search” step. During training we employed a manually annotated dataset of 1300 webpages (940 negative and 360 positive) that were retrieved from the general purpose search engine after submitting around 40 weather related queries for the sub-region of Uusimaa (southern Finland). In this experiment, Yahoo! BOSS³ search engine was employed. As already mentioned (section 4) the hierarchical classification scheme is divided into two layers. In the top layer, the following two SVM models were developed with different optimum feature vectors (i.e. lexicons) and training feature vectors: a) Ontology-driven model: the feature vector of this model consists of 55 terms obtained from the PESCaDO ontology as described in section 4.3; b) Content driven model: the feature vector of this model consists of the most frequent terms extracted from a set of 20 typical weather sites. In this case 667 terms were considered. Based on our previous work [2] the large lexicons (around 700 words) outperformed the performance of smaller feature vectors. The frequency of the terms that were embedded in tags such as title, headings was boosted by a weighted factor, which was set experimentally to 3.

For cross-validation purposes, we created five different variations for each of the aforementioned cases by splitting randomly the training and validation set. Table 1 contains the precision, recall, F-score and accuracy average values (i.e. averaging the results of the 5-fold cross validation) of all the models created.

In the top layer the content-driven model performs better than the ontology-driven by reporting an improved F-score of around 9.7% (Table 1). In the bottom layer, we apply late fusion by training an additional SVM model with the results (i.e. the distance from the hyperplane) obtained by the two classifiers of the top layer using the aforementioned data set. The late fusion results are presented in Table 2 and Figure 5, where it is clear that the hybrid model outperforms the classifiers of the top layer. Finally, Table 3 contains the c and g parameters of all the SVM models.

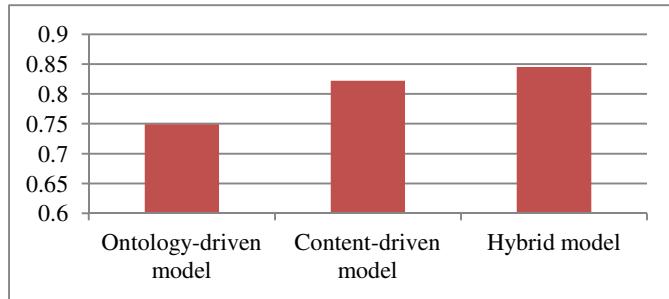
Table 1. Average metrics for the SVM models in the top layer

| Top Layer SVM models | Precision | Recall | Accuracy | F-score |
|-----------------------|-----------|--------|----------|---------|
| Ontology-driven model | 0.873 | 0.658 | 0.879 | 0.749 |
| Content-driven model | 0.930 | 0.736 | 0.912 | 0.822 |

Table 2. Average metrics for the Hybrid SVM classification model in the bottom layer

| Bottom Layer SVM model | Precision | Recall | Accuracy | F-score |
|------------------------|-----------|--------|----------|---------|
| Hybrid model | 0.901 | 0.797 | 0.919 | 0.845 |

³ <http://developer.yahoo.com/boss/search/>

**Fig. 5.** Average F-score metrics for all the models created**Table 3.** SVM parameters used for the different models

| SVM Parameters | Ontology-driven | Content-driven | Hybrid |
|----------------|-----------------|----------------|--------|
| c | 1 | 1 | 1 |
| g | 0.0015 | 0.018 | 0.5 |

5.2 Weather Forecast Discovery Results

Finally, the proposed framework is employed to discover weather forecast nodes for south Finland. The selection of the cities was based exclusively on population criteria and they are the following: Helsinki, Espoo, Vantaa, Porvoo, Lohja, Tuusula, Kirkkonummi, Kerava, Vihti and Sipoo. It should be noted that these cities were not included in the queries submitted during the evaluation and the dataset creation in section 5.1. Overall 100 queries were generated and submitted to Yahoo! BOSS search engine. For each query 40 results were retrieved, which leads to a total of 3224 webpages retrieved after removing the duplicate URLs. Indicative queries that were used for retrieving the URLs of the testing set are the following: humidity+Espoo, "sky+condition"+Espoo. The retrieved results from the general purpose search engine include 968 positive and 2256 negative websites. Thus, the initial precision (i.e. without performing website classification) of the initial results during "Web Search" is 29.5%. Then we apply website classification using the models trained in section 5.1 (the best performing models were selected from the 5 model variations trained during the 5-fold cross-validation). Table 4 contains the precision, recall, accuracy and F-score metrics of the dataset after each model of the hierarchical classification scheme is applied. The precision is improved reaching 90% compared to the initial precision of 29.5%, while still keeping a decent recall (78.7%). As already shown in 5.1, the application of the hybrid model improves the overall performance in terms of F-score. We also present the confusion matrix of the hybrid SVM model used in the classification step including the true and false positives (TP, FP) and the true and false negatives (TN, FN). Finally, in Table 6 we compare the F-score of the proposed approach with the one reported in our previous work [2]. It should be noted that both works aim at weather forecast discovery in the same region. Based on the results, the proposed

approach reports higher F-score compared to both of the techniques (i.e. classification based on multi-word features produced by KX (concept extraction tool) and keyword splices) employed in [2]. The improved results show that the ontology-based queries, as well as the late fusion of content and ontology-driven classification perform better than the single step classification using multi-word key phrases as features.

Table 4. Performance after the post-processing step

| Models | Precision | Recall | Accuracy | F-score |
|-----------------------------------|-----------|--------|----------|---------|
| Ontology-driven model (top layer) | 0.886 | 0.659 | 0.872 | 0.756 |
| Content-driven model (top layer) | 0.941 | 0.739 | 0.907 | 0.828 |
| Hybrid SVM (bottom layer) | 0.898 | 0.787 | 0.909 | 0.839 |

Table 5. Confusion matrix of the hybrid SVM model after the post-processing step

| | | Predicted Class | |
|--------------|-----|-----------------|-----------|
| | | Yes | No |
| Actual Class | Yes | 762 (TP) | 206 (FN) |
| | No | 87 (FP) | 2169 (TN) |

Table 6. F-score comparison with [2]

| Domain-specific approaches | F-score |
|---|---------|
| Hybrid Hierarchical Classification (proposed work) | 0.839 |
| Classification using KX and basic queries [2] | 0.725 |
| Classification with KX and extended queries using keyword splices [2] | 0.795 |

6 Conclusions

In this paper, we propose a domain-specific search framework for discovering weather forecast websites based on a two layer hierarchical supervised classification scheme. Although the selection of Finland and the focus on weather forecast webpages was dictated by the needs of the FP7 project PESCaDO⁴ [17], the same techniques could be applied to deal with other geographical areas and environmental aspects such as air quality and pollen, which are factors that directly affect the quality of human life. Future work includes the discovery of air quality and pollen nodes by considering multimodal information, such as the images contained in the webpages.

Acknowledgments. This work was supported by the project PESCaDO (FP7-248594) funded by the EC.

⁴ <https://www.pescado-project.eu/>

References

1. Epitropou, V., Karatzas, K.D., Bassoukos, A., Kukkonen, J., Balk, T.: A new environmental image processing method for chemical weather forecasts in Europe. In: Proceedings of 5th International Symposium on Information Technologies in Environmental Engineering, Poznan. Springer Series: Environmental Science and Engineering, pp. 781–791 (2011)
2. Moumtzidou, A., Vrochidis, S., Tonelli, S., Kompatsiaris, I., Pianta, E.: Discovery of Environmental Nodes in the Web. In: Proceedings of 5th IRF Conference, Austria, Vienna (2012)
3. Oyama, S., Kokubo, T., Ishida, T.: Domain-Specific Web Search with Keyword Spices. *IEEE Transactions on Knowledge and Data Engineering* 16, 17–27 (2004)
4. Menemenis, F., Papadopoulos, S., Bratu, B., Waddington, S., Kompatsiaris, Y.: AQUAM: Automatic Query Formulation Architecture for Mobile Applications. In: Proceedings of 7th International Conference on Mobile and Ubiquitous Multimedia, MUM 2008, December 3–5. ACM, New York (2008)
5. Chen, H., Fan, H., Chau, M., Zeng, D.: MetaSpider: Meta-Searching and Categorization on the Web. *Journal of the American Society for Information Science and Technology* 52(13), 1134–1147 (2001)
6. Luong, H.P., Gauch, S., Wang, Q.: Ontology-Based Focused Crawling. In: Int. Conference on Information, Process, and Knowledge Management, pp. 123–128 (2009)
7. Qi, X., Davison, B.D.: Web page classification: Features and algorithms. *ACM Comput. Surv.* 41(2), 31 pages, Article 12 (2009)
8. Mladenic, D.: Turning Yahoo into an automatic Web-page classifier. In: Proceedings of the European Conference on Artificial Intelligence, pp. 473–474 (1998)
9. Golub, K., Ardö, A.: Importance of HTML structural elements and metadata in automated subject classification. In: Rauber, A., Christodoulakis, S., Tjoa, A.M. (eds.) ECDL 2005. LNCS, vol. 3652, pp. 368–378. Springer, Heidelberg (2005)
10. Kwon, O.-W., Lee, J.-H.: Text categorization based on k-nearest neighbor approach for Web site classification. *Inform. Process. Manage.* 29(1), 25–44 (2003)
11. Shen, D., Chen, Z., Yang, Q., Zeng, H.-J., Zhang, B., Lu, Y., Ma, W.-Y.: Web-page classification through summarization. In: Proceedings of 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 242–249. ACM Press, New York (2004)
12. Jet Propulsion Laboratory: Semantic Web for Earth and Environmental Terminology (SWEET), <http://sweet.jpl.nasa.gov/ontology/>
13. Rospocher, M., Serafini, L.: An Ontological Framework for Decision Support. In: 2nd Joint International Semantic Technology Conference (JIST 2012), Nara, Japan (2012)
14. Girardi, C.: The HLT Web Manager. FBK Technical Report n. 23969 (2011)
15. Calado, P., Cristo, M., Moura, E., Ziviani, N., Ribeiro-Neto, B., Gonçalves, M.A.: Combining link-based and content-based methods for web document classification. In: Proceedings of 12th International Conference on Information and Knowledge Management, New Orleans, LA, USA, pp. 394–401 (2003)
16. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 27:1–27:27 (2011)
17. Wanner, L., et al.: Personalized environmental service orchestration for quality of life improvement. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H., Karatzas, K., Sioutas, S. (eds.) Artificial Intelligence Applications and Innovations, Part II. IFIP

Towards a Wearable Coach: Classifying Sports Activities with Reservoir Computing

Stefan Schliebs, Nikola Kasabov, Dave Parry, and Doug Hunt^{*}

Auckland University of Technology, New Zealand

{sschlieb, nkasabov, dparry, dphunt}@aut.ac.nz

Abstract. This paper employs a Liquid State Machine (LSM) to classify inertial sensor data collected from horse riders into activities of interest. Since LSM was shown to be an effective classifier for spatio-temporal data and efficient hardware implementations on custom chips exist, we argue that LSM would be relative easy to integrate into wearable technologies. We explore here the general method of applying LSM technology to domain constrained activity recognition using a real-world data set. The aim of this study is to provide a proof of concept illustrating the applicability of LSM for the chosen problem domain.

Keywords: Wearable Computing, Liquid State Machine, Reservoir Computing, Spatio-temporal data processing, Equestrian sport.

1 Introduction

Spatio-temporal data from inertial sensors (accelerometers, gyroscopes, magnetometers) worn on the human body have recently been used as the input to gesture recognition systems [5], orientation tracking systems [19] and activity recognition systems [1].

Unconstrained activity recognition presents a number of challenges including a general lack of overall context in many situations that makes it difficult to distinguish two similar movements (e.g. turning a door knob to open a door and turning a key to start a vehicle). Domain constrained activity recognition, however, has been shown to be more achievable and reliable [6, 18]. Constraining the domain to a particular sport (such as equestrian sport) is also potentially useful, particularly if the rules or traditions of that sport add further activity and style constraints.

We propose to construct a domain constrained, activity classification system to classify the inertial data from wearable sensors, collected from horse riders to recognise activities of interest within equestrian sport.

2 Wearable Coach

In this section we explain the bigger picture of the research presented in this paper and how it fits into the concept of a wearable coach. If successful the

* Corresponding author.

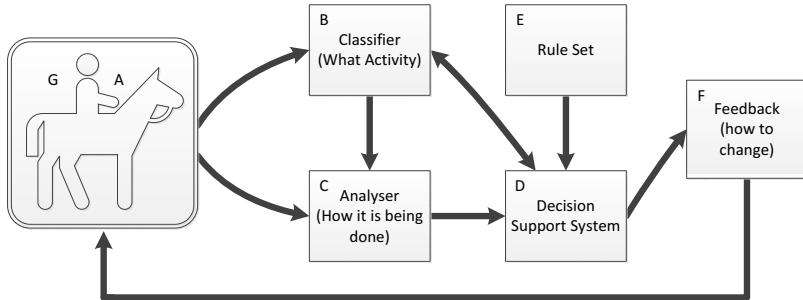


Fig. 1. Proposed concept of a wearable coach

classification system described here will become part of a larger, wearable riding coaching system.

Our concept of a wearable coach is shown in Figure 1 and includes small sensors of various types (A), worn by athletes as they train. The output from the sensors is fed into a classifier (B) that classifies the athlete's current or emerging activity and into an analyser (C) that analyses the athletes performance (or style) of the current activity. The outputs from the analyser is then accumulated into a decision support system (D) that includes predefined rule-sets (E) that decide what feedback may be usefully sent back to the athlete. This feedback system (F) then provides the feedback to the athlete using various on-body methods (G) and the result of the feedback is measured via the sensors (A) the analyser (C) and decision support system (D).

In this study, we concentrate on the classifier and build on earlier work [14] to investigate the suitability of a Liquid State Machine (LSM) [7] as the core of this classification system. LSM technology has been shown to be effective at continuous speech recognition [16] and is implementable on custom chips that enable relative easy integration into wearable technologies. Sensor data streams for speech have some aspects in common with inertial data streams (spatio-temporal, continuous data stream, real time requirements, digitally encoded analogue data). In addition the ability to possibly implement the LSM on chip for wearable applications has further potential benefits.

The aim of this study is to test the applicability of LSM as a reliable classifier for the chosen problem domain using semi-realistic data from real, scripted activities. In the next sections, we first explain the proposed Classifier framework and then provide some experimental results on the continuous classification of a spatio-temporal data set obtained from scripted human activities in a laboratory setting that emulates part of the equestrian sport domain.

3 Experimental Setup

An LSM consists of two main components, a “liquid” (also called reservoir) in the form of a recurrent Spiking Neural Network (SNN) [3] and a trainable readout function. The liquid is stimulated by spatio-temporal input signals causing neural activity in the SNN that is further propagated through the network due to its recurrent topology. Therefore, a snapshot of the neural activity in the reservoir contains information about the current and past inputs to the system.

The function of the liquid is to accumulate the temporal and spatial information of all input signals into a single high-dimensional intermediate state in order to enhance the separability between network inputs. The readout function is then trained to transform this intermediate state into a desired system output.

3.1 Data

Data was recorded from two laboratory and fifty-five real life riding sessions from twenty participants (and their horses) over a three month period using a commercially available, six degrees of freedom inertial sensor from SparkFun [17]. Data collection was done as part of a Masters project [4] and all data was collected within Sweden. Each session was videoed so that activities could be manually classified by participants, riding domain experts and the research team.

The SparkFun 6DoF sensor contains a Freescale MMA7260Q triple-axis accelerometer, two InvenSense IDG300 500° per second gyroscopes and both a Honeywell HMC1052L and a HMC1051Z magnetic sensor. The sensor outputs three axis of magnetic, three axis of accelerometer and pitch, roll and yaw readings based on a 12 bit analogue to digital converter; an unsigned 16 bit serialised sample number and simple start and stop characters (for data validation); giving a total of 12 data fields per sample. Sensor readings were sampled at 10Hz and broadcast via Bluetooth to an on-body receiver for later analysis. Figure 2 shows a horse rider in a typical data collection session.

The data for this set of experiments was taken from one of the two laboratory sessions and during this session the rider wore the sensor on her right wrist using a simple stretchable Velcro bandage for attachment. The “horse” used during the laboratory sessions was a built-for-purpose wooden framed horse of approximately 16 hands in height (163cm at the “shoulder”), draped with a standard European riding saddle and stirrups. During this particular laboratory session the rider mounted and dismounted 17 times.

Figure 3 depicts part of the recorded time series. The three upper panels show the 3-dimensional recordings from the magnetometer, accelerometer and gyroscope, respectively. The bottom panel shows the activity undertaken during recording (mounting/dismounting). The figure shows two mounts and two dismounts.

3.2 Encoding

The time series data obtained from the sensors are presented to the reservoir in the form of an ordered sequence of real-valued data vectors. In order to compute



Fig. 2. Recording of a horse rider mounting a horse

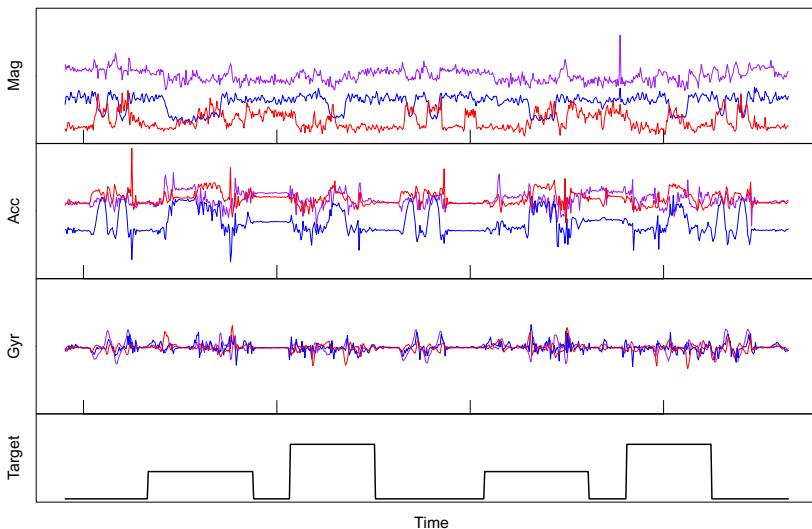


Fig. 3. Inertial sensor data collected from a horse rider in one of two laboratory sessions

an input compatible with the SNN, each real value of a data vector is transformed into a spike train using a spike encoding. In [14], the authors explored two different encoding schemes, namely Ben's Spike Algorithm (BSA) [15] and a population encoding technique. Since only the latter one reported satisfying results on a temporal classification task, we restrict our analysis to this technique.

Population encoding uses more than one input neuron to encode a single time series. The idea is to distribute a single input to multiple neurons, each of them being sensitive to a different range of real values. Our implementation is based on arrays of receptive fields with overlapping sensitivity profiles as described

in [2, 12]. We refer to the mentioned references for further details and examples of this encoding algorithm. As a result of the encoding, input neurons emit spikes at predefined times according to the presented data vectors.

3.3 Reservoir

For the reservoir, we employ the Leaky Integrate-and-Fire neuron with exponential synaptic currents and a dynamic firing threshold [13] along with dynamic synapses based on the short-term plasticity (STP) proposed by Markram et al. [8]. We follow the initiative recently proposed in [11] that promotes reproducible descriptions of neural network models and experiments. The initiative suggests the use of specifically formatted tables explaining neural and synaptic models along with their parametrization. We use the setup outlined in Table 1 for the experiments described below.

We construct a reservoir having a small-world inter-connectivity pattern as described in [7]. A recurrent SNN is generated by aligning 500 neurons in a three-dimensional grid of size $10 \times 10 \times 5$. In this grid, two neurons A and B are connected with a connection probability

$$P(A, B) = C \times e^{\frac{-d(A, B)}{\lambda^2}} \quad (1)$$

where $d(A, B)$ denotes the Euclidean distance between two neurons and λ corresponds to the density of connections which was set to $\lambda = 3$ in our simulations. Parameter C depends on the type of the neurons. We discriminate into excitatory (ex) and inhibitory (inh) neural types resulting in the following parameters for C : $C_{ex-ex} = 0.3$, $C_{ex-inh} = 0.2$, $C_{inh-ex} = 0.4$ and $C_{inh-inh} = 0.1$. The network contained 70% excitatory and 30% inhibitory neurons that were chosen randomly.

3.4 Readout and Learning

In this study, we use the typical analogue readout function in which every spike is convolved by a kernel function that transforms the spike train of each neuron in the reservoir into a continuous analogue signal. We use an exponential kernel with a time constant of $\tau = 50\text{ms}$. The convolved spike trains are then sampled using a time step of 10ms resulting in 500 time series – one for each neuron in the reservoir. In these series, the data points at time t represent the readout for the presented input sample. A very similar readout was used in many other studies, e.g. in [16] for a speech recognition problem.

Readouts were labelled according to their readout time t . If the readout occurred at the time when a sensor signal of interest (e.g. mounting/dismounting the horse) was fed into the reservoir, then the corresponding readout is labelled accordingly. Consequently, a readout belongs to class 0 (base class), if it was obtained during the presentation of a part of the input signal that is of no particular interest.

Table 1. Tabular description of the experimental setup

| Model Summary | |
|------------------|---|
| Neural model | Leaky integrate-and-fire with dynamic firing threshold [13] |
| Synaptic model | Exponential synaptic currents |
| Input | Normalized real-valued data encoded using population encoding |
| Connectivity | Small-world inter-connectivity between reservoir neurons |
| Neural Model | |
| Type Description | <p>Leaky integrate-and-fire (LIF) neuron</p> <p>Dynamics of membrane potential $V(t)$:</p> <ul style="list-style-type: none"> – Spike times: $t^{(f)} : V(t^{(f)}) = \vartheta(t^{(f)})$ – Sub-threshold dynamics: $\tau_m \frac{dV}{dt} = -V(t) + R I^{\text{syn}}(t)$ $\tau_\vartheta \frac{d\vartheta}{dt} = \theta^{\min} - \vartheta(t)$ <ul style="list-style-type: none"> – Reset and refractoriness $\forall f : t \in (t^{(f)}, t^{(f)} + \tau_{\text{ref}})$: $\vartheta(t) \leftarrow \vartheta(t) + \Delta\theta$ $V(t) \leftarrow V_r$ – Exact integration with temporal resolution dt |
| Parameters | <p>Membrane time constant $\tau_m = 30\text{ms}$</p> <p>Membrane resistance $R = 1\text{M}\Omega$</p> <p>Threshold: $\theta^{\min} = 0.5\text{mV}$, $\Delta\theta = 5\text{mV}$, $\tau_\vartheta = 50\text{ms}$</p> <p>Refractory period $\tau_{\text{ref}} = 1\text{ms}$, reset potential $V_r = 0\text{mV}$</p> <p>Time resolution $dt = 0.1\text{ms}$, simulation time $T = 6500\text{ms}$</p> |
| Synaptic Model | |
| Type Description | <p>Current synapses with exponential post-synaptic currents</p> <p>Synapse modelled using Short-term Plasticity (STP) [8]:</p> $\tau_s \frac{dI^{\text{syn}}}{dt} = -I^{\text{syn}}(t)$ $\tau_D \frac{dx_i}{dt} = 1 - x_i$ $\tau_F \frac{du_i}{dt} = U_i - u_i$ <p>Pre-synaptic spike from neuron j to neuron i triggers:</p> $I_i^{\text{syn}} \leftarrow I_i^{\text{syn}} + w_{ji} x_i u_i$ $u_i \leftarrow u_i + U_i(1 - u_i)$ $x_i \leftarrow x_i(1 - u_i)$ |
| Parameters | <p>Synaptic weight $w \in \mathbb{R}$, uniformly initialized in $[-50, 50]\text{nA}$</p> <p>Synaptic time constant $\tau_s = 5\text{ms}$</p> <p>Utilization $U = 0.1$, depression $\tau_D = 500\text{ms}$, facilitation $\tau_F = 200\text{ms}$</p> |

The final step of the LSM framework consists of a mapping from a readout sample to a class label. The general approach is to employ a machine learning

algorithm to learn the correct mapping from the readout data. In fact, since the readout samples are expected to be linearly separable with regard to their class label [7], a comparably simple learning method can be applied for this task. From the labelled readouts, we obtained a ridge regression model for each activity of interest mapping a reservoir readout sample to the corresponding class label. Ridge regression is essentially a regularized linear regression that has been reported to counteract model over-fitting. We experimented with the regularization parameter α and used $\alpha = 10$ for the experiments presented below.

4 Results

Figure 4 shows the outputs obtained from each of the individual processing steps of the LSM framework. Our data consists of a set of nine time series over a time window of 6500ms of simulation time which included 17 occurrences of two alternating mounting and dismounting patterns, cf. Figure 4A. The encoded spike trains (population encoding) derived from the time series are depicted in 4B. The figures show a raster plot of the neural activity of the input neurons over time. A point in these plots indicates a spike fired by a particular neuron at a given time.

The obtained spike trains were then fed into a reservoir resulting in characteristic response patterns of the reservoir neurons, cf. Figure 4C. The reservoir is continuously read out every 10ms of the time simulation using the technique described in section 3.4. Figure 4D shows the readouts over time for the population-encoded reservoir inputs. The colour in these plots indicates the value of the readout obtained from a certain neuron; the brighter the colour, the larger the readout value. The bright horizontal lines in this plot indicate the reservoir neurons that are directly stimulated from the encoded spike trains of the input neurons. The stimulus causes a characteristic readout pattern in which the mount and dismount signals are detectable.

The learning and classification step of the LSM framework is presented in the last plot of Figure 4. We used two ridge regression models, one for learning the mounting and the other for learning the dismounting class. Both models were trained on the first 3250ms of readout data and then tested on the entire set of time series. For the testing, we obtain the output of both regression models and choose the one reporting the larger output in order to decide the class label (winner-takes-all strategy). A sample was considered as correctly classified, if the corresponding model output was larger than a threshold of 0.6 for mount/dismount samples.

The model reported excellent classification on the training data (97.2%) and a satisfying classification accuracy on the testing data (85.1%). Errors usually occurred immediately after the onset of a signal when the reservoir had not yet accumulated sufficient information about the pattern. Considering the short training period, this result is very encouraging. The classification accuracy decreases significantly at the very end of the simulation. Here the horse rider actually removes the sensor from her wrist which is an activity not trained within the system resulting in highly variable model outputs.

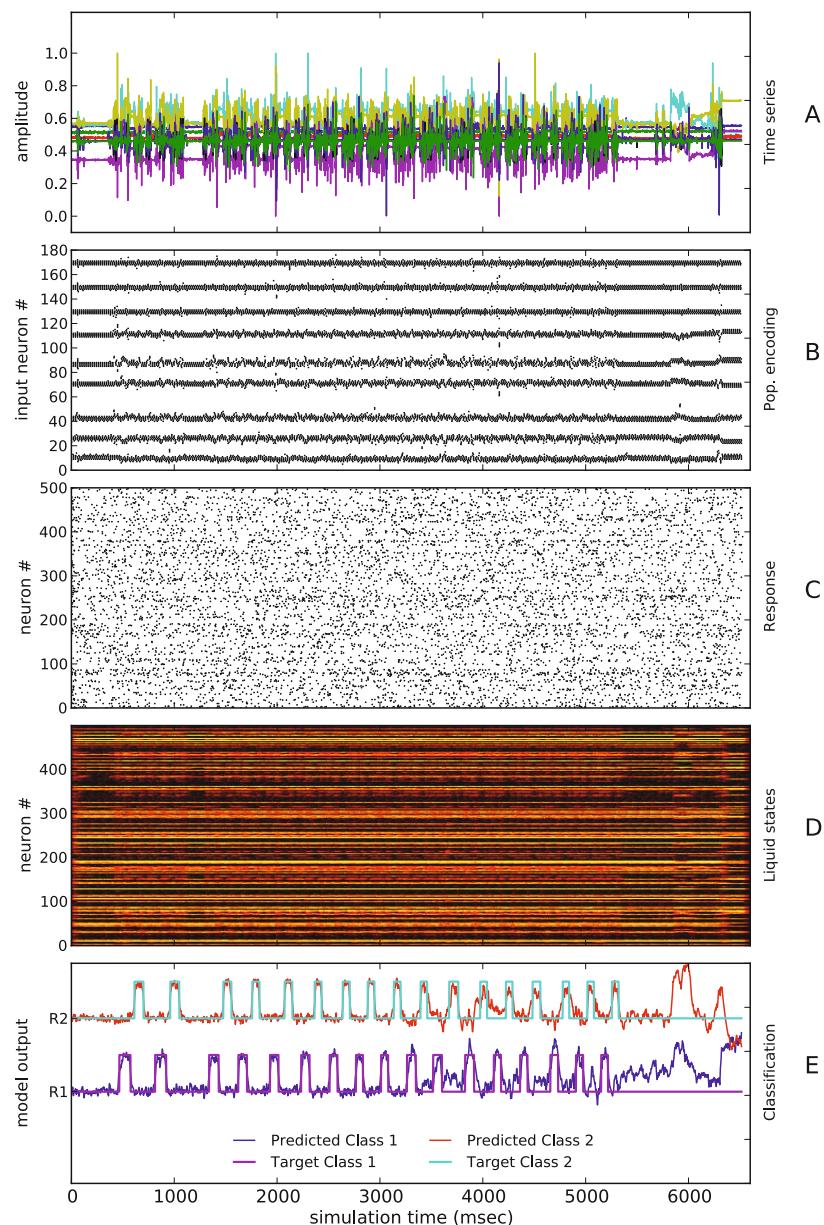


Fig. 4. Experimental results obtained from the horse rider activity recognition. See text for detailed explanations on the figure.

5 Conclusions and Future Directions

In this paper, we have proposed an LSM based, continuous classification method to detect spatio-temporal patterns in inertial sensor data obtained from accelerometers, gyroscopes and magnetometers. We have presented results on a complex real-world data set of scripted movements in the context of Equestrian sports. Despite only having chosen a meaningful default configuration for the large number of parameters of the LSM, the method reported very promising results.

A core part of the method is the simulation of a spiking neural network for which efficient implementations in hardware can be realized [16]. The low energy profile and the real-time processing capabilities of customized hardware are highly desired features of a wearable system of continuous classification of inertial sensor readouts.

Future studies will investigate the applicability of reservoir computing techniques for classifying unscripted human activities within equestrian sport and possibly combining the classifier, decision support and analyser modules into a single SPAN¹ [9, 10] reservoir model that has the sensor data as input and the feedback signals as the output spikes when the desired patterns are recognised.

We are specifically interested in a robust configuration of the reservoir and an efficient implementation on a mobile platform. Contemporary bluetooth enabled mobiles may have sufficient computational resources to allow the simulation of the reservoir and the continuous classification of streamed inertial sensor data in real time.

References

1. Bao, L., Intille, S.S.: Activity recognition from user-annotated acceleration data. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 1–17. Springer, Heidelberg (2004)
2. Bohte, S.M., Kok, J.N., Poutré, J.A.L.: Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing 48(1-4), 17–37 (2002)
3. Gerstner, W., Kistler, W.M.: Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press, Cambridge (2002)
4. Hunt, D.: A heuristic method to distinguish horse rider mounts using a single wrist mounted inertial sensor. Master's thesis, Auckland University of Technology, Auckland, New Zealand (2009)
5. Junker, H., Amft, O., Lukowicz, P., Tröster, G.: Gesture spotting with body-worn inertial sensors to detect user activities. Pattern Recognition 41(6), 2010–2024 (2008)
6. Lukowicz, P., Ward, J.A., Junker, H., Stäger, M., Tröster, G., Atrash, A., Starner, T.: Recognizing workshop activity using body worn microphones and accelerometers. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 18–32. Springer, Heidelberg (2004)

¹ Spike Pattern Association Neuron.

7. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation* 14(11), 2531–2560 (2002)
8. Markram, H., Wang, Y., Tsodyks, M.: Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences* 95(9), 5323–5328 (1998)
9. Mohammed, A., Schliebs, S., Matsuda, S., Kasabov, N.: Method for training a spiking neuron to associate input-output spike trains. In: Iliadis, L., Jayne, C. (eds.) *EANN/AIAI 2011, Part I. IFIP AICT*, vol. 363, pp. 219–228. Springer, Heidelberg (2011)
10. Mohammed, A., Schliebs, S., Matsuda, S., Dhoble, K., Kasabov, N.: Span: Spike pattern association neuron for learning spatio-temporal spike patterns. *International Journal on Neural Systems* 22, 04 (2012)
11. Nordlie, E., Gewaltig, M.O., Plesser, H.E.: Towards reproducible descriptions of neuronal network models. *PLoS Comput. Biol.* 5(8), e1000456 (2009)
12. Schliebs, S., Defoين-Plateau, M., Kasabov, N.: Integrated feature and parameter optimization for an evolving spiking neural network. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) *ICONIP 2008, Part I. LNCS*, vol. 5506, pp. 1229–1236. Springer, Heidelberg (2009)
13. Schliebs, S., Fiasché, M., Kasabov, N.: Constructing robust liquid state machines to process highly variable data streams. In: Villa, A.E.P., Duch, W., Érdi, P., Masulli, F., Palm, G. (eds.) *ICANN 2012, Part I. LNCS*, vol. 7552, pp. 604–611. Springer, Heidelberg (2012)
14. Schliebs, S., Hunt, D.: Continuous classification of spatio-temporal data streams using liquid state machines. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) *ICONIP 2012, Part IV. LNCS*, vol. 7666, pp. 626–633. Springer, Heidelberg (2012)
15. Schrauwen, B., Van Campenhout, J.: BSA, a fast and accurate spike train encoding scheme. In: *Proceedings of the International Joint Conference on Neural Networks*, vol. 4, pp. 2825–2830 (July 2003)
16. Schrauwen, B., D'Haene, M., Verstraeten, D., Campenhout, J.V.: Compact hardware liquid state machines on fpga for real-time speech recognition. *Neural Networks* 21(2-3), 511–523 (2008)
17. SparkFun Electronics Inc: IMU 6 degrees of freedom - v4 with bluetooth capability - SparkFun electronics (2008), <http://www.sparkfun.com/products/8454>
18. Stiefmeier, T., Ogris, G., Junker, H., Lukowicz, P., Troster, G.: Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. In: *2006 10th IEEE International Symposium on Wearable Computers*, pp. 97–104. IEEE (2006)
19. Zhu, R., Zhou, Z.: A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 12(2), 295–302 (2004)

Real-Time Psychophysiological Emotional State Estimation in Digital Gameplay Scenarios

Pedro A. Nogueira¹, Rui Rodrigues², and Eugénio Oliveira²

¹ LIACC – Artificial Intelligence and Computer Science Lab., University of Porto, Portugal

² INESC TEC and Faculty of Engineering, University of Porto, Portugal

{pedro.alves.nogueira, rui.rodrigues, eco}@fe.up.pt

Abstract. Despite the rising number of emotional state detection methods motivated by the popularity increase in affective computing techniques in recent years, they are yet faced with subject and domain transferability issues. In this paper, we present an improved methodology for modelling individuals' emotional states in multimedia interactive environments. Our method relies on a two-layer classification process to classify Arousal and Valence based on four distinct physiological sensor inputs. The first classification layer uses several regression models to normalize each of the sensor inputs across participants and experimental conditions, while also correlating each input to either Arousal or Valence – effectively addressing the aforementioned transferability issues. The second classification layer then employs a residual sum of squares-based weighting scheme to merge the various regression outputs into one optimal Arousal/Valence classification in real-time, while maintaining a smooth prediction output. The presented method exhibited convincing accuracy ratings – 85% for Arousal and 78% for Valence –, which are only marginally worse than our previous non-real-time approach.

Keywords: Affect recognition, regression analysis, affective computing, games, physiology, galvanic skin response, heart rate, electromyography.

1 Introduction

Video games have pioneered the most recent breakthroughs in various computer science fields, such as computer graphics, artificial intelligence and human-computer interaction. Their popularity, along with their considerable emotional influence potential [1], have made them popular study cases for affective computing experiments. However, despite the consecutive advances in gaming technology, there is still a distinct lack of affective experience evaluation tools, which are needed to perform usability tests on traditional applications, but are also a crucial and necessary first step for more complex affective systems. Given their wide applicability, it means that emotional detection systems must not only be sufficiently accurate (depending on each applicational case), but also present a general approach that can easily be adapted to various scenarios, while requiring minimal calibration steps.

Thus, we are interested in how to develop a physiologically-based emotion detection system that provides a lightweight, objective, real-time estimation of users'

emotional states. Throughout this paper we describe the various approaches taken in the literature, along with the process through which we designed the proposed method.

2 Emotional Recognition Techniques

Various taxonomies exist for emotional detection systems and each one has its own dedicated (and in some cases extensive) literature. These taxonomies rely on diverse input signals, such as interaction features, body motion, facial expressions or physiological measures. The latter ones have proven to be the most reliable and adaptable and, as such, are the ones we have based our system on. Unfortunately, there are no rigid guidelines regarding experimental protocol or training material for building these systems and, as such, comparing the results presented in the literature is fundamentally unfeasible since the type of detected emotions, dimensions and scales vary, with each proposed method.

2.1 Physiological Emotion Recognition Techniques

Physiological emotion recognition techniques attempt to explore possible – usually phasic – correlations between game, or otherwise, events and physiological alterations. This type of approach usually employs multiple input modalities for both real and non-real-time applications [2–5]. These techniques can be further divided into model-based and model-free types. While model-based approaches link physiological changes to popular models derived from emotion theories (e.g. Russell's popular Arousal and Valence dimensions [6] or Plutchik's emotion wheel [7]), model-free techniques build their mappings on almost exclusively subjective ground truth annotations. However, systems may instead rely on a hybrid approach by assuming a theoretical model of emotion as their structure and building the mapping functions in a data-driven manner, by asking users to rate their experiences on the emotion model.

Various successful attempts have been made in the field of emotion recognition using the various types of objective modelling techniques aforementioned – although the most popular ones are clearly the hybrid ones. For instance, Chanel [2] was able to classify Arousal using naïve Bayes classifiers and Fisher Discriminant Analysis (FDA), based on Electroencephalogram (EEG), skin conductance (SC), blood volume pressure (BVP), heart rate (HR), skin temperature and respiration rate measures. Complementary to this work, Leon [5] proposes the classification of Valence in three intensity levels, using different measures (SC, its time gradient and derivative, HR and BVP), along with auto-associative neural networks. Also using neural networks, Haag et al. [8] propose employing EMG, SC, skin temperature, BVP, ECG and respiration rates to classify emotional states, reporting encouraging results (89% accuracy for Arousal and 63% for Valence, with a 10% error margin).

Within the proposed line of low calibration approaches, the work by Vinhas et al. [4] proposes a system capable of measuring both Arousal and Valence in real-time, using the subject's SC response and HR derivative. A key factor of this work is that it introduced a continuous classification of Arousal and Valence, thus increasing the

state of the art's maximum granularity. Finally and similar to Vinhas, Mandryk presents an approach based on fuzzy logic that classifies ECG, EMG and SC measurements in terms of both Arousal and Valence [3].

3 Methods

In order to gather enough ground truth data to determine whether we could build an emotion recognition system for our purposes, we conducted a series of controlled experiments with a total of twenty-two healthy participants (15 males and 7 females). Participants ranged from undergraduate students to more senior researchers and were aged 22 to 31 ($M=24.83$, $SD=2.29$). In line with the literature, the applied experimental protocol was initially tested and refined in an iterative prototypical cycle using several pilot studies comprising a total of 10 participants. The results reported in this paper apply to the data collected and processed for the remaining twelve participants in the final iteration of the experimental procedure [2, 4, 5, 8]. As with other studies [2–4], given the population distribution, we limit our findings to this specific demographic. Seven of the participants reported playing video games at least monthly, while the remaining ones reported sporadic activity, whenever big titles came out.

3.1 Experimental Conditions

Sessions were divided into three conditions. The first two conditions were aimed at eliciting extreme Arousal and Valence ratings: the first one being a session of relaxing music and the second playing the horror video game Slenderman, by Parsec Productions. The third one aimed at eliciting neutral to mild reactions using 36 emotionally-charged images from the International Affective Picture System (IAPS) library [9].

In each of the experimental conditions, the participant's SC, facial EMG and BVP readings were recorded. SC was measured at the subject's index and middle fingers using two Ag/AgCL surface sensors snapped to two Velcro straps. BVP was measured at the thumb using a clip-on sensor. Facial EMG was measured at the zygomaticus major (cheek) and the corrugator supercilii (brow) muscles and, as indicated in the literature, correlated with positive and negative Valence, respectively [10].

Sessions were timed and conducted in a room with acoustic insulation, controlled lighting and temperature conditions. Participants were left alone in the room at the beginning of each section. The only human interaction was during the relaxation and briefing periods in between conditions.

3.2 Experimental Procedure and Apparatus

After signing an informed consent form, participants underwent the experimental protocol. Each condition was preceded by a relaxation period of approximately 5 minutes, through which baseline (averaged) values for each channel were extracted. The participants then underwent each of the experimental conditions, whilst reporting their affect ratings. Experimental conditions were sub-divided into its constituent training samples, each with the same length as the event, plus a buffer length of 5 seconds at both its extremities. Regarding training samples, since we were interested

in the participant's lowest emotional activation values for the relaxing music condition, in this case the sample was equal to the whole condition. On the remaining two conditions, each event – images for the IAPS condition and gameplay events for the Slenderman condition – was isolated and independently rated by the participants.

Regarding the annotation procedure, participants rated the training samples immediately after their presentation. The exception to this was the Slenderman condition, since interrupting the gameplay activity to rate each event was not only intrusive; it also implied our physical presence, which could contaminate the experience. As such, for this condition, the gameplay session was recorded by a commercial frame grabber (Fraps) at 30Hz and analysed in conjunction with the participant in a post-gameplay interview. Participants reported their absolute maximum Arousal and Valence ratings in a 10-point Likert scale, ranging from -5 to 5. Since it is harder for individuals to rate their mean affect over a 10 or 20-second time window than to isolate a single, more intense emotional peak, we chose to ask participants to rate each event according to their absolute maximum, as it introduced the least noise in the obtained ratings.

All of the sessions were performed on a MacBook Pro computer running Windows 7. The monitor was a 17" LCD display running at a resolution of 1080p in order to potentiate a higher sensorial immersion. Physiological data was collected using the Nexus-10 hardware by Mind Media. Each condition had an average duration of 10 to 12 minutes, with the exception of the terror videogame, which usually took 15 to 20 minutes. Overall, from setup to debriefing, the experiment had an approximate duration of 2 hours.

3.3 Data Analysis and Feature Extraction

Regarding data analysis and feature extraction, HR, HRV and SC readings were collected at 32Hz, while facial EMG was collected at 2048 Hz. HR and HRV (R-R intervals) readings were computed from the raw BVP readings using the BioTrace+ software suite. All of the physiological signals were then exported to a tab-delimited text file sampled at 32 Hz for future analysis.

The exported raw data was then filtered for anomalies, which were deleted. The exception to this rule was the HR readings, which were filtered using an irregularity detection method that estimated the past variation of the HR signal and only allowed a 25% variation, as described in [9]. HRV was then recomputed from the corrected HR values. Raw SC were corrected by subtracting baseline values and EMG amplitude values were then extracted from the raw EMG readings using the Root-Mean Square procedure. Subsequent signal analysis revealed no additional filtering was necessary. Sensor readings were then smoothed using a moving average filter [3, 10]. HR and HRV were smoothed over a 2-second moving window, SC over a 5-second window and EMG over a 0.125-second window [10].

As previously mentioned, each condition was segmented into several training samples that were independently rated by participants using a 10-point Likert scale. As participants were asked to rate their absolute emotional peaks, so were these values extracted from each channel's training sample. Overall, an average of 230 data points were collected per participant: the minimum values for the relaxing music condition (5 data points, one per channel), 36 samples for the IAPS condition (180 data points),

an average of 6 samples (30 data points) in the terror videogame condition (number of gameplay events varied) and 3 neutral baseline samples (15 data points).

4 Detecting AV States

This section details how the obtained annotated ground truth was used to estimate the participants' Arousal and Valence (AV) states from their physiological data. The developed method categorizes participants' AV ratings through a two-layer classification process (Fig. 1). The first classification layer applies several regression models to each of the four physiological inputs, which allow us to simultaneously normalize these inputs and correlate them to the AV dimensions. The second classification layer then combines the Arousal and Valence ratings obtained from the previous step into one final rating by averaging the models' residual sum of squares error.

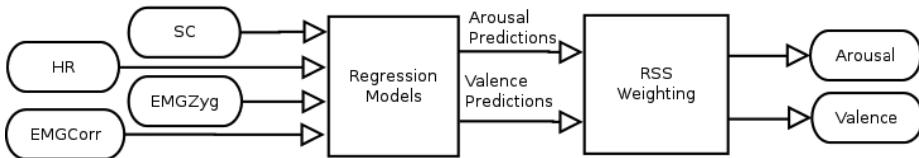


Fig. 1. High-level overview of the proposed system

4.1 Physiological Input Regression Models

One of the most common issues with emotional recognition systems is the difficulty in obtaining an absolute scaling for the reported measures [3–5, 8]. This is due to either *a*) the used material's inability to elicit meaningful Arousal and Valence alterations or *b*) by failing to annotate how each recorded event actually impacted the participant (i.e. assuming that a full emotional range elicitation occurred); usually, a combination of both these factors. Scaling is traditionally addressed by normalising recorded values across participants; a process that introduces a fairly high amount of noise in the data since not only assumes that all participants experienced the same overall emotional ranges, but more importantly assumes all of them experienced these ranges to their fullest extent. To counter this problem we approached it from a different perspective. We addressed insufficient emotional elicitation by exposing the participants to the already discussed wide range of emotional content and, instead of a simple normalisation, explored the correlation functions between each of the physiological channels and the participants' ratings using regression models.

By using the annotated data as our ground truth we aimed at reflecting each participant's characteristic physiological activation functions in their own model and, at the same time, relate them to their corresponding AV dimensions. We proceeded to explore the correlations of each annotated physiological channel to the AV dimensions and – apart from the HRV channel – have confirmed each of the correlations referred in the literature. However, despite HR negatively correlating with Valence, this was not observed for all three experimental conditions, as HR did not significantly fluctuate in the IAPS condition. As such, the results reported in Table 1 for the

HR-Valence correlation refer only to the training samples extracted in the first two experimental conditions, not all three as per the remaining described correlations. In this exploratory phase we used both linear and non-linear (polynomial) models. A regression model is of the form $Y \approx f(X, \beta)$, where $Y=\{\text{Arousal, Valence}\}$ is the dependent variable, $X=\{\text{SC, HR, EMG}_{\text{Zyg}}, \text{EMG}_{\text{Corr}}\}$ is the dependent variable, β is a tuple denoting the unknown parameters and f defines the form of the correlation function. As such, the following model describes the system's first layer:

$$y = f(x, \beta) = \begin{cases} \beta_0 + \beta_1 x_i + \varepsilon_i, & \text{if } (X, Y) = (\text{SC, Arousal}) \\ \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \dots + \beta_n x_i^n + \varepsilon_i, & \text{otherwise} \end{cases}$$

In both cases, ε_i is an error term and the subscript i indexes the particular observation of the sampled physiological metric set being analysed. Given we use third-order polynomials, in this paper $n=3$.

Model complexity was kept in check using bidirectional stepwise regression. This procedure was based on their adjusted-R² values in order to minimise the effects of a large number of predictors on the polynomial models. The final models were re-evaluated using a 10-fold cross-validation scheme, as shown below on Table 1.

Table 1. Ten-fold cross validation fitness values obtained for each regression model that obtained significance ($p<0.05$). Model complexity was controlled using stepwise regression.

| Physiological Channel | AV Space Dimension | Adjusted-R² Model Values (μ, σ) | |
|------------------------------|---------------------------|---|---------------------|
| | | Linear | Polynomial |
| SC | Arousal | 0.90 ± 3.8^{-2} | 0.95 ± 3.0^{-2} |
| HR | Arousal | 0.68 ± 7.1^{-2} | 0.74 ± 8.9^{-2} |
| EMG _{Zyg} | Valence (positive) | 0.84 ± 1.4^{-2} | 0.92 ± 1.6^{-1} |
| EMG _{Corr} | Valence (negative) | 0.83 ± 7.9^{-2} | 0.95 ± 7.5^{-2} |
| HR | Valence | 0.88 ± 1.0^{-1} | 0.96 ± 6.4^{-2} |

Although there are multiple accounts of a linear correlation between SC and Arousal [7-9,10,14], there is no common evidence that any of the remaining collected metrics correlate linearly with any of the AV dimensions. Upon a statistical analysis between the linear and polynomial models we found non-linear correlations are indeed supported by our data. Two-tailed paired t-tests using the models' adjusted-R² values as within-subject conditions revealed statistically significant ($p<0.05$) differences between the linear and polynomial models for: SC-Arousal ($t = -2.397, p = 0.035$), HR-Arousal ($t = -2.393, p = 0.036$), EMG_{Zyg}-Valence ($t = -2.396, p = 0.038$), EMG_{Corr}-Valence ($t = -2.825, p = 0.018$) and HR-Valence ($t = -3.297, p = 0.007$). Upon closer inspection, we found that although the polynomial SC-Arousal models presented a significant improvement over the linear ones, they were marginally different from the latter and only presented a better fit (5% better), while the same ratio on the remaining models presented improved fitness values from 9 to 14%. Thus, we decided to only maintain the linear regression model for correlating SC and Arousal.

4.2 AV Rating Fusion Models

Having obtained the various AV ratings from the regression models in the previous step it becomes then necessary to fuse them in order to obtain a final AV classification. In our previous work [11], we have adopted regression trees for this task. However, doing so limits the real-time usefulness of the method because while regression trees are able to handle continuous inputs, their output is limited to a fixed number of classes that is – at most – equal to the number of leaf nodes in the tree. Increasing the number of predicted classes somewhat attenuates this issue, but does not eliminate it.

Having this in mind, we decided to adopt a weighted voting scheme based on the regression models' error functions (see Fig. 2).

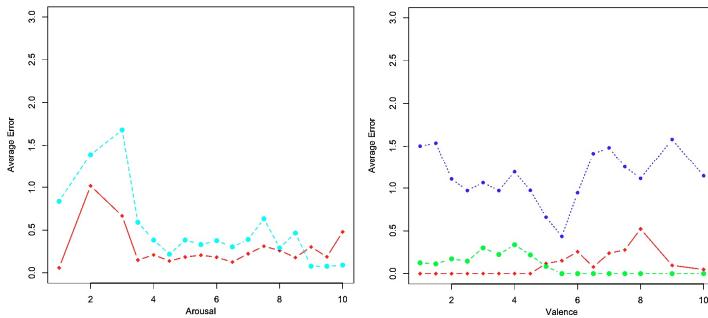


Fig. 2. Average error values for each of the five regression models employed in the first classification layer. *Left panel:* SC (red) shows a much smaller error than HR (blue) in classifying Arousal, except for high Arousal values, perhaps due to the HR's more immediate response. *Right panel:* HR (blue) exhibits much higher relative error than both the EMG channels (Zygomaticus in red and Corrugator in green). However, since EMG values are null a large percentage of the time for some participants, HR presents itself as a potentially vital fallback.

The rationale behind this step was that the weighting scheme would be able to minimise the contribution of correlation models with high error probabilities towards the final AV classification value. For this, we used the normalised residual sum of squares (RSS) as a relative indicator of the model's error along the predicted variable. The normalised residual sum of squares for the set of observed data points $p_m = \{p_{m1}, p_{m2}, \dots, p_{mn}\}$ for a specific regression model m and for a specific value y of the predicted variable Y is given by:

$$RSS(p_m, y) = \frac{\sum_{j=0}^n (y - \hat{y}_{p_{mj}})^2}{n}$$

Where $\hat{y}_{p_{mj}}$ is the model's predicted value for its j^{th} observation of y and n is the number of available observations of y in the observation set p_m . Given the normalised RSS of each model for a particular observation y_i the prediction fusion process assumes the form of a weighted voting function $V(x)$ of the form:

$$V(x) = \frac{\sum_{i=0}^k f(x_i, \beta) \cdot RSS(p_{mi}, f(x_i, \beta))}{\sum_{i=0}^k RSS(p_{mi}, f(x_i, \beta))}$$

Where x is a tuple containing a set of sampled physiological metrics for each AV dimension that are given as input to the correlation functions $f(x_i, \beta)$, and k denotes the number of employed correlation models for estimating either Arousal or Valence. An important aspect of the extracted RSS values is that since each EMG channel was only used to classify either positive or negative Valence, they trivially exhibit symmetrical nil error rates between them. Thus, the voting function was programmed to disregard the EMG channel with a null error value.

In order to evaluate the proposed method's performance a cross-validation approach was adopted. Given that folds were previously computed individually for each participant in the previous layer, these were reused by randomly merging one fold from each participant to generate "population" folds for the 10-fold scheme. Folds for the 3-fold scheme were computed in a similar fashion. While this served no significant computational purpose or gain, it avoided injecting unseen data into the second layer's training samples. Care was also taken to, as much as possible, equally divide the training samples across classes, so as to not bias the classifiers. Each predicted AV rating was evaluated through the following binary thresholding function:

$$E(v) = \begin{cases} 1, & \text{if } |v - y| \leq t \\ 0, & \text{if } |v - y| > t \end{cases}$$

Where v is the predicted AV rating by the weighted voting function $V(x)$, and t is the maximum acceptable error threshold for v to be considered correct. Following the observed error margins in the literature, t was set at values between 0.2 and 1.0. We consider that these values provide a good overview of how the method performs in varying levels of granularity and, as such, represent its adequacy for these scenarios. The detailed results can be examined in Table 2.

Table 2. Accuracy ratings for the AV correlation model fusion process

| Error Threshold (t) in AV points | Arousal Accuracy (%) | | Valence Accuracy (%) | |
|---|----------------------|------------|----------------------|------------|
| | 3 fold CV | 10 fold CV | 3 fold CV | 10 fold CV |
| 0.2 | 73.4 | 83.7 | 74.6 | 69.8 |
| 0.5 | 76.8 | 85.3 | 78.2 | 73.5 |
| 1.0 | 94.7 | 95.9 | 80.3 | 78.8 |

A closer inspection of the depicted results in Table 2 indicates we are able to correctly identify Arousal with accuracy percentages between 73.4% and 85.3%, and Valence between 69.8% and 78.2%, when considering an acceptable error threshold of 0.2 and 0.5 AV points, respectively. These results represent an average accuracy decrease of 4.5% and 5.5% in estimating Arousal and Valence respectively, when compared to our previous results [11]. However, the observed accuracy ratings are still in line with the state of the art and thus highly acceptable for the intended purposes. Furthermore, the added continuous output capabilities that the voting scheme offers enable the method to provide a smooth, continuous emotional state estimation (see Fig. 3), which make it invaluable for real-time scenarios.

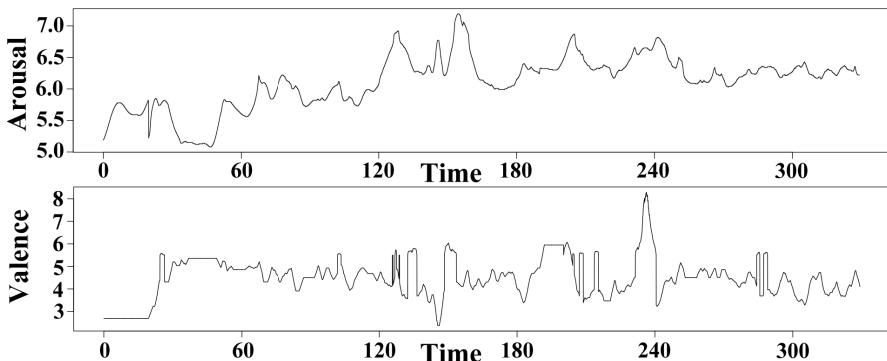


Fig. 3. Continuous ES estimation for a participant over a 300-second epoch. Analysing the mean value can provide insight into how the participant felt over this period, but analysing the continuous time series can also reveal information of his phasic reactions to specific stimuli.

Our second finding relates to the observed correlation indexes between recorded physiological metrics and AV dimensions. Although these metrics showed high correlation indexes across participants and despite doing so for the remaining two conditions, HR did not significantly correlate with Valence in the IAPS images condition. We consider this is possibly due to two factors: 1) their low emotional elicitation potential, and 2) due to the nature of the elicited events. It seems the Slenderman condition was unable to elicit events with both positive Arousal and Valence, thus exhibiting the observed correlation between HR and Valence. The same seems to have occurred in the relaxing music condition as only negative Arousal and positive Valence were elicited. Drachen et al. have previously found this same correlation in first-person shooter videogames [12], and thus we believe it is an interesting parallel research avenue. This finding also has a significant impact on this particular implementation of the system, as it implies that while it is possible to estimate Valence based on HR for the present scenario, the correlation must be verified in future adaptations – possibly outside of the first-person shooter game genre, given Drachen's results.

5 Conclusions

In this paper, we have proposed a data-driven, multi-modal method to interpret psychophysiological measures in a continuous fashion. The obtained results show that we are able to perform this process with adequate accuracy, all the while providing the user with a continuous, real-time estimation of the participants' emotional state.

As emotional detection is not only a critical component of a wider range of affective computing applications, but also a highly complex task, we expect this method will contribute to the standardization of their development guidelines. This will thus translate into a quickening of their implementation cycle, ultimately allowing for a higher percentage of the time to be allocated to the creation of more complex affective computing systems or affect-related experimental studies.

References

1. Ermi, L., Mäyrä, F.: Fundamental components of the gameplay experience: Analysing immersion. In: Digital Games Research Association Conference: Changing Views - Worlds in Play (2005)
2. Chanel, G., Kronegg, J., Grandjean, D., Pun, T.: Emotion assessment: Arousal Evaluation using EEG's and Peripheral Physiological Signals. In: Gunsel, B., Jain, A.K., Tekalp, A.M., Sankur, B. (eds.) MRCS 2006. LNCS, vol. 4105, pp. 530–537. Springer, Heidelberg (2006)
3. Mandryk, R., Atkins, M.: A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies. International Journal of Human-Computer Studies 65, 329–347 (2007)
4. Vinhas, V., Silva, D., Oliveira, E., Reis, L.: Biometric Emotion Assessment and Feedback in an Immersive Digital Environment. Social Robotics, 307–317 (2009)
5. Leon, E., Clarke, G., Callaghan, V., Sepulveda, F.: A user-independent real-time emotion recognition system for software agents in domestic environments. Engineering Applications of Artificial Intelligence 20, 337–345 (2007)
6. Russel, J.A.: A Circumplex Model of Affect. Journal of Personality and Social Psychology 39, 1161–1178 (1980)
7. Plutchik, R.: A General Psychoevolutionary Theory of Emotion. Emotion: Theory, Research, and Experience 1, 3–33 (1980)
8. Haag, A., Goronzy, S., Schaich, P., Williams, J.: Emotion recognition using bio-sensors: First steps towards an automatic system. In: André, E., Dybkjær, L., Minker, W., Heisterkamp, P. (eds.) ADS 2004. LNCS (LNAI), vol. 3068, pp. 36–48. Springer, Heidelberg (2004)
9. Lang, P.J., Bradley, M.M., Cuthbert, B.N.: International affective picture system (IAPS): Affective ratings of pictures and instruction manual. University of Florida, Gainesville (2008)
10. Stern, R.M., Ray, W.J., Quigley, K.S.: Psychophysiological recording. Oxford University Press, New York (2001)
11. Nogueira, P.A., Rodrigues, R., Oliveira, E., Nacke, L.E.: A Regression-based Method for Lightweight Emotional State Detection in Interactive Environments. In: Proceedings of the XVI Portuguese Conference on Artificial Intelligence (EPIA), Angra do Heroísmo, Açores, Portugal (to appear, 2013)
12. Drachen, A., Nacke, L.E., Yannakakis, G., Pedersen, A.L.: Correlation between Heart Rate, Electrodermal Activity and Player Experience in First-Person Shooter Games. In: Proc. of the 5th ACM SIGGRAPH Symposium on Video Games, pp. 49–54 (2010)

Probabilistic Prediction for the Detection of Vesicoureteral Reflux

Harris Papadopoulos¹ and George Anastassopoulos²

¹ Computer Science and Engineering Department, Frederick University,
7 Y. Frederickou St., Palouriotisa, Nicosia 1036, Cyprus
h.papadopoulos@frederick.ac.cy

² Medical Informatics Laboratory, Democritus University of Thrace,
GR-68100, Alexandroupolis, Greece
anasta@med.duth.gr

Abstract. Vesicoureteral Reflux (VUR) is a pediatric disorder in which urine flows backwards from the bladder into one or both ureters and, in some cases, into one or both kidneys. This has potentially very serious consequences as in the case of a Urinary Tract Infection, which is the main symptom of VUR, bacteria have direct access to the kidneys and can cause a kidney infection (pyelonephritis). The principal medical examination for the detection of VUR is the voiding cystourethrogram (VCUG), which is not only a painful procedure, but also demands the exposure of the child to radiation. In an effort to avoid the unnecessary exposure of children to radiation, this study examines the use of a novel machine learning framework, called *Venn Prediction*, for reliably assessing the risk of a child having VUR. Venn prediction is used for obtaining lower and upper bounds for the conditional probability of a given child having VUR. The important property of these bounds is that they are guaranteed (up to statistical fluctuations) to contain well-calibrated probabilities with the only requirement that observations are independent and identically distributed (i.i.d.).

Keywords: Vesicoureteral Reflux, Venn Prediction, Probabilistic Classification, Multiprobability Prediction, Medical Decision Support.

1 Introduction

Vesicoureteral Reflux (VUR) is a pediatric disorder that has potentially very serious consequences as it might lead to a kidney infection. Unfortunately its detection requires a painful medical examination that exposes the child to x-ray radiation or radioactive material. For this reason, in an effort to avoid the unnecessary exposure of children to radiation some recent studies [4–6] examined the use of computational intelligence techniques for the diagnosis of VUR. This work takes one step further and proposes an approach that provides probabilistic outputs rather than the plain yes/no answers of conventional machine learning techniques. This is achieved with the use of a recently developed framework called *Venn Prediction* (VP).

Venn Prediction was proposed in [11] while a detailed description of the framework can be found in [10]. It provides a way of extending conventional classifiers to develop techniques that produce *multiprobability predictions* without assuming anything more than i.i.d. observations. In effect multiprobability predictions are a set of probability distributions for the true classification of the new example, which can be summarized by lower and upper bounds for the conditional probability of each new example belonging to each one of the possible classes for the task in question. The resulting bounds are guaranteed to contain well-calibrated probabilities (up to statistical fluctuations).

Until now the VP framework has been combined with the k -Nearest Neighbour classifier in [10] and [1], with Support Vector Machines in [3, 12], with Logistic Regression in [7] and with Artificial Neural Networks (ANN) in [8, 9]. In this work we apply the Artificial Neural Network Venn Predictor to the problem of detecting VUR based on a dataset consisting of children diagnosed with UTI and further examined with VCUG. The data were collected by the Pediatric Clinical Information System of Alexandroupolis University Hospital, Greece. We follow a slightly modified version of the approach proposed in [8], so as to address the class imbalance problem of the particular dataset. Our experiments demonstrate that the probability bounds produced by Venn Prediction are well-calibrated as opposed to the probabilistic outputs produced by conventional ANN which can be very misleading. Furthermore the proposed approach achieves much higher sensitivity than that of previous studies on the same dataset.

2 Vesicoureteral Reflux Disease

Vesicoureteral Reflux (VUR) is the abnormal flow of urine from the bladder to the upper urinary tract. The urinary tract is the body's drainage system for removing wastes and extra water. The urinary tract includes two kidneys, two ureters, a bladder, and a urethra. Blood flows through the kidneys, and the kidneys filter out wastes and extra water, making urine. The urine travels down the ureters and stored in a balloon-like organ called the bladder. When the bladder empties, urine flows out of the body through a tube called the urethra at the bottom of the bladder.

In VUR, urine may flow back into one or both ureters and, in some cases, to one or both kidneys. VUR is usually diagnosed in infants and children. The disorder increases the risk of Urinary Tract Infections (UTIs), which, if left untreated, can lead to kidney damage. A UTI is a bacterial infection of the urinary tract and may involve the kidney, the bladder or both. The bacteria that cause UTIs are typically in the child's own feces. Even with excellent hygiene, bacteria may gather in the genital area (with no external signs of infection) and ultimately enter the urethra and bladder. If the child has reflux, the bacteria may be transported to the kidney(s) and result in kidney infection. Therefore young children diagnosed with UTI should be further examined for VUR.

The most common method to diagnose VUR is a test called a voiding cystourethrogram (VCUG), which is an X-ray of the bladder. A thin plastic tube

Table 1. VUR clinical and laboratorial parameters together with their values

| No. | Parameter | Possible values | | | | | |
|-------|------------|-----------------|----------------|-------------|-------------|--------------|--------|
| 1 | Sex | Boy | Girl | | | | |
| 2 | Age | < 1 year | 1-5 years | > 5 years | | | |
| 3 | Siblings | 1 | 2 | 3 | | | |
| 4-8 | Systsypm | Fever | Vomit/diarrhea | Anorexia | Weight loss | Others | |
| 9 | WBC | < 4500 | 4500-10500 | > 10500 | | | |
| 10 | WBC type | n | L | m | E | b | |
| 11 | Ht | < 37 | 37-42 | > 42 | | | |
| 12 | Hb | < 11.5 | 11.5-13.5 | > 13.5 | | | |
| 13 | PLT | < 170 | 170-450 | > 450 | | | |
| 14 | ESR | < 20 | 20-40 | > 40 | | | |
| 15 | CRP | + | - | | | | |
| 16 | Bacteria | E.coli | Proteus | Kiebsielas | Strep | Stapf | Psedom |
| 17-22 | Sensitiv | Penicillin | Kefalosp2 | Kefalosp3 | Aminoglyc | Sulfonamides | Other |
| 23 | Ultrasound | Rsize nrm | Rsize abn | Rstract nrm | Rstract abn | Normal | Other |
| 24 | Dursymp | 2 days | 3 days | 4 days | 5 days | > 5 days | |
| 25 | Starttre | 2 days | 3 days | 4 days | 5 days | | |
| 26-27 | Riskfact | Age < 1 year | | Ttreat | | | |
| 28 | Collect | U-bag | Catheter | Suprapubic | | | |
| 29-34 | Resistan | Penicillin | Kefalosp2 | Kefalosp3 | Aminoglyc | Sulfonamides | Other |

called a catheter is inserted into the urethra. Fluid containing an X-ray dye is injected through the tube until the bladder is full, and then the child is asked to urinate. Pictures of the bladder are taken to see if the dye goes backward up to one or both kidneys. The VCUG usually takes 15 to 20 minutes. In some instances, the test is performed with fluid containing a tiny amount of radioactive tracer and the test is monitored with a special camera. Another method for VUR diagnosis is a Radionuclide Cystogram (RNC). This is a type of nuclear scan that involves placing radioactive material into the bladder. A scanner then detects the radioactive material as the child urinates or after the bladder is empty. RNC is more sensitive than VCUG but does not provide as much detail of the bladder anatomy. Also, Abdominal ultrasound can be used for VUR diagnosis. An abdominal ultrasound can create images of the entire urinary tract, including the kidneys and bladder. Ultrasound may be used before VCUG or RNC if the child's family or health care provider wants to avoid exposure to x-ray radiation or radioactive material. Based on above methods, VUR can be classified into five grades - grade 1 being the least and grade 5 is the worst.

The VUR data used in this paper, were obtained from the Pediatric Clinical Information System of Alexandroupolis University Hospital, Greece. The dataset consists of 162 child patients with UTI, of which 30 were diagnosed with VUR. The clinical and laboratorial parameters that were considered for VUR diagnosis were 19, namely: sex, age, number of siblings, clinical presentation (systsypm), white blood cell count (WBC), WBC type, haematocrit (Ht), haemoglobin (Hb), platelets (PLT), Erythrocyte Sedimentation Rate (ESR), C-Reactive Protein (CRP), bacteria, sensitivity, ultrasound, symptoms duration (Dursymp), start of treatment (Starttre), risk factor (Riskfact), collect and resistance. A list

of these parameters and their values is given in Table 1. It is emphasized that some of the parameters may take more than one values simultaneously. These parameters were transformed to a binary set of sub-parameters, one for each of their possible values, and for this reason they have a range in the first column of Table 1. As a result the total number of parameters was extended to 34.

Due to the large number of parameters and the relatively small number of cases, feature selection was applied to the data so as to avoid overfitting. Specifically, Correlation-based Feature Subset Evaluation [2] was used in conjunction with Best-first search, which resulted in the selection of the features 11, 21, 22, 23 and 27. These were the features used for all experiments of this work.

3 The Venn Prediction Framework

This section gives a brief description of the Venn prediction framework; for more details the interested reader is referred to [10]. We are given a training set $\{(x_1, y_1), \dots, (x_l, y_l)\}$ of examples, where each $x_i \in \mathbb{R}^d$ is the vector of attributes for example i and $y_i \in \{Y_1, \dots, Y_c\}$ is the classification of that example. We are also given a new unclassified example x_{l+1} and our task is to predict the probability of this new example belonging to each class $Y_j \in \{Y_1, \dots, Y_c\}$ based only on the assumption that all $(x_i, y_i), i = 1, 2, \dots$ are i.i.d.

The Venn Prediction framework assigns each one of the possible classifications $Y_j \in \{Y_1, \dots, Y_c\}$ to x_{l+1} in turn and generates the extended set

$$\{(x_1, y_1), \dots, (x_l, y_l), (x_{l+1}, Y_j)\}. \quad (1)$$

For each resulting set (1) it then divides the examples into a number of categories and calculates the probability of x_{l+1} belonging to each class Y_k as the frequency of Y_k 's in the category that contains it.

To divide each set (1) into categories it uses what is called a *Venn taxonomy*. A Venn taxonomy is a measurable function that assigns a category $\kappa_i^{Y_j}$ to each example z_i in (1); the output of this function should not depend on the order of the examples. Every Venn taxonomy defines a different Venn Predictor. Typically each taxonomy is based on a traditional machine learning algorithm, called the *underlying algorithm* of the VP. The output of this algorithm for each attribute vector $x_i, i = 1, \dots, l + 1$ after being trained either on the whole set (1), or on the set resulting after removing the pair (x_i, y_i) from (1), is used to assign $\kappa_i^{Y_j}$ to (x_i, y_i) . For example, a Venn taxonomy that can be used with every traditional algorithm assigns the same category to all examples that are given the same classification by the underlying algorithm. The Venn taxonomy used in this work is defined in the next section.

After assigning the category $\kappa_i^{Y_j}$ to each example (x_i, y_i) in the extended set (1), the empirical probability of each classification Y_k among the examples assigned $\kappa_{l+1}^{Y_j}$ will be

$$p^{Y_j}(Y_k) = \frac{\left| \{i = 1, \dots, l + 1 | \kappa_i^{Y_j} = \kappa_{l+1}^{Y_j} \& y_i = Y_k\} \right|}{\left| \{i = 1, \dots, l + 1 | \kappa_i^{Y_j} = \kappa_{l+1}^{Y_j}\} \right|}. \quad (2)$$

This is a probability distribution for the label of x_{l+1} . After assigning all possible labels to x_{l+1} we get a set of probability distributions that compose the multiprobability prediction of the Venn predictor $P_{l+1} = \{p^{Y_j} : Y_j \in \{Y_1, \dots, Y_c\}\}$. As proved in [10] the predictions produced by any Venn predictor are automatically valid multiprobability predictions. This is true regardless of the taxonomy of the Venn predictor. Of course the taxonomy used is still very important as it determines how efficient, or informative, the resulting predictions are. We want the diameter of multiprobability predictions and therefore their uncertainty to be small and we also want predictions to be as close as possible to zero or one.

The maximum and minimum probabilities obtained for each class Y_k define the interval for the probability of the new example belonging to Y_k :

$$\left[\min_{k=1,\dots,c} p^{Y_j}(Y_k), \max_{k=1,\dots,c} p^{Y_j}(Y_k) \right]. \quad (3)$$

If the lower bound of this interval is denoted as $L(Y_k)$ and the upper bound is denoted as $U(Y_k)$, the Venn predictor finds

$$k_{best} = \arg \max_{k=1,\dots,c} \overline{p(Y_k)}, \quad (4)$$

where $\overline{p(Y_k)}$ is the mean of the probabilities obtained for Y_k , and outputs the class $\hat{y} = Y_{k_{best}}$ as its prediction together with the interval $[L(\hat{y}), U(\hat{y})]$ as the probability interval that this prediction is correct. The complementary interval $[1 - U(\hat{y}), 1 - L(\hat{y})]$ gives the probability that \hat{y} is not the true classification of the new example and it is called the *error probability interval*.

4 Artificial Neural Networks Venn Prediction with Minority Oversampling

This section describes the Venn Taxonomy used in this work, which is based on ANN, and gives the complete algorithm of the proposed approach. The ANNs used were 2-layer fully connected feed-forward networks with tangent sigmoid hidden units and a single logistic sigmoid output unit. They were trained with the variable learning rate backpropagation algorithm minimizing cross-entropy error. As a result their outputs can be interpreted as probabilities for class 1 and they can be compared with those produced by the Venn predictor. Early stopping was used based on a validation set consisting of 20% of the corresponding training set. In order to address the class imbalance problem of the data, before training the examples belonging to the minority class of the training set were oversampled so as to train the ANN with an equal number of positive and negative examples.

After adding the new example (x_{l+1}, Y_j) to the oversampled training set and training the ANN, the output o_i produced by the ANN for each input pattern x_i is used to determine its category κ_i . Specifically the range of the ANN output $[0, 1]$ is split to a number of equally sized regions λ and the same category is assigned to the examples with output falling in the same region. In other words each one of these λ regions defines one category of the taxonomy.

Using this taxonomy we assign a category κ_i^0 to each example (x_i, y_i) in the oversampled training set extended with $(x_{l+1}, 0)$ and a category κ_i^1 to each example (x_i, y_i) in the oversampled training set extended with $(x_{l+1}, 1)$. We then follow the process described in Section 3 to calculate the outputs of the Artificial Neural Network Venn Predictor with Minority Oversampling (ANN-VP with MO), which is presented in Algorithm 1.

Algorithm 1. ANN-VP with MO

Input: training set $\{(x_1, y_1), \dots, (x_l, y_l)\}$, new example x_{l+1} , number of categories λ .

Oversample the minority class generating the training set $\{(x_1, y_1), \dots, (x_n, y_n)\}$, where $n > l$;

Denote the new example x_{l+1} by x_{n+1} ;

for $k = 0$ **to** 1 **do**

- Train the ANN on the extended set $\{(x_1, y_1), \dots, (x_n, y_n), (x_{n+1}, k)\}$;
- Supply the input patterns x_1, \dots, x_{n+1} to the trained ANN to obtain the outputs o_1, \dots, o_{n+1} ;
- for** $i = 1$ **to** $n + 1$ **do**

 - Assign κ_i to (x_i, y_i) based on the output o_i ;

- end**
- $p^k(1) := \frac{|\{i=1, \dots, n+1 | \kappa_i^k = \kappa_{n+1}^k \& y_i = 1\}|}{|\{i=1, \dots, n+1 | \kappa_i^k = \kappa_{n+1}^k\}|}$;
- $p^k(0) := 1 - p^k(1)$;
- end**
- $L(0) := \min_{k=0,1} p^k(0)$; and $L(1) := \min_{k=0,1} p^k(1)$;

Output:

Prediction $\hat{y} = \arg \max_{j=0,1} L(j)$;

The probability interval for \hat{y} : $[\min_{k=0,1} p^k(\hat{y}), \max_{k=0,1} p^k(\hat{y})]$.

5 Experimental Results

We performed experiments in both the on-line setting and in the batch setting. The former demonstrate the empirical validity of the multiprobability outputs produced by the ANN-VP, while the latter examine its performance mainly in terms of its accuracy.

Before each training session all attributes were normalised setting their mean value to 0 and their standard deviation to 1. For all experiments with conventional ANN minority oversampling was performed on the training set in the same way as for the proposed approach. The number of categories λ of ANN-VP was set to 6, which seems to be a good choice for small to moderate size datasets [8].

For finding the best number of hidden units to use we tested the values from 3 to 20 following a 10-fold cross-validation process with the original ANNs. This experiment was repeated 10 times with different divisions of the dataset into the

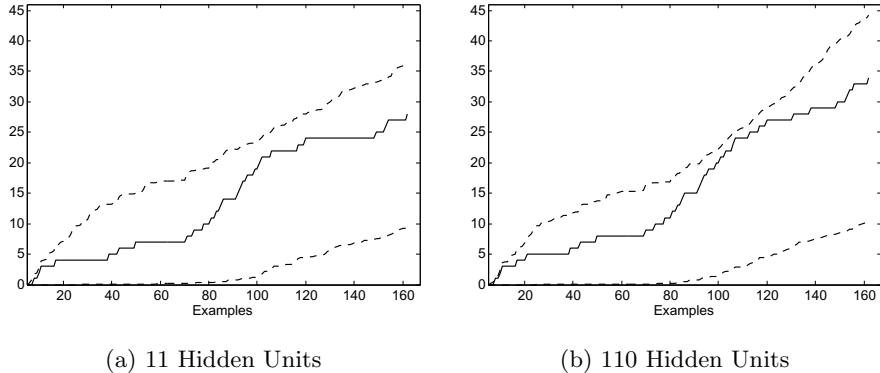


Fig. 1. On-line performance of ANN-VP with MO on the two datasets. Each plot shows the cumulative number of errors E_n with a solid line and the cumulative lower and upper error probability curves LEP_n and UEP_n with dashed lines.

10 folds. The choice of 11 hidden units gave the best mean accuracy. This process involves data snooping since the whole dataset was used to select the number of hidden units, but this is in favour of the original ANN classifier, to which we compare, rather than the proposed approach. Furthermore, the divisions into folds used in our batch setting experiments were created with different random seeds than the ones used for this purpose.

5.1 On-Line Setting Results

This subsection presents the results obtained when applying the ANN-VP and conventional ANN approaches (with minority oversampling) in the on-line setting. Specifically, in this setting each experiment started with an initial training set of 5 examples and one by one the remaining 157 examples were predicted in turn and immediately after prediction their true classification was revealed and they were added to the training set for predicting the next example. In order to demonstrate that the choice of hidden units does not affect the validity of the resulting probabilistic outputs, we performed this experiment not only with 11 hidden units, but also with ten times this number. The results are presented in Figure 1 in the form of the following three curves for each experiment:

- the cumulative error curve

$$E_n = \sum_{i=1}^n err_i, \quad (5)$$

– where $err_i = 1$ if the prediction \hat{y}_i is wrong and $err_i = 0$ otherwise,
 – the cumulative lower error probability curve

$$LEP_n = \sum_{i=1}^n 1 - U(\hat{y}_i) \quad (6)$$

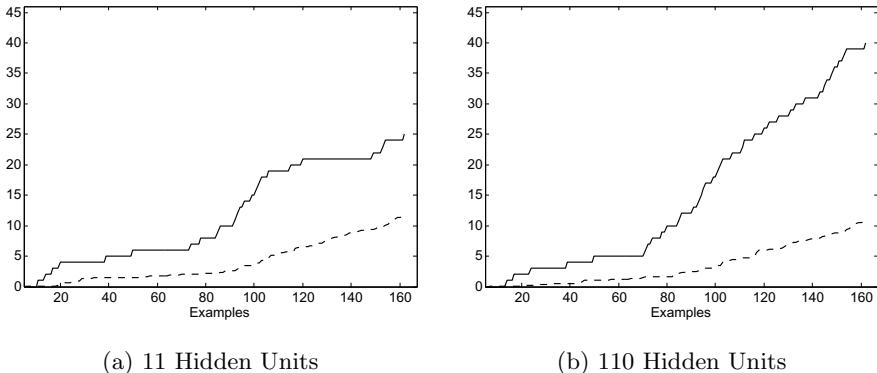


Fig. 2. On-line performance of the original ANN classifier on the two datasets. Each plot shows the cumulative number of errors E_n with a solid line and the cumulative error probability curve EP_n with a dashed line.

– and the cumulative upper error probability curve

$$UEP_n = \sum_{i=1}^n 1 - L(\hat{y}_i). \quad (7)$$

The two plots confirm that the probability intervals produced by ANN-VP are well-calibrated since the cumulative errors are always included inside the cumulative upper and lower error probability curves. Note that although the use of a much larger number of hidden units generates more errors, the bounds of the VP remain well-calibrated.

The same experiment was performed with the original ANN classifier and two analogous plots are displayed in Figure 2. In this case the cumulative error curve (5) is plotted together with the cumulative error probability curve

$$EP_n = \sum_{i=1}^n |\hat{y}_i - \hat{p}_i|, \quad (8)$$

where $\hat{y}_i \in \{0, 1\}$ is the ANN prediction for example i and \hat{p}_i is the probability given by ANN for example i belonging to class 1. In effect this curve is a sum of the probabilities of the less likely class for each example according to the ANN. One would expect that this curve would be near the cumulative error curve if the probabilities produced by the ANN were well-calibrated. The plots of Figure 2 show that this is not the case. The ANNs underestimate the true error probability in both cases since the cumulative error curve is much higher than the cumulative error probability curve. To check how misleading the probabilities produced by the ANN are, the 2-sided p -value of obtaining the resulting total number of errors E_N with the observed deviation from the expected errors EP_N given the

Table 2. Results of ANN-VP with MO in the batch setting and their comparison with those of conventional ANN and those of the best previously proposed approach

| | Accuracy | Sensitivity | Specificity |
|----------------------|----------|-------------|-------------|
| ANN-VP with MO | 84.07% | 72.00% | 86.89% |
| Original ANN with MO | 82.78% | 68.33% | 86.06% |
| PNN 3 from [4] | 85.56% | 53.00% | 92.95% |

probabilities produced by the ANN was calculated for each case. The resulting p -values were 0.000011792 with 11 hidden units and much smaller with 110 hidden units. This shows how misleading the probabilities produced conventional ANNs are, as opposed to the well-calibrated bounds produced by VPs.

5.2 Batch Setting Results

This subsection examines the performance of the proposed approach in the batch setting and compares its results with those of conventional ANN and those of the best approach developed in previous studies on the same data. Specifically we compare it to the third PNN proposed in [4] (we actually tried out all five PNNs of the same study and this was the one that gave the best results).

For these experiments we followed a 10-fold cross-validation process for 10 times with different divisions of the dataset into the 10 folds and the results reported here are the mean values over all runs. Table 2 reports the accuracy, sensitivity and specificity of the three techniques, which are typical metrics for assessing the performance of classifiers. The values reported here show that the proposed approach did not have much lower accuracy than that of the PNN. Nevertheless it had the highest sensitivity, which is of great importance for the particular problem, with quite a significant difference especially from that of the PNN. It should be noted of course that the key advantage of Venn prediction is not its better performance, but the important additional information it provides.

6 Conclusions

This paper applied a Venn Predictor based on ANN and minority oversampling to the problem of VUR detection. Unlike conventional classifiers the proposed approach produces lower and upper bounds for the conditional probability of each child having VUR, which are valid under the general i.i.d. assumption. Our experimental results in the on-line setting demonstrate that the probability bounds produced by ANN-VP with MO are well-calibrated regardless of the number of hidden units used in the underlying ANN. On the contrary, the single probabilities produced by conventional ANN can be very misleading. Moreover, the comparison performed in the batch setting shows that the proposed approach has much higher sensitivity than the other techniques. The better sensitivity is partly due to the use of minority oversampling and partly due to the way Venn

Prediction computes its predictions, since it also improves over the sensitivity of the original ANN, for which minority oversampling was also used.

The main directions for future work include the exploration of more sophisticated techniques for handling the class imbalance problem of the data and experimentation with VPs based on other conventional classifiers.

References

1. Dashevskiy, M., Luo, Z.: Reliable probabilistic classification and its application to internet traffic. In: Huang, D.-S., Wunsch II, D.C., Levine, D.S., Jo, K.-H. (eds.) ICIC 2008. LNCS, vol. 5226, pp. 380–388. Springer, Heidelberg (2008)
2. Hall, M.A.: Correlation-based Feature Subset Selection for Machine Learning. Ph.D. thesis, University of Waikato, Hamilton, New Zealand (1998)
3. Lambrou, A., Papadopoulos, H., Nouretdinov, I., Gammerman, A.: Reliable probability estimates based on support vector machines for large multiclass datasets. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H., Karatzas, K., Sioutas, S. (eds.) AIAI 2012, Part II. IFIP AICT, vol. 382, pp. 182–191. Springer, Heidelberg (2012)
4. Mantzaris, D., Anastassopoulos, G., Adamopoulos, A.: Genetic algorithm pruning of probabilistic neural networks in medical disease estimation. *Neural Networks* 24(8), 842–851 (2011)
5. Mantzaris, D., Anastassopoulos, G., Iliadis, L., Tsalkidis, A., Adamopoulos, A.: A probabilistic neural network for assessment of the vesicoureteral reflux's diagnostic factors validity. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010, Part I. LNCS, vol. 6352, pp. 241–250. Springer, Heidelberg (2010)
6. Mantzaris, D., Anastassopoulos, G., Tsalkidis, A., Adamopoulos, A.: Intelligent prediction of vesicoureteral reflux disease. *WSEAS Transactions on Systems* 4(9), 1440–1449 (2005)
7. Nouretdinov, I., et al.: Multiprobabilistic venn predictors with logistic regression. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H., Karatzas, K., Sioutas, S. (eds.) AIAI 2012, Part II. IFIP AICT, vol. 382, pp. 224–233. Springer, Heidelberg (2012)
8. Papadopoulos, H.: Reliable probabilistic prediction for medical decision support. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H. (eds.) EANN/AIAI 2011, Part II. IFIP AICT, vol. 364, pp. 265–274. Springer, Heidelberg (2011)
9. Papadopoulos, H.: Reliable probabilistic classification with neural networks. *Neurocomputing* 107, 59–68 (2013)
10. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World. Springer, New York (2005)
11. Vovk, V., Shafer, G., Nouretdinov, I.: Self-calibrating probability forecasting. In: Advances in Neural Information Processing Systems 16, pp. 1133–1140. MIT Press (2004)
12. Zhou, C., Nouretdinov, I., Luo, Z., Adamskiy, D., Randell, L., Coldham, N., Gammerman, A.: A comparison of venn machine with platt's method in probabilistic outputs. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H. (eds.) EANN/AIAI 2011, Part II. IFIP AICT, vol. 364, pp. 483–490. Springer, Heidelberg (2011)

Application of a Neural Network to Improve the Automatic Measurement of Blood Pressure

Juan Luis Salazar Mendiola¹, José Luis Vargas Luna^{1,*},
José Luis González Guerra², and Jorge Armando Cortés Ramírez^{1,*}

¹ Centro de Innovación en Diseño y Tecnología, Tecnológico de Monterrey, Monterrey, México
^{jl.vargas.phd.mty, jcortes}@itesm.mx}

² Instituto de Cardiología y Medicina Vascular, Hospital Zambrano Hellion,
Tecnológico de Monterrey, Monterrey, México

Abstract. Nowadays, Blood Pressure (BP) is an established major risk factor to determine cardiovascular accidents. Current BP monitors are not able to deal with noisy situations such as those present in stress tests. The aim of this study is to develop a system able to measure the BP even under these conditions. A device and an algorithm based on an Artificial Neural Network (ANN) are proposed as a feasible solution for BP measurement. Different ANN structures were trained to optimize the recognition of the Korotkoff sounds and the best implemented in the final system.

The system generates measurements similar to those of the physician, yet the differences increase with high deflation rates or when the staff lecture is not reliable. This work demonstrates the feasibility of an ANN application to improve BP measurement in situations in which the most reliable method known at present is conducted manually.

Keywords: ANN, ECG, Korotkoff Sounds, Oscillometry, Blood Pressure.

1 Introduction

Nowadays, Blood Pressure (BP) is an established major risk factor to determine coronary heart disease and cardiovascular accidents. Regardless of the huge technological advance in medicine, the most commonly used method for BP measurement in the daily clinical activity is still with the use of a manual sphygmomanometer and a stethoscope to detect the Korotkoff sounds [1]. These sounds are heard in the brachial artery when a pressure between the systolic and diastolic pressure occludes the blood flow in the upper arm [2].

Although used for over 100 years, this technique is still the most reliable way to monitor BP. Proof of this is the clear preference of clinicians to use it over automatic methods during effort tests [1], in which the patient's movements generate huge artifacts that most automatic monitors are not able to discriminate. Since BP can be a key factor for diagnosis, the intra and inter-variability of conventional methods is not

* Corresponding authors.

trivial since reliable BP readings can be affected by many factors, such as lack of experience or the physical and psychological states of the medical staff. Even if the staff is in optimal conditions, the accurate measurement of BP cannot be assured because the deflation rate of the cuff and the heart rate generate some degree of uncertainty.

The current automatic BP monitors only use oscillometry (pressure measurement) as the input signal. This limits their capability to distinguish between artifacts and the desired signal. At least two factors play an important role in the performance of algorithms for oscillometry. First, oscillometry waves are susceptible to artifacts and noise from a variety of sources. The typical algorithms are unable to cope with artifacts, such as common noise and other variations that reflect on the waves of the oscillometry [3]. In fact, most of these algorithms assume directly or indirectly that the blood pressure remains constant during the registration period (up to 30s). The variations in blood pressure during this time and the signal noise definitely corrupt the measurement. The second factor that limits the performance of conventional algorithms is the simple way in which the BP-Oscillometry relationship is considered; for instance, it is a nonlinear relationship.

Many studies have used artificial intelligent techniques such as the artificial neural networks (ANNs) to detect the BP variations in a more accurate way [3–7]. [4] models the variations in systolic, diastolic and mean pressure applying statistical regressions and ANNs to parameters such as age, pulse and alcohol addiction. However, the study threw a general expression since it was not possible to generalize the BP only under such parameters. [6] developed a more complex method with the use of multiple physiological signals. In this study, blood pressure was estimated based on the assumption that a relationship exists between the time of pulse wave propagation (PWTT) and blood pressure.

However, the algorithms developed are not stable enough to deal with the noise that occurs, for example, in a stress test. For this reason, the aim of this study is to develop a system able to emulate the human (manual) process to measure blood pressure even under highly noisy conditions. A device and an algorithm based on an artificial neural network are proposed as a feasible solution for BP measurement. It combines data of heart sounds (HS), electrocardiography (ECG) and oscillometry. The algorithm uses and trains an ANN to analyze the sounds of auscultation and discriminates between valid and invalid Korotkoff sounds. It also implements an algorithm of level of confidence and presents the deflation rate of the cuff in order to evaluate its performance. Unlike conventional BP monitors, this system's main field of application is to work under conditions in which the measurements are complicated even for highly qualified staff.

2 Methodology

For this viability study, the necessary hardware to achieve the signal acquisition required for the proposed method was developed. An experimental setup was defined in order to generate a database with the appropriate information. Then, the software was

developed to extract the most representative information on the database and feed the ANN. Different ANN structures were proposed and their performances compared. Finally, an algorithm that implements the signal acquisition and the best ANN was developed and validated.

2.1 Hardware Development

The hardware developed consists of a device capable of recording oscilometry, a 2-derivation electrocardiogram (ECG) and heart sounds (auscultation). The three stages are based on instrumentation amplifiers (INA114AP, Burr-Brown Corp., USA) and notch-filtered by a second order filter with central frequency of 60Hz. The oscilometry signal is acquired with the implementation of a pressure sensor (MPX2050, Motorola, Inc., USA) on a conventional baumanometer. The heart sounds are acquired with the help of a commercial electronic stethoscope.

To digitalize and transmit the data to a PC, an acquisition card NI-USB6008 (National Instruments, Inc., USA) was used. All the signals were digitalized using a sampling rate of 1kHz through a USB transmission port.

2.2 Experimental Setup

Data were collected from male subjects: 6 healthy volunteers and 4 patients with different cardiovascular pathologies. Their ages ranged between 20-25 and 50-60 years respectively. The volunteers were healthy people with no diagnosis of cardiovascular problems, while the patients were selected by a cardiologist.

The participants were subjected to an effort test protocol and medically monitored by highly qualified staff before and during the test. The data acquisition for BP measurement was done simultaneously to the measurement made by the hospital staff. Highly qualified personnel registered the blood pressure with the conventional method (cuff placed in the left upper arm and the stethoscope placed over the brachial artery in the antecubital fossa [2]) with the use of a sphygmomanometer and a digital stethoscope. The measurements were done once under 8 different stages of the protocol:

1. REP: Rest
2. V1.7: Treadmill at a speed of 1.7km/h and no inclination
3. V2.5: Treadmill at a speed of 2.5km/h and no inclination
4. V3.4: Treadmill at a speed of 3.4km/h and no inclination
5. V4.2: Treadmill at a speed of 4.2km/h and no inclination
6. V2.5+: Treadmill at a speed of 2.5km/h and inclination at 2%
7. V4.2+: Treadmill at a speed of 4.2km/h and inclination at 2%
8. RDAF: 2 minutes after the tests on the treadmill

2.3 Software

Although some signal conditioning is done in the hardware stage, the recorded signals are often contaminated by noise. In order to extract useful information from the raw

data, two processing stages were implemented: preprocessing and feature extraction. The whole development of the software was done by implementing and combining LabView™ (National Instruments, Austin, USA) and MATLAB™ (Mathworks, Inc., Natick, USA).

Signal Pre-processing. In this stage, all the signal components unrelated to the phenomenon of interest are removed. In order to clean the signals from the main noises, digital filters or 12th order were implemented. A high-pass filter with cut frequency of 5Hz was implemented to remove the baseline wandering and the respiration effect in the ECG. The electromagnetic noise is removed from all signals with a 12th order notch filter centered in 60Hz.

Feature Extraction. This stage aims to extract the most representative data of each signal in order to provide more useful data feeding to the ANN. This extraction is made through the analysis of the segmented signal. The segmentation is done by cutting the signal in the middle points between each sound peak.

Each of the used signals has multiple metrics related to statistics, time-domain, frequency-domain, among others. However, a detailed selection of these features was done according to the physiological understanding of the cardiovascular system and the origin of the signals on it. Then an evaluation of the metrics with biological plausibility to describe the variables of interest as well as literature research were done [8–12]. A total of 16 metrics related with the QRS complex, peak and valley values of oscilometry, sound amplitude, frequency domain of sound and the combination between them were used. A summary of the metrics is shown in Table 1.

2.4 Artificial Neural Network

The development of a high performance ANN able to identify the Korotkoff sounds is crucial to the success of the system. Therefore, several steps were considered for its definition. The input matrix consisted of 2131 metric vectors obtained from the feature extraction, while the target matrix was generated by a human expert that made a visual recognition of the Korotkoff sounds. Different ANN structures were implemented, tested and compared with MATLAB™ (Mathworks, Inc., Natick, USA). All the ANNs were trained with the same training matrix using the Levenberg-Marquardt optimization algorithm and hyperbolic tangent as transfer function for all the neurons. All the evaluated structures had been defined with backpropagation and contained 2 or 3 hidden layers. The training matrix was randomly split in 3 parts: 70% was used for training, 15% for validation, and 15% for testing. Their performances were compared by means of the Minimum Mean Square Error of its output and the best one was selected for further analysis.

Table 1. Summary of features extracted from the physiological signals

| Name | Signal | Unit | Description |
|-------|-------------|------|--|
| QRSH | ECG | V | Height of interval QRS for the first lead |
| QRSW | ECG | Seg | Width of interval QRS for the first lead |
| QRSH2 | ECG | V | Height of interval QRS for the second lead |
| QRSW2 | ECG | Seg | Width of interval QRS for the second lead |
| BPH | BP | mmHg | Difference between peak and valley of oscillometry |
| SH | HS | V | Amplitude difference between maximum and minimum point of each sound segment |
| HR | HS | Seg | Time between the current and past sounds |
| SM-R | HS-ECG | Seg | Time between the middle of the sound and R-peak |
| IS-R | HS-ECG | Seg | Time between sound beginning R-peak |
| IDO-R | BP-ECG | Seg | Time between the beginning of the pressure wave and the R-peak |
| FDO-R | BP-ECG | Seg | Time between the end of pressure wave and the peak R |
| FMAX | Freq-Domain | Hz | Maximum frequency in characteristic isocurve of the frequency spectrogram |
| FMIN | Freq-Domain | Hz | Minimum frequency in characteristic isocurve of the frequency spectrogram |
| P75 | Freq-Domain | Hz | Percentile 75 of frequency in characteristic isocurve of the frequency spectrogram |
| MED | Freq-Domain | Hz | Median frequency in characteristic isocurve of the frequency spectrogram |
| P25 | Freq-Domain | Hz | Percentile 25 of frequency in characteristic isocurve of the frequency spectrogram |

2.5 System Algorithm

The global algorithm, concatenated all the previous stages, embedded the signal acquisition, processing, sound identification and BP evaluation. As the method proposes, this algorithm implements the ANN to identify the Korotkoff sounds and relates them to the oscillometry pressure. The blood pressure values and the level of confidence provided by the system are calculated according to the flow diagram shown in Fig. 1 and considers the following assumptions:

- The first and last detected sounds are considered as systolic and diastolic pressures respectively.
- If both sounds are validated by the ANN, then the level of confidence is considered high.
- If only one sound is validated by the ANN, then the level of confidence is considered medium, since the other value is given only by the oscillometry.
- If none of the sounds are validated by the ANN, the level of confidence is considered low, since only the oscillometry is considered for the BP measurement.

The performance of the final system was tested with 15 new measurements, and the results were compared with human-specialist results. The measurements conditions like deflation rate and protocol stage are also presented for discussion.

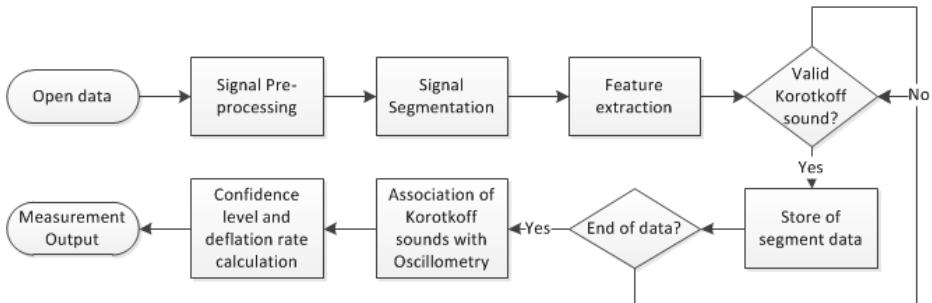


Fig. 1. General process of the BP measurement algorithm

3 Results

3.1 Hardware Development

A functional portable prototype was successfully built. Although the use of a commercial digital acquisition card was used, the whole circuitry was packaged in a portable box. This made the measurements more robust and easier to set up. Also, the coupling of the system sensors with the devices used during the effort test (sphygmomanometer, ECG and stethoscope) avoids the addition of more sensors to the subject. This also reduces errors since the human and the device use the same signals to evaluate the BP.

3.2 Experimental Setup

The selection of the study protocol resulted ideal due to its complicated conditions. It also showed that even for highly qualified medical staff it is difficult to make accurate measurements.

During the initial phase of this study, a total of 90 BP measurements were made; 2131 sounds were obtained (valid and invalid) from those measurements. This data were considered enough as initial training data set.

3.3 Software

The implementation of the pre-processing filters resulted to be sufficient to provide clean data to the feature extraction stage.

The feature extraction for each segment was also obtained with the software. Fig. 2 shows a typical example. Other metrics that correlate the 3 signals are mainly in terms of time differences.

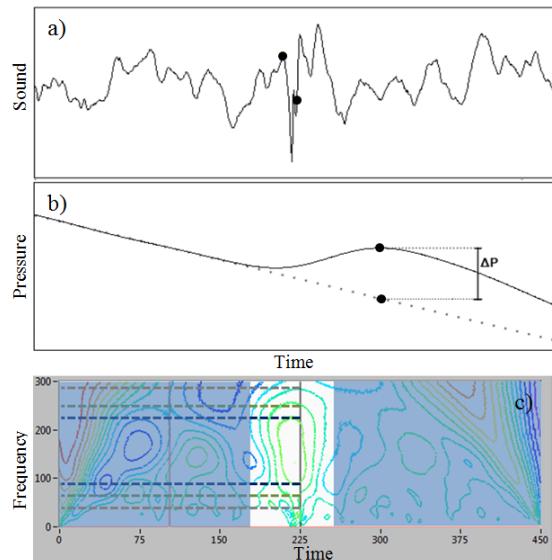


Fig. 2. Punctual metrics obtained from the signals a) auscultation; b) oscillometry (with linear prediction); c) Frequency regions of the sound's wavelet

3.4 Artificial Neural Network

A total of 10 different structures of ANN were tested. A summary of the structures and their performance expressed in the MSE value is shown in Table 2. The best trained network (by means of MSE value) was No. 10, so this structure was chosen for an extra validation test.

3.5 System Algorithm

The performance of the final system was tested using fifteen new BP measurements. The system's output is compared with the results of the human specialist and presented in Table 3, which also displays the condition of measurement (Cond), the number of pulses registered in the sample, the blood pressure (systolic-diastolic) provided by the doctor who performed the measurement (BP-H), the blood pressure provided by the system (BP-S), the difference of the systolic and diastolic blood pressure (BP-Err), the cuff deflation rate in mmHg per second (Def), and the level of confidence (Low, Medium or High) of the measurement (Conf).

Table 2. Comparison of different artificial neural networks architectures tested to improve the system performance

| # | Number of Neurons in the hidden layer: | | | MSE |
|----|--|----|----|-------|
| | 1 | 2 | 3 | |
| 1 | 3 | 3 | 3 | 0.191 |
| 2 | 5 | 3 | 2 | 0.175 |
| 3 | 5 | 5 | 5 | 0.172 |
| 4 | 5 | 5 | 10 | 0.174 |
| 5 | 5 | 10 | 5 | 0.184 |
| 6 | 10 | 5 | 0 | 0.201 |
| 7 | 10 | 5 | 3 | 0.175 |
| 8 | 10 | 5 | 10 | 0.185 |
| 9 | 10 | 10 | 10 | 0.189 |
| 10 | 12 | 5 | 3 | 0.159 |

Table 3. Results provided by the system compared to the human expert. NA marks the events in which the staff was not able to give a lecture.

| Case | Cond | Pulses | BP-H | BP-S | BP-Err | Def | Conf |
|-----------------|-------|--------|---------|---------|--------|-------|------|
| 1 ¹ | REP | 27 | NA | 91-79 | NA | 9.60 | L |
| 2 | REP | 24 | 110-80 | 108-76 | 2-4 | 5.62 | H |
| 3 | V1.7 | 33 | 120-90 | 111-72 | 9-18 | 8.20 | H |
| 4 | V1.7 | 44 | 110-70 | 105-73 | 5-3 | 8.30 | H |
| 5 | V2.5 | 25 | 110-70 | 120-78 | 10-8 | 10.18 | H |
| 6 | V3.4 | 32 | 110-70 | 106-76 | 4-6 | 9.16 | M |
| 7 | V4.2 | 16 | 130-60 | 124-77 | 6-17 | 15.91 | M |
| 8 ¹ | V4.2 | 54 | NA | 111-71 | NA | 6.77 | M |
| 9 | V2.5+ | 10 | 130-70 | 110-70 | 20-0 | 8.85 | L |
| 10 | V4.2+ | 50 | 130-70 | 119-78 | 11-8 | 8.70 | L |
| 11 ¹ | RDAF | 31 | 130-110 | 103-103 | 27-8 | 4.62 | M |
| 12 ¹ | RDAF | 47 | 140-95 | 118-94 | 22-1 | 3.80 | M |
| 13 ¹ | RDAF | 39 | NA | 85-70 | NA | 2.65 | H |
| 14 | RDAF | 37 | 110-70 | 103-65 | 7-5 | 6.57 | H |
| 15 | RDAF | 38 | 100-90 | 92-88 | 8-2 | 5.33 | M |

4 Discussion

A functional prototype was fully developed. The design was successfully used and accepted by the hospital staff. Although a more robust electronic design is necessary, the device could record the three physiological signals required.

The experimental protocol was correctly defined and provided enough data under a range of conditions. It also demonstrated that even with the use of the gold standard

¹ Cases where the physician did not obtain a clear record of the measurement.

techniques by highly qualified personnel, the accuracy of the measurement was not guaranteed; and moreover, it shows that the resolution given by medical staff is hardly better than 10mmHg. The number of subjects was enough to train the ANN and to produce good results for concept proof; however, it is necessary to increase the database to improve the accuracy of the measurements.

The feature extraction became a challenge on account of the plethora of available algorithms for many different metrics. But once the final selection of metrics had been defined, the calculation was finalized without further complications. However, many possible metrics were not considered in this first stage. Further research mainly in the sound/speech recognition field should be done in order to identify potential features to recognize the Korotkoff sounds.

Although the exploration of feasible ANN structures was limited by time constraints, an excessively time consuming structure selection was made. The selected structure showed an acceptable performance, but since the best ANN was the largest, it is clear that the application of an optimization algorithm is still possible to increase the efficiency and reduce the dimensionality of the metrics used.

The final system was able to proof the concept of the artificial intelligence capabilities in the BP measurement. Although some measurements differed significantly from the data measured by the medical staff, the most important errors were found in two main situations: when the human observer was not sure of his measurements, and with high deflation rates of the cuff. In neither case could the error be completely attributed to the device. However, the second situation can be easily solved if the following prototype takes control of the cuff's inflation/deflation rate.

5 Conclusions

A functional prototype for the BP measurement was developed. Although the output results are acceptable, they can be clearly improved. A lot of areas of opportunity were identified in the electronic, sound/speech recognition, digital signal processing and ANN optimization fields. Also, new possible robust mechanisms were found mainly in the direction of the deflation rate. It was also remarkable the acceptance of the device by the hospital staff. However, new user-requirements data were collected and considered for further developments.

This work demonstrates the feasibility of an ANN application to improve the BP measurement in clinical situations in which the only reliable method until now is done manually.

References

1. Kurl, S., Laukkanen, J.A., Rauramaa, R., Lakka, T.A., Sivenius, J., Salonen, J.T.: Systolic Blood Pressure Response to Exercise Stress Test and Risk of Stroke. *Stroke* 32, 2036–2041 (2001)
2. Guyton, A.C., Hall, J.E.: *Tratado de Fisiología Médica* (2011)

3. Baker, P., Orr, J.: Method for determining blood pressure utilizing a neural network (1994),
<http://www.google.com/patents?hl=en&lr=&vid=USPAT5339818&id=yXwmAAAAEBAJ&oi=fnd&dq=Method+for+determining+blood+pressure+utilizing+a+neural+network&printsec=abstract>
4. Bhaduri, A., Bhaduri, A., Bhaduri, A., Mohapatra, P.K.: Blood Pressure Modeling using Statistical and Computational Intelligence Approaches. In: 2009 IEEE International Advance Computing Conference, pp. 1026–1030. IEEE (2009)
5. Wang, F., Syeda-Mahmood, T., Beymer, D.: Finding disease similarity by combining ECG with heart auscultation sound. Computers in Cardiology, 261–264 (2007)
6. Lass, J., Meigas, K., Karai, D., Kattai, R., Kaik, J., Rossmann, M.: Continuous blood pressure monitoring during exercise using pulse wave transit time measurement. In: Conference Proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society Conference, vol. 3, pp. 2239–2242 (2004)
7. Matakgah, M., Knopp, J., Mawaghed, S.: Iterative Processing Method Using Gabor Wavelets and the Wavelet Transform for the Analysis of Phonocardiogram. In: Akay, M. (ed.) Time Frequency and Wavelets in Biomedical Signal Processing, pp. 271–304 (1998)
8. Alexakis, C., Nyongesa, H.O., Saatchi, R., Harris, N.D., Davies, C., Emery, C., Ireland, R.H., Heller, S.R.: Feature extraction and classification of electrocardiogram (ECG) signals related to hypoglycaemia. Computers in Cardiology, 537–540 (2003)
9. Ahmad, S., Chen, S., Soueidan, K., Batkin, I., Bolic, M., Dajani, H., Groza, V.: Electrocardiogram-assisted blood pressure estimation. IEEE Transactions on Bio-medical Engineering 59, 608–618 (2012)
10. Chen, S., Groza, V.Z., Bolic, M., Dajani, H.R.: Assessment of algorithms for oscillometric blood pressure measurement. In: 2009 IEEE Instrumentation and Measurement Technology Conference, pp. 1763–1767. IEEE (2009)
11. Izeboudjen, N., Farah, A.: A New Neural Network System for Arrhythmias Classification. In: Proceedings of the Mediterranean Conference on Medical and Biological Engineering and Computing (1998)
12. Omran, S., Tayel, M.: A heart sound segmentation and feature extraction algorithm using wavelets. In: 2003 46th Midwest Symposium on Circuits and Systems, pp. 392–395. IEEE (2004)

An Immune-Inspired Approach for Breast Cancer Classification

Rima Daoudi^{1,2}, Khalifa Djemal¹, and Abdelkader Benyettou²

¹ IBISC Laboratory, University of Evry Val d'Essonne, 40, Rue du Pelvoux,
CE1455 Courcouronnes 91020 Evry Cédex, France

{Rima.Daoudi,khalifa.djemal}@ibisc.univ-evry.fr

² SIMPA Laboratory, University of Sciences and Technologies,
BP 1505 Mnaouer, Oran, Algeria
aek.benyettou@univ-usto.dz

Abstract. Many pattern recognition and machine learning methods have been used in cancer diagnosis. The Artificial Immune System (AIS) is a novel computational intelligence technique. Designed by the principles of the natural immune system, it is able of learning, memorize and perform pattern recognition. The AIS's are used in various domains as intrusion detection, robotics, illnesses diagnostic, data mining, etc. This paper presents a new immune inspired idea based on median filtering for cloning, and applied for benign/malignant breast cancer classification. The classifier was tested on Wisconsin Diagnostic Breast Cancer Database using classification accuracy, sensitivity and specificity, and was found to be very competitive when compared to other classifiers.

Keywords: Breast cancer, Clonal selection, Local sets, Median filter, Clone, Mutate.

1 Introduction

Every two minutes a woman is diagnosed with breast cancer. Breast cancer has become one of the major causes of mortality around the world since the last decades. Research into cancer diagnosis and treatment have become an important issue for the scientific community. Due to its late diagnosis, the result is often a heavy treatment, mutilating and expensive which is accompanied by a high mortality rate. Although breast cancer can be fatal, women have the highest chance of survival if cancer could be detected at the early stages. Early diagnosis and treatment play critical roles in increasing the chance of survival. As such, the classification of breast cancer, diagnosis and prediction techniques have been the focus of much researches in the world of medical informatics. There is no doubt that the evaluation of data from patients and experts decision are the most important factors in diagnosis. However, it has been

proven in the literature that different artificial intelligence techniques are of a precious help to the experts for decision making. Indeed, tiredness, lack of experience and microscopic analysis are factors that can mislead the classification. Therefore, support systems for diagnosis have been effective in minimizing errors that can be done while providing more detailed medical data in a shorter period [1].

Several articles have been published attempting to classify the datasets for breast cancer, using various techniques such as Neural Networks [1-2], Support Vector Machines [3-4], Genetic Algorithms [5-6], Expert Systems [7] and Artificial Immune Systems [8-9].

The vertebrate immune system is composed of diverse sets of cells and molecules that work together with other systems (like neural and endocrine) for maintaining homeostatic state. Its primary function is to protect the body from foreign substances called antigens by recognizing and eliminating them. This process is known as the immune response. It makes use of a huge variety of antibodies to neutralize these antigens [10].

Inspired by biological immune systems, Artificial Immune Systems have emerged during the last decade. They are incited by many researchers to design and build immune-based models for a variety of application domains. Artificial immune systems can be defined as a computational paradigm that is inspired by theoretical immunology, observed immune functions, principles and mechanisms [11].

In artificial intelligence language, Artificial immune systems (AIS) is a diverse and maturing area of research that bridges the disciplines of immunology, biology, medical science, computer science, physics, mathematics, and engineering. The most used models in the field of AIS are immune networks, clonal selection and negative selection [12].

Negative selection is a mechanism that protects the human body against self-reactive lymphocytes. It uses the immune system's ability to detect unknown antigens without reacting to the self cells [13]. The immune network theory was first proposed by Jerne [14]. This theory proposes that the immune system maintains an idiotypic network on interconnected B-cells for antigen recognition.

The Implications of the clonal selection theory were explored on 1959 by F.Macfarlane Burnet in [15]. It states that an antigen selects from among a variety of antibodies those with receptors capable of reacting with part of the antigen. There are two processes: 1) pattern recognition and 2) selection. Only cells able to recognize an antigen will proliferate (*cloning + mutation*). Fig 1 describes the characteristics of the clonal selection theory.

In 2002, De Castro and Von Zuben proposed CLONALG. The algorithm works by retaining only one memory cell for each antigen presented to it and makes use of a ranking system to determine the rate at which clones can be produced. The clones, in turn, are mutated using a multipoint mutation method. Whereby, they are mutated if a randomly generated control number exceeds a given threshold. In this paper, we propose an improvement of CLONALG by introducing a new concept based on median filter to choose the cell that will be cloned.

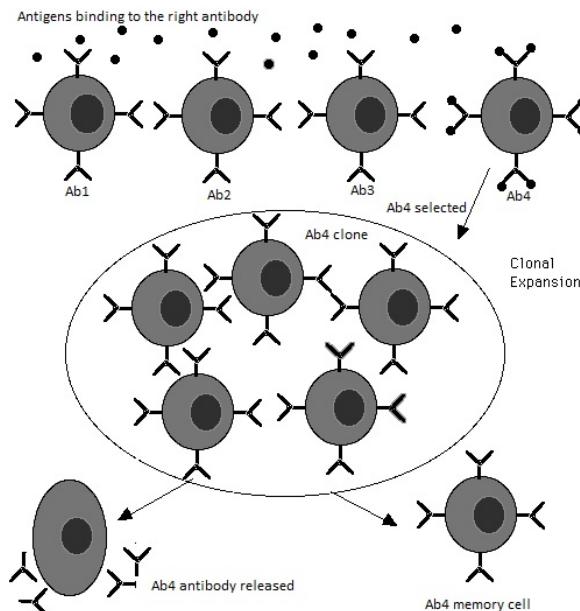


Fig. 1. Clonal Selection Principle: antibody selection, clonal expansion and memory cells generation

The rest of this paper is organized as follows: section 2 presents a theoretical background of CLONALG algorithm, detailing its principle and methods, and the proposed idea introduced to improve the algorithm; in section 3 we present the dataset used and give and discuss the results obtained; section 4 presents highlights directions for future work and forms the conclusion of this work.

2 Selection and Cloning Based Median Filter

2.1 CLONALG : Principle and Methods

The clonal selection algorithm, CSA, was first proposed by de Castro and Von Zuben in [16] and was later enhanced in [17] and named CLONALG. The principle of the algorithm is to take a population of antibodies and by repeated exposure to antigens, over a number of generations, develops a population more sensitive to the antigenic stimulus. The general immune aspects used in CLONALG are:

- Preservation of a set of memory cells,
- Selection and cloning of the most representative antibodies,
- Rejection of the less stimulated cells,
- Reselection of the nearest generated clones to the antigen in progress,
- Generation and maintenance of diversity.

CLONALG algorithm takes a random population of antibodies in the initialization step of learning. These cells are taken from the set of training examples, which means that the initial cells do not represent necessarily all cells to learn. Learning will then depend on this set of randomly initialized cells. In the following steps, the algorithm build for each training example a set of memory cells that represent it the most in similarity measure, by selecting N-best candidate cells and generating a set of clones for each cell in proportion to its affinity these clones will be mutated each one inversely relative to its similarity measure, and added to the set of N cells. A reselection of the best cells from this set is made thereafter and these latest will be added to all memory cells. The worst P cells will be replaced by randomly created ones, even if those who are rejected can have a best representation of other examples of the training dataset in succeeding generations, no checking is done.

In order to improve CLONALG and treat negative points mentioned above, we propose some modifications that can enhance the learning algorithm.

2.2 CLONALG Improvement

As in CLONALG, The aim of learning is to build a set of memory cells for each training class, these cells will be used in the classification step thereafter. The first proposed change is in the initialization step of learning. Instead of taking a random set of cells from the training data, a creation of local sets of these examples will be executed, such that each example belongs at least one of these groups. The center of each group is calculated, these average cells represent the memory cells for the initial launch of the learning algorithm, this will allow a representation of all cells to learn before starting iterations of the algorithm. Figure 2 shows the initialization step, and the creation of initial antibodies.

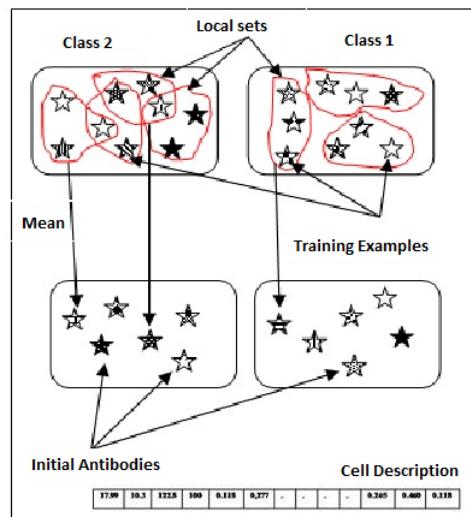


Fig. 2. Creation of initial antibodies: Local sets creation and calculation of centers

The second change introduced to CLONALG is during learning, more precisely in the choice of the cell to be cloned. CLONALG selects the n cells closest to the antigen, clone and mutate them; The best mutated clones are added to the set of memory cells. The idea that we propose is to create a median cell which will be compared to the best antibody, if it produces a higher affinity, this median cell is cloned and mutated, the set of mutated clones will be added to the memory cells. Otherwise, the nearest antibody to the antigen is cloned and mutated.

Let n be the number of attributes in the database to learn, creating the median cell is made by selecting n antibodies representing the most the antigen in process, and by taking the median value of each attribute. This is done by sorting each column of the matrix of the selected antibodies and taking the middle line. Figure3 explains the creation process of the median cell.

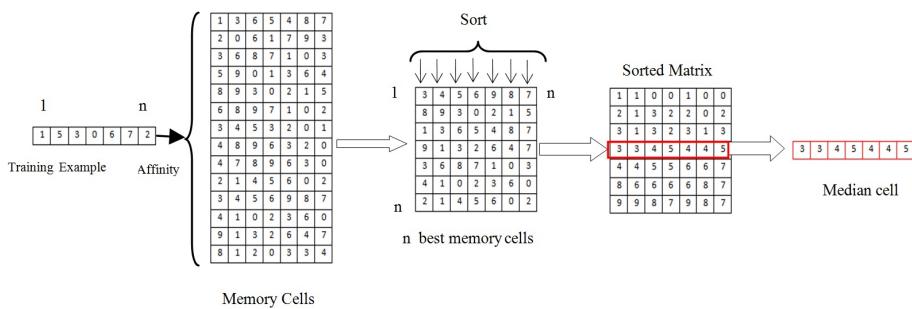


Fig. 3. Creation of Median Cell: N best antibodies selection, matrix sorting and median cell picking

After the creation of the median cell, we calculate its affinity with the training example. This value is compared to the affinity value of the closest antibody to the antigen, if it is better, median cell is added to the set of memory cells, cloned and mutated, and its best clones are added to antibodies thereafter. Otherwise if the best antibody's affinity is greater, it will be cloned and mutated, and best clones will be added to the set of memory cells. This procedure to create diversity without rejecting any cell that can be effective in succeeding generations.

The classification step consists of comparing each example to classify to all memory cells obtained at the end of learning, and the example is assigned to the class of the memory cell that maximizes the affinity.

3 Experimental Results

In this section we give the different obtained results improving breast cancer classification, first we present the used database in section 3.1, all achieved tests are presented in section 3.2, in section 3.3 we discuss the advantages of the proposed approach.

3.1 Wisconsin Diagnostic Breast Cancer

In order to test the approach, we used the Wisconsin Diagnosis Breast Cancer Database (WDBC), made publicly available by Dr. William H. Wolberg of the Department of Surgery of the University of Wisconsin Medical School, the database is available at [18].

WDBC presents parameters calculated from a scanned image of a fine needle aspiration (FNA) taken from the breast. WDBC consists of 569 samples (357 benign and 212 malignant), each with 32 values of attributes. The attributes information's are detailed below:

1. Identification number
2. Diagnosis (M = malignant, B = benign)
- 3-32. Ten real-valued features are computed for each cell nucleus:
 - a. radius (mean of distances from center to points on the perimeter)
 - b. texture (standard deviation of gray-scale values)
 - c. perimeter
 - d. area
 - e. smoothness (local variation in radius lengths)
 - f. compactness (perimeter² / area - 1.0)
 - g. concavity (severity of concave portions of the contour)
 - h. concave points (number of concave portions of the contour)
 - i. symmetry
 - j. fractal dimension ("coastline approximation" - 1)

The performance of the algorithm is studied using WDBC, the training data are antigens represented by feature vectors, also, the antibodies have the same shape as the antigenic vectors, the Euclidean distance was used as a measure of similarity.

3.2 Results

We randomly shared our database into 75% for training data and 25% for the test, and then the average of 10 times of successive runs is taken as classification result. After 5, 8, 10, 15 and 20 iterations, memory cells generated at the end of learning are used in classification, by comparing each cell to classify to all of the created memory cells, and assigning it to the class containing the memory cell with the highest measure of similarity, simulation and implementation are done using MATLAB 7.11.0. Table 1 summarizes the results obtained:

Classification accuracy: In this study, classification accuracy for the data sets are measured using the equation:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

- True positive (TP): An input is detected as a patient with breast cancer, as diagnosed by the expert clinicians.
- True negative (TN): An input is detected as normal and also labeled as a healthy person by the expert clinicians.
- False positive (FP): An input is detected as a patient with breast cancer, although labeled as a healthy person by the expert clinicians.
- False negative (FN): An input is detected as normal, although diagnosed by the expert clinicians as having breast cancer.

Sensitivity and specificity: For sensitivity and specificity analysis, we use the following expressions:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

So as to compare the results obtained, Table 2 shows the test results of various AIS classification algorithms obtained through Weka 3.4.11 in [15]

Table 1. Classification accuracies of Median Filter AIS used for detection of breast cancer

| Iterations N° | Classification accuracies (%) | | |
|---------------|-------------------------------|-------------|-------------------------------|
| | Specificity | Sensitivity | Total classification accuracy |
| 5 | 95.55 | 88.67 | 93 |
| 8 | 92.22 | 92.44 | 92.30 |
| 10 | 91.11 | 92.45 | 91.60 |
| 15 | 94.44 | 92.45 | 93.70 |
| 20 | 93.33 | 96.22 | 94.40 |

Table 2. Classification accuracies of different AIS algorithms on WDBC

| AIS Algorithms | % Accuracy |
|-------------------|------------|
| CLONALG | 71.9 |
| CSA | 72.8 |
| AIRS1 | 90.3 |
| IMMUNOS1 | 71.0 |
| CLONAX | 93.4 |
| Median Filter AIS | 94.40 |

3.3 Discussion

The results obtained show a great performance of the algorithm. Indeed, the initialization step played an important role, instead of taking a random set of antibodies that does not really represent all the data to learn, we create local sets to generate antibodies to launch initial learning. These middle cells obtained represent all the training data unlike other clonal selection algorithms. Also, choosing a median cell for cloning has prevented several iterations performed in CLONALG and have better results with lower complexity. CLONALG chooses N cells close to the antigen being processed, and passed to operators of cloning and mutation. Reselection is done thereafter with a worst cell replacement, which can mislead the classification, because these rejected cells can be the best ones in subsequent generations. Our approach avoids the rejection, and the stages of selection and reselection, by filtering the cells and selecting a cell to be cloned while maintaining diversity and reduce complexity. The classification rate obtained after 20 iterations is 94.40%, higher performance compared with other AIS algorithms applied to the same database.

4 Conclusion

This paper has focused on a supervised learning system based on immunological principles, The idea of median filter was introduced to the artificial clonal selection for breast cancer recognition, and another of local sets creation for the generation of initial cells before starting learning. The results obtained have shown promising in comparison to other AIS algorithms applied to the same database, which implies that the proposed idea can be useful for experts for a second opinion on their diagnosis of breast cancer. However, the algorithm has some negatives in terms of execution time, and the number of memory cells generated, these items will be discussed in our next work.

References

1. Marcano-Cedeno, A., Quintanilla-Dominguez, J., Andina, D.: WBCD breast cancer database classification applying artificial metaplasticity neural network. *Expert Systems with Applications* 38(8), 9573–9579 (2011)
2. Huang, M.-L., et al.: Usage of case-based reasoning, neural network and adaptive neuro-fuzzy inference system classification techniques in breast cancer dataset classification diagnosis. *Journal of Medical Systems* 36(2), 407–414 (2012)
3. Maglogiannis, I., Zafiroopoulos, E., Anagnostopoulos, I.: An intelligent system for automated breast cancer diagnosis and prognosis using SVM based classifiers. *Applied Intelligence* 30(1), 24–36 (2009)
4. Rocca, P., et al.: An integration between SVM Classifiers and Multi-Resolution Techniques for Early Breast Cancer Detection. In: 2008 IEEE Antennas and Propagation Society International Symposium, vols. 1-92008, pp. 4114–4117 (2008)

5. Jain, R., Mazumdar, J.: A genetic algorithm based nearest neighbor classification to breast cancer diagnosis. *Australasian Physical & Engineering Sciences in Medicine/Supported by the Australasian College of Physical Scientists in Medicine and the Australasian Association of Physical Sciences in Medicine* 26(1), 6–11 (2003)
6. Mazurowski, M.A., et al.: Case-base reduction for a computer assisted breast cancer detection system using genetic algorithms. In: *Proceedings 2007 IEEE Congress on Evolutionary Computation*, vols. 1-10, pp. 600–605 (2007)
7. Karabatak, M., Ince, M.C.: An expert system for detection of breast cancer based on association rules and neural network. *Expert Systems with Applications* 36(2), 3465–3469 (2009)
8. Polat, K., Sahan, S., Kodaz, H., Günes, S.: A new classification method for breast cancer diagnosis: Feature selection artificial immune recognition system (FS-AIRS). In: Wang, L., Chen, K., S. Ong, Y. (eds.) *ICNC 2005, Part II. LNCS*, vol. 3611, pp. 830–838. Springer, Heidelberg (2005)
9. Sharma, A., Sharma, D.: Clonal Selection Algorithm for Classification. In: Liò, P., Nicosia, G., Stibor, T. (eds.) *ICARIS 2011. LNCS*, vol. 6825, pp. 361–370. Springer, Heidelberg (2011)
10. Leung, K., Cheong, F., Cheong, C.: Generating compact classifier systems using a simple artificial immune system. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics* 37(5), 1344–1356 (2007)
11. de Castro, L.N., Timmis, J.I.: Artificial immune systems as a novel soft computing paradigm. *Soft Computing* 7(8), 526–544 (2003)
12. Cutello, V., Nicosia, G.: An immunological approach to combinatorial optimization problems. In: Garijo, F.J., Riquelme, J.-C., Toro, M. (eds.) *IBERAMIA 2002. LNCS (LNAI)*, vol. 2527, pp. 361–370. Springer, Heidelberg (2002)
13. Somayaji, A., Hofmeyr, S., Forrest, S.: Principles of a Computer Immune System. In: *Proceedings of the Second New Security Paradigms Workshop*, pp. 75–82 (1997)
14. Jerne, N.K.: Towards a Network Theory of Immune System. *Annales D Immunologie* C125(1-2), 373–389 (1974)
15. Burnet, F.: The clonal selection theory of acquired immunity. University Press, Cambridge (1959)
16. De Castro, L.N., Von Zuben, F.J.: The clonal selection algorithm with engineering applications. In: *Proceedings of GECCO 2000, Workshop on Artificial Immune Systems and their Applications*, Las Vegas, USA, pp. 36–37 (2000)
17. De Castro, L.N., Von Zuben, F.J.: Learning and Optimization Using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems (IEEE)* 6(3), 239–251 (2002)
18. <http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29>

Classification of Arrhythmia Types Using Cartesian Genetic Programming Evolved Artificial Neural Networks

Arbab Masood Ahmad, Gul Muhammad Khan, and Sahibzada Ali Mahmud

University of Engineering and Technology, Peshawar, Pakistan
{arbabmasood,gk502,sahibzada.mahmud}@nwfpuet.edu.pk
<http://www.nwfpuet.edu.pk>

Abstract. Cartesian Genetic programming Evolved Artificial Neural Network (CGPANN) is explored for classification of different types of arrhythmia and presented in this paper. Electrocardiography (ECG) signal is preprocessed to acquire important parameters and then presented to the classifier. The parameters are calculated from the location and amplitudes of ECG fiducial points, determined with a new algorithm inspired by Pan-Tompkins's algorithm [14]. The classification results are satisfactory and better than contemporary methods introduced in the field.

Keywords: CGPANN, Artificial Neural Network, Neuro-evolution, CVD, Cardiac Arrhythmias, Classification, Fiducial points, LBBB beats, RBBB beats.

1 Introduction

Cardiovascular diseases (CVD) are a major health issue worldwide. The heart beats with a rate that is controlled in certain limits by brain and may change with physical activities or emotions. However if a person's heart beats too fast, tachycardia($HR > 100$ beats per minute (bpm)); too slow, bradycardia ($HR < 60$ bpm) or irregularly, we say the person is having cardiac arrhythmias¹. Although most arrhythmia are less serious and transitory, some are very serious and may cause sudden cardiac death. In such cases instant action is needed in the form of medication or cardiac resuscitation. Often such arrhythmia episodes last for only a few minutes and cannot be captured in a normal ECG. A 24 hours ECG recording (Holter monitoring)is needed in such cases, that is later on analyzed.

In modern Computer based arrhythmia classification systems a digitized ECG is first preprocessed to make it clean from noise. The fiducial points of all the ECG complexes in a signal are then determined. From the fiducial points some morphological features or time/frequency patterns are extracted and fed as input to a classification system. A number of computational intelligence techniques have been adapted for classifying the ECG beats. A brief literature survey on this is given in section 3.

¹ <http://www.mayoclinic.com/health/heart-arrhythmias/DS00290>

The classification algorithm that we adapted is Cartesian Genetic Programming evolved Artificial Neural Network (CGPANN). The reason for using this algorithm is that it is less computation intensive than the others [9]. The algorithm evolves an optimized network with nodes that are not all connected.

The ECG signals that we used in our experiments were downloaded from the MIT-BIH website². We determined the fiducial points through our new algorithm and calculated the important morphological features of the ECG. With this data we trained our network so that it can classify ECG Arrhythmias.

The rest of the paper is arranged as follows: Section 2 describes the concept of neuroevolution, previous research done in the area and details of the algorithm that we used. Section 3 describes the cardiac arrhythmias, the algorithm that we developed for arrhythmia Classification and previous work done in this field. Section 4 is about the results and their analysis. Section 5 is the concluding section. Future directions for our research are also presented in this section.

2 Neuroevolution

The design of efficient architecture of an artificial neural network (ANN) has in the past depended on the experience of the researchers. However efforts have been made to develop algorithms that can automatically evolve the topology and weights of the ANN and are referred to as TWEANNs (Topology Weight Evolved Artificial Neural Networks) [21]. Neuro Evolution (NE) is the term used for artificial evolution of ANNs. In Neuro-Evolution [21] a number of aspects, such as functions, weights, inputs and ANN topology are evolved. The term used for the genetic representation of an ANN is genotype while that for the network itself as the phenotype. Following are a few popular neuroevolutionary techniques: Symbiotic Adaptive Neural Evolution (SANE) [13], Enforced Sub-Population (ESP) [4], EuSANE (Eugenic SANE)[9], conventional neuro-evolution (CNE) [4], Neuro Evolution of Augmenting Topologies (NEAT) [18], real time NEAT (rt-NEAT) [16], Evolution of recurrent systems with linear outputs (EVOLINO)[15], HyperNEAT [17], HyperGP [2]. The neuroevolution algorithm that we used is CGPANN. The benefits of using this algorithm are as follows:

1. This is a fast learning algorithm.
2. It evolves all the parameters of the network, generating the most efficient network, resulting in minimum number of nodes.
3. Previous work with CGPANN shows very good results [1]

2.1 Cartesian Genetic Programming Evolved Artificial Neural Network (CGPANN)

CGPANN is an ANN system that uses Cartesian genetic programming (CGP) for its training[11]. We replace the nodes of CGP which are necessarily digital logic or mathematical functions, with artificial neurons that have weighted connections and non-linear activation functions and term the network as

² <http://www.physionet.org/physiobank/database/mitdb/>

CGPANN[9]. Unlike a tree based genetic programming [10] its structure is made up of nodes placed in a two dimensional graph, connected under certain rules. The main parameters of the network are as follows: number of nodes, number of rows, number of columns, number of inputs per node (arity) and levels-back. A variety of networks can be produced by assigning different values to these parameters. The genotype of CGPANN is a fixed length array of integers that represent the input connections to nodes, node functions and network outputs. The integers (genes) in this array can take values that are limited by certain constraints. In the network (phenotype) produced from a genotype there are many nodes that do not lie in the path starting at inputs and ending at outputs. Nodes that do not lie in the path are called non-coding or inactive nodes. During the process of evolution the non-coding nodes might become coding. A fixed percentage of the genes (10% in our case) in a genotype are mutated to produce offspring that can be very much different than the parent. By evolving CGPANNs we allow topology, neural functions and weights to evolve.

Let m be the number of nodes in a CGPANN genotype and a be the number of inputs to a node, also called its arity. The node genes for a node N_i are: $F, I_1, W_1, C_1, I_2, W_2, C_2, I_3, W_3, C_3 \dots I_a, W_a, C_a$, while the genotype $G(m)$ is written as $G(m) = N_1, N_2, \dots, N_m, O_1, O_2, \dots, O_p$ where:

F is the activation function (either sigmoid or hyperbolic tangent), represented by a 0 or a 1,

I_i : represents the node input.

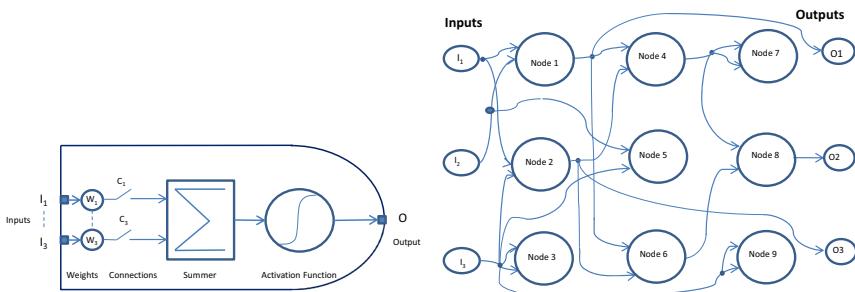
W_i : weight with I_i , having value -1 to +1,

C_i : indicates whether I_i is connected to the node or disconnected. Its value is either 0 or 1,

O_i : represent neurons that are directly connected to the network outputs.

Figure 1 shows a CGPANN neuron with arity $a = 3$. It shows the input ports, weights, connection switches, summer and activation function. The rows and columns of a network are connected such that a node in a certain column can get its input from a node to its left. A node in a column cannot connect to a node in the same column. Similarly a network output can be linked to any node or input of system. A chromosome contains genes in such an order that the genes for $Node_1$ are placed first followed by other nodes in the column and then the nodes in second column and so on, these are followed by the output genes. The nodes that lie in the paths between the phenotype inputs and outputs are the active nodes, while all the others are termed as inactive or junk nodes. In Figure 2 3, 5, 7 and 9 are junk nodes in a typical phenotype.

We used a $1 + \lambda$ evolutionary strategy for our system, where $\lambda = 9$ in our case and is the number of offspring [1]. CGPANN is explored in a range of applications including control, pattern recognition, load forecasting, multimedia traffic estimation and bandwidth management [9], [8], [1], [7].

**Fig. 1.** CGPANN Neuron**Fig. 2.** CGPANN Phenotype

3 Cardiac Arrhythmias

Arrhythmias are caused by disturbances in electrical pathways of the heart. The most effective technique for their detection is the ECG. An ECG is heart's electrical signal that is picked from the body surface and is represented by a graph in which the different sections have the following meanings: P-wave is produced when atria activate, QRS wave is produced when the ventricles depolarize and T-wave is produced when ventricles recover or repolarize. Conventionally a physician diagnoses the heart's condition from the shape of patient's ECG. However in many emergency cases a fast and automatic detection method is required. Automatic Detection of arrhythmias using computational intelligence is the subject of this paper. In this paper we discuss a new algorithm that we developed and whereby we can extract some important features of the ECG complexes, and subsequently apply them to a CGPANN to classify the complexes for different arrhythmia types.

3.1 Proposed Algorithm for Extracting ECG Parameter

We developed an algorithm that is inspired by Pan-Tompkins's algorithm [14]. Our algorithm matches the Pan-Tompkin's algorithm only to the extent of pre-processing ECG signal to detect the location of R-peaks, inside the QRS wave. We however go further to detect the locations and amplitudes of P, Q, S and T waves also, so as to determine different morphological features of a beat. Fig 3 shows a typical ECG complex with fiducial points and a number of time domain parameters. Different arrhythmia types were classified using these parameters with CGPANN as discussed in the next subsection. Following are the key features of our algorithm:

The MIT-BIH database contains 48 ECG recordings under its arrhythmia category, which we used in our project. These 360 samples/sec recordings have their associated beat to beat attribute files, that have the following important annotation fields:

- R-peak sample number.

- Arrhythmia type Mnemonics e.g. V=PVC beat, N=Normal beat etc. There are a total of nineteen such type mnemonics, each beat has an associated mnemonic added to its R-peak by experts. We encoded the type mnemonics with numbers (1-19).

Our program code reads each ECG sample file and the attribute file containing type mnemonic of each beat, that are initially downloaded from the database. These type mnemonics which we encode to numbers are used as target values for training our network. The fiducial points that include the R-peaks besides others are then determined. These R-peaks perfectly matched those that we downloaded. The steps for fiducial points detection are shown in figure 4 and explained below.

The signal is band limited by an IIR low pass and a high pass filter. The 5-12 Hz pass band contains enough information to process. The filtered ECG signal is differentiated to extract high sloped QRS wave. It is then squared to make positive and intensify the QRS slope and integrated with a sliding window integrator, to obtain hump shaped pulses that correspond to QRS complexes. These steps are similar to the Pan-Tompkins's algorithm. Using thresholds for the slopes of the resulting signal and some logic functions, a rectangular window signal is formed at the QRS location. The width of these pulses equal the QRS width. Highest point of filtered ECG in this window is the R-peak. The lowest point between the window pulse onset and the R-peak is the Q-point, while that between the R-peak and the window pulse offset is the S-point.

The P-peak is the highest point, 120 ms to 200 ms before QRS onset, and is the maximum value found after gating the filtered ECG with a window pulse in this time slot. The T-peak occurs after QRS complex. Its exact location depends on heart rate and is determined by finding the highest point following the QRS offset. The points with zero value before and after the P and T peaks are the onset and offset points of these peaks, respectively. An ECG waveform with fiducial points marked x by the proposed algorithm is shown at bottom left of Figure 4.

With R-peak at center, the fiducial points of each complex are arranged as follows: P_onset, P_peak, P_offset, QRS_onset, Q, R_peak, QRS_offset, T_onset, T_peak and T_offset. After experimentation and some domain knowledge we got best results with the parameters: QRS-width, RQ length, RS length, RQ-slope and RS-slope. The parameters along with the corresponding beat type are grouped together for all the beats extracted from the database. Based on frequency of occurrence, we chose five important types out of the nineteen arrhythmias found in the record. Other arrhythmia types can be explored similarly. We determined Parameters for 200 beats of each arrhythmia type and grouped them together. All the arrhythmia types were uniformly distributed by reshuffling them before training the CGPANN.

3.2 Arrhythmia Detection Using CGPANN

Each parameter is normalized before it is applied as input to the network, using the following formula.

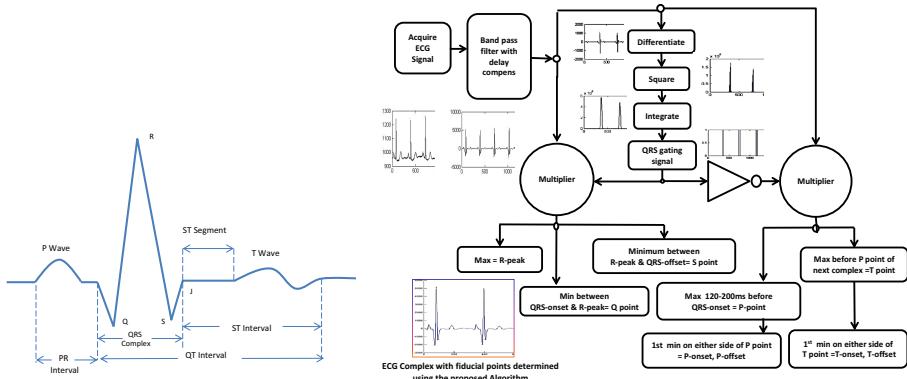


Fig. 3. QRS Complex with different parameters

Fig. 4. Block Diagram of ECG fiducial point detection using the proposed algorithm

$$\text{Param}(Norm.) = [\text{Param}-\text{Param}(\min)]/[\text{Param}(\max)-\text{Param}(\min)]$$

Out of the many other strategies we adapted the two fold cross validation strategy for its simplicity. In this strategy we run two experiments, with half of the data set for training and the other half for testing in one experiment while reversing the order in the second experiment. Hence there are 500 beats(100 beats of each arrhythmia type) for training and 500 beats for testing.

We chose the following five types of beats for classification, based on the frequency of their occurrence in the database: Normal (N) beat, Premature ventricular contraction (PVC) beat, Left bundle branch block (LBBB) beat, Atrial premature (AP) beat and Right bundle branch block (RBBB) beat.

A separate network is trained and tested for each arrhythmia type. All the networks are then connected in parallel, each getting the same inputs but having a separate output. The advantage of this modular approach is that more networks for different arrhythmia types can be added to it, if required. An output value above 0.5 indicates the presence of arrhythmia and made equal to 1, while a value below 0.5 indicates no arrhythmia and is made equal to 0.

After experimentation, the network with $10 \text{ rows} \times 10 \text{ columns}$ i.e. 100 nodes, each node having two inputs, gave the best results. We applied parameters for each beat and its related arrhythmia type, as the input and target output respectively. The actual and target outputs were compared. The fitness of a candidate network was determined from the percentage of correct matches. An initial population of ten random networks is generated and their fitness calculated. In the next generation, the fittest genotype becomes parent. Nine different offspring are generated by randomly mutating (mutation rate=10%) the weights, connections and activation functions of the parent network. Nine offspring and the parent form the next generation. Again the fitness of all the genotypes is determined. This process repeats for one million generations. We then tested the performance of our trained system with the test data.

3.3 Related Work

Mohammad Zadeh et al. in [12] discuss heart rate variability (HRV), a type of arrhythmia. As it is non-linear in nature, it can be determined using non-linear techniques like ANN. They used a multilayer neural network trained with error back propagation method.

In [22] Zhang et al. applied wavelet transform to ECG complexes obtained from the MIT-BIH database. The coefficients so obtained were optimized using principal component analysis (PCA) and Independent component analysis (ICA). These coefficients together with R-R interval are fed as input to a network running ID3, thus classifying the waveforms for arrhythmia. In [14] Pan et al. implemented an algorithm that exploits the high slope of the R-peak. The ECG signal is passed through a bandpass filter followed by squaring and integration. Applying a two level thresholding the QRS waves are determined.

In [6] the authors exploited Pan-Tompkins's algorithm to form a set of inputs for feed forward ANN using Levenberg-Marquardt training algorithm. The set comprises of the RQ and RS slopes of QRS complex, T-peak and P-wave coefficients from the modelled parabola and the average R-R interval. The network classified Normal, Ventricular Ectopic and Supraventricular beats. In [20] a real time wearable arrhythmia detection system is presented. A PSoC system filters the ECG and determines its Discrete Fourier Transform coefficients. The beats were classified for arrhythmia using many small Kohonen's Self Organizing Maps (KSOM).

Table 1. Testing results for five independent evolutionary runs using Two fold cross validation and result comparison with contemporary work **N:** Normal Beat, **LBBB:** Left Bundle Branch Block Beat, **RBBB:** Right Bundle Branch Block Beat, **Acc:** Accuracy, **Sen:** Sensitivity, **Spec:** Specificity

| Fold-1 | N | | | LBBB | | | RBBB | | |
|-------------------|-------------|-----------|-----------|------------|------------|------------|-------------|------------|------------|
| | Acc | Sen | Spec | Acc | Sen | Spec | Acc | Sen | Spec |
| Exp 1 | 93.4 | 79 | 97 | 100 | 100 | 100 | 96.6 | 99 | 96 |
| Exp 2 | 94.8 | 83 | 97.75 | 100 | 100 | 100 | 96.8 | 96 | 97 |
| Exp 3 | 95.2 | 85 | 97.75 | 100 | 100 | 100 | 97.2 | 98 | 97 |
| Exp 4 | 94.8 | 82 | 98 | 100 | 100 | 100 | 96.8 | 100 | 96 |
| Exp 5 | 94.6 | 88 | 96.25 | 100 | 100 | 100 | 96.8 | 100 | 96 |
| Average | 94.56 | 83.4 | 97.35 | 100 | 100 | 100 | 96.84 | 100 | 96.4 |
| Fold-2 | | | | | | | | | |
| Exp 1 | 88 | 73 | 91.75 | 100 | 100 | 100 | 99.8 | 100 | 99.75 |
| Exp 2 | 96 | 92 | 97 | 100 | 100 | 100 | 95.4 | 78 | 99.75 |
| Exp 3 | 91 | 76 | 94.75 | 100 | 100 | 100 | 89.2 | 100 | 86.5 |
| Exp 4 | 95.6 | 92 | 96.5 | 100 | 100 | 100 | 97.8 | 99 | 97.5 |
| Exp 5 | 95.4 | 92 | 96.25 | 99.8 | 99 | 100 | 99.4 | 97 | 100 |
| Average | 93.2 | 85 | 95.25 | 99.96 | 99.8 | 100 | 96.32 | 94.8 | 96.7 |
| Result Comparison | | | | | | | | | |
| Masih et al. [19] | 98.75 | 98.41 | 99.03 | 98.93 | 94.23 | 99.41 | 99.10 | 96.43 | 99.40 |
| Proposed Method | 96 | 92 | 98 | 100 | 100 | 100 | 99.8 | 100 | 100 |

In [3] the authors classified six different arrhythmia types using Radial Basis function Neural Network (RBFNN). Four time based morphological features of a beat were used as the inputs to the network. In [5] the authors classified the ECG for Bradycardia, Tachycardia, Bundle Branch Block (BBB), Incomplete Bundle Branch Block, Supraventricular Tachycardia (SVT) and Ventricular Tachycardia (VT). Using frequency and time based features of the beat as inputs to an ANN they performed the classification.

In [19] the authors applied HRV and ECG features to genetic programming (GP) algorithm. The best features are automatically selected by the GP. The classified arrhythmia types are: Normal Beats (NB), Left Bundle Branch Block Beats (LBBB), Right Bundle Branch Block Beats (RBBB), Premature Ventricular Contraction (PVC), Fusion of Ventricular and normal beats (FUSION), Atrial Premature Contraction (APC) and Paced Beats (PB). Table 1 shows results for the first three types and are presented for comparison with our proposed algorithm.

4 Results and Analysis

Table 1 shows our testing results and those of Masih et al. [19] for the same arrhythmia types, for comparison. Three of the five beat types i.e. Normal, LBBB and RBBB were classified with high accuracy. The results of the other two types were not upto the mark and they need more experimentation. The unsatisfactory results for the later two were obtained because the parameter set chosen for input bears less information for them as compared to the former three. Conventionally physicians look for the ECG complexes in different ECG leads (I, II, III, aVR, aVL, aVF, V1-V6) to diagnose a patient for arrhythmia. In our case we trained our system with features extracted from only ECG lead II in the dataset. Still we expect good results if different feature subsets are used for classifying each arrhythmia type. For this we plan to omit those features for each beat type, for which the number of events that their values lie outside standard deviation $\pm 20\%$, are higher than a preset threshold. This way only the most relevant features shall be applied as input, making the training faster. We are working to classify a total of ten arrhythmia types using the feature selection strategy discussed above. The feature set shall also be extended, so that it shall include not only more time based features but also time-frequency features like wavelet transform coefficients.

It can be observed that the accuracy for LBBB obtained with our proposed algorithm is higher than that obtained by Masih et al. while it is lower for N and RBBB. We hope to improve the accuracy for all the types by implementing the options discussed above. Following are the performance metrics for the experiments.

Accuracy = $(TP + TN) / N$; Sensitivity = $TP / (TP + FN)$; Specificity = $TN / (FP + TN)$; Where N : Number of samples; TP : True Positive; TN : True Negative;

FP : False Positive; FN : False Negative

5 Future Directions and Conclusion

We developed an arrhythmia classification system using CGPANN. The idea is to determine the fiducial points of an ECG complex, and from them some morphological features. We selected the set of best parameters for classifying major arrhythmia types. The ECG signals with arrhythmia cases were picked from the MIT-BIH database and processed in a number of steps before the parameters were determined and applied to the network. Out of five arrhythmia types, the normal beat, LBBB beat and RBBB beat, were classified with very high accuracy on average. Experiments with the other types shall be carried out after adding more features to the input parameter set and picking the best input set for each type using the procedure outlined in the previous section.

In the next phase of our research we intend to classify the beats with parameters determined using time-frequency analyses techniques like wavelet transform, in addition to the existing ones. The techniques developed in this work will also open the way for classifying other physiological signals like those from brain and eyes.

References

1. Ahmad, A.M., Khan, G.M., Mahmud, S.A., Miller, J.F.: Breast cancer detection using cartesian genetic programming evolved artificial neural networks. In: Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference, GECCO 2012, pp. 1031–1038. ACM, New York (2012)
2. Buk, Z., Koutník, J., Šnorek, M.: NEAT in hyperNEAT substituted with genetic programming. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) ICANNGA 2009. LNCS, vol. 5495, pp. 243–252. Springer, Heidelberg (2009)
3. Dogan, B., Korurek, M.: Performance evaluation of radial basis function neural network on ecg beat classification. In: 14th National Biomedical Engineering Meeting, BIYOMUT 200, pp. 1–4. IEEE (2009)
4. Gomez, F., Schmidhuber, J., Miikkulainen, R.: Accelerated neural evolution through cooperatively coevolved synapses. *J. Mach. Learn. Res.* 9, 937–965 (2008)
5. Gothwal, H., Kedawat, S., Kumar, R.: Cardiac arrhythmias detection in an ecg beat signal using fast fourier transform and artificial neural network. *Journal of Biomedical Science and Engineering* 4(4), 289–296 (2011)
6. Jokić, S., Krčo, S., Delić, V., Sakacć, D., Lukić, Z., Turukalo, T.: An efficient approach for heartbeat classification. *Computers in Cardiology*, 991–994 (2010)
7. Khan, G.M., Khan, S., Ullah, F.: Short-term daily peak load forecasting using fast learning neural network. In: 2011 11th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 843–848. IEEE (2011)
8. Khan, G.M., Miller, J.F., Halliday, D.M.: Evolution of cartesian genetic programs for development of learning neural architecture. *Evolutionary Computation* 19(3), 469–523 (2011)
9. Khan, M.M., Khan, G.M., Miller, J.F.: Evolution of neural networks using cartesian genetic programming. In: IEEE Congress on Evolutionary Computation, pp. 1–8 (2010)
10. Koza, J.R.: *Genetic Programming II: Automatic Discovery of Reusable Subprograms*. MIT Press (1994)

11. Miller, J.F., Thomson, P.: Cartesian genetic programming. In: Poli, R., Banzhaf, W., Langdon, W.B., Miller, J., Nordin, P., Fogarty, T.C. (eds.) EuroGP 2000. LNCS, vol. 1802, pp. 121–132. Springer, Heidelberg (2000)
12. Mohammadzadeh-Asl, B., Setarehdan, S.: Neural network based arrhythmia classification using heart rate variability signal. In: Proceedings of the EUSIPCO (2006)
13. Moriarty, D.: Symbiotic Evolution of Neural Networks in Sequential Decision Tasks. PhD thesis, University of Texas at Austin (1997)
14. Pan, J., Tompkins, W.: A real-time qrs detection algorithm. IEEE Transactions on Biomedical Engineering (3), 230–236 (1985)
15. Schmidhuber, J., Wierstra, D., Gomez, F.: Evolino: Hybrid neuroevolution/optimal linear search for sequence prediction. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI), pp. 853–858. Professional Book Center (2005)
16. Stanley, K.O., Bryant, B.D., Miikkulainen, R.: Real-time neuroevolution in the nero video game. IEEE Transactions on Evolutionary Computation 9, 653–668 (2005)
17. Stanley, K.O., D'Ambrosio, D.B., Gauci, J.: A hypercube-based encoding for evolving large-scale neural networks. Artif. Life 15(2), 185–212 (2009)
18. Stanley, K.O., Miikkulainen, R.: Evolving neural network through augmenting topologies. Evolutionary Computation 10(2), 99–127 (2002)
19. Tavassoli, M., Ebadzadeh, M., Malek, H.: Classification of cardiac arrhythmia with respect to ecg and hrv signal by genetic programming
20. Valenza, G., Lanatà, A., Ferro, M., Scilingo, E.: Real-time discrimination of multiple cardiac arrhythmias for wearable systems based on neural networks. In: Computers in Cardiology, pp. 1053–1056. IEEE (2008)
21. Yao, X.: Evolving artificial neural networks. Proceedings of the IEEE 87(9), 1423–1447 (1999)
22. Zhang, L., Peng, H., Yu, C.: An approach for ecg classification based on wavelet feature extraction and decision tree. In: 2010 International Conference on Wireless Communications and Signal Processing (WCSP), pp. 1–4. IEEE (2010)

Artificial Neural Networks and Principal Components Analysis for Detection of Idiopathic Pulmonary Fibrosis in Microscopy Images

Spiros V. Georgakopoulos¹, Sotiris K. Tasoulis¹, Vassilis P. Plagianakos¹,
and Ilias Maglogiannis²

¹ Department of Computer Science and Biomedical Informatics,
University of Central Greece, Papaxiropoulou 2–4, Lamia, 35100, Greece

{spirosgeorg, stas, vpp}@ucg.gr

² Department of Digital Systems,

University of Piraeus,
Grigoriou Lampraki 126, Piraeus, 18532, Greece
imaglo@unipi.gr

Abstract. In this study we present a computer assisted image identification and recognition tool that aims to help the diagnosis of idiopathic pulmonary fibrosis in microscopy images. To this end, we use principal components analysis to reduce the dimensionality of the data and subsequently we perform classification using Artificial Neural Networks. The proposed approach succeeded in locating the pathological regions and achieved high quality results in terms of classification accuracy.

Keywords: Pattern Recognition, Artificial Neural Network, Principal Components Analysis, Idiopathic Pulmonary Fibrosis, Classification.

1 Introduction

For reasons that are yet unknown [2], lungs are suffering from a chronic and progressive disease called Idiopathic Pulmonary Fibrosis (IPF), also referred to as cryptogenic fibrosis alveolitis. Patients with IPF have median survival, about 3 years, similar to that of clinical non-small cell lung cancer. IPF usually affects patients between the age of 50 to 70 years, with male preponderance, evidenced by a male to female ratio 2:1. The number of patients diseased by IPF is estimated at 10.7 cases per 100,000 men and 7.4 cases per 100,000 women and keeps rising [8]. An efficient therapy for IPF still remains to be found [21].

Advanced developments in technology enabled the designing of better cameras [24] and improved fluorescent probes [7]. This has strongly influenced medical and biological research due to expanding capabilities of the light microscope. All these advancements have made it feasible for the science of pattern recognition to contribute to the diagnostic process through computer imaging and image analysis [14]. In [25] an image characterization tool have been developed based on Fractal Analysis and Fuzzy clustering for the quantification of degree of the

IPF. Also in [14], a method that classifies similar microscopy images of lung tissues section has been proposed. Additional tools have been presented for the assessment of testicular interstitial fibrosis [23,22], liver fibrosis [3,6,15], lung fibrosis [11] and the study of micro vascular circulating leukocytes [10].

The improvement of quality imaging in digital cameras is linked to the increased image resolution and thus the dimensionality. Processing and handling data becomes more challenging as dimensionality grows due to the effect high dimensionality has on distance or similarity. It has been shown that the difference between data becomes negligible in such cases [4]. In order to deal with this problem, we usually reduce the dimensionality of the data. A very effective technique to achieve this is to project the high dimensional data onto a lower dimensional subspace. One of the most widely used methods for this purpose is Principal Components Analysis (PCA) [13]. The basic idea of PCA is that it reduces the dimensionality of the data, while retain as much of the variation as possible. Dimensionality reduction with PCA has great significance, since the amount of memory required as well as the computational cost is reduced. Furthermore, using the lower dimensionality data we can avoid overfitting.

In this study we present a computer assisted image identification and recognition tool that aims to help diagnose the idiopathic pulmonary fibrosis in microscopy images. To this end, we use Artificial Neural Networks (ANN) for the classification of the microscopy images. ANN provide a powerful tool to analyse and model clinical data in many medical applications [1]. The basic idea of ANN is that it defines a non-linear decision surface for each training medical dataset and new instances are classified into known classes. Here we use one of the most common structure of ANN, named Feedforward Neural Network (FNN), which has been used in several disease diagnosis tasks [1].

The rest of this paper is structured as follows. In Section 2, the artificial neural networks and the principal components analysis method are presented. In Section 3, we introduce the proposed methodology for the detection and recognition of IPF and in Section 4 we present the experimental results and analysis. We conclude with pointers for future work.

2 Background Methods

In this section we review the basic tools used in the proposed methodology. In particular, we firstly present ANN for classification and then the PCA method for dimensionality reduction is reviewed. Finally, we provide brief introduction to the Cellular automata utilized for post-processing the classification results.

2.1 Artificial Neural Networks

Artificial Neural Networks are classification methods that have inherent parallelism in their structure and high performance rate compared to other classifiers. We use them to handle complex and difficult real-life problems in many fields of science [9].

One critical parameter that can affect the prediction of the network is its architecture. The nodes (neurons) of a network are organized into layers and the connections between neurons possess connection weights. The most common used structure is the Feedforward Neural Network (FNN), which consists of one input layer, one output layer and some intermediate layers, referred to as the hidden layers. The information in FNN spreads from each neuron of a layer to all neurons of the next layer (fully interconnected FNNs). As we increase the number of neurons in the hidden layers the number of weights also increases. Thus, training is more difficult, but the FNN has more degrees of freedom and is capable to solve more complex problems. However, if too many hidden neurons are utilized, the generalization capability of FNN may be hindered due to overfitting.

To succeed in training FNNs, we have used the backpropagation (BP) training algorithm [19]. The requirements of the BP algorithm is a vector with input patterns, x and a vector with targets, y , respectively. The input x_i is associated with the output o_i . Each output is compared to its corresponding desirable target and the difference give the error. Our goal is to find weights that minimize the cost function, through a number of iterations

$$E(w) = \frac{1}{n} \sum_{p=1}^P \sum_{j=1}^{N_L} (o_{j,p}^L - y_{j,p})^2 \quad (1)$$

where P the number of patterns, $o_{j,p}^L$ the output of j neuron that belongs to L layer, N_L the number of neurons in output layer, $y_{j,p}$ the desirable target of j neuron of pattern p .

The only parameter that can be minimized in the error function $E(w)$ is the weight vector w . This transforms the neural network training problem to a non-linear problem of least squares [27]. In this way we benefit from the existence of several methods to solve it [26].

2.2 Principal Component Analysis

The aim of the PCA method is the dimension reduction of the dataset, without significant loss of information. That is achieved by transforming the original data to a new set of variables, the principal components [13]. The principal components correspond to the direction of higher variance of covariance matrix of data. Lets suppose a dataset $X_{L \times J}$, with L patterns and J variables, μ the mean of X . We compute the covariance matrix

$$C = \frac{1}{L} \sum_{i=1}^L (X - \mu)(X - \mu)^T. \quad (2)$$

Then we compute the eigenspace $PC_{k \times L}$ which is consisted by the k eigenvectors that correspond to the k largest eigenvalues of the covariance matrix. Subsequently, the dataset in projected on the eigenspace follows:

$$w_i = PC_i(X - \mu), \quad (3)$$

where w_i is the projection of i th pattern onto $PC_{k \times L}$ (usually referred as weighted vector).

2.3 Cellular Automata

Cellular automata (CA) are discrete dynamical system that model complex behavior based on simple, local rules animating cells on a lattice. They were introduced by Ulam and von Neumann [16].

The cellular automata consist of a regular grid of cells and each of these cells can be in only one of the finite number of possible states. The state of each cell specified by the previous states of its surrounding cells, which called neighborhoods and is updated synchronously in discrete time steps. The rules that describe the update of each cell usually determined as a rule table and contain every possible neighborhood configuration of states [18].

The most common structure of neighborhoods in 2-dimensional grid are the von Neumann and Moore neighborhoods. The von Neumann neighborhoods consist of the 4 cells orthogonally surrounding the central one and Moore neighborhoods consist of the 8 cells surrounding the central cell. In this study, we apply CA principals to post-process the classification results.

3 The Proposed Method

Images captured using Nikon ECLIPSE E800 microscope and a Nikon digital camera DXM1200 at a magnification of 4x were used in this study. The task is to detect and recognize idiopathic pulmonary fibrosis, whose variability concerning the severity of the lesions it incurs in the lung is great, when assessed by microscopy histological images [11]. The basic tools in the proposed method, is PCA to reduce the dimensionality of image dataset and FNN to perform classification (Figure 1). The main idea is to disassemble each image into blocks of specified window size and reduce their dimensionality using PCA and then classify each window using an FNN.

Age- and sex- matched, 6-8 week-old mice were used for the induction of pulmonary fibrosis by a single intravenous injection with a dose of 100mg/Kg of body weight (100 mg/kg body weight; 1/3 LD50; Nippon Kayaku Co. Ltd., Tokyo). Bleomycin administration initially induces lung inflammation that is followed by a progressive destruction of the normal lung architecture. To monitor disease initiation and progression, mice were sacrificed at 7, 15 and 23 days after bleomycin injection. Mice injected with saline alone and sacrificed 23 days post injection, served as the control group. For pathology assessment, at each time point, bronchoalveolar lavage has been performed (3x 1ml Saline) for the estimation of total and differential cell populations. Subsequently, after perfusion of lungs via the heart ventricle with 10ml Phosphate Buffer Saline, lungs were then removed, weighed dissected and collected, for histology. Sagittal sections from

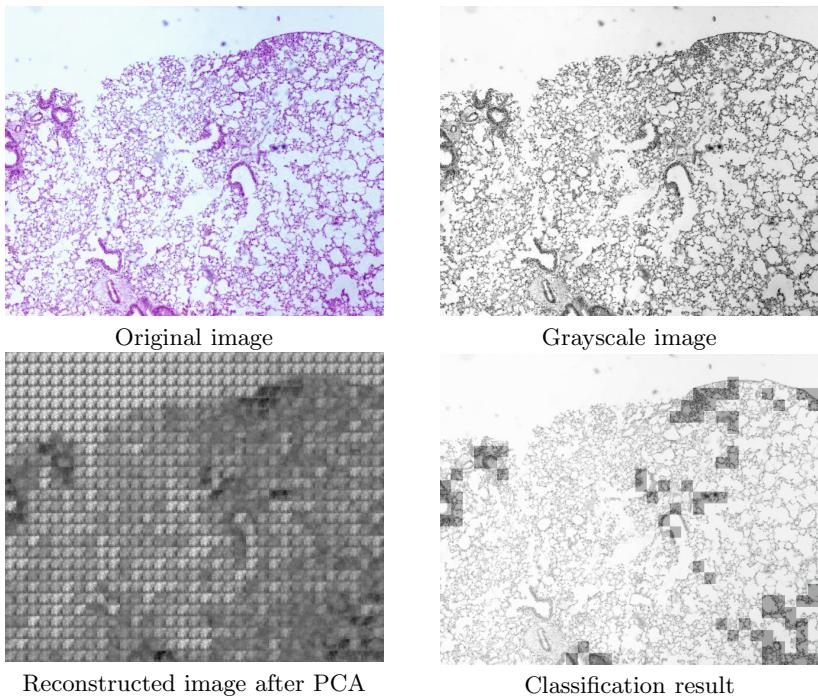


Fig. 1. The original microscopy image (top left), the grayscale image (top right), the reconstructed image after the application of the PCA procedure (lower left) and the classification result (lower right)

the right lung were used for Hematoxylin and Eosin staining and histopathologic analysis. Finally, we retrieve the images of these lungs with the assistance of light microscope.

The proposed method consists of three phases as presented in Figure 2. The first phase is the preprocess, that refers to the training set selection. Each image is partitioned to $n \times n$ subimages - blocks. Then, the physicians, select the m most representative subimages for each category (normal or pathological). Subsequently, the subimages are converted to grayscale in order to represent them as a matrix of integer values that are within the range [0, 255]. We have then created the training set consisted of m subimages (patterns) of n^2 size. For the physician to classify a block of the image, the block needs to be of sufficient size n . If it is too small, it may be difficult to characterize it with confidence. On the other hand, if it is too big, it may contain parts of normal, as well as pathological region tissue. After consultation with the physicians, we have created 35×35 pixel blocks.

The second phase is the classification. Here we use PCA to reduce the dimensionality of the m blocks from n^2 to k , where k is the number of the most important principal components, given by Kaiser criterion [12]. The visual difference between the normal and pathological subimages after the PCA procedure

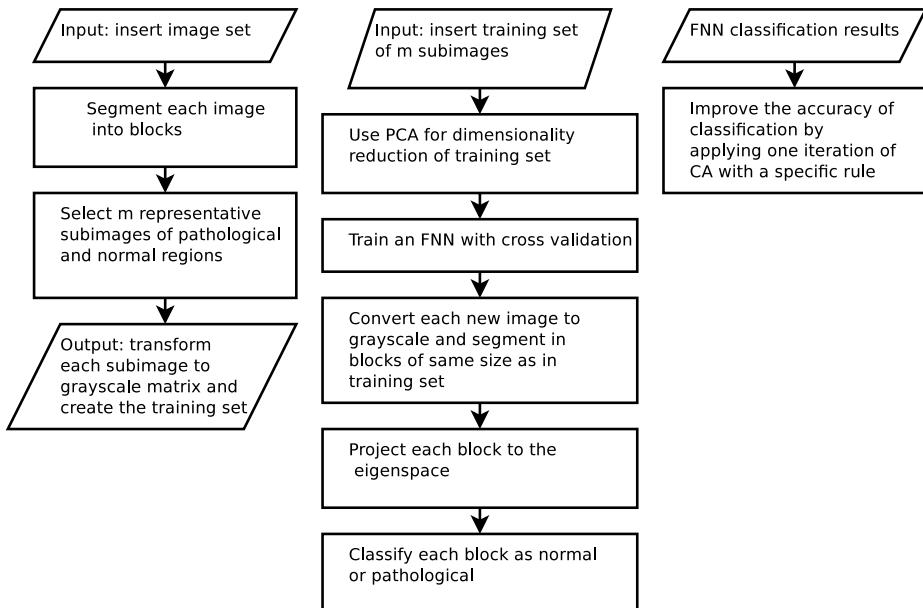


Fig. 2. The preprocess (left), the classification (middle) and the postprocess (right) phases of the proposed methodology for IPF microscopy image classification

is further highlighted. We train an FNN with the transformed training set using k-fold cross validation. The next step is the identification and characterization of normal and pathological regions of a new input image. Firstly, we divide the new image into subimages of same size as those in the training set and convert them to grayscale. Then, the dimensionality of the subimages is reduced using PCA as explained above and finally the trained FNN classify them.

Finally, to further improve the classification accuracy, we apply one iteration of cellular automata.

4 Experimental Results

In this section, we present the results of our methodology for identification and recognition of IPF into lung images that correspond to 7, 15 and 23 days after bleomycin administration to mice (see left column in Figure 3). The lung images are captured using Nikon ECLIPSE E800 microscope and Nikon DXM1200 digital camera.

The proposed method is implemented in C++ with the aid of Armadillo [20] and OpenCV [5] libraries. The program was built using gcc version 4.7.2 and run on a Intel Core i3-2100 computer with Linux Mint 14 operating system. The CPU time for the training of the FNN was 4 sec.

After we have constructed the training set of lung images produced with the same procedure as the left column images in Figure 3, we apply the PCA for

dimension reduction and proceed with the training of an FNN according to the methodology described in Section 3.

In the classification stage, we utilize an FNN with k neurons on the input layer (given by the Kaiser's criterion), one hidden layer with 10 neurons having sigmoid activation functions and one output neuron with sigmoid activation function. The training algorithm used was the RPROP [17] with learning rate 0.2 and 4-fold cross validation.

After the successful training, the FNN is used to identify pathological regions into the new set of microscopy images (see left column in Figure 3) according to the methodology described above. Figure 3 exhibits the reconstruction of each image and the classification result of the FNN for each of the images retrieved after 7, 15 and 23 days, respectively. Subsequently, we perform one iteration of cellular automaton, with the rule: *if all neighbors of predicted block are normal, then you are normal too*. The reason of applying the cellular rule is to reduce the outliers and the classification error. The CA actually filters the results.

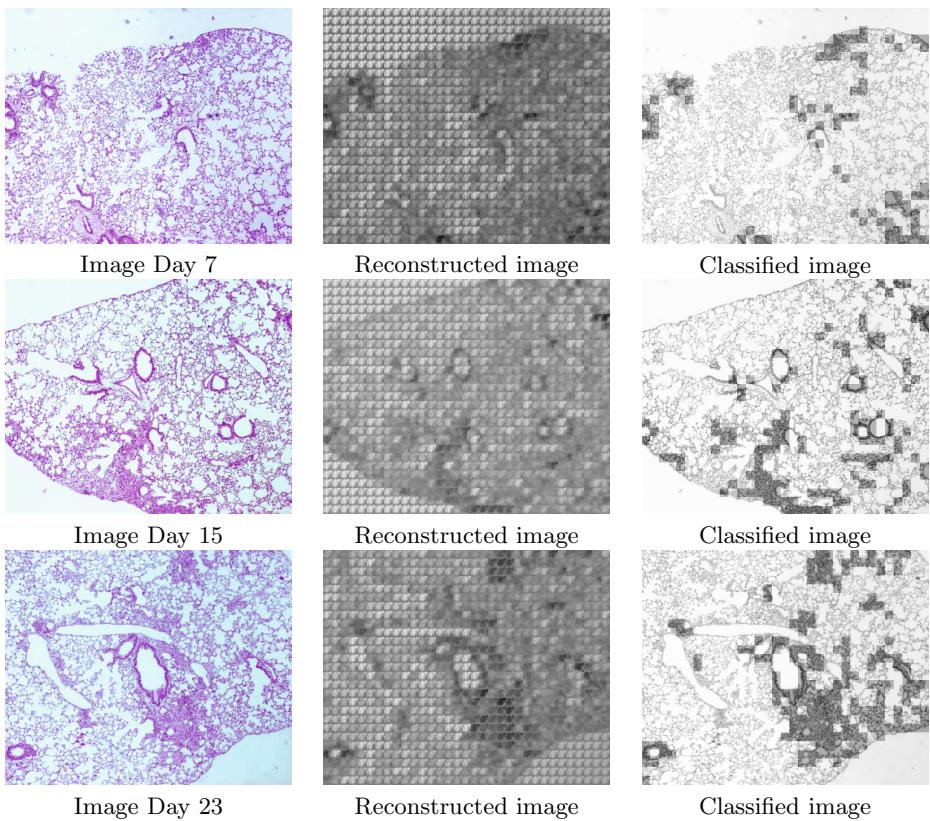


Fig. 3. Original, reconstructed and classified images after 7, 15 and 23 days

Table 1. The ratio of pathological over normal region calculated by the proposed method in comparison with the real score given by physicians for each day

| | Pathological blocks (%) | | |
|-------------------|-------------------------|--------|--------|
| | Day 7 | Day 15 | Day 23 |
| Proposed Approach | 11.09 | 19.34 | 28.57 |
| Physicians Score | 12.46 | 19.01 | 23.81 |

For every image, the physicians give a score percentage for the pathology region of the image over the normal. The proposed method achieved satisfactory result, close to real ones [2]. In Table 1, we report the score percentage of pathological over normal blocks for all images. As it is evident from the Table 1, the ratio of the pathological over the normal regions calculated by our approach is close to the real percentage given by the physicians. Moreover, it is obvious from Figure 3 that the method succeeds in correctly identifying the pathological regions.

5 Conclusions

In the present paper, a computer-assisted image identification and characterization tool was introduced for the quantification of degree of Idiopathic Pulmonary Fibrosis in medical images. The proposed method uses Principal Components Analysis for dimensionality reduction and Artificial Neural Networks to solve the classification task. The results are very promising regarding of the automatic assessment and identification of pathological regions.

Future research directions include characterization of large image sets as well as comparison against other similar approaches.

Acknowledgments. The authors would like to thank the European Union (European Social Fund ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: "Heracleitus II. Investing in knowledge society through the European Social Fund." for financially supporting part of this work. Part of this work has been also funded by Program: Thalis - Interdisciplinary Research in Affective Computing for Biological Activity Recognition in Assistive Environments / Operational Program "Education and Lifelong Learning".

References

1. Al-Shayea, Q.K.: Artificial neural networks in medical diagnosis. International Journal of Computer Science Issues 8(2) (2011)
2. Antoniou, M.K., Pataka, A., Bouros, D., Siafakas, M.N.: Pathogenetic pathways and novel pharmacotherapeutic targets in idiopathic pulmonary fibrosis. Pulmonary Pharmacology and Therapeutics 20(5), 453–461 (2007)

3. Bedossa, P., Dargère, D., Paradis, V.: Sampling variability of liver fibrosis in chronic hepatitis c. *Hepatology* 38(6), 1449–1457 (2003)
4. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is "nearest neighbor" meaningful. In: Int. Conf. on Database Theory, pp. 217–235 (1999)
5. Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media (2008)
6. Caballero, T., Pérez-Milena, A., Masseroli, M., O'Valle, F., Salmerón, F.J., Del-Moral, R., Sánchez-Salgado, G.: Liver fibrosis assessment with semiquantitative indexes and image analysis quantification in sustained-responder and non-responder interferon-treated patients with chronic hepatitis c. *Journal of Hepatology* 34(5), 740–747 (2001)
7. Chalfie, M., Tu, Y., Euskirchen, G., Ward, W.W., Prasher, D.C.: Green fluorescent protein as a marker for gene expression. *Science* 263, 802–805 (1994)
8. Coultas, D.B., Zumwalt, R.E., Black, W.C., Sobonya, R.E.: The epidemiology of interstitial lung diseases. *American Journal of Respiratory and Critical Care Medicine* 150(4), 967–972 (1994)
9. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall PTR, Upper Saddle River (1998)
10. Hussain, M.A., Merchant, S.N., Mombasawala, L.S., Punyani, R.R.: A decrease in effective diameter of rat mesenteric venules due to leukocyte margination after a bolus injection of pentoxifylline digital image analysis of an intravital microscopic observation. *Microvascular Research* 67(3), 237–244 (2004), <http://www.sciencedirect.com/science/article/pii/S0026286204000081>
11. Izicki, G., Segel, M.J., Christensen, T.G., Conner, M.W., Breuer, R.: Time course of bleomycin-induced lung fibrosis. *International Journal of Experimental Pathology* 83(3), 111–119 (2002)
12. Jackson, D.A.: Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches. *Ecology* 74(8) (1993)
13. Jolliffe, T.I.: Principal Component Analysis. Springer (2002)
14. Maglogiannis, I., Sarimveis, H., Kiranoudis, C.T., Chatzioannou, A.A., Oikonomou, N., Aidinis, V.: Radial basis function neural networks classification for the recognition of idiopathic pulmonary fibrosis in microscopic images. *IEEE Transactions on Information Technology in Biomedicine* 12(1), 42–54 (2008)
15. Masseroli, M., Caballero, T., O'Valle, F., Del-Moral, R., Perez-Milena, A., Del-Moral, R.: Automatic quantification of liver fibrosis: design and validation of a new image analysis method: comparison with semi-quantitative indexes of fibrosis. *Journal of Hepatology* 32(3), 453–464 (2000)
16. Neumann, J.V.: Theory of Self-Reproducing Automata. University of Illinois Press, Champaign (1966)
17. Riedmiller, M., Heinrich, B.: A direct adaptive method for faster backpropagation learning: The rprop algorithm. In: IEEE International Conference on Neural Networks, pp. 586–591 (1993)
18. Rosin, L.P.: Training cellular automata for image processing. *IEEE Transactions on Image Processing* 15(7), 2076–2087 (2006)
19. Rumelhart, E.D., Hinton, E.G., Williams, J.R.: Learning representations by back-propagating errors. In: Anderson, J.A., Rosenfeld, E. (eds.) *Neurocomputing: Foundations of Research*, pp. 696–699. MIT Press, Cambridge (1988)
20. Sanderson, C.: Armadillo: An open source c++ linear algebra library for fast prototyping and computationally intensive experiment. Tech. rep., NICTA (2010), <http://arma.sourceforge.net>

21. Selman, M., Pardo, A.: Idiopathic pulmonary fibrosis: an epithelial/fibroblastic cross-talk disorder. *Respiratory Research* 3(1) (2002)
22. Shiraiishi, K., Takihara, H., Naito, K.: Influence of interstitial fibrosis on spermatogenesis after vasectomy and vasovasostomy. *Contraception* 65(3), 245–249 (2002)
23. Shiraiishi, K., Takihara, H., Naito, K.: Quantitative analysis of testicular interstitial fibrosis after vasectomy in humans. *Aktuelle Urologie* 34(4), 262–264 (2003)
24. Shotton, D.: Image resolution and digital image processing in electronic light microscopy. *Cell Biology, a Laboratory Handbook* 3, 85–98 (1998)
25. Tasoulis, S.K., Maglogiannis, I., Plagianakos, V.P.: Unsupervised detection of fibrosis in microscopy images using fractals and fuzzy c-means clustering. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H. (eds.) *AIAI 2012. IFIP AICT*, vol. 381, pp. 385–394. Springer, Heidelberg (2012)
26. Wilamowski, M.B.: Neural network architectures and learning. In: *2003 IEEE International Conference on Industrial Technology*, vol. 1 (2003)
27. Yi, L.: A note on margin-based loss functions in classification. *Tech. rep., Statistics and probability letters* (2002)

Prediction of Air Quality Indices by Neural Networks and Fuzzy Inference Systems – The Case of Pardubice Microregion

Petr Hájek and Vladimír Olej

Institute of System Engineering and Informatics,
Faculty of Economics and Administration, University of Pardubice, Studentská 84,
532 10 Pardubice, Czech Republic
{petr.hajek, vladimir.olej}@upce.cz

Abstract. This paper presents a design of models for air quality prediction using feed-forward neural networks of perceptron and Takagi-Sugeno fuzzy inference systems. In addition, the sets of input variables are optimized for each air pollutant prediction by genetic algorithms. Based on data measured by the monitoring station of the Pardubice city, the Czech Republic, models are designed to predict air quality indices for each air pollutant separately and consequently, to predict the common air quality index. Considering the root mean squared error, the results show that the compositions of individual prediction models outperform single predictions of common air quality index. Therefore, these models can be applied to obtain more accurate one day ahead predictions of air quality indices.

Keywords: Air quality indices, modelling, prediction, optimization, neural networks, Takagi-Sugeno fuzzy inference systems.

1 Introduction

Air quality indices (AQIs) [1 to 10] are used to report on the state of air pollutants. The air pollutants are widely accepted as important determinants of adverse health effect [5]. AQIs use both a direct numerical expression and a linguistic description to help the public better understand the environmental conditions. The values of air pollutants are transformed into a dimensionless number characterizing the state of air pollution using the standards defining the maximum concentrations allowed. Therefore, increasing attention has been paid to the prediction of individual air pollutants recently. Recent studies have shown that predicting the air pollutants is a complex problem and, thus, computational intelligence and machine learning approaches have provided promising results [11,12]. The problem becomes even more complex when developing models for the prediction of AQIs that combine several air pollutants. Only few attempts have been made to address this issue [7]. So far these models have only been applied to classification tasks although much information is lost when transforming the values of AQIs into nominal values.

This paper will examine the prediction of AQIs using both feed-forward neural networks (FFNNs) of perceptron type and fuzzy inference systems (FISs) of Takagi-Sugeno type. Apart from the introduction section, the paper reviews the AQIs in section 2. Section 2 also introduces the AQI of the Czech National Institute of Public Health (CNIPH) and limit concentrations of substances in the air developed by the Czech Hydro-meteorological Institute (CHI). Then 34 input variables are consequently employed to the modelling in section 3. Data pre-processing is carried out using the imputation of missing values [13] and the optimization of the original set of input variables using correlation based filter and genetic algorithm [11,14]. Section 4 contains the design of models for AQI prediction. Basic notions of the methods for modelling, i.e. FFNNs [15,16,17] and Takagi-Sugeno FISs [18,19] are introduced in this section, too. Section 5 is focused on the modelling and the comparison of results obtained by FFNNs and Takagi-Sugeno FISs. Conclusion section discusses the contributions and limitations of the proposed models.

2 Air Quality Indices

Currently, several AQIs are monitored with different scales [1]. A widely used index is the AQI developed by Environmental Protection Agency (EPA) [2]. It is based on the Pollution Standards Index (PSI) which was initially established in response to health issues related to the deteriorating air quality. It is defined with respect to the five main common pollutants: carbon monoxide CO, nitrogen dioxide NO₂, ozone O₃, particulate matter PM₁₀ and sulphur dioxide SO₂. Modified versions of the AQI of EPA were developed by [3] taking into consideration the limit values ruling in Europe. The Revised AQI (RAQI) is derived from the AQI, and is a background arithmetic mean index and a background arithmetic mean entropy index [4]. In [5], an aggregate AQI was developed to represent long-term exposure to air pollutants. A uniform indexing scale using well pre-established air quality standards and, at the same time, accounting for local conditions assessed via statistical analysis of data in the Athens metropolitan area was proposed by [6]. A daily AQI was proposed by [7] to show exceeding limit values. In [8], an aggregated AQI was developed that takes into account the combined effects of five criteria pollutants (CO, SO₂, NO₂, O₃ and PM₁₀). The example of the development of an alternative AQI is used by [9] to illustrate issues related to quantifying the public health burden attributable to air pollution. There are also AQIs that use fuzzy logic evaluation instead of exact value calculation [20].

The AQI of the CNIPH was used as target variable in this study. The daily AQIs for individual air pollutants (AP) are expressed as AQIAP^t=AP^t/AP_{24Hmax}, where AP^t is the maximum value of AP in day t, and AP_{24Hmax} denotes maximum allowed daily concentration of the AP. If PM₁₀ and SO₂ are monitored at the same time, the index of synergy should be calculated as AQISNG^t=(PM₁₀^t+SO₂^t)/AP_{24Hmax}, where AP_{24Hmax} is the lower value of the two maximum allowed concentrations of PM₁₀ and SO₂. The common AQI (further only AQI^t) is then expressed as follows

$$AQI^t = 1/n \sum_{i=1}^n AQIAP_i^t, \quad (1)$$

where AQI_{AP}^t is the value of AQI for the i-the AP, and n denotes the number of APs. The AQI^t can be further transformed into $\langle 0,6 \rangle$ interval using linear discontinuous functions, for details see [21]. Based on the value of the AQI^t the state of air pollution can be classified into six classes (Table 1).

Table 1. AQI^t classes $\omega_i^t \in \Omega$ of the CNIPH

| AQI^t | Class description |
|-----------------------|--|
| $\langle 0,1 \rangle$ | Clean air, very healthy environment. |
| $\langle 1,2 \rangle$ | Satisfactory air, healthy environment. |
| $\langle 2,3 \rangle$ | Slightly polluted air, acceptable environment. |
| $\langle 3,4 \rangle$ | Polluted air, environment dangerous for sensitive population. |
| $\langle 4,5 \rangle$ | High polluted air, environment dangerous for the whole population. |
| $\langle 5,6 \rangle$ | Very high polluted air, harmful environment. |

Table 2 presents the maximum limits of the examined APs developed by the CHI. Where the daily limits were not available, the value of AP_{max} for the nearest time interval was used. This evaluation takes the possible influence of human health into account [10].

Table 2. Limits of air pollutants AP_{max} proposed by CHI

| AP | SO_2 | NO_2 | NO_x | O_3 | PM_{10} |
|---------------------------|-------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|
| average for AP_{max} | 24h [$\mu\text{g.m}^{-3}$] 125 | 1h [$\mu\text{g.m}^{-3}$] 200 | 1yr [$\mu\text{g.m}^{-3}$] 30 | 8h [$\mu\text{g.m}^{-3}$] 120 | 24h [$\mu\text{g.m}^{-3}$] 50 |

3 Problem Formulation

The data for our investigations are represented by the measurements from the city of Pardubice, Dukla station, the Czech Republic, for years 2009–2011. The CO substance was not measured in this locality. The data were obtained from the CHI and contain the average daily meteorological variables (such as T_{2m} – temperature 2m above terrain, V – wind velocity, θ – wind direction, H – relative air humidity, and SR – solar radiation), maximum daily emission variables (NO_2 , NO, NO_x , SO_2 , PM_{10} , O_3 , toluene (TLN) and benzene (BZN)), and other variables (working day). All of the variables used in this study were measured as presented in Table 3. Basic information about the studied locality is provided in Table 4. The Pardubice Dukla station is situated in a residential zone close to the Pardubice airport.

The input variables for the modelling of AQIs in the city of Pardubice, Dukla station, the Czech Republic, were obtained using measuring and calculations as follows: workingday (0,1); air pollutants (O_3 , TLN, BZN, NO_x , NO_2 , NO, SO_2 , PM_{10}); meteorological parameters (T_{2m} , SR, H, θ – average 6hr values of parameters (s_a, c_a) ($s_{a(0-5)}$, $s_{a(6-11)}$, $s_{a(12-17)}$, $s_{a(18-23)}$, $c_{a(0-5)}$, $c_{a(6-11)}$, $c_{a(12-17)}$, $c_{a(18-23)}$), average 6hr angle θ_a ($\theta_{a(0-5)}$, $\theta_{a(6-11)}$, $\theta_{a(12-17)}$, $\theta_{a(18-23)}$), average 6hr standard deviation σ_θ ($\sigma_{\theta(0-5)}$, $\sigma_{\theta(6-11)}$,

$\sigma_{\theta(12-17)}$, $\sigma_{\theta(18-23)}$), V – average 6hr angle θ_v ($\theta_{v(0-5)}$, $\theta_{v(6-11)}$, $\theta_{v(12-17)}$, $\theta_{v(18-23)}$)); and seasonal effect ($\sin(\text{day})$, $\cos(\text{day})$). The calculation of the standard deviation σ_θ of the θ was carried out based on the EPA recommendation [2] using [22,23] from the sequence of n angles θ_i . The average angle θ_a was determined from the average values of (s_a, c_a) , i.e. from $\sin(\theta_i)$ and $\cos(\theta_i)$, as follows

$$s_a = 1/n \sum_{i=1}^n \sin(\theta_i), \quad c_a = 1/n \sum_{i=1}^n \cos(\theta_i), \quad \theta_a = \arctg(s_a/c_a). \quad (2)$$

Table 3. Information about the measured variables

| Measured | | | | | |
|-----------------------|------------------|--|--|-------------------|----------|
| Instrument | variable | Method | | Unit | Interval |
| TEI, M49 | O ₃ | UVABS [UV-absorption] | | µg/m ³ | 1h |
| BTX ¹⁾ | TLN | GC [gaseous chromatography -R] | | µg/m ³ | 1h |
| BTX ¹⁾ | BZN | GC [gaseous chromatography -R] | | µg/m ³ | 1h |
| TEI, M42 | NO | CHLM [chemiluminescence] | | µg/m ³ | 1h |
| TEI, M42 | NO ₂ | CHLM [chemiluminescence] | | µg/m ³ | 1h |
| TEI, M17 | NO _x | CHLM [chemiluminescence] | | µg/m ³ | 1h |
| TEI, M43 | SO ₂ | UVFL [UV-fluorescence] | | µg/m ³ | 10min |
| TESMA, | PM ₁₀ | RADIO [radiometry–beta ray absorption] | | µg/m ³ | 1h |
| Vaisala ²⁾ | V | OPEL [optoelectronic method] | | m/s | 10min |
| Vaisala ²⁾ | θ | OPEL [optoelectronic method] | | deg | 10min |
| TM, HS 420 | H | CAP [capacitance sensor] | | % | 1h |
| TM, SG 420 | SR | TDM [temperature difference method] | | W/m ² | 10min |
| TM, PT 420 | T _{2m} | PT100 [resistance method] | | K | 1h |

Legend: TEI is Thermo Environmental Instrument, TESMA is Thermo ESM Andersen FH 62 I-R, ¹⁾BTX Syntech Spectras, ²⁾Vaisala-WAA25, WAW15, WAT12.

Table 4. Information about the locality

| | |
|--------------------|---|
| EOI classification | station type – background, zone type – urban, zone characteristics – residential |
| Localization | 50° 1' 26.531" North latitude 15° 45' 48.776" East longitude, altitude = 239 m |
| Supplementary data | plain terrain, poly-storeyed built-up area |

Legend: EOI is Exchange of Information classification.

Then the standard deviation σ_θ can be expressed according to [22,23] as

$$\sigma_\theta = \arcsin(\varepsilon)(1+(2/\sqrt{3}-1))\varepsilon^3, \text{ where } \varepsilon = \sqrt{1-(s_a^2 + c_a^2)}. \quad (3)$$

Similarly, wind velocity V can be determined as follows

$$V_{xa} = -1/n \sum_{i=1}^n V_i \sin(\theta_i), \quad V_{ya} = -1/n \sum_{i=1}^n V_i \cos(\theta_i), \quad \theta_v = \arctg(V_{xa}/V_{ya}), \quad (4)$$

where V_i is wind velocity, and the +x and +y directions and velocities align with east and north unit vector, respectively.

About 1.7% of data were missing. As shown in [13], an improvement in the accuracy of air quality prediction can be achieved using multiple imputations where a final estimate is composed of the outputs of several multivariate fill-in methods. This approach solves the problem of underestimation of the error variance. Therefore, the multiple imputation scheme was used to fill in the missing values.

The original set of input parameters was optimized using correlation based filter [11,14]. This filter optimizes the set of input parameters so that it evaluates the worth of a subset of input parameters (features) by considering the individual predictive ability of each feature along with the degree of redundancy between them. The objective function $f(\lambda)$, also known as Pearson's correlation coefficient, is optimized in this filter. The objective function can be expressed as

$$f(\lambda) = \frac{\lambda \times \zeta_{cr}}{\sqrt{\lambda + \lambda \times (\lambda - 1) \times \zeta_{rr}}}, \quad (5)$$

where λ is the subset of features, ζ_{cr} is the average feature to output correlation, and ζ_{rr} is the average feature to feature correlation. Genetic algorithm was used as a search method to maximize the objective function $f(\lambda)$. For further modelling it was necessary to optimize the set of input variables for each AQI so that the function f was maximized. After the optimization of input variables, the individual prediction models were designed for: $AQIO_3^{t+1}$, $AQINO_2^{t+1}$, $AQINO_x^{t+1}$, $AQISO_2^{t+1}$, $AQIPM_{10}^{t+1}$, and $AQISNG^{t+1}$ (synergy) in time $t+1$ (the time horizon was 1 day). The AQI^{t+1} was also calculated as a composition (eq. (12)), i.e. as the average (eq. (1)) of APs' predictions (obtained from eq. (6-11)). The output indices were the functions of the following input variables

$$AQIO_3^{t+1} = f(\text{workingday}(0,1)^t, O_3^t, c_{a(12-17)}^t, s_{a(6-11)}^t, \theta_{v(0-5)}^t, \text{sinday}^t, \text{cosday}^t, \theta_{a(12-17)}^t, \sigma_{\theta(0-5)}^t, \sigma_{\theta(6-11)}^t, \sigma_{\theta(18-23)}^t), \quad (6)$$

$$AQINO_2^{t+1} = f(TLN^t, BZN^t, NO_2^t, T_{2m}^t, c_{a(0-5)}^t, s_{a(6-11)}^t, s_{a(12-17)}^t, \text{cosday}^t, \theta_{a(0-5)}^t), \quad (7)$$

$$AQINO_x^{t+1} = f(BZN^t, NO_2^t, NO^t, T_{2m}^t, c_{a(12-17)}^t, s_{a(6-11)}^t, s_{a(12-17)}^t, \text{cosday}^t, \theta_{a(0-5)}^t, \theta_{a(12-17)}^t, \sigma_{\theta(0-5)}^t), \quad (8)$$

$$AQISO_2^{t+1} = f(\text{workingday}(0,1)^t, BZN^t, SO_2^t, s_{a(18-23)}^t, \text{sinday}^t, \theta_{a(0-5)}^t, \sigma_{\theta(0-5)}^t), \quad (9)$$

$$AQIPM_{10}^{t+1} = f(NO_2^t, c_{a(6-11)}^t, c_{a(12-17)}^t, c_{a(18-23)}^t, s_{a(6-11)}^t, \theta_{v(18-23)}^t, \text{cosday}^t, PM_{10}^t, \theta_{a(0-5)}^t, \sigma_{\theta(0-5)}^t), \quad (10)$$

$$AQISNG^{t+1} = f(BZN^t, NO_2^t, SO_2^t, \theta_{v(12-17)}^t, \text{cosday}^t, PM_{10}^t, \theta_{a(0-5)}^t, \sigma_{\theta(0-5)}^t), \quad (11)$$

$$AQI^{t+1} = K(AQIO_3^{t+1}, AQINO_2^{t+1}, AQINO_x^{t+1}, AQISO_2^{t+1}, AQIPM_{10}^{t+1}, AQISNG^{t+1}). \quad (12)$$

4 Model Design and Methods for Modelling

The prediction model for the AQI^{t+1} , based on eq. (12), is depicted in Fig.1. First, individual partial indices have to be predicted, namely $AQIO_3^{t+1}$, $AQINO_2^{t+1}$, $AQINO_x^{t+1}$, $AQISO_2^{t+1}$, $AQIPM_{10}^{t+1}$, $AQISNG^{t+1}$ using the models M_1, M_2, \dots, M_6 . The input variables for individual models are defined in eq. (6-11). Second, the composition (common average) AQI^{t+1} is calculated according to the eq. (1) from the individual prediction of $AQIO_3^{t+1}$, $AQINO_2^{t+1}$, $AQINO_x^{t+1}$, $AQISO_2^{t+1}$, $AQIPM_{10}^{t+1}$, and $AQISNG^{t+1}$, i.e. $n=6$. Then the output of the composition prediction, obtained by the model defined in eq. (12) and Fig. 1, represents the AQI^{t+1} for the city of Pardubice, Dukla Station, the Czech Republic.

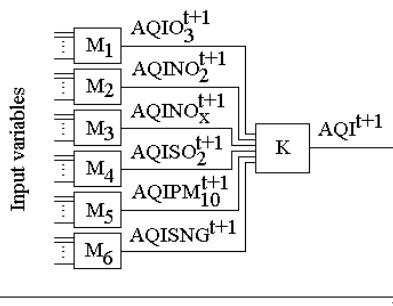


Fig. 1. Model for the prediction of AQI^{t+1} with composition

For further comparison, it was also possible to design the second model whose set of input variables was optimized in the same manner as for models M_1, M_2, \dots, M_6 , i.e. using the correlation based filter with genetic algorithm. After the optimization, the AQI^{t+1} is function of the following input variables in time t

$$AQI^{t+1} = f(\text{workingday}(0,1)^t, TLN^t, NO_2^t, T_{2m}^t, c_{a(6-11)}^t, c_{a(12-17)}^t, s_{a(6-11)}^t, \text{sinday}^t, PM_{10}^t, \theta_{a(0-5)}^t, \sigma_{\theta(0-5)}^t), \quad (13)$$

The analysis of results showed that the data for the AQI^{t+1} prediction change over time, and have nonlinear character. They are also heterogeneous, inconsistent, missing and uncertain. This character of data indicates that the models using neural networks and methods with uncertainty might perform well in the case of AQIs' prediction. Both the perceptron type FFNNs and Takagi-Sugeno FISs were employed to predict the target AQI^{t+1} . The structure of perceptron type FFNN [15,16,17] is given by the task which it executes. The output of perceptron type FFNN can be expressed for example as follows

$$y = \sum_{k=1}^K v_k \left(d \left(\sum_{j=1}^J w_{j,k} x_{j,k}^t \right) \right), \quad (14)$$

where y is the output of the FFNN (individual AQIs defined by eq. (6-11,13)), v_k is the vector of synapses' weights among neurons in the hidden layer and output neuron, $w_{j,k}$ is the vector of synapses' weights among input neurons and neurons in the hidden

layer, k is the index of neuron in the hidden layer, K is the number of neurons in the hidden layer, d is the activation function, j is the index of the input neuron, J is the number of the input neurons and per one neuron in the hidden layer, and $x_{j,k}^t$ is the input vector of the FFNN (the number of input variables for each model). In the process of learning, the values of synapse weights among neurons are adjusted. The most often used ones are gradient methods.

The general structure of FIS is defined in [18,19]. The base of rules consists of if-then rules. The rules are used for creating predicate clauses representing the base of FIS. The k -th if-then rule R_k in Takagi-Sugeno FIS can be written down in the following form

$$R_k: \text{if } x_1 \text{ is } A_{1,i(1,k)} \text{ AND } x_2 \text{ is } A_{2,i(2,k)} \text{ AND } \dots \text{ AND } x_j \text{ is } A_{j,i(j,k)} \text{ AND } \dots \text{ AND } x_m \text{ is } A_{m,i(m,k)} \text{ then } y_k = f(x_1, x_2, \dots, x_m), \quad (15)$$

where $A_{1,i(1,k)}, A_{2,i(2,k)}, \dots, A_{j,i(j,k)}, \dots, A_{m,i(m,k)}$, represent the values of linguistic variables, $f(x_1, x_2, \dots, x_m)$ is a linear or polynomial function (the function of input variable of individual models). The output level y_k of the k -th if-then rule R_k is weighted by the $w_k = \mu(x_1) \text{ AND } \mu(x_2) \text{ AND } \dots \text{ AND } \mu(x_m)$, where μ represents membership function. The final output y (AQI) of the Takagi-Sugeno FIS is the weighted average of all N outputs y_k , $k=1,2, \dots, N$.

5 Modelling and Analysis of the Results

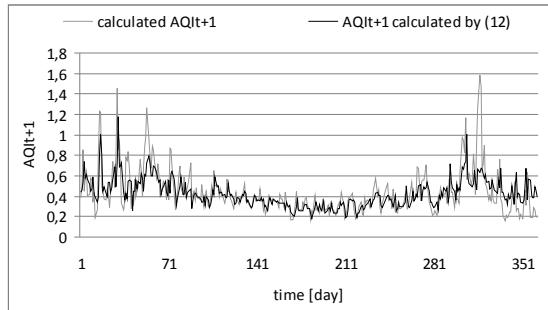
The quality of the AQIs' predictions is determined by the relation between training and testing set $O_{\text{train}}:O_{\text{test}}$. The most usual procedure in air quality prediction was followed. Therefore, the set of training data O_{train} (used in the process of learning) contained data from the years 2009 and 2010, and the set of testing data O_{test} covered the year 2011.

Further, the quality of FFNNs' prediction depends both on its structure and on the parameters of learning process. Based on numerous experiments, the back-propagation algorithm with momentum was chosen as the learning algorithm of the FFNN from the set of learning algorithms. The number of neurons in the hidden layer K was heuristically set as $J-1$, where J is the number of neurons in the input layer, i.e. $K=10$ for the model M_1 , $K=8$ for the model M_2 , etc. Other parameters were tested for the following values: the number of cycles in the learning process $c=\{50,100, \dots, 800\}$; learning rate $\eta=\{0.05,0.1,0.3\}$; momentum $m=\{0.2,0.3\}$, the constant term added to the derivation of the activation function before its value calculation for back-propagation from each neuron (increasing the resistance against overlearning) $k=0.1$; and the maximum difference between the target value and the value of neuron in the output layer $d_{\max}=0.1$. To avoid overlearning, the 10-fold cross-validation was used to find the optimum values of these parameters on the training set O_{train} .

Table 5 presents the $\text{RMSE}_{\text{test}}$ (R^2) on testing data O_{test} for AQIO_3^{t+1} , AQINO_2^{t+1} , AQINO_x^{t+1} , AQISO_2^{t+1} , AQIPM_{10}^{t+1} , and AQISNG^{t+1} . For the AQI^{t+1} obtained using the composition of the outputs of the models M_1, M_2, \dots, M_6 (eq. (12)), $\text{RMSE}_{\text{test}} = 0.1635$ ($R^2=0.4372$) was achieved by the FFNNs (see Fig. 2). On the other hand, the $\text{RMSE}_{\text{test}}$ obtained from the single model (eq. (13)) was equal to 0.2064 ($R^2=0.4125$).

Table 5. RMSEtest for partial AQIs obtained using FFNNs

| | AQIO ₃ ^{t+1} | AQINO ₂ ^{t+1} | AQINO _x ^{t+1} | AQISO ₂ ^{t+1} | AQIPM ₁₀ ^{t+1} | AQISNG ^{t+1} |
|----------------------|----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|------------------------------------|-----------------------|
| RMSE _{test} | 0.1421 | 0.0381 | 0.5095 | 0.0361 | 0.3158 | 0.3211 |
| R ² | 0.5162 | 0.3843 | 0.4217 | 0.1835 | 0.5633 | 0.5544 |

**Fig. 2.** Calculated (target) values AQI^{t+1} and values obtained by FFNNs according to eq. (12) for testing data

For the Takagi-Sugeno FISs, the same data division was used as for the FFNNs. The FISs (the parameters of triangular membership functions μ and the base of if-then rules) were optimized using the ANFIS method, namely with the hybrid algorithm combining back-propagation with least square method. The remaining parameters of the prediction using Takagi-Sugeno FISs is presented in Table 6.

Table 6. The setting of parameters for the prediction of AQIs using Takagi-Sugeno FISs

| O _{train} :O _{test} | c | Number of μ for each input | Shape of μ | Output y | Design of base rules | Optimization |
|---------------------------------------|----|--------------------------------|----------------|------------------|----------------------|--------------|
| 2:1 | 50 | 3 | triangular | weighted average | historical data | ANFIS |

Table 7 provides the RMSE_{test} (R^2) of predictions for partial AQIO₃^{t+1}, AQINO₂^{t+1}, AQINO_x^{t+1}, AQISO₂^{t+1}, AQIPM₁₀^{t+1}, and AQISNG^{t+1}. For the AQI^{t+1} obtained using the composition (eq. (12)), RMSE_{test} was equal to 0.2174 ($R^2=0.3987$), while RMSE_{test}=0.2587 ($R^2=0.1594$) for the single model (eq. (13)). The predictions of the composition model using Takagi-Sugeno FIS are depicted in Fig. 3.

Table 7. RMSEtest of partial AQIs obtained using Takagi-Sugeno FISs

| | AQIO ₃ ^{t+1} | AQINO ₂ ^{t+1} | AQINO _x ^{t+1} | AQISO ₂ ^{t+1} | AQIPM ₁₀ ^{t+1} | AQISNG ^{t+1} |
|----------------------|----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|------------------------------------|-----------------------|
| RMSE _{test} | 0.1623 | 0.0409 | 0.4779 | 0.0379 | 0.4281 | 0.3435 |
| R ² | 0.3126 | 0.2314 | 0.3722 | 0.1472 | 0.2212 | 0.4300 |

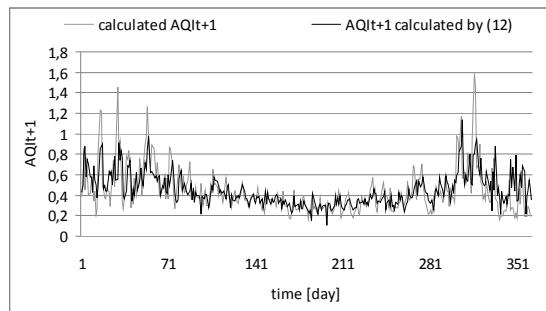


Fig. 3. Calculated (target) values AQI^{t+1} and values obtained by Takagi Sugeno FISs according to eq. (12) for testing data

6 Conclusion

We used two models for the prediction of AQI^{t+1} (the city of Pardubice, Dukla station, the Czech Republic). The first model was based on partial AQIs (AQIO_3^{t+1} , AQINO_2^{t+1} , AQINO_x^{t+1} , AQISO_2^{t+1} , AQIPM_{10}^{t+1} , and AQISNG^{t+1}) and on the consequent composition according to eq. (12). This model was more complex since the set of input variables was optimized separately for each subsystem. Following this procedure, both the FFNNs and Takagi-Sugeno FISs performed better compared with the single prediction model (eq. (13)) where the set of input variables was optimized only once from the original set of all input variables. Except for the AQINO_x^{t+1} , the FFNN performed better than Takagi-Sugeno FIS. The main limitation of the composition approach is that it does not take into account the final error between the target and predicted value of AQI^{t+1} . Therefore, future research should be done using multiple output SVMs [24]. Further, radial basis function neural networks [25], ϵ -insensitive support vector machine regression [25,26] with supervised and semi-supervised learning could be used due to their generalization capability and possibility to use the unlabelled data. Finally, the uncertainty in the data could be effectively modelled using the generalizations of Takagi-Sugeno FISs [11,27].

The experiments were carried out in Windows 7, Weka 3.6.8 (correlation based filter and FFNNs), and Matlab 7.1 Fuzzy Logic Toolbox (Takagi-Sugeno FISs).

Acknowledgment. This work was supported by the scientific research project of the Technology Agency of the Czech Republic under Grant No: TD-010130 and we thank the CHMI for providing the data.

References

1. Elshout, S., Leger, K., et al.: Comparing Urban Air Quality in Europe in Real Time – A Review of Existing Air Quality Indices and the Proposal of a Common Alternative. *Environment International* 34(5), 720–726 (2008)
2. Environmental Protection Agency, Guideline for Reporting of Daily Air Quality – Air Quality Index, US Environmental Protection Agency (1999)

3. Murena, F.: Measuring Air Quality over Large Urban Areas: Development and Application of an Air Pollution Index at the Urban Area of Naples. *Atmospheric Environment* 38, 6195–6202 (2004)
4. Cheng, W., Kuo, Y., et al.: Revised Air Quality Index Derived from an Entropy Function. *Atmospheric Environment* 38, 383–391 (2004)
5. Kyle, A.D., Woodruff, T.J., et al.: Use of an Index to Reflect the Aggregate Burden of Long-term Exposure to Criteria Air Pollutants in the United States. *Environmental Health Perspectives* 110, 95–102 (2002)
6. Kassomenos, P., Skouloudis, A.N., et al.: Air Quality Indicators for Uniform Indexing of Atmospheric Pollution over Large Metropolitan Areas. *Atmospheric Environment* 33, 1861–1879 (1999)
7. Kyriakidis, I., Karatzas, K., Papadourakis, G., Ware, A., Kukkonen, J.: Investigation and Forecasting of the Common Air Quality Index in Thessaloniki, Greece. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H., Karatzas, K., Sioutas, S. (eds.) *AIAI 2012, Part II*. IFIP AICT, vol. 382, pp. 390–400. Springer, Heidelberg (2012)
8. Kyrkilis, G., Chaloulakou, A.: Development of an Aggregate Air Quality Index for an Urban Mediterranean Agglomeration: Relation to Potential Health Effects. *Environment International* 33(5), 670–676 (2007)
9. Stieb, D.M., Doiron, M.S., et al.: Estimating the Public Health Burden Attributable to Air Pollution: An Illustration using the Development of an Alternative Air Quality Index. *Journal of Toxicology and Environmental Health* 68(13–14), 1275–1288 (2005)
10. State Policy of Environment in Czech Republic 2004–2010. Ministry of Environment, Prague (2004) (in Czech)
11. Hajek, P., Olej, V.: Ozone Prediction on the basis of Neural Networks, Support Vector Regression and Methods with Uncertainty. *Ecological Informatics* 12, 31–42 (2012)
12. Iliadis, L.S., Papaleonidas, A.: Intelligent Agents Networks Employing Hybrid Reasoning: Application in Air Quality Monitoring and Improvement. In: Palmer-Brown, D., Draganova, C., Pimenidis, E., Mouratidis, H. (eds.) *EANN 2009. CCIS*, vol. 43, pp. 1–16. Springer, Heidelberg (2009)
13. Junninen, H., Niska, H., et al.: Methods for Imputation of Missing Values in Air Quality Data Sets. *Atmospheric Environment* 38(18), 2895–2290 (2004)
14. Hall, M.A.: Correlation-Based Feature Subset Selection for Machine Learning. Hamilton, University of Waikato (1998)
15. Lippman, R.P.: An Introduction to Computing with Neural Nets. *IEEE ASSP Mag.* 4, 4–22 (1987)
16. Rosenblatt, F.: The Perceptron, a Probabilistic Model for Information Storage and Organization in the Brain. *Psychol. Rev.* 65(6), 386–408 (1958)
17. Haykin, S.: *Neural Networks: A Comprehensive Foundation*. Prentice-Hall Inc., New Jersey (1999)
18. Pedrycz, W.: *Fuzzy Control and Fuzzy Systems*, 2nd edn. John Wiley and Sons Inc., New York (1993)
19. Takagi, T., Sugeno, M.: Fuzzy Identification of Systems and its Applications to Modeling and Control. *IEEE Trans. on Syst. Man and Cybern.* 15(1), 116–132 (1985)
20. Hajek, P., Olej, V.: An Air Pollution Index based on Hierarchical Fuzzy Inference Systems. In: Mastorakis, N. (ed.) *Proc. of the 5th International Conference on Energy, Environment, Ecosystems and Sustainable Development*, pp. 83–77. WSEAS (2009)
21. Air Quality Index (in Czech), http://www.szu.cz/uploads/documents/chzp_ovzdusi/organizace_mzso/index_kvality_ovzdusi.pdf

22. Yamartino, R.J.: A Comparison of Several Single Pass Estimators of the Standard Deviation of Wind Direction. *Journal of Climate and Applied Meteorology* 23, 1362–1366 (1984)
23. Farrugia, P.S., Micallef, A.: Comparative Analysis of Estimators for Wind Direction Standard Deviation. *Meteorological Application* 13, 29–41 (2006)
24. Sanchez-Fernandez, M., de-Prado-Cumplido, M., et al.: SVM Multiregression for Non Linear Channel Estimation in Multiple-Input Multiple-Output Systems. *IEEE Transactions on Signal Processing* 52(8), 2298–2307 (2004)
25. Olej, V., Filipová, J.: Modelling of Web Domain Visits by Radial Basis Function Neural Networks and Support Vector Machine Regression. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H. (eds.) *EANN/AIAI 2011, Part II. IFIP AICT*, vol. 364, pp. 229–239. Springer, Heidelberg (2011)
26. Camps-Valls, G., Muñoz-Marí, J., et al.: Biophysical Parameter Estimation with a Semisupervised Support Vector Machine. *IEEE Geoscience and Sensing Letters* 6(2), 248–252 (2009)
27. Hájek, P., Olej, V.: Adaptive Intuitionistic Fuzzy Inference Systems of Takagi-Sugeno Type for Regression Problems. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H. (eds.) *AIAI 2012. IFIP AICT*, vol. 381, pp. 206–216. Springer, Heidelberg (2012)

Novel Neural Architecture for Air Data Angle Estimation

Manuela Battipede, Mario Cassaro, Piero Gili, and Angelo Lerro

Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino 10128, Italy

{manuela.battipede,mario.cassaro,piero.gili,
angelo.lerro}@polito.it

Abstract. This paper presents a novel architecture for air-data angle estimation. It represents an effective low-cost low-weight solution to be implemented in small, mini and micro Unmanned Aerial Vehicles (UAVs). It can be used as a simplex sensor or as a voter in a dual-redundant sensor systems, to detect inconsistencies of the main sensors and accommodate the failures. The estimator acts as a virtual sensor processing data derived from an Attitude Heading Reference System (AHRS) coupled with a dynamic pressure sensor. This novel architecture is based on the synergy of a neural network and of an ANFIS filter which acts on the noise-corrupted data, cancelling the noise contribution without interfering with the turbulence frequencies, which must be preserved as key information for the AFCS activity.

Keywords: Air-Data sensor, virtual sensor, analytical redundancy, ANFIS filtering.

1 Introduction

The increasing need of modern UAVs to reduce cost and complexity of on-board systems has encouraged the practice to substitute, whenever feasible, expensive, heavy and sometimes even voluminous hardware devices with executable software code. Virtual sensors can be used as voters in dual-redundant or simplex sensor systems, to detect inconsistencies of the hardware sensors and accommodate the sensor failures. This practice is commonly referred to as analytical redundancy. More generally, analytical redundancy identifies with the functional redundancy of the system.

The virtual sensor for air-data estimation presented in this paper is based on neural networks (NN) which are an extremely powerful tool to reduce the discrepancies between the mathematical model and the real plant, which is the main drawback of the majority of the model-based techniques. The current literature is rich of examples of NNs used as emulators to estimate aerodynamic coefficients [1], angle of attack [2], [3], and sideslip angle [4] from data derived from other sources that are not the classic vanes, differential pressure sensors or expensive modern multifunction probes. Many of them, however, rely on the dynamic pressure measure [5], [6], which represents a criticality, as the dynamic pressure is usually measured by external devices, which are not very suitable on modern UAVs: for example they might

interfere with the pilot camera angle of view or other payload sensors, or simply they might limit the UAV stealth capabilities. When the dynamic pressure signal is not directly required, it is calculated correlating information related to the center of gravity position, the engine torque and the actual thrust [7], which must be measured with great accuracy to provide the virtual air-data sensor with acceptable performance.

The neural Air-Data Sensors (ADS) presented in this paper is based on data derived from the Attitude Heading Reference System (AHRS) coupled with the dynamic pressure signal. AHRS consist of either solid-state or MEMS gyroscopes, accelerometers and magnetometers mounted in the orthogonal 3D frame. The key difference between an AHRS and an IMU, which simply consists of three angular rate sensors and three accelerometers, is the addition of an on-board processing system in an AHRS which provides also attitude and heading information.

A sensitivity analysis is performed to select the minimum data set to be included among the NN input variables. The effect of sensor accuracy is investigated. One of the most critical issues related to real applications, is quality of the flight data, which are usually noise-corrupted [8]. An ANFIS filter is thus developed to cut-off the sensor noise without interfering with the turbulence frequencies, which must be preserved as key information for the AFCS activity.

2 Sensitivity Analysis on Air-Data Angle Models

In respect to the definition of the aircraft body and wind axis reference system, as reported in Fig.1 the angle-of-attack and the sideslip angle can be expressed by the following relations:

$$\begin{aligned}\alpha &= \tan^{-1} \frac{w}{u} \\ \beta &= \tan^{-1} \frac{v}{\sqrt{u^2 + w^2}}\end{aligned}\quad (1)$$

where u, v , and w are the body-axis component of the airspeed V . By differentiating the first equation and imposing the force equilibrium along the x and z body axes it is possible to find an explicit equations for $\dot{\alpha}$:

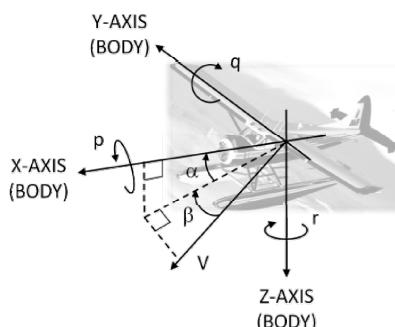


Fig. 1. Wind-axis and body-axis reference frames

$$\dot{\alpha} = \frac{\left(qu - pv + g_0 \cos \phi \cos \vartheta + \frac{F_z}{m} \right) \cos \alpha}{V \cos \beta} - \frac{\left(rv - qw - g_0 \sin \vartheta + \frac{F_x}{m} \right) \sin \alpha}{V \cos \beta} \quad (2)$$

where F_x and F_z are the resultant external forces along the x and z body axis, which accounts for the aerodynamic and propulsive contributions, including the pilot commands, given through the control surface deflections. $\dot{\alpha}$ can be either integrated to obtain the angle-of-attack trend or used to infer it through a neural emulator, for example. In this context Eq. 2 is essential to understand the dependence of α on the state and control variables. According to Eq. 2, namely, it is possible to reconstruct the α signal through the measurements of a given set of variables:

$$\alpha = f_\alpha(q_c, a_x, a_z, \beta, \vartheta, \phi, p, q, r, \delta_f) \quad (3)$$

where the dynamic pressure q_c is representative of the total speed V . In Eq. 6 the contribution of the external forces F_x and F_z have been substituted by the accelerations a_x and a_z respectively, which are measured by the on-board accelerometers, under the assumption that the AHRS platform is located in the close proximity of the aircraft centre of gravity, and aligned with the body axes. Eq. 6 has been extended to include the effects of flaps which can considerably change, if extracted, the aerodynamic configuration during the take-off, approach and landing phases.

Eq. 2 takes into account the contribution of a very wide set of variables under the assumption that they can be measured and adequately filtered to reconstruct a signal for the angle-of-attack, which can be used with a high level of confidence for purposes of guidance and control of the aircraft. The high number of variable involved, though, suggests that a further analysis can be conducted to investigate the actual necessity of including all the variables and the impact of each of them on the angle-of-attack estimation.

The sensitivity analysis can be performed in different ways depending on the flight condition characteristics: steady flight, for example, can be analyzed through a local method involving the linearization of the equations of motions:

$$f(\dot{X}, X, U) = 0 \quad (4)$$

where X and U are respectively the state and control variables and f indicated the implicit nonlinear body-axis first order differential equations of motion. Linearization around the steady condition implies calculating the partial derivatives of each equation with respect to each variable:

$$\nabla_{\dot{X}} f \cdot \delta \dot{X} + \nabla_X f \cdot \delta X + \nabla_U f \cdot \delta U = 0 \quad (5)$$

where ∇ represents a row vector of first partial derivative operators.

Analytical and numerical investigations reveal that, under the specific assumption that the stability-axis inertia matrix J_s is symmetric, the longitudinal and lateral-directional equations are decoupled [18]. This implies that $\dot{\alpha}$ is affected mainly by the longitudinal variables q_c, a_x, a_z, q, θ and δ_f , among which the least influential term is the pitch angle θ .

In unsteady conditions the non linear dynamic equations must be considered and the sensitivity analysis can be performed through the uncertainty propagation method, which assumes independence among the measured variable: test manoeuvres are simulated in the time-domain and the sensor signal y is modelled assuming that the uncertainties have a Gaussian standard probability distribution, where the root mean square deviation is given by the sensor accuracy. For the sake of generality, in the present analysis accuracy is given as a percentage of the full scale range of each sensor and comparison is carried out considering the same accuracy percentage for each sensor. Indeed, the accuracy is a specific characteristic of the particular sensor, which is declared by manufacturers in terms of non-linearity, temperature and hysteresis effects. The uncertainty analysis performed in unsteady conditions on all the variables included in Eq.2 reveals that the angle-of-attack errors resulting from the inaccuracy propagation of q_c and a_z , considered separately, are one order of magnitude greater than the errors caused by all the other variables. Inaccuracy of the lateral-directional variables does not affect the angle-of-attack estimation. Eq. 8, thus, can reduce to the following relation:

$$\alpha = f_\alpha(q_c, a_x, a_z, q, \delta_f) \quad (6)$$

The same analysis performed on the sideslip angle leads to the following relation:

$$\beta = f_\beta(q_c, a_y, a_z, p, r) \quad (7)$$

3 Neural Air-Data Sensors

The implemented neural Air-Data Sensors (ADS) is made of two multi-layer perceptrons (MLP) network, one for the angle-of-attack and one for the sideslip angle, with a single non linear hidden layer containing 10 neurons and one linear output layer.

The single perceptron model was introduced and demonstrated by Rosenblatt [16], and it remains the simplest form of neural network, and it has been used as the smallest unit of the neural networks used for this work. Multilayer perceptron models were successfully applied in the past to solve several problems by means of training with very popular algorithms[12,19]. The single perceptron model is represented in Figure 2. It is generally characterized by one single output, y_j , one bias, b_j , and one activation function, f_j , which processes the signal, v_i . The variable v_i is the sum of the matrix product between a weight matrix, w_{ji} , and n-inputs, y_i . It is clear that the key-factors for the success of the single perceptron are the values of weight matrix coefficients. In order to obtain the best neural network weights, several optimization, or training algorithms exist [12,19] which have the aim to optimize the neural

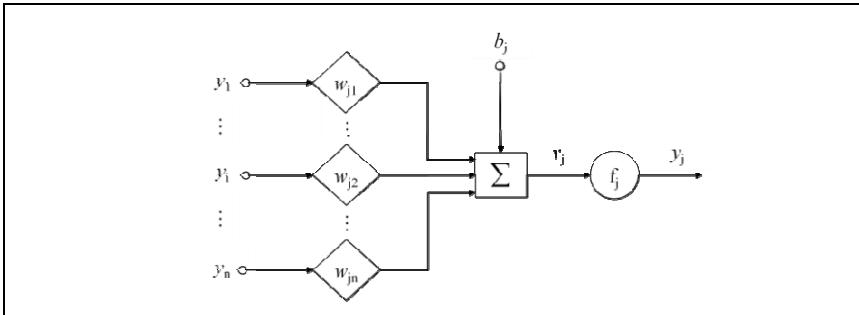


Fig. 2. Example of general neural perceptron with n inputs

network weights in order to minimize the error between the input-output pattern used for training. The learning algorithm used for this work has been the Levenberg-Marquardt algorithm [17,18].

To assess the validity of the model it has decided to simulate firstly the best-case scenario, training and testing the network on noise-free signals. Turbulence and noise have been addressed as occurring both separately and simultaneously. Performance are evaluated on the basis of ± 1 deg error specification.

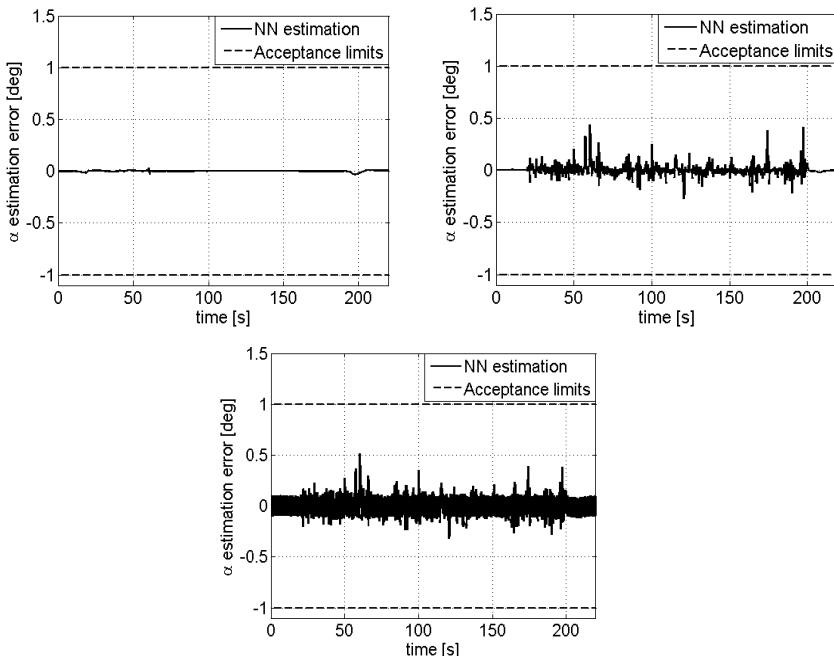


Fig. 3. Estimation error obtained using the test manoeuvre for the neural network NNA: (a) without turbulence and noise; (b) with turbulence and without noise; (c) with turbulence and noise.

The neural network acts as a time-invariant neural ARX (AutoRegressive with eXternal input)[10-11] predictor:

$$\hat{\alpha}(t, W) = g(\varphi(t), W) \quad (8)$$

(in the specific case of the angle-of-attack) where W is the vector containing the neural network parameters and φ is the regressor vector, characterized by zero input and output lag:

$$\varphi(t) = [q_c(t) a_x(t) a_z(t) q(t) \theta(t) \delta_f(t)] \quad (9)$$

This structure has the enormous advantage of being light in terms of computational resources as the estimation of $\alpha(t)$ is performed without using any memory delay, processing only the variables at the current time step. From now on, neural networks will be identified as NNA, if designed for the angle-of-attack estimation α , and NNB, if designed for sideslip angle β .

As shown in Fig 3 and 4, the optimal response is obviously obtained in the noise-free situation (a), where the maximum peak to peak error is less than 0.1 deg . The estimation is progressively less accurate, when the signal is corrupted by turbulence and noise (b or c). The estimation error of 0.5 deg for the worst condition demonstrates high sensitivity of the network to the external noise.

The comparison of performance obtained on noise-corrupted signal suggest that data should be filtered before being fed to the neural air data sensors.

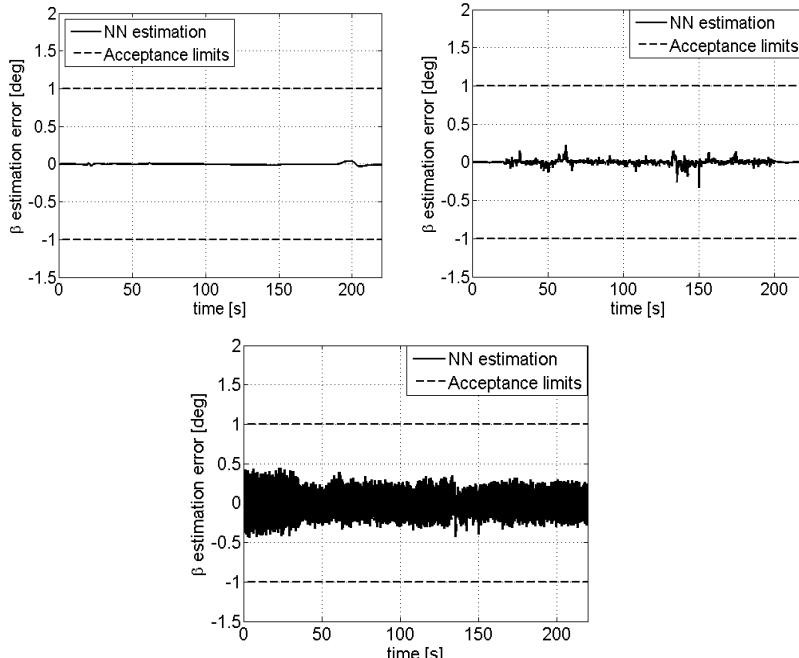


Fig. 4. Estimation error obtained using the test manoeuvre for the neural network NNB: (a) without turbulence and noise; (b) with turbulence and without noise; (c) with turbulence and noise

4 ANFIS Filter Design

An Adaptive Neuro Fuzzy Inference System (ANFIS) [12] is proposed for filtering the signals generated from the AHRS, which must be used as inputs for the neural virtual sensor. Small UAVs can be a challenging application as they represent an environment where the spectrum of noise vibrations and aircraft dynamic frequencies can overlap in several flight conditions. Conventional filters used for noise rejection and based on the Fast Fourier transform technique, usually introduce time delays, as they tend to affect the phase signal. However, even small time delays sometimes can be unacceptable, especially when the signal is fed to an artificial flight control system [13].

ANFIS filters reported in literature [14-15], are based on the knowledge of the noise affecting the signal. The membership functions of the input variables are usually trained using the noise-corrupted signals whereas the noise-free filtered signal is used to optimize the output part. Noise-free signals can be obtained by a classical post-processing method, based on an off-line signal re-phasing. Once it is trained, the ANFIS is able to associate a noise-corrupted signal to a noise-free one and to act as an on-line filter. Filtering is performed without introducing delay, besides the one brought about by the discretization process.

Performance, however, can deteriorate considerably if the input signal exceeds the universe of discourse, or, strictly speaking, if the amplitude and frequency of the input signal are not included among the examples of the training domain. For this reason it is important to identify the sources of noise that must be filtered, together with the range of the characteristic parameters. AHRS signals can be affected by several noise sources such as the electronic one, typical of each sensor, and the one introduced by the structural vibrations, typical of the installation. The latter may vary as a function of the structure architecture, of the installed propulsion system and of the flight condition which implies a certain degree of aerodynamic unsteadiness.

It is important to notice that oscillations due to turbulence shall not be filtered as they are key information to reconstruct the air-data angle signals. Turbulence conditions, in fact, require a more intensive autopilot control action and, for this reason, the signals used as feedback for the control law (i.e. α and β) must contain all the relevant information.

The a_x , a_y and a_z acceleration signals are processed by 3 MISO ANFIS filters, one for each acceleration sensor. Each filter receives two different input signals: the first, coming from the AHRS, contains the measured noisy acceleration a_M ; the second, coming from the vibration sensor, is used to characterize the signal noise w_M .

The vibration sensor must be secured to the structure in the AHRS platform proximity, in case the noise is generated by structural vibrations that are not naturally cancelled by the AHRS platform dampers. Otherwise, the vibration sensor can be located close to noise source, such as the propulsion system.

In detail, each acceleration ANFIS filter, uses a hybrid learning algorithm to optimize the membership function parameters of a Takagi-Sugeno type fuzzy inference systems, each of which has a proper universe of discourse, imposed by the aircraft flight performance. As an example the a_z component of the aircraft under analysis ranges $\pm 2 g$ from the level flight condition, and 7 bell-shape membership

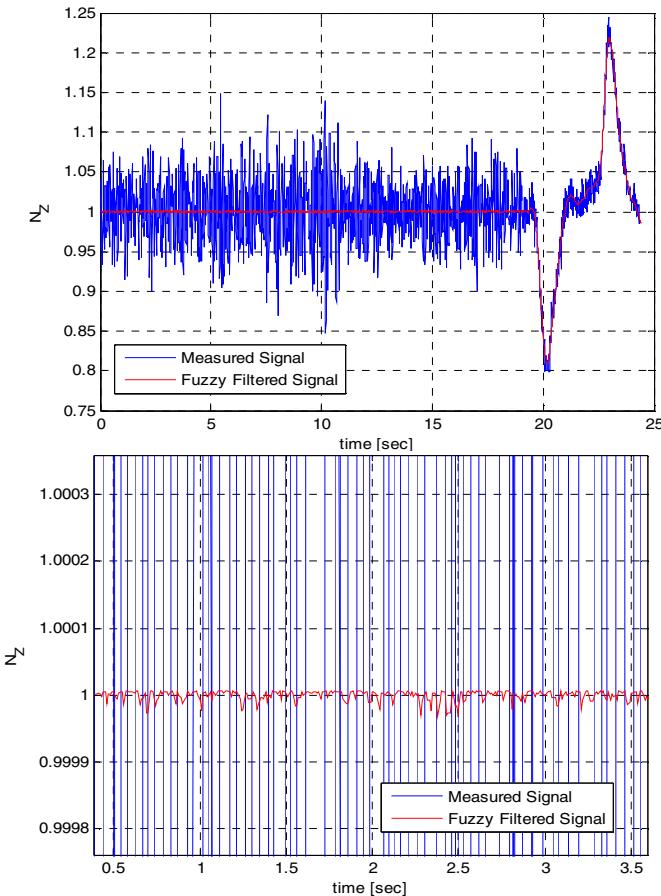


Fig. 5. (a) Comparison between measured and filtered n_z during take-off; **(b)** Zoom for error evaluation

functions, plus 2 z/s-shape at the boundary, were used to discretize the input domain. The filtered signal used to train the membership function parameters was obtained by re-phasing a third order Butterworth filter result.

To test the filter performance a take-off manoeuvre, registered for a small UAV was used, just after a short training conducted with data representative of the whole flight envelope.

The first nineteen seconds of Figure 5 refer to the ground run and are characterized by strong vibrations due to the gear loads which sum up to the structural vibrations induced by the propeller. From the take-off onwards only aerodynamic and structural loads are captured by the accelerometer. Globally, the maneuver is characterized by a wide variation of the noise characteristics, but the ANFIS filter shows its ability to cut off undesirable frequencies with no appreciable delay other than the sampling time.

5 Conclusion

A neural Air-Data Sensors has been presented, based on data derived from the Attitude Heading Reference System (AHRS) coupled with the dynamic pressure signal. Accelerometer signal are commonly noise-corrupted and needs an accurate filtering process to cut-off the sensor noise without interfering with the turbulence frequencies, which must be preserved as key information for the AFCS activity.

The ANFIS filter has proven effective to provide the neural network with noise-free data for the real time on-board estimation of the air-data angles, without introducing time delays. The ANFIS filter is based on the knowledge of the noise affecting the signal and its correct modeling. Even when the sources of undesirable noise have been correctly identified and included in the training domain, however, there might be a criticality, associated to a structural or propulsive failure, which might shift unpredictably the noise frequency beyond the filter notches, and deteriorate the filter performance. For this reason, before the virtual sensor can become commercially valuable, a quantitative hazard analysis should be performed to evaluate the impact of different sources of failure and implement the required mitigating factors.

References

1. Norgaard, M., Jorgensen, C.C., Ross, J.C.: Neural Network Prediction of New Aircraft Design Coefficients. NASA Technical Memorandum 112197 (1997)
2. Oosterom, M., Babuska, R.: Virtual Sensor for the Angle-of-Attack Signal in Small Commercial Aircraft. In: IEEE International Conference on Fuzzy Systems (2006)
3. Samara, P.A., Sakellariou, F.J.S., Fassois, S.D.: Aircraft Angle-Of-Attack Virtual Sensor Design via a Functional Pooling Narx Methodology. In: Proceedings of the European Control Conference (ECC), Cambridge, UK (2003)
4. Xiaoping, D., et al.: A prediction model for vehicle sideslip angle based on neural network. In: IEEE Information and Financial Engineering (ICIFE), pp. 451–455 (2010)
5. Rohloff, T.J., Whitmore, S.A., Catton, I.: Air Data Sensing from Surface Pressure Measurements Using a Neural Network Method. AIAA Journal 36(11) (1998)
6. Samy, I., Postlethwaite, I., Green, J.: Neural-Network-Based Flush Air Data Sensing System Demonstrated on a Mini Air Vehicle. Journal of Aircraft 47(1) (2009)
7. McCool, K., Haas, D.: Neural network system for estimation of aircraft flight data, United States Patent 6.466.888 (2002)
8. Svobodova, J., Koudelka, V., Raida, Z.: Aircraft equipment modeling using neural networks. In: Electromagnetics in Advanced Applications, ICEAA (2011)
9. Roskam, J.: Airplane Flight Dynamics and Automatic Flight Controls, Design, Analysis and Research Corporation, Lawrence, KS (2001)
10. Narendra, K.S., Parthasarathy, K.: Identification and Control of Dynamical Systems Using Neural Networks. In: Proceedings of the IEEE First Annual International Conference on Neural Networks, vol. 1(1), pp. 4–27 (1990)
11. Chen, S., Billings, S.A.: Neural Networks for Nonlinear Dynamic System Modeling and Identification. In: Advances in Intelligent Control, pp. 85–112. Taylor and Francis (1994)

12. Jang, Sun, Mizutani: Neuro- fuzzy and soft computing: a computational approach to learning and machine intelligence. Prentice-Hall, Upper Saddle River (1997)
13. Rohac, J., Reinstein, M.L., Draxler, K.: Data Processing of Inertial Sensors in Strong-Vibration Environment. In: 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, Technology and Applications (2011)
14. He, Y., Manful, D., Bárdossy, A.: Fuzzy Logic Based De-Noising of Ultrasound Signals From Non- destructive Testing. Otto-Graf-Journal 15 (2004)
15. Balaiah, P., Ilavennila: Comparative Evaluation of Adaptive Filter and Neuro-Fuzzy Filter in Artifacts Removal From Electroencephalogram Signal. American Journal of Applied Sciences 9(10), 1583–1593 (2012)
16. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review 65(6), 386–408 (1958)
17. Levenberg, K.: A method for the solution of certain problems in least squares. Quart. Applied Math. 2, 164–168 (1944)
18. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. SIAM Journal on Applied Mathematics 11(2), 431–441 (1963)
19. Galushkin, A.I.: Neural Networks Theory. Springer (2007) ISBN 978-3-540-48124-9

Audio Data Fuzzy Fusion for Source Localization

Mario Malcangi

Department of Computer Science, Università degli Studi di Milano,
Via Comelico 39, 20135 Milano, Italy
malcangi@di.unimi.it

Abstract. This work addresses the problem of audio source localization in multiple speakers indoor scenarios. Three different direction of arrival (DOA) algorithms are applied to measure the angular position of the primary audio source with respect to a reference microphone, then a fuzzy logic-based method is applied to fuse the crisp measurements. The model-free estimation capability of the fuzzy logic enables to gain a good degree of precision keeping low the computational complexity of the system. This two level audio source localization system approach is robust and reliable because each module operates independently from the other, and the fuzzy logic inferential engine has the capability to evaluate qualitatively the performance of each of the DOA measurement subsystems.

Keywords: Fuzzy logic data fusion, audio source localization, direction of arrival, audio beamforming.

1 Introduction

The sound is the most important information media in the interaction between human beings and the physical world surrounding them, including others human beings. Among the several information that the sound embeds, the spatial location of the source is of primary importance in some human beings capabilities such as the attention-less monitoring of the surrounding environment, the source localization, and the beamforming towards the source. This auditory capabilities are effective thanks to the binaural nature of the auditory system and to its ability to fuse smartly the amplitude, time, and phase information coming from the auditory perception [1].

Human beings are able to recognize which sound is which, to locate where each sound is coming from and to recognize which sounds can be ignored, all simultaneously. This ability is evident when a person is able to understand what another person says in a group of many persons chatting.

Audio-based tasks of human beings, such as speech recognition, speaker tracking, and speaker identification use sound source location and data fusion strategies to successfully interact and communicate with other human beings. To enable the interaction of a human being with the machine in a natural way, a similar functionality needs to be developed in terms of sound source localization capabilities and data fusion strategy.

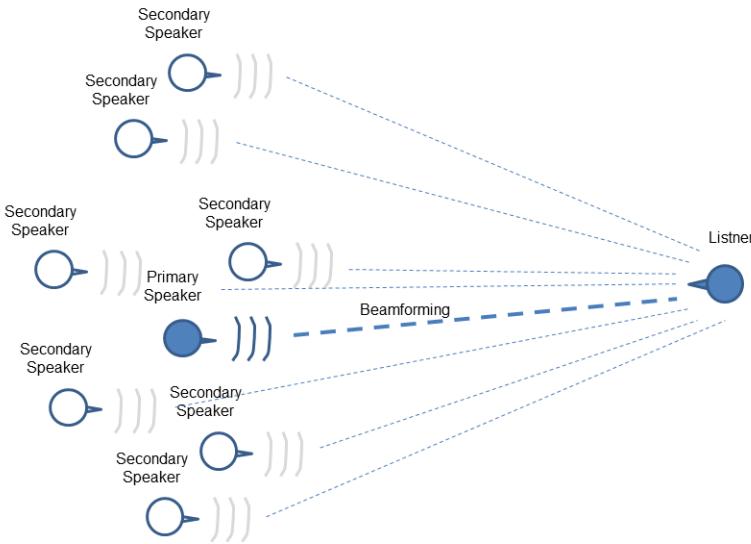


Fig. 1. Human beings are able to recognize which sound is which, to locate where each sound is coming from and to recognize which sounds can be ignored, beamforming towards the sound source target

The sound source localization (SSL) is a signal processing task based on various signal processing methods for time delay estimation applied to signals received by an array of microphones. Many audio-based applications, such as the automatic speech recognition (ASR), the automatic speaker identification (ASI), the audio noise cancellation (ANC), the indoor audio navigation (IAN), can be improved using SSL [2]. Audio-visual ASR [3] integrating SSL can improve its performance.

Using proper microphone geometry, 2D and 3D sound source location can be detected and the related measurements applied to recognize and isolate the sound source. Beamforming can be executed on the sound source, either by moving the primary microphone in the direction of the sound source, or executing application specific signal processing algorithms on the captured sound.

We present a framework to implement a sound source localization system that fuzzy fuse [3] [4] [5] the sound time delay measurements executed by a set of three algorithms processing at the same time the sound signal captured by a couple of microphones. The purpose is to implement a basic SSL capability useful to design a front-end between an array of microphones and the sound processing application.

2 System Framework

The system framework consists of two layers. The lower layer consists of three subsystems, each one implementing a sound time delay measurement. The upper layer implements a fuzzy logic-based inferential engine tuned to fuse the decision of the three sound delay measurements.

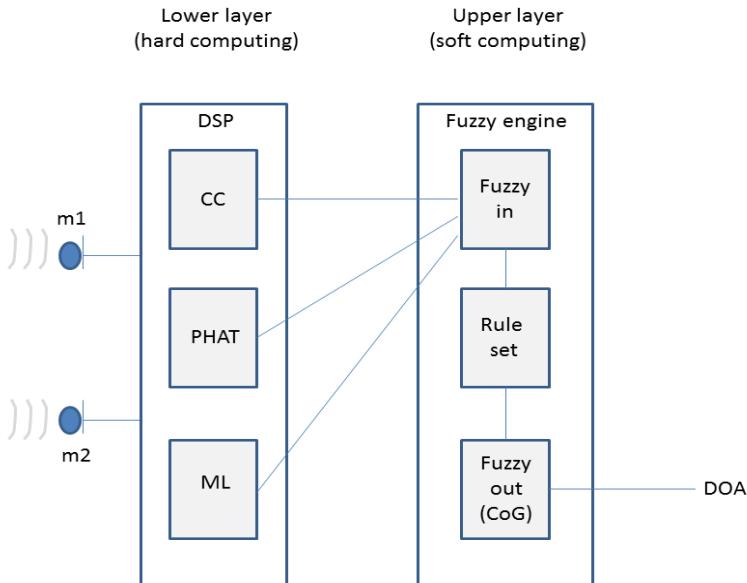


Fig. 2. System framework implementing the sound time delay measurement

2.1 Time Delay Measurement of Sound Sources

Among several time delay measurement algorithms [6], three of them have been selected as independent estimator methods: cross-correlation (CC), phase transform (PHAT), and maximum likelihood (ML) [7][8]. The CC method has been selected because it is simple and effective. As the CC shows many dominant peaks, and the highest is not always referred to DOA, then the PATH and the ML methods have been selected to match the true peak related to DOA [9][10].

The first, the CC method, combines the signals received from two microphones (m_1, m_2) so that the time delay corresponds to the maximum peak in the cross correlation function. It can be modeled as follows:

$$\begin{aligned} R_{m_1 m_2}(\tau) &= E[m_1(t)m_2(t-\tau)] \\ D_{CC} &= \arg \max_{\tau} [R_{m_1 m_2}(\tau)] \end{aligned} \quad (1)$$

where the D_{CC} is the measured delay between m_1 and m_2 signals.

The second, the PHAT method, is a generalized CC (GCC) that combines the signals received from two microphones eliminating the multiple peaks of the CC, so that its output presents only a relevant maximum peak in correspondence of the time delay between the two sounds captured by the microphones. Its model is:

$$R_{m_1 m_2}(\tau) = \int_{-\infty}^{+\infty} \frac{G_{m_1 m_2}(f)}{|G_{m_1 m_2}(f)|} e^{j2\pi f\tau} df \quad (2)$$

$$D_{PHAT} = \arg \max_{\tau} [R_{m_1 m_2}(\tau)]$$

where $G_{m_1 m_2}(f)$ is the cross-spectrum of the signal captured by the microphones pairs and D_{PHAT} is the measured delay between m_1 and m_2 signals.

The third, the ML method is another generalized CC (GCC). It gives the maximum likelihood solution to the problem of delay estimation between two signals. Its model is:

$$R_{m_1 m_2}(\tau) = \int_{-\infty}^{+\infty} \frac{1}{|G_{m_1 m_2}(f)|} \frac{\left|G_{m_1 m_2}(f)\right|^2}{1 - \frac{\left|G_{m_1 m_2}(f)\right|^2}{G_{m_1 m_1}(f)G_{m_2 m_2}(f)}} G_{m_1 m_2}(f) e^{j2\pi f\tau} df \quad (3)$$

$$D_{ML} = \arg \max_{\tau} [R_{m_1 m_2}(\tau)]$$

where D_{ML} is the measured delay between m_1 and m_2 signals.

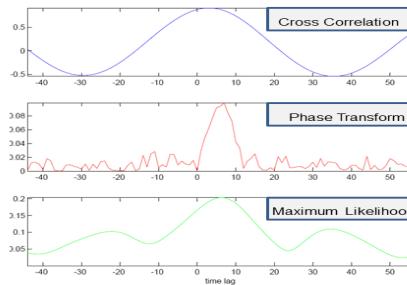


Fig. 3. Cross correlation, phase transform, and maximum likelihood functions combining the signal captured from a pair of microphones

2.2 Fuzzy Logic-Based Fusion

The three methods perform well in low noise and low reverberant rooms, highlighting a unambiguous position of the maximum, but if noise increases and reflections occur, then each method decrease its performance. To improve the performance of the delay estimation, the three algorithms run independently each other, measuring the time delay of a sound frame while an upper layer executes the fusion of the time delay measured for the previous sound frame.

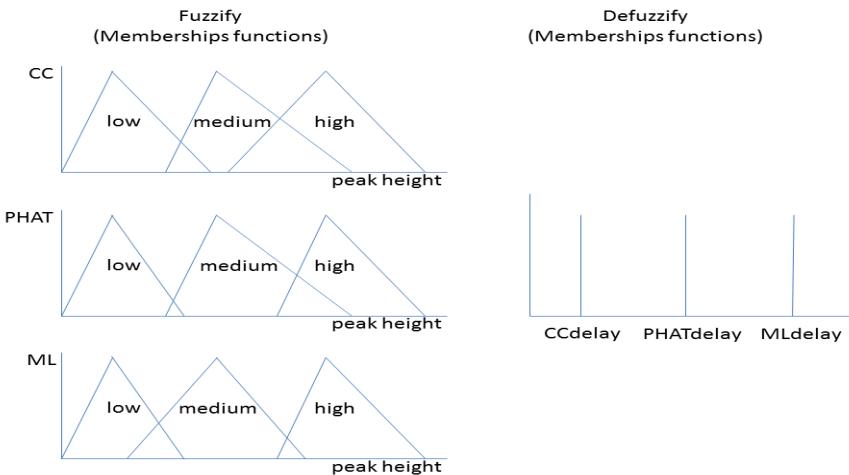


Fig. 4. Membership functions to fuzzify the measurements and defuzzify the decision

These crisp measurements are fuzzified by a set of membership functions (triangular shaped). A set of rules has been compiled to fuse the information at decision level.

The fuzzy decision is then defuzzified by a singleton membership function that produces as output the crisp measurement of the estimated delay. The center of gravity (CoG) defuzzification method has been applied in its weighted average (VA) implementation for singleton membership functions:

$$\text{Crisp_out} = \frac{\sum(\text{fuzzy_out}) (\text{singleton_ position})}{\sum \text{fuzzy_out}} \quad (4)$$

The set of rules considers the delay measured at the lower layer by each of the sound direction estimation algorithm and some other ancillary information such as the peak height and the primary to secondary peak height ratio. In this first release of the system, the rules are hand tuned at compile time, but an adaptive tuning process will be implemented to take in count the evolving nature of sound scenarios [11].

The decision rules look like this (the strongest):

```

IF
  CCpeak1 IS High AND
  PHATpeak1 IS High AND
  MLpeak1 IS High AND
  CCpeak12 IS Medium AND
  PATHpeak12 IS High AND
  MLpeak12 IS Medium
THEN
  Delay is PATHdelay
  
```

Only two parameters have been used in the ruleset, peak1 and peak12. Peak1 is the primary prominent peak amplitude. Peak12 is the secondary prominent peak amplitude, closer to the primary. Basic ruleset consists of 9 rules, but more rules are required when more secondary sound sources are close to the primary sound source.

3 Test and Performance

A set of tests has been executed in a simulated context using pure tone sound sources and short uttered frames. The simulation has been executed in Matlab environment, with an STMicroelectronics 8 MEMS microphonic array protoboard connected to Audacity acquisition and editing IDE for simultaneous 8 channel data acquisition, and a loudspeaker as sound source.

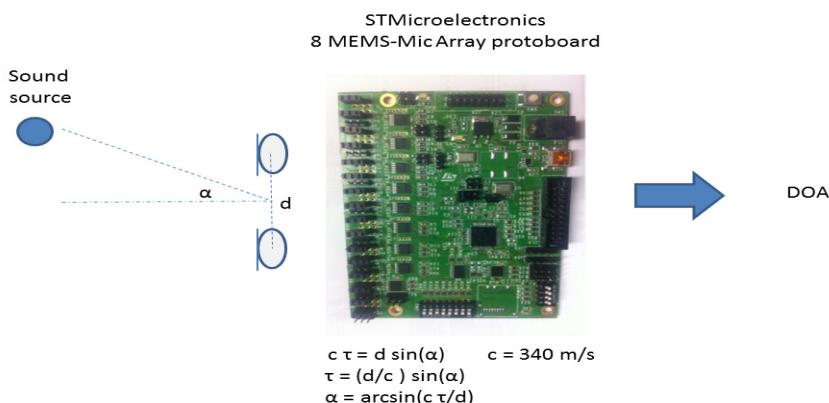


Fig. 5. Test setup for the measurement sound direction of arrival

The pure tone has been played at three different frequencies (500, 1000, 2000 Hz) and positioned at three different angles (0, 15, 30 degree). The same test has been executed for a short utterance. The two tests have been executed with and without noise. The following success rate resulted:

| | CC | CCwithPHAT&ML | FuzzyFusion |
|-------------------------|------|---------------|-------------|
| • Tone test noiseless | 100% | 100% | 100% |
| • Speech test noiseless | 95% | 97% | 97% |
| • Tone test noisy | 85% | 88% | 93% |
| • Speech test noisy | 75% | 77% | 91% |

The tests have confirmed that CC supported by PATH and ML works well than CC alone and that fuzzy fusion improves the performance of DOA measurements, mainly in noisy contexts.

4 Conclusions and Future Developments

Sound source localization is a very complex task which, at its core, involves the interaction among multiple microphones. Smart data fusion based on a fuzzy logic solution is an effective approach to manage and fuse the gathered data, mainly because a multiple level data fusion can be implemented.

To improve the reliability of the system, future developments will concern the reengineering of the fuzzy logic-based data fusion, splitting the fuzzy data fusion engine in a multi-level hierarchy, the lower level for feature fusion and the upper level for decision fusion. Each pair of microphones will work like a sub-module to compose a specific geometry to match a specific application. To this purpose, a third fuzzy fusion level layer will be developed to fuse the DOA decisions of the microphones pairs.

References

1. Parvaneh, P.: Binaural Hearing - Human Ability of Sound Source Localization. Master Thesis, in Electrical Engineering. Blekinge Institute of Technology, Karlskrona, Sweden (December 2008)
2. Scola, C.F., Bolaños Ortega, M.D.: Direction of Arrival Estimation – A two Microphones Approach. Master Thesis, Degree of Master of Science in Electrical Engineering, Blekinge Institute of Technology, Karlskrona, Sweden (September 2010)
3. Akhoundi, M.A.A., Valavi, E.: Multi-Sensor Fuzzy Data Fusion Using Sensors with Different Characteristics. Submitted to the CSI Journal on Computer Science and Engineering, JCSE (2010)
4. Xia, Y.S., Leung, H., Bossé, E.: Neural Data Fusion Algorithms Based on a Linearly Constrained Least Square Method. *IEEE Trans. Neural Netw.* 13(2), 320–329 (2002)
5. Stover, J., Hall, D., Gibson, R.: A fuzzy-logic Architecture for Autonomous Multisensor Data Fusion. *IEEE Transactions on Industrial Electronics* 43(3), 403–410 (1996)
6. Zhang, Y., Abdulla, W.H.: A Comparative Study of Time-Delay Estimation Techniques Using Microphone Arrays. Master Thesis, Department of Electrical and Computer Engineering, The University of Auckland, Auckland, New Zealand (2005)
7. Zhang, W., Ra, B.D.: Two Microphone Based Direction of Arrival Estimation for Multiple Speech Sources using Spectral Properties of Speech. In: ICASSP 2009 IEEE International Conference, pp. 2193–2196 (April 2009) ISSN 1520
8. Pollefeyns, M., Nister, D.: Direct Computation of Sound Microphone Locations From Time Difference of Arrival Data. In: ICASSP 2008 IEEE International Conference, March 31–April 4, pp. 2445–2448 (2008)
9. Blandin, C., Ozerov, A., Vincent, E.: Multi-source TDOA Estimation in Reverberant Audio Using Angular Spectra and Clustering. *Signal Processing* 92(8), 1950–1960 (2012)
10. Jiang, H., Mathews, B., Wilford, P.: Sound Localization Using Compressive Sensing. In: Proc. SENSO NETS, pp. 159–166 (2012)
11. Kwok, N.M., Buchholz, J., Fang, G., Gal, J.: Sound Source Localization: Microphone Array Design and Evolutionary Estimation. In: ICIT 2005, IEEE International Conference, pp. 14–17 (December 2005)

Boosting Simplified Fuzzy Neural Networks

Alexey Natekin¹ and Alois Knoll²

¹ Fortiss GmbH, Guerickstr. 25, Munich, Germany
natekin@fortiss.org

² Technical University Munich, Boltzmannstr. 3, Garching, Germany
knoll@in.tum.de

Abstract. Fuzzy neural networks are a powerful machine learning technique, that can be used in a large number of applications. Proper learning of fuzzy neural networks requires a lot of computational effort and the fuzzy-rule designs of these networks suffer from the curse of dimensionality. To alleviate these problems, a simplified fuzzy neural network is presented. The proposed simplified network model can be efficiently initialized with considerably high predictive power. We propose the ensembling approach, thus, using the new simplified neural network models as the type of a general-purpose fuzzy base-learner. The new base-learner properties are analyzed and the practical results of the new algorithm are presented on the robotic hand controller application.

Keywords: neural network, gradient boosting, fuzzy neural network, neural ensemble, boosting, robotic control, machine learning.

1 Introduction

Fuzzy neural networks have found successful applications in a wide range of domains. The most commonly used models are the additive fuzzy system models, rule-based universal approximators [2,1]. Fuzzy rules are the core of the fuzzy neural models and can be interpreted as the local submodels, interacting with each other through the fuzzy inference design.

Fuzzy neural networks suffer from a number of problems. First of all, the desired number of the fuzzy rules is typically not known beforehand. This problem is very similar to the problem of choosing the number of neurons in a common neural network, which is typically approached by trial and error or by means of cross-validation. In fuzzy neural networks this problem can be efficiently circumvented with clustering procedures [1,3], used for rule-base estimation. Using Gaussian mixture models, one can efficiently initialize the rule-base in an unsupervised manner and to inspect the optimal number of rules by means of some information criteria, like Bayesian Information Criterion. However, mixture models in their turn greatly suffer from the curse of dimensionality, making them less applicable to a large portion of real-world datasets.

The second problem of the fuzzy neural networks is their learning speed, which is typically higher, when compared to other methods like common MLPs, SVMs

or Random Forests. The fuzzy neural network model is simply more complex, which leads to increased learning time, in order to achieve a similar level of generalization accuracy.

We propose the solution to the described fuzzy neural network problems by designing a specific low-cost fuzzy neural network architecture, which will be used as a base-learner in the ensemble models. For this purpose we have developed a simplified fuzzy neural network model, efficiently initialized with k-means clustering and only one least-squares estimation. Under different hyperparameter settings, the proposed simplified network model can have different properties, which are addressed in this paper.

The ensembling approach, based on the gradient boosting algorithm [4] is introduced for fuzzy neural networks. Previous approaches to form fuzzy-based ensembles were mostly based on the bagging procedure, i.e. Fuzzy Random Forests [5]. In our approach, we want to fix the fuzzy base-learner complexity and take advantage of boosting for efficient learning of the details, not captured with other models in the ensemble. In this paper, we will provide both the algorithmic description of the proposed model, and the justifications about the choice of the hyperparameters with the resulting model properties.

In Section 2, we describe the fuzzy neural network models. In Section 3, we provide the basic gradient boosting algorithm description. In Section 4, the proposed fuzzy neural base-learner is presented and its properties are analyzed. Section 5 provides the application example of the GBM ensembles with the proposed fuzzy base-learners on the robotic hand controller. In Section 6, the results and conclusions are discussed.

2 Fuzzy Neural Networks

In the course of this article, only the regression problem will be considered. Let's suppose that we are given the dataset $(x, y)_{i=1}^N$, where $x = (x_1, \dots, x_d) \in R^d$ refers to the explanatory input variables and $y \in R$ to the corresponding response variable. The goal is to reconstruct the unknown functional dependence $x \xrightarrow{f} y$ with the parameterized estimate $\hat{f}(x, \theta)$, such that the empirical squared error function is minimized. Within the regression problem context, we will consider the estimate $\hat{f}(x, \theta)$ to be the additive fuzzy system.

2.1 Additive Fuzzy System Model

Fuzzy system is a set of fuzzy "IF-THEN" rules that maps the explanatory input variables x to the response output variable y . Additive fuzzy systems reconstruct the underlying functional dependence by covering the joint input-output distribution with fuzzy patches which encode these fuzzy rules. Fuzzy patches form coordinate-wise fuzzy sets in the "IF", or premise, part of the fuzzy rules, and local regression models in the "THEN", or consequent, part. Given G fuzzy rules, the i -th fuzzy rule is given in (1), $i = \overline{1..G}$.

$$\text{Rule}_i : \text{IF } x \text{ Is } A_i \text{ THEN } y \text{ is } \hat{f}_i(x, \theta_i), \quad (1)$$

- A_i is a fuzzy set defined on x , $i = \overline{1..G}$;
- $\hat{f}_i(x, \theta_i)$ is a consequent model of the rule.

We will consider the additive model, where the fuzzy sets A_i are defined on R^d by cartesian product coordinate-wise: $A_i = A_{i1} \times A_{i2} \times \dots \times A_{id}$. Each fuzzy set A_{ij} , defined on x_j , is in its turn characterized with some membership function $\mu_{A_{ij}}(x_j) \in [0, 1]$, $i = \overline{1..G}, j = \overline{1..d}$;

There are different design choices for the membership functions to be used. In this paper we will focus on the Gaussian model of the membership function, parameterized with its mean m_{ij} and standard deviations s_{ij} :

$$\mu_{A_{ij}}(x_j) = e^{-\frac{(x_j - m_{ij})^2}{2s_{ij}^2}}, i = \overline{1..G}, j = \overline{1..d} \quad (2)$$

The consequent models $\hat{f}_i(x)$ used can also be of different form and complexity. We will consider the commonly used linear regression consequents, as they are easy to estimate by solving LSE problem:

$$\hat{f}_i(x) = \sum_{j=1}^d c_{ij} x_j + c_{i0}, i = \overline{1..G} \quad (3)$$

After all the G fuzzy rules are fired, having calculated all the $\mu_{A_{ij}}$ and \hat{f}_i , the aggregated memberships $\mu_{A_i} = \prod_{j=1}^n \mu_{A_{ij}}(x_j)$ are calculated.

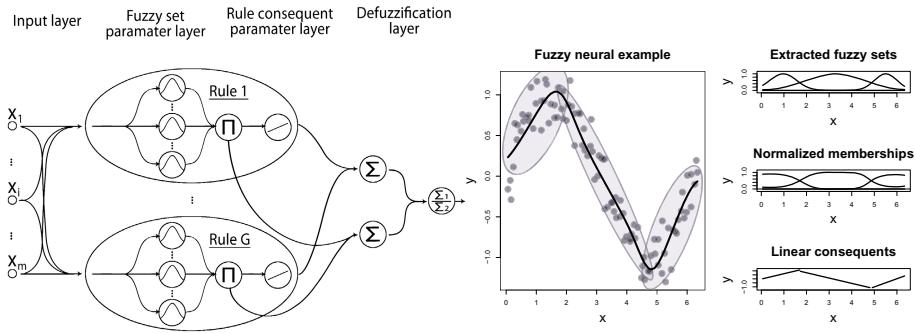
At last, the overall output of the network is calculated as the weighted average of consequents, with weights equal to normalized aggregated memberships. The overlapping fuzzy rules are fuzzed with respect to their relative certainty. The overall functional model of the additive fuzzy model is given in (4):

$$\hat{y} = \frac{\sum_{i=1}^G \mu_{A_i} \left(\sum_{j=1}^d c_{ij} x_j + c_{i0} \right)}{\sum_{i=1}^G \mu_{A_i}}, i = \overline{1..G}, j = \overline{1..d} \quad (4)$$

2.2 Fuzzy Neural Network Model

The (4) fuzzy function estimator can be represented in the form of a feedforward neural network with two parametric layers, containing the evaluation of the premise and consequent parts of fuzzy rules. The resulting neural network architecture is shown on Figure (1a).

For learning in these networks, hybrid learning algorithms are used. At each iteration, at first the fuzzy membership function parameters of the Gaussians m_{ij}, s_{ij} are optimized by a gradient descent step, $i = \overline{1..G}, j = \overline{1..d}$. Then, the consequent parameters c_{ij} are optimized by solving a single least squares linear regression problem, $i = \overline{1..G}, j = \overline{0..d}$. Due to the limited size of the article, we omit describing the details of the learning procedure. One can find the learning algorithms thoroughly described in [11].



(a) Neural network representation of the additive fuzzy system

(b) Example of the fitted fuzzy neural network

Fig. 1. Fuzzy neural network model

One of the most efficient and compact ways to approach fuzzy rule initialization is based on unsupervised clustering procedures. The joint input-output $x \times y$ space is covered with cluster patches in an unsupervised way. Then, the obtained clusters are marginalized over y , in order to get cluster projections on x and initialize the fuzzy rule's membership functions.

Having the membership function parameters initialized, one can proceed to the hybrid learning. Demonstration of the initialization process is shown on Figure (1b). For the demonstration purposes, synthetic noisy $\sin(x)$ function was used as the data input.

3 Gradient Boosting

Gradient boosting machines, or simply GBMs, are a family of efficient ensemble models that can capture complex nonlinear function dependencies. This family of models has shown considerable success in not only various practical applications, but also in various machine-learning and data-mining challenges [9,10].

The GBM models are based on a constructive strategy of ensemble formation. The main idea is to add new models to the ensemble sequentially. The learning procedure consecutively fits new models to provide more accurate estimates. The base-learners used can be chosen in various ways, the most commonly used base-learner models are the decision trees and generalized additive models [6].

The principle idea behind the GBM algorithm is to construct the new base-learners to be maximally correlated with the negative gradient of the loss function, associated with the whole ensemble. We will consider the squared error loss, where the learning procedure would result in consecutive error-fitting.

The GBMs have some hyperparameters. The most important hyperparameter is the number of base-learners, or boosting iterations in the ensemble M . Larger numbers of M increase the model complexity and can lead to overfitting. Also, two regularization hyperparameters have been introduced. The first of them is

the shrinkage λ , which penalizes the effect of each boosting iteration. The second regularization parameter is the bag fraction bag , which specifies the percentage of the data to be used at each iteration. The overall description of the GBM algorithm, used in this article is described in Algorithm (1). For more details about the GBMs we recommend the reader the [4,6,7] articles.

Algorithm 1. GBM Algorithm with squared error loss function

Inputs:

- input data $(x, y)_{i=1}^N$
- number of iterations M
- regularization parameters λ and bag
- choice of the base-leaner model $h(x, \theta)$
- base-leaner hyperparameters θ

Algorithm:

- 1: initialize \hat{f}_0 with a constant
 - 2: **for** $t = 1$ to M **do**
 - 3: sample $Bag_t = bag \cdot N$ indices, used for fitting a base-learner
 - 4: compute the negative gradient $g_t(x) = \sum_{i \in Bag_t} (y_i - \sum_{j=0}^{t-1} \hat{f}_j(x_i))$
 - 5: fit a new base-learner function $h(x, \theta)$ to the gradient g_t
 - 6: find the best gradient descent step-size ρ_t :
- $$\rho_t = \arg \min_{\rho} \sum_{i \in Bag_t} [g_t(x_i) - \rho h(x_i, \theta)]^2$$
- 7: update the ensemble:
- $$\hat{f}_t \leftarrow \hat{f}_{t-1} + \rho_t \lambda h(x, \theta)$$
- 8: **end for**
-

4 Simplified Fuzzy Neural Base-Learner

We now aim at designing a simplified fuzzy neural network model, which will have two important properties. First of all, it is going to be a fuzzy non-linear model, which can capture complex non-linearities. And second, it will be very fast to initialize and learn, in order to be the base-learner, used for boosting thousands of neural network models in reasonable time.

If we consider the model in (4) and it's demonstration on Figure (1b), we can denote that the model tends to fit fuzzy patches with local linear models, slightly smoothing the models with respect to their memberships. An obvious way to simplify the model will be to cut the consequent model to the constant terms c_{i0} only, $i = \overline{1..G}$.

This model will still remain smooth and continuous due to taking memberships into account. In hybrid learning, to estimate the parameters c_{ij} , one has to solve a LSE problem for a very large matrix, having dimensions $n \times (d+1)G$, $i = \overline{1..G}$, $j = \overline{0..d}$. Using only the intercept parameters in each rule's consequent

will dramatically reduce the size of this matrix to $n \times G$. This particular simplification is equivalent to the Takagi-Sugeno model of the 0-th order, but we aim at simplifying the model even further.

The next problem is the initialization of the fuzzy system. Mixture-based clustering, currently considered in the model, is a computationally expensive procedure. Moreover, inferring the number of clusters is also a problematic task.

To deal with these issues, one can consider the k-means clustering as an example of a simple mixture model, where the Gaussian clusters have all equal covariance structure of the form $\Sigma = vI$. This spherical simplification of the model significantly decreases the initialization time. The parameter $v > 0$ is held out as a hyperparameter that describes the uncertainty of the fuzzy rules.

Exploitation of the k-means clustering also allows us to use larger number of fuzzy rules, which in the clustering context is equal to the number of clusters. The larger the number of rules, the more complicated patterns one can capture. This particular design choice allows us to reduce the number of fuzzy rule parameters from $G \times d \times 2$ to $F \times d + 1$, having only the Gaussian center parameters and the only instance of v considered.

The last constraint that we haven't touched upon yet is the curse of dimensionality: the geometrical structure, captured by the clustering algorithm vanishes with the dimensionality of data increased. We will use a simple heuristic, used in Random Forests: for each of the base-learners in the ensemble we will randomly sample $p = \lceil \sqrt{d} \rceil$ dimensions, used for fitting. We constrain ourselves with the dimensionality p and number of clusters G for each of our base-learners.

To summarize, the simplified fuzzy neural (SFN) model is given in (5). The fitting procedure of one SFN base-learner is organized in the Algorithm (2).

$$\begin{aligned} \mu_{A_{ij}} &= e^{-\frac{(x_j - m_{ij})^2}{v}}, \mu_{A_i} = \prod_{j=1}^n \mu_{A_{ij}}(x_j), \\ \hat{y} &= \frac{\sum_{i=1}^G \mu_{A_i}(c_i)}{\sum_{i=1}^G \mu_{A_i}}, i = \overline{1..G}, j \in R^p \subset R^d \end{aligned} \quad (5)$$

4.1 SFN Properties

The hyperparameter v defines the uncertainty of the whole fuzzy model, behaving like a fuzzy membership modifier, either sharpening or smoothing the membership values. However, there is one important effect, connected with this parameter. To illustrate it we define the average winning normalized membership $awnm$ as (6).

$$awnm = \frac{1}{N} \sum_{j=1}^N \max_{i \in 1..G} \frac{\mu_{A_i}(x_j)}{\sum_{k=1}^G \mu_{A_k}(x_j)} \quad (6)$$

Algorithm 2. Fitting a simplified fuzzy base-learner**Inputs:**

- input data $(x, y)_{i=1}^N$
- number of fuzzy rules \sim clusters G
- number of dimensions $p < d$, used for fitting. default value: $p = \lceil \sqrt{d} \rceil$
- uncertainty hyperparameter $v > 0$

Algorithm:

- 1: sample p dimensions of x , used for fitting x^p
- 2: apply k-means clustering with G clusters to the joint (x^p, y) distribution
- 3: marginalize the y variable from the obtained cluster centers:

$$(m_{ij}, m_{iy}) \rightarrow (m_{ij}), i = \overline{1..G}, j = \overline{1..p}$$

- 4: calculate the average cluster standard deviations:

$$\sigma^2 = \frac{1}{pN} \sum_{i=1}^G \sum_{j=1}^p \sigma_{ij}^2, \sigma_{ij}^2 = \frac{1}{N-1} \sum_{k \in \text{Cluster}_i} (x_{kj} - m_{ij})^2$$

- 5: initialize the parameter $V = \frac{v}{\sigma^2}$

- 6: estimate the c_i parameters by fitting a linear regression model to normalized memberships $\mu_{A_i}, i = \overline{1..G}$

When the value of $awnm$ is near to 1, the smooth borders between rules are nearly diminished and the SFN model has a sharp discontinuous form, similar to the decision trees. On the contrary, when the $awnm$ is near to $\frac{1}{G}$, constructive membership information vanishes and the coefficients c_i become unstable. Visualization of these effects for fixed number of rules $G = 20$ on the same simulated $\sin(x)$ data is given on Figure (2).

From Figure (2a) we can see that starting from $awnm = 0.7$ the coefficients become less stable. If we consider the dependence between G , $awnm$ and MSE error, we can see that from some number G the model stabilizes and converges

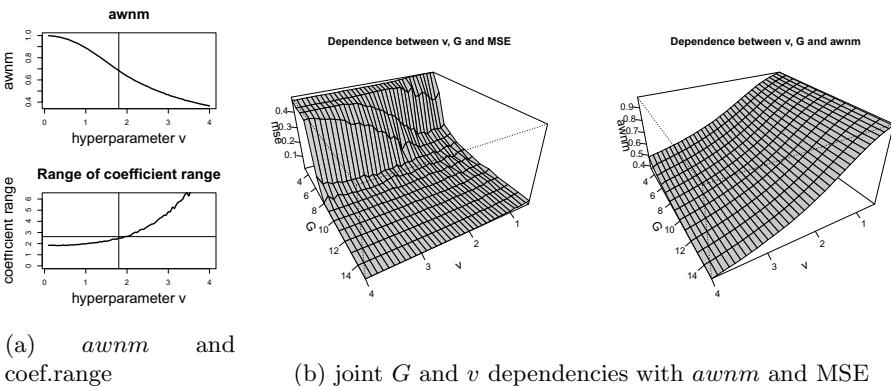


Fig. 2. SFN hyperparameter dependencies

to the same output, regarding complexity. And the *awnm* has a similar behavior all across the hyperparameter grid. The plot with the described dependencies is given on Figure (2b). In future works, parametrization based on *awnm* will be considered, as this statistic is more descriptive than parameter v .

4.2 SFN's Connection to Other Base-Learners

SFNs have one important common property with decision trees: they extrapolate the data with constant values. This means that the extrapolation with the SFN model will most likely result in exactly the coefficient value of the closest fuzzy rule, on the border of the observed data., just like a decision tree. This can be considered both a shortcoming and a feature.

Another property of the SFNs is that if the boosted model is considered to be additive, they can be used as a substitute for spline base-learners. Together with the variety of continuity in shape, this makes the SFNs a general purpose base-learner, lacking only the purely linear effects.

It feasible to interpolate a linear function with the SFNs, unlike the decision trees, but the linear effects can be intuitively added to the model. Thus, forming a model that captures nonlinear SFN effects, accounted after a linear fit.

5 Application Example

We will consider building a regression model to map the EMG signals to the robotic hand controller, in a similar manner as described in [8]. The data was provided by the TUM Roboroterhalle machine learning laboratory. 8 surface EMG electrodes, positioned on the hand, were used to record the muscular activity of the person performing different hand movements. These movements were then visually tracked to gather the actual spatial positions of the hand. Combined, this data was used to design a robotic arm control system. The machine learning task was to reconstruct the hand's position and orientation from the EMG channels and then to use it online as the robotic hand control. In our application we will consider a slightly altered experiment setup than the originally described one. We will focus on predicting the 3D hand positions only.

For each of the hand position coordinates, the data comprises 8 pre-processed signal features and 27,161 observations. Originally, the controller was first trained on all of the available data and then tested live, without knowing the correct labels, as the hand-tracking device was not available. In our case, we train the models on one half the available data and validate it on the other part. The train/test separation is organized sequentially: the first 100 points are used for training, the following 100 for validation, the next 100 points are used for training again and so on. As a consequence, the training set consists of 13,561 points and the test set consists of 13,600 points.

The model performance metric to be used will be the root mean squared error (RMSE), evaluated for each of the position variables y_i , $i = 1, 2, 3$ and the compiled, 3D error metric:

$$RMSE_i = \sqrt{\sum_{j=1}^N \frac{1}{N} (y_{ij} - \hat{y}_{ij})^2} \quad (7)$$

$$M3DE = \sum_{i=1}^N \frac{1}{N} \sqrt{(y_{1i} - \hat{y}_{1i})^2 + (y_{2i} - \hat{y}_{2i})^2 + (y_{3i} - \hat{y}_{3i})^2}$$

We consider 3 SFN models with different number of rules G . The regularization parameters shrinkage $\lambda = 0.01$ and $bag = 0.5$ were chosen beforehand because they are a sensible guess of GBM model's detailization. The optimal number of base-learners was chosen by cross-validation as $M = 1000$, number of dimensions used $p = 4$. We have also cross-validated the v parameter and for this application selected it equal to $v = 2$.

To make the model evaluation fair, we will compare the SFN model with other popular machine learning techniques: SVMs, ANFIS, Random Forests and tree-based GBMs. For each of these models the optimal hyperparameters were chosen by the 5-fold cross-validation, applied to the grid-search. The algorithm accuracy comparisons are given in Table (1).

Table 1. Machine learning algorithm accuracy

| Method | $RMSE_1$ | $RMSE_2$ | $RMSE_3$ | $M3DE$ |
|------------------------|----------|----------|----------|--------|
| Boosted SFN, G=10, v=2 | 0.073 | 0.057 | 0.078 | 0.087 |
| Boosted SFN, G=20, v=2 | 0.069 | 0.054 | 0.071 | 0.084 |
| Boosted SFN, G=30, v=2 | 0.065 | 0.052 | 0.067 | 0.081 |
| GBM, trees | 0.063 | 0.054 | 0.066 | 0.081 |
| Random Forests | 0.062 | 0.054 | 0.067 | 0.081 |
| Support Vector Machine | 0.076 | 0.069 | 0.084 | 0.100 |
| ANFIS | 0.074 | 0.069 | 0.089 | 0.103 |
| Linear Regression | 0.100 | 0.087 | 0.095 | 0.136 |

From Table (1) we can see that the new base-learner can efficiently compete with other machine learning techniques, delivering the same high level of accuracy. And the accuracy level is significantly higher than the accuracy of a single ANFIS, trained with all the necessary parameters and no simplifications. At last, the although the accuracy is nearly the same as the GBM and RF, the predictions of the new model are much smoother.

The high SFN performance can be explained by accurately catching nonlinear interactions. RF and GBM also exploit the interaction structure of the data in the ensemble, but due to decision tree limitations in approximating continuous functions, Boosted SFNs outperform these methods.

To summarize, the developed SFN model allowed us not only to design a competitive machine learning algorithm, but also to efficiently exploit all of the available information and build an accurate predictive model.

6 Conclusion

We have developed a new type of base-learner, which is not only interesting from a theoretical perspective, but has shown considerable success in practice, compared to other machine learning algorithms. The new model allows to have a fast fit of smooth interactions between variables, like the currently used generalized additive base-learners, but in multiple dimensions. Varying the hyperparameter v , one can also achieve different model behavior. Besides generalizing the model to the classification task, a straightforward extension to the algorithm would be to estimate the optimal parameter v with several gradient steps.

Acknowledgment. The authors would like to thank Prof. Patrick van der Smagt, Jorn Vogel and Justin Bayer from TUM Roboterhalle for providing the robotic control data. The authors also thank anonymous reviewers for their valuable comments.

References

1. Gan, M.-T., Hanmandlu, M., Tan, A.H.: From a Gaussian mixture model to additive fuzzy systems. *Trans. Fuz Sys.* 13(3), 303–316 (2005)
2. Kosko, B.: Fuzzy Systems as Universal Approximator. *B.s. IEEE Transactions on Computers* 43, 1329–1333 (1994)
3. George, E., Tsakouras, H.S.: A hierarchical fuzzy-clustering approach to fuzzy modeling. *Fuzzy Sets and Systems* 150, 245–266 (2005)
4. Friedman, J.: Greedy Boosting Approximation: A Gradient Boosting Machine. *The Annals of Statistics* 29, 1189–1232 (2001)
5. Bonissone, P., Cadenas, J.M., Garrido, M.C., Diaz-Valladares, A.R.: A fuzzy random forest. *International Journal of Approximate Reasoning* 51, 729–747 (2010)
6. Hothorn, T., Buhlmann, P., Kneib, T., Schmid, M., Hofner, B.: Model-based boosting 2.0. *Journal of Machine Learning Research* 11, 2109–2113 (2010)
7. Natekin, A., Knoll, A.: Gradient Boosting Machines, A Tutorial. *Frontiers in Neurorobotics* (2013), doi:10.3389
8. Vogel, J., Castellini, C., van der Smagt, P.: EMG-based teleoperation and manipulation with the DLR LWR-III. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS (2011)
9. Johnson, R., Zhang, T.: Learning Nonlinear Functions Using Regularized Greedy Forest. arXiv:1109.0887 (2012)
10. Bissacco, A., Yang, M.-H., Soatto, S.: Fast Human Pose Estimation using Appearance and Motion via Multi-Dimensional Boosting Regression. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007 (2007)
11. Jang, J.-S.R.: ANFIS: Adaptive-Network-based Fuzzy Inference Systems. *IEEE Transactions on Systems, Man and Cybernetics* 23, 665–685 (1993)

A Parallel and Hierarchical Markovian RBF Neural Network: Preliminary Performance Evaluation

Yiannis Kokkinos and Konstantinos Margaritis^{*}

Parallel and Distributed Processing Laboratory, Department of Applied Informatics,
University of Macedonia, 156 Egnatia str., P.O. Box 1591, 54006, Thessaloniki, Greece

Abstract. This paper presents a hierarchical Markovian Radial Basis Function (RBF) Neural Network, having embedded nature with many levels. The hidden RBF neurons in all the tree nodes of the hierarchy are composed of fully functional RBF Neural Networks that have the classical two synaptic link sets, one for the RBF centers and the other for the linear output weights. The Markov chain rule is specifically used in the RBF summations and permits this hierarchical functionality in a fractal-like fashion which supports a clear recursion. Thus the Neural Network operation is exactly the same at all levels. We further analyze the general framework where classical RBF Neural Network training algorithms can be applied, specifically for finding the centers and calculating the linear output weights. This framework is fast, simple and facilitates easy parallel implementations which are the main targets of this study.

Keywords: Hierarchical, parallel processing, radial basis function, Neural Networks, scalability analysis.

1 Introduction

Hierarchical structures are often strong candidates for parallel processing [1][2] when their applied algorithms permit it. During training they can inherently facilitate parallelism and naturally alleviate many common scalability problems through the divide and conquer strategy [3]. Although data can hide multiple levels of analysis from coarse-grained to fine-grained the Neural Network learning algorithms typically correspond to one, two or three hidden layers architectures. In principal a hierarchical Neural Network [4] decomposes a task into a series of simpler computations at each level. The outcomes of the previous levels must be combined into the next levels of the hierarchy. While this offers a major advantage for faster implementations there are fundamental issues concerning sample complexity, functional decomposition, hierarchical representation and training.

The well known Radial Basis Function Neural Network (RBFNN) [5][6][7][8][9] is a special type model that uses radial basis functions as its activation units and has been extensively used for function approximation, classification or system identification and control tasks. RBFNN was motivated by the presence of many local response

* Corresponding author.

units that exist in the human brain. Because of their universal approximation, compact topology and fast learning RBFNN have been widely applied in many science and engineering fields [5][6][7][8][9]. Since the locality principle operation of RBF units implies that a hierarchy is possible some interesting hierarchical RBF Neural Network examples have been suggested during the last years. For function approximation and 3-D meshing the hierarchical model of Ferrari and co-workers [10][11][12][13]. For clinical diagnosis the hierarchical model of Mat Isa et al. [14]. For system identification, classification and time-series forecasting the flexible hierarchical model by Chen and co-workers [15][16][17]. These models use different training algorithms than that of the original RBFNN since the first is a multi-gridding approach, the second is a cascading approach and the third is an evolving neural tree. We present here a Hierarchical Markovian RBF topology and a training framework that can use the same training stages and algorithms as in the conventional RBFNNs [5][6][7][8][9] and can be efficiently parallelized.

A distinct feature of our Hierarchical Markovian RBF approach is that each hidden ‘neuron’ in every level of the hierarchy is another fully functional RBF Neural Network and not just a simple summation neuron that combines the previous level outputs. Thus all RBFNNs have the classical two synaptic weight sets, one for the RBF centers and the other for the linear output weights and perform the same operation. The hierarchical levels can be defined during the clustering stage. All those RBFNNs are individually trained in the same classical way, using conventional algorithms well known from the literature. Thus the internal symmetry of the hierarchical topology and the simplicity of the training strategy renders the method promising.

Finding centers is performed top-down and hierarchically decomposes a dataset into several parts from coarse grained to fine grained levels, by employing conventional clustering algorithms. Then the calculation of linear output weights starts from the bottom level RBF by using conventional regularized least squares and continues up towards the higher levels. Such a hierarchical Markovian structure of RBF Neural Networks can potentially become practical in overcoming common scalability issues that arise in RBF training on large datasets.

2 Related Work in Hierarchical Learning Neural Networks

Although the idea of hierarchical learning approximations is rather old [4] in the Neural Network community, the innovative studies in hierarchical Neural Network models emerge only recently, during the last years. Designed for function approximation and 3-D meshing a multi-scale model called Hierarchical Radial Basis Function (HRBF), was studied in [10][11][12][13] by Ferrari and co-workers. Proposed for clinical diagnosis the model of Mat Isa et al. [14] uses two RBF Neural Networks cascading together, where the first classifies and filters the data and the second uses only the particular attributes that provided by the first. For system identification, classification and time-series forecasting [15][16][17] the flexible hierarchical RBF Neural Network model proposed by Chen and co-workers, also called flexible Neural tree [15][17] from which was originated. Some distinct differences of those pioneering works from our study should be mentioned at this point for clarity.

The model of Ferrari et al. [11][12][13] which was proposed as a hierarchical RBF for multiscale function approximation is constituted of hierarchical layers, each containing a Gaussian grid at a decreasing scale. That is why it is suitable for 3D processing. The scaling sigma parameter is the same for all Gaussians in a particular layer in the gridding approach. The training phase is based on a hierarchical gridding of the input space where additional layers of Gaussians at lower scales are added when the residual error is higher. The weight of each Gaussian is estimated through a maximum a posteriori estimate of Nadaraya-Watson regression type, carried out locally on a sub-set of the data points. During the reconstruction phase, all the multi-resolution different RBF layers work on the same input data and the same hidden layer. This phase is additive as it is based on residuals. Thus only the training phase is hierarchical (top-down).

Our hierarchical Markovian RBF approach for classification differs since: (1) it is not restricted to 3-Dimensional datasets, (2) in contrast to gridding it allocates centers in an adaptive sense via clustering, (3) the multi-resolution RBF layers are in different hidden layer in the hierarchy, (4) it can estimate the weights via conventional linear least squares, (5) The function supports a clear RBF recursion throughout the hidden neurons. In the model of Ferrari et al. the outputs of the RBF nodes in one level are re-weighted in the next level by the Nadaraya-Watson estimator. In our hierarchical markovian RBF model these RBF outputs are reweighted by another RBF Neural Network. That is why the proposed method can also potentially work as a mixture of experts. As stated in [10] such hierarchical RBF Networks can also be seen as a special case of the mixture of experts models [18] [19] where each hidden layer can be seen as a sub-Network specialised in a certain range.

In the model of Chen et al. [15][16][17] multiple Neural Networks are assembled in the form of an acyclic graph. Leaf nodes may have totally different input features. The hierarchical neural structure was evolved through various evolutionary algorithms. Also the output of one RBFNN node (or neural tree node) becomes the input of another. It is a cascading network operation. The input neurons of the next level receive the outputs of the previous level, like the cascading machines. In our hierarchical Markovian RBF model in fig. 1 the input neurons of all levels are fed with the unknown x sample (much like the hierarchical mixture of experts) and the hidden neurons of the next level are those that sum up the combined outputs of the previous level nodes. It is not a cascaded machine since the operation is recursive.

Within the hierarchical Markovian RBF we propose in this work and analyze in the next sections the computationally intensive calculations are likely to be the inversion of large Gram matrices ($G^T G$), or long gradient descent run times, since small sets of RBF centers will be exist in every level nodes. Notice here that the Markovian property we specifically employ in the method is a key ingredient without which it would be rather difficult to achieve pure recursion. Thus the proposed training procedures would be basically transformed into simple Gaussian summations needed to compute the several smaller regressor matrices in every level. Such summations are especially known for their ability to be performed in parallel.

3 Hierarchical Markovian RBF Neural Network Topology

A Radial Basis Function Neural Network (RBFNN) is composed of three layers of input neurons, hidden neurons and output neuron respectively, with two synaptic weight sets, \mathbf{C} (for RBF centers) and \mathbf{w} (for linear output weights), in between them. For a two-class problem there is one output. For an M class-problem there are M outputs. The hidden RBF layer is nonlinear, whereas the output layer is linear. The hidden RBF neurons form K receptive fields and map the input space of an unknown input vector x onto a new space. For RBFs popular choices are Gaussian functions with appropriate centers and covariances. The RBF outputs are communicated via weighted links w_k to the output layer where the sum $f()$ is calculated as:

$$f(\vec{x}) = \sum_{k=1}^K w_k \cdot \exp(-\|\vec{c}_k - \vec{x}\|^2 / \sigma_k^2) \quad (1)$$

With a given training set $\{\vec{x}_i, y_i\}_{i=1}^N$, the training algorithm search the parameter space to find the best of all centers c_k , sigmas σ_k and weights w_k (the center coordinates, widths and heights of the Gaussian bells). There are various different algorithms for optimized training in such RBF Neural Networks [5][6][7][8][9]. Several such RBFNNs in a tree structure can form a Hierarchical RBF NN.

Fig. 1 illustrates, for the two-class problem, the symmetric architecture of the proposed Hierarchical RBF NN as well as its operation that adopts the Markovian property. Note that as the Neural Networks at the bottom level are purely RBF in nature the same holds also for all the Neural Networks in all levels. They are characterized by radial basis functions with specific RBF centers that form the hidden neurons.

The divide and conquer strategy is widely used in many scientific fields such as machine learning [3] in order to divide complex problems into a set of simpler ones. A Hidden Markov model, in the form of a series of naive Bayes classifiers, is such an example application [3] that uses merging of simple models to learn a complex concept. Hidden Markov models as well as Markov random fields known from image analysis make use of the Markov property which says that contextual constraints are given by local interactions, giving rise to Markov chain rule. The proposed Hierarchical Markovian structure (fig. 1) exploits the simple Markov property that sums over all intermediate centers \vec{c}'_i between the vector \vec{x} and the center \vec{c} in order to explicitly output the response function $\varphi(\|\vec{c} - \vec{x}\|)$ as:

$$\varphi(\|\vec{c} - \vec{x}\|) = \sum_{i=1}^{K'} w'_i \cdot \varphi''(\|\vec{c} - \vec{c}'_i\|) \cdot \varphi'(\|\vec{c}'_i - \vec{x}\|) \quad (2)$$

The first term φ'' in the right side of the function in eq. 2 is the cost of choosing \vec{c} when true pattern is \vec{c}'_i , given by a Gaussian distribution multiplied by its weight, while the second term φ' is the response of the previous level. The \vec{c}'_i centers are those that belong to the \vec{c} cluster group of size K' . Hence a center \vec{c} response sums all the contributions from the child centers \vec{c}'_i that belong to its cluster group.

Thus the Markovian RBF NN facilitates an explicit hierarchically structured framework where classical RBF NN training algorithms can be applied.

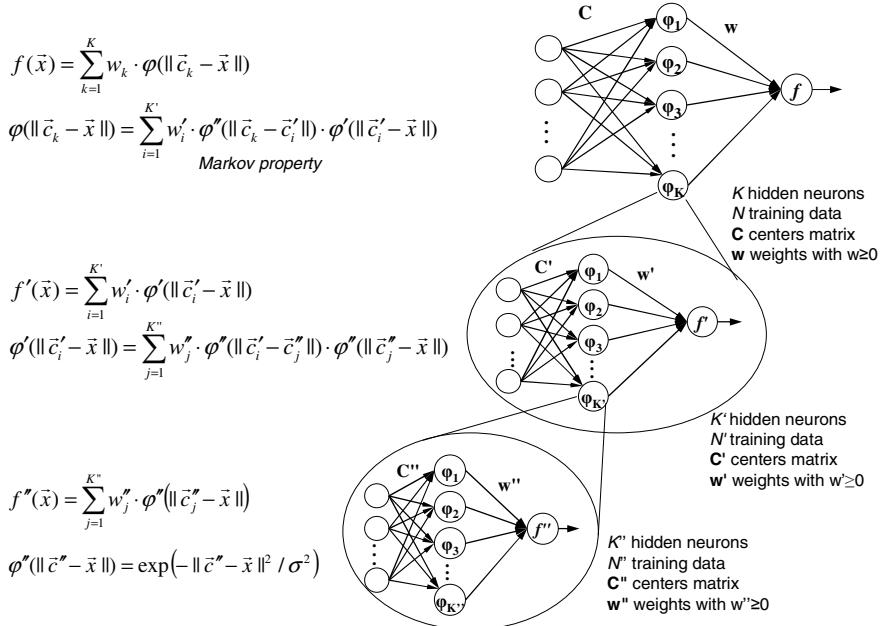


Fig. 1. Hierarchical Markovian RBF Neural Network topology and operation for the two-class problem. This hierarchy is on the hidden neurons, and consequently it allows the recursion.

In fig.1 the top level is composed of K clusters. Each cluster has a corresponding center. All centers are stored in the \mathbf{C} matrix (of size K centers $\times d$ dimensions). By clustering the points inside each of these clusters one can define the clusters in the next level of whom the K' in number centers are stored in their corresponding \mathbf{C}' matrices. At the third level in the bottom the \mathbf{C}'' matrices holds the K'' in number cluster centers. The bottom level outputs all the $f''()$ functions, the middle level outputs $f'()$ functions and the top level outputs all the $f()$ functions. In order to find the linear output weights \mathbf{w} , \mathbf{w}' and \mathbf{w}'' the computation starts from the bottom level for each RBF Neural Network using the training samples it holds. When the hierarchical Markovian RBF Neural Network is in operation each level requires all the outputs of the previous level in order to work. Optimization is done with the constraint that all the linear weights \mathbf{w} , \mathbf{w}' and \mathbf{w}'' entries must be nonnegative ($w_i \geq 0$, $w'_i \geq 0$, $w''_i \geq 0$). The imposed restrictions for nonnegative weights prevent the negative class outputs that otherwise could occur. Then a non negative quadratic optimization solver can compute the linear weights.

The advantage of allowing multiple local models is counterbalanced by the cost of storing and querying the local models for each unknown sample, and these kinds of applications can scale well by means of parallelism.

4 Hierarchical Markovian RBF Neural Network Training

4.1 Top-Down RBF Centers Finding

The RBF center selection is performed top-down through the hierarchy in order to find the centers and thus fill in the matrices \mathbf{C} , \mathbf{C}' , \mathbf{C}'' that illustrated in the fig.1 example. A tree structure is formed. Several training methods can be applied. Popular algorithms [8] select the RBF centers randomly, or employ supervised center selection or most commonly use unsupervised clustering for center selection. Clustering algorithms such as k-Means, X-means, fuzzy c-Means or Subtractive Clustering are all well documented for this RBF training stage. We use the last one. Note that cover trees are also strong candidates. Once the clustering is completed, the mean of each cluster is used as RBF center. To finally construct a three level hierarchical RBF Neural Network in fig. 1 the training data inside each cluster are recursively clustered towards the low levels. Specifically for the top level K clusters are formed, having centers that are stored in the associated \mathbf{C} matrix. Then, the clustering is repeated inside each one of the K cluster group of points of the top level in order to find K' clusters inside them and to store their centers in the corresponding \mathbf{C}' matrices. In consequence the same clustering is repeated in all the K' clusters to find K'' clusters inside them and to store their centers to the \mathbf{C}'' matrices of the bottom level. A typical stopping criterion can be defined here to terminate the recursive clustering procedure if any cluster has less than a predefined number of points, or at the opposite to further continue clustering into extra than three levels in depth, if a cluster holds more than another predefined large number of points. As a demonstration we can utilize the subtractive clustering [9], which selects the denser samples and assigns to them the nearest surrounding points to form each cluster.

Apparently this hierarchy promotes the master/worker parallel architecture. A master processor, the parent RBFNN node, splits the data into K clusters and assigns jobs to many worker processors, the child RBFNN nodes. When the workers finish the local clustering into K' clusters they assign the partitions into their child process nodes to continue. A synchronization point is not required at each level of the top down approach since the jobs are processed independently. The hierarchical Markovian RBF NN in fig. 1 has a weak point. For symmetry it needs from all RBFNN nodes to own data points from all classes. This problem can simply be solved during clustering by merging a cluster that has a missing class with its closer cluster that holds points from this class.

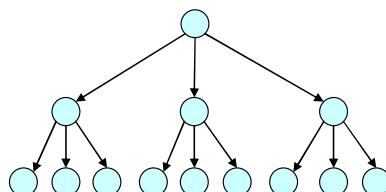


Fig. 2. The top down parallel Master/worker partitioning

4.2 Bottom-Up Linear Weights Finding

The output linear weights \mathbf{w} , \mathbf{w}' and \mathbf{w}'' for the RBF Neural Network nodes at each level need to be determined. This step is performed bottom-up through the hierarchy by following classical regularized linear least squares [6][7][8][9]. The Markovian RBF makes possible the bottom-up merging of the lower level models into higher ones. The linear weights $\mathbf{w} = \{\mathbf{w}_i, i=1, \dots, K\}$ are computed as $\mathbf{w} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T \mathbf{y}$ where $\mathbf{y} = \{y_i, i=1, \dots, N\}$ are the desired labels, \mathbf{I} is the unit matrix and λ a small positive regularization parameter. The usual regressor matrix \mathbf{G} holds the outputs of the RBF units from all N samples with entries $g_{ij} = \varphi_{ij}(\|\vec{c}_j - \vec{x}_i\|)$: $\{i=1, \dots, N, j=1, \dots, K\}$.

An example of the bottom-up merging is the training of the three level hierarchical RBF in fig.1. For a bottom level RBF Neural Network the \mathbf{w}'' weight vector would be given, in terms of the \mathbf{G}'' matrix of size $N'' \times K''$:

$$\mathbf{w}'' = (\mathbf{G}''^T \mathbf{G}'' + \lambda \mathbf{I})^{-1} \mathbf{G}''^T \mathbf{y}'', \text{ with } \mathbf{G}''[i][j] = \varphi''(\|\vec{c}_j'' - \vec{x}_i\|) \text{ and } (\mathbf{w}'' \geq 0)$$

where optimization is done with the constraint that the \mathbf{w}'' entries to be nonnegative. To this end we utilize a simple non negative quadratic optimization solver.

Similarly for a middle level RBF Neural Network the \mathbf{w}' output weight vector would be given, in terms of the \mathbf{G}' matrix of size $N' \times K'$, by:

$$\mathbf{w}' = (\mathbf{G}'^T \mathbf{G}' + \lambda \mathbf{I})^{-1} \mathbf{G}'^T \mathbf{y}', \text{ with } \mathbf{G}'[i][j] = \varphi'(\|\vec{c}_j' - \vec{x}_i\|) \text{ and } (\mathbf{w}' \geq 0).$$

Note that in this way there is a different \mathbf{G} matrix for each class since the outputs $f_j()$ of the previous level are class conscious. At the root RBFNN the \mathbf{w} weight vector would be given, in terms of \mathbf{G} matrix of size $N \times K$, by:

$$\mathbf{w} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T \mathbf{y}, \text{ with } \mathbf{G}[i][j] = \varphi(\|\vec{c}_j - \vec{x}_i\|) \text{ and } (\mathbf{w} \geq 0).$$

In each case \mathbf{y} , \mathbf{y}' , and \mathbf{y}'' are the desired label vectors (hot encoding is 1 for the same class labels and 0 for the other classes) for the corresponding training datasets of size N , N' and N'' respectively. Since a level requires all the outputs of the previous level, this training starts from bottom and continues up.

This bottom-up training stage favours the master/worker architecture. Each leaf of the tree in fig. 3 is assigned into one worker processor. When workers finish, the local results are merged into the local masters. A synchronization point exists in every tree level. When the level nodes complete their job, then their masters, the parent nodes, become workers with their corresponding parents to play now the role of the master.

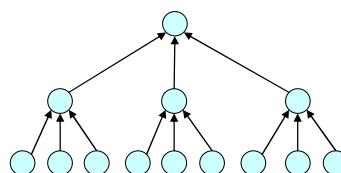


Fig. 3. The bottom-up parallel Master/worker merging

5 Experimental Simulations and Discussion

Experimental simulations for the parallel master/worker implementations are performed for the parallel Hierarchical Markovian RBF Neural Network training. We test both the top-down parallel partitioning via the subtractive clustering algorithm [9] for the RBF centers finding and the bottom-up parallel Master/worker merging for the linear weights solving in every tree level. The performance tests are conducted on a cluster of workstations of Intel Pentium with a clock speed of 2.5 GHz, memory 2GB and Linux system. The machines are interconnected with 1000 Mbps Ethernet. We measure the total parallel execution time versus the number of processors and the speedup S / P , where S the sequential run time in a single processor and P the time that simulates the Network in parallel.

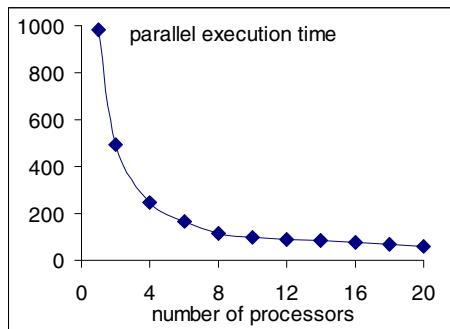


Fig. 4. Total parallel execution time versus the number of processors

Fig. 4 illustrates the performance evaluation using the parallel execution time versus the number of processors for one artificial dataset with $N=500000$. Using 20 processors the parallel execution time was reduced by 94%. While 984 seconds were needed for training the sequential Hierarchical Markovian RBF NN, only 61 seconds were required for the parallel one to finish. The speedup is close to linear.

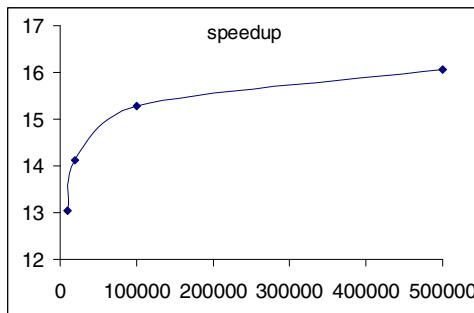


Fig. 5. Speedup as measured on different datasets

In fig. 5 one can see the ratio of sequential execution time per parallel execution time using 20 processors when tested in variable size datasets. This ratio assists in the scalability analysis. By increasing the dataset size the parallel hierarchical Markovian RBFNN improves its scalability towards linear speedup.

The complexity in the first part of the training phase depends on the type of algorithm used for top-down RBF centers finding and the number of levels. The second part of the training is the bottom-up linear weights finding in every node and depends on the number of nodes. Assuming a node has K RBF centers and holds N training data, then the computational complexity for every node (see fig. 1 for K, N notations) is $O(KN)$ to compute the regressor matrix \mathbf{G} and $O(K^3)$ to invert the gram matrix $(\mathbf{G}^T \mathbf{G})$, much like the single RBFNN case. However the work here is distributed and thus can be parallelized. Moreover the K number in every level is usually much smaller than N ($K \ll N$). For example inverting a single matrix of size $K \times K$ with $K=100000$ is rather difficult to implement. On the other hand, by having 1000 nodes in the hierarchy and solving 1000 matrices with $K=100$ is much faster.

When the network is in operation the complexity is bounded by the number of the RBF centers in the leaf nodes that compute $\varphi''(\|\vec{c}_j'' - \vec{x}\|)$, the responses to unknown x , since the other computations $\varphi''(\|\vec{c} - \vec{c}'_i\|)$ between the centers in different levels are fixed and thus can be pre-computed and permanently stored. If the population of all leaf RBF centers is L then the operation cost for classifying an unknown pattern is $O(L)$ equal to that of a single RBFNN that has L hidden units. In other words while internally, during training, it is a hierarchical Markovian structure, externally, during operation, it can be seen by an outsider as a single RBFNN. The hierarchy is hidden.

6 Conclusions

This work proposes a new hierarchical Markovian RBF Neural Network approach, suitable for parallelization studies. The Markovian property permits the hierarchical RBF function in a fractal-like fashion that supports a clear recursion. It also permits the separation of the internal view (training) from the external view (operation). The hierarchical levels can be determined during a clustering stage. We further analyze the general framework and some potential training methods. All the hidden ‘neurons’ in the hierarchy levels are composed of fully functional RBF in nature Neural Networks having the two classical synaptic weight sets, namely the \mathbf{C} matrix that holds the RBF centers and the \mathbf{w} vector that holds the linear output weights. Thus the Neural Network operation is exactly the same at all levels of the hierarchical integration.

Experimental simulations which evaluate performance are conducted for the parallel implementation of this hierarchical Markovian structure. By putting many processors in the calculations the parallel execution time is well improved over the sequential execution time. Although preliminary performance results of this approach seem promising further and more extensive experiments are needed. In the future we plan to study and explore the capabilities of the proposed hierarchical Markovian RBF in order to improve its scalability, to find out further potential applications and to examine its effectiveness on real world problems.

References

1. Wilson, G.: *Parallel Programming for Scientists and Engineers*. MIT Press, Cambridge (1995)
2. Pacheco, P.: *Parallel Programming with MPI*. Morgan Kaufmann, San Francisco (1997)
3. Dietterich, T.G.: The Divide-and-Conquer Manifesto. *Algorithmic Learning Theory*, 13–26 (2000)
4. Moody, J.E.: Fast learning in multi-resolution hierarchies. In: *Neural Information Processing Systems*, pp. 29–39 (1988)
5. Han, M., Xi, J.: Efficient clustering of radial basis perceptron Neural Network for pattern recognition. *Pattern Recognition* 37, 2059–2067 (2004)
6. Hu, Y.H., Hwang, J.N. (eds.): *Handbook of Neural Network signal processing*. CRC Press LLC (2002)
7. Wang, L., Fu, X.: *Data Mining with Computational Intelligence*. Springer (2005)
8. Karayiannis, N., Randolph-Gips, M.: On the Construction and Training of Reformulated Radial Basis Function Neural Networks. *IEEE Transactions on Neural Networks* 14(4), 835–844 (2003)
9. Sarimveis, H., Alexandridis, A., Bafas, G.: A fast training algorithm for RBF Networks based on subtractive clustering. *Neurocomputing* 51, 501–505 (2003)
10. Borghese, N.A., Ferrari, S.: Hierarchical RBF Networks and local parameters estimate. *Neurocomputing* 19, 259–283 (1998)
11. Ferrari, S., Maggioni, M., Borghese, N.A.: Multi-scale approximation with hierarchical radial basis functions Networks. *IEEE Transactions on Neural Networks* 15(1), 178–188 (2004)
12. Ferrari, S., Frosio, I., Piuri, V., Borghese, N.A.: Automatic multiscale meshing through HRBF Networks. *IEEE Trans. on Instr. and Meas.* 54(4), 1463–1470 (2005)
13. Ferrari, S., Bellocchio, F., Piuri, V., Borghese, N.A.: A hierarchical RBF online learning algorithm for real-time 3-D scanner. *IEEE Transactions on Neural Networks* 21(2), 275–285 (2010)
14. Mat Isa, N.A., Mashor, M.Y., Othman, N.H.: Diagnosis of Cervical Cancer using Hierarchical Radial Basis Function (HiRBF) Network. In: Yaacob, S., et al. (eds.) *Proceedings of the International Conference on Artificial Intelligence in Engineering and Technology*, pp. 458–463 (2002)
15. Chen, Y., Yang, B., Dong, J., Abraham, A.: Time-series forecasting using flexible Neural tree model. *Information Science* 174(3-4), 219–235 (2005)
16. Chen, Y., Peng, L., Abraham, A.: Hierarchical Radial Basis Function Neural Networks for Classification Problems. In: Wang, J., Yi, Z., Žurada, J.M., Lu, B.-L., Yin, H. (eds.) *ISNN 2006, Part I. LNCS*, vol. 3971, pp. 873–879. Springer, Heidelberg (2006)
17. Chen, Y., Yang, B., Meng, Q.: Small-time scale Network traffic prediction based on flexible Neural tree. *Applied Soft Computing* 12(1), 274–279 (2012)
18. Jacobs, R.A., Jordan, M., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. *Neural Computation* 3, 79–87 (1991)
19. Jordan, M.I., Jacobs, R.A.: Hierarchical mixtures of experts and the EM algorithm. *Neural Computation* 6, 181–214 (1994)

Data Mining and Modelling for Wave Power Applications Using Hybrid SOM-NG Algorithm

Mario J. Crespo-Ramos¹, Iván Machón-González¹, Hilario López-García¹,
and Jose Luis Calvo-Rolle²

¹ Universidad de Oviedo

² Universidad de A Coruña

Abstract. Renewable energies are increasing their importance in the current society so technical research is increasingly focusing on all of them. An uncommon one is wave power which obtains energy from wave displacements instead of sea level as tidal power. In this work a one year long data set containing hourly measures done by two different buoys was studied. Variable selection was done using a hybrid Self-Organizing Map with a model based method and the same algorithm was used to create a more accurate model to estimate the wave power from atmospheric data. The goals are to demonstrate the adaptability of the algorithm and to increase the knowledge about wave power.

1 Introduction

Due to the actual interest in renewable energies many research projects have put their attention in the sea, focusing on tidal power and wave power. Wave power converts the energy of sea displacements into useful energy, usually electricity.

Choosing the location for a wave farm is a difficult task. A lot of information is usually needed to take such kind of decisions, so usually sensors are located for data acquisition during long times. This is a long and expensive phase in the process of locating a power plant.

Using simulations and models is interesting to avoid buying and installing buoys in every possible location. In this work data mining techniques are used to study oceanographic data obtained with two buoys located in the coast of Asturias (Spain). They measure atmospheric, oceanographic and biological data. For energy purposes biological data is not taken in care.

As it was stated before, the goal is to create a model to estimate the wave energy. Due to the different states of the sea and its behaviour local models are used. Local models have two main challenges, clustering and creating the model for each cluster. For both of them it is important to select the most useful information. Feature selection, or variable selection, is the determination of an appropriate set of inputs for a defined task. In forecasting tasks variable selection is a trade-off between the accuracy of the prediction and the amount of information needed to create the model. Usually a small amount of variables can provide a fair prediction and adding extra inputs can give only a slightly better model that does not worth the computational cost.

Feature selection methods are divided into two main categories: model based and model free [11]. Model free methods are usually related with statistical tools, leaded by Pearson's correlation. Mutual Information (MI) [3] is a non-linear estimator of the effect of a variable's probability distribution into another's distribution. Mutual Information provides an approximation of the relationship between variables not only linear ones. To improve MI for variable selection the minimal-redundancy-maximal-relevance criterion [15] chooses the variables with the greatest amount of new information, comparing them with the previously selected ones.

Model based methods are usually searching algorithms used in combination with a quality measure of the obtained models. The most reliable and inefficient method is the exhaustive search, or brute-force, which checks every possible combination of variables. With the current data acquisition technologies is very common to have great amounts of variables, hundreds or even thousands of them, so heuristics are needed. A common search heuristic is the “Branch and Bound” algorithm [14] which requires a monotonically growing quality measurement, i.e. more variables must produce better quality. In the last years many metaheuristics have been developed and used to improve feature selection, probably the best example is the use of Genetic Algorithms (GA), proposed in [5]. In [6] a statistic that takes in care the relevancy of the variables and their redundancy is used with a philosophy similar to [15].

In this work, variable selection will be done using a model that includes all the candidate variables checking redundancies with component planes and relevancy with gradient analysis.

Self-Organizing Maps are a kind of artificial neural networks (ANN) that provides a set of prototypes, also called codewords or neurons, to represent the original data set. A classical algorithm of this kind is k-means [10,8] which creates a new Voronoi tessellation based on the position of the prototypes and relocates the prototypes in the mean of the data points included in each Voronoi region repeating the process until an acceptable solution is found. This algorithm is still very common and used for many clustering applications because of its simplicity and speed even when it has some drawbacks like the influence of the initial values and the employed metric or providing local minima solutions.

Neural Gas (NG) [12] is a more developed algorithm which includes a fuzzy membership to the Voronoi regions based on a distance ranking with the prototypes. NG provides optimum quantization for the determined training conditions. It also has a supervised version, proposed in the original paper [12], which estimates an output variable using a Local Lineal Model (LLM) associated to each prototype.

Kohonen's Self-Organizing Map (SOM) [7] or Kohonen's Self-Organizing Feature Map (SOFM) is a different algorithm that preserves topological neighbourhood between the high dimensional input data and a low dimensional grid called “map”. SOM is an unsupervised algorithm that returns the position of the prototypes but many other algorithms based on SOM provide supervised learning. For time-series prediction there are special versions, like the Temporal Kohonen Map

(TKM) [1], the Recurrent SOM [16] and the Recursive SOM [17], all of them working with temporally ordered chunks of data instead of using single data vectors. To estimate general data there are other algorithms, like the Hybrid SOM [13] which combines a unsupervised SOM with a multi-layer perceptron (MLP) or the Continuous Interpolating SOM (CI-SOM) [4] which interpolates the output variable between adjacent neurons to estimate the value in an intermediate point.

A hybrid algorithm which combines the quantization quality of NG and the topological preservation of SOM was proposed in [9]. A neighbourhood function is calculated taking in care both the proximity of the prototypes in the input space and their distance in the grid, tuning the importance of each effect with a topology preservation parameter γ . The algorithm was improved adding estimation capability in [2] with local linear models. In this work the information provided by the gradient and its components will be used as data mining tool.

The hybrid algorithm will be shown in Section 2 and the data will be presented in Section 3. The data mining process starts with variable selection in Section 4 and continues with modelling and estimation in Section 5. Finally, conclusions are discussed in Section 6.

2 The Hybrid SOM-NG Algorithm

The hybrid SOM-NG algorithm [9,2] is a hybrid Self-Organizing Map that combines SOM and NG to obtain a trade-off between quantization and topological preservation. It uses the following neighbourhood functions:

$$h_{NG}(v, w_i) = \exp\left(-\frac{k(v, w_i)}{\gamma^2 \cdot \sigma(t)}\right) \quad (1)$$

$$h_{SOM}(v, w_i) = \exp\left(-\frac{s(r_{i*}, r_i)}{\sigma(t)}\right) \quad (2)$$

where $k(v, w_i)$ and $s(r_{i*}, r_i)$ are rank functions, γ is the topology preservation constant and $\sigma(t)$ is the neighbourhood radius in training epoch t . In the h_{NG} term, the usual NG ranking is used, only the parameter γ was introduced to tune the general behaviour of the algorithm. The neighbourhood function for h_{SOM} term is slightly different to the usual SOM, instead of using the Euclidean distance a rank function is used. The expression is:

$$s(r_i, r_{i*}) = |\{r_j / d(r_j, r_{i*}) < d(r_i, r_{i*})\}| \quad (3)$$

which means that the ranking of a neuron in the lattice is equal to the amount of neurons closer to the best matching unit (bmu). So in a rectangular lattice the bmu will have ranking 0, the four perpendicular ones will have ranking 1, the diagonal ones will have ranking 5 and so on. For the sake of simplicity the parameters will be removed from expressions and h_{NG} and h_{SOM} will be used instead of $h_{NG}(v_j, w_i)$ and $h_{SOM}(v_j, w_i)$ in the whole paper.

Using (1) and (2) the learning rule for prototype positions is:

$$w_i = \frac{\sum_j h_{SOM} \cdot h_{NG} \cdot v_j}{\sum_j h_{SOM} \cdot h_{NG}} \quad (4)$$

where w_i is the prototype vector and v_j are the data vectors.

The supervised version of the algorithm calculates a LLM for each prototype, using the expression:

$$\tilde{f}(v) = y_{i*} + a_{i*} \cdot (v - w_{i*}) \quad (5)$$

where $\tilde{f}(v)$ is the estimated function, w_{i*} is the best matching unit, y_{i*} is the reference value $f(w_{i*})$ and a_{i*} is the approximated gradient for its Voronoi region. Using this nomenclature the learning rule for y_i is:

$$y_i = \frac{\sum_j h_{SOM} \cdot h_{NG} \cdot f(v_j)}{\sum_j h_{SOM} \cdot h_{NG}} \quad (6)$$

and the updating rule for a_i is:

$$\Delta a_i = \frac{\sum_j h_{SOM} \cdot h_{NG} \cdot (v_j - w_i) \cdot (f(v_j) - \hat{f}(v_j))}{\sum_j h_{SOM} \cdot h_{NG} \cdot ((v_j - w_i) \cdot (v_j - w_i))} \quad (7)$$

Detailed explanations of the algorithm and how the learning rules are obtained can be found in [9,2].

3 The Data

All the used data was measured by two buoys located in the coast of Asturias. There are 18 atmospheric and oceanographic variables available that will be analysed in this work.

The wave power is not obtained as a direct measure, so it has to be calculated using the measured variables. The wave power is calculated with the Bretschneider-Mitsuyasu model, using the expression:

$$N_L = 0.458 \cdot H_s^2 \cdot T_z \quad (8)$$

where N_L is the wave power in kW/m , H_s is the significant wave height in m and T_z is the mean spectral period in s .

Being rigorous these variables should be removed from the candidates to make the model but in the first phase of this study correlations and redundancies are also analysed, so they will be kept as initial candidates in order to detect redundant variables that should be removed as well.

4 Variable Selection

To make a correct variable selection a model based method will be used. A LLM will be created using all the candidate variables using a very topological restrictive training, i.e. it will be similar to a SOM map but including the estimation

part. This model will be done using a high value of γ , keeping high values of σ using a slow decreasing function with high starting value and train for a larger number of epochs.

The model obtained with this cooperative training is mainly focused on visual inspection for data mining, but using all the variables will also give better apparent estimation errors. In this stage of the study, errors will not be taken in care.

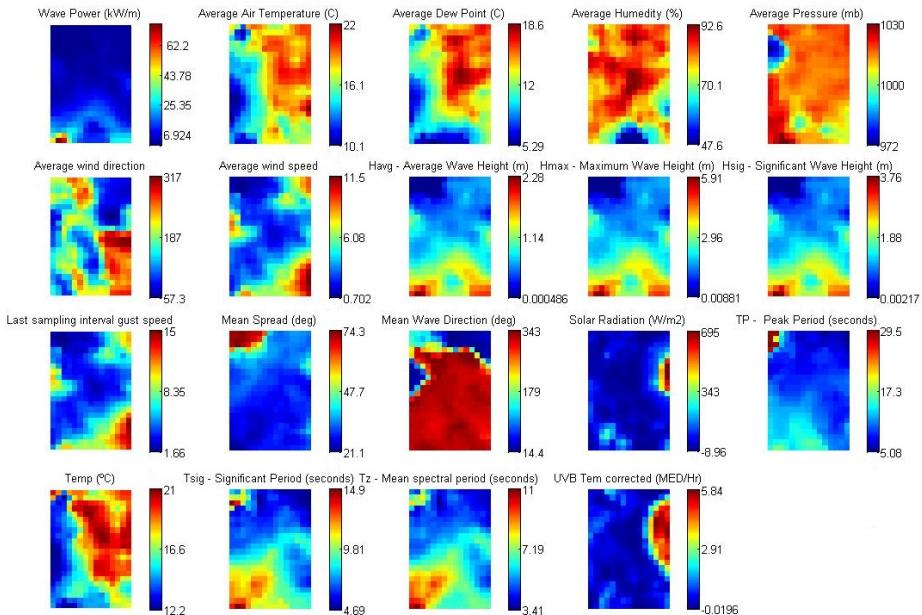


Fig. 1. Codebook using all the candidate variables

In Figure 1 the component planes of the output variable and all the input variables can be seen. It is easy to see that all the wave height variables are highly correlated, and the same with the period ones. As the output variables is calculated using the significant wave height and the mean spectral period all of them should be removed to obtain a real estimator. Other highly correlated variables are Average Air Temperature, Water Temperature (Temp) and Average Dew Point. The radiation variables are also highly correlated: Solar Radiation and UVB Tem corrected.

The local influence of each variable can be seen in the gradient vector. Representing in a box plot the value of each component of the gradient vector, i.e. the approximate partial derivatives, the relevance of each variable is seen.

In Figure 2 gradient components point that the most relevant variables are those related with the wave heights (variables 7, 8 and 9) and periods (variables 16 and 17). These 5 variables are the discarded ones. If they were not used to

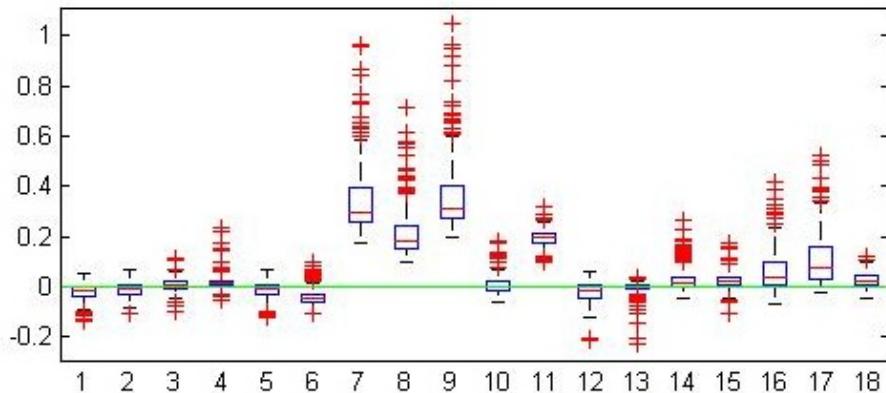


Fig. 2. Gradient components box plot for all the candidate variables

obtain the output variable only one of each should be kept as the weighted sum of several highly correlated variables can be approximated using only one of them. There are other variables that show relevant influence even in presence of the redundant ones, specially the wind related ones (variables 5 and 12).

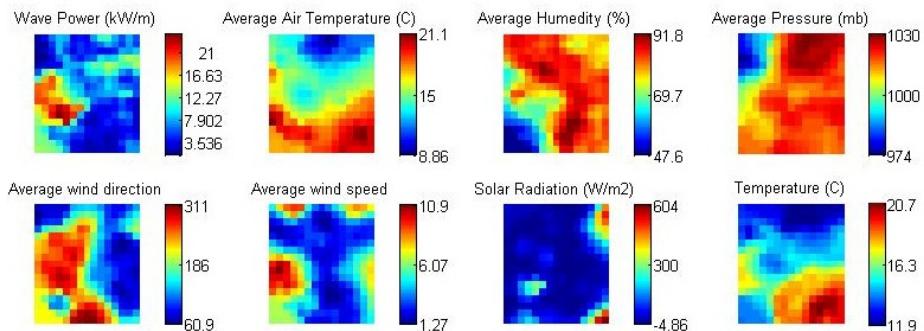


Fig. 3. Codebook using atmospheric variables

After removing the wave related variables only atmospheric data is available, so a new model is needed. This new model will be done using topology preserving parameters too. Now only non-redundant atmospheric variables will be used, keeping only 7 input variables. In Figure 3 the new component planes are represented. As there are no evidences of high correlation, with the only exception of air and water temperatures, relevance will be analysed. Gradient components are represented in Figure 4 and only Solar Radiation seems to have a small relevance. Air Temperature and Humidity have negative influence, Wind Speed

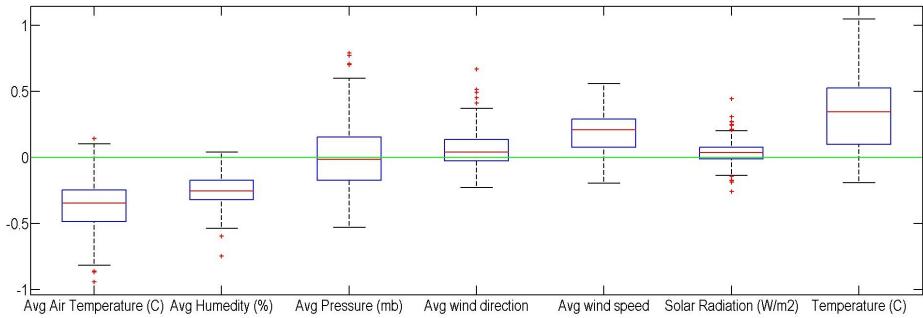


Fig. 4. Gradient components box plot for atmospheric variables

and Water Temperature have positive influence and Pressure has both positive and negative influence depending on the conditions. Wind Direction has smaller gradient components but it is important to determine the model to be used, i.e. the position in the map, as fast winds only produce high power waves if the direction is the right one.

Variable selection was compared with others methods. After removing those variables related to wave properties only 9 variables were left. Genetic algorithms and Branch & Bound method were used to find the optimal choice. It was the same solution than GA, as it was the best estimation variable selection, tested using Brute Force, i.e. checking all possible combinations. These solutions were achieved using a SOM-NG model with $m = 17 \times 15$ and $\gamma = 10$ for each variable group and using their apparent $rmse$ as only quality measure. In that optimum variable selection humidity and wind direction were removed and the solar radiation was replaced by its redundant variable UVB Tem.

5 Estimation Models

Once the relevant variables have been chosen the same algorithm will be used to make an estimation model. In this use topographic preservation is not as important as accuracy but its influence over estimation will be studied. To avoid getting better than real estimation errors cross-validation training was applied. Each model will be done leaving 10 data samples out of the training set, using the k-fold cross-validation method. Estimation errors are divided into “Apparent” which are calculated using the training data and cross-validation error calculated with those removed data samples. After calculating 269 models for each training conditions, the average of the $rmse$ values was done and taken as K-fold error.

Wave height is estimated using the atmospheric variables. The estimated variable is the significant wave height, as it is the one needed to calculate the wave power. This intermediate step has promising results, as the apparent rms error is about 18cm and the cross validation average rms error is about 39cm. Results for three different values of γ are shown in Table 1. These errors point to the fact that wave height can be estimated with atmospheric variables.

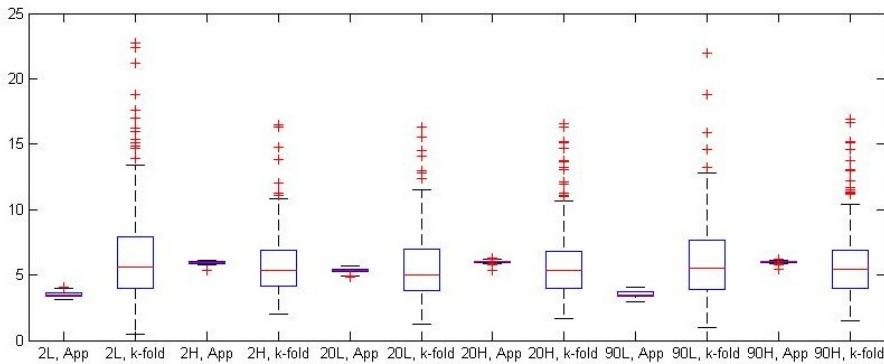
Table 1. Root mean squared estimation error for significant wave height in m

| | Apparent | K-folds |
|---------------|----------|---------|
| $\gamma = 2$ | 0.17741 | 0.38459 |
| $\gamma = 20$ | 0.17757 | 0.39544 |
| $\gamma = 90$ | 0.17843 | 0.38168 |

After estimating the wave height different models were done using three different values of γ and using “high” and “low” neighbourhood radius profiles to estimate the wave power. Using “High” profile neighbourhood radii start with five times the number of prototypes and decreases linearly to 0.01 and with “Low” profile neighbourhood radii have an initial value of half the number of prototypes and decreases exponentially to 0.001. The estimation results are shown in Table 2 and the box plot is represented in Figure 5. Axes in Figure 5 focus on the quartiles, but there are 7 outliers above $25kW/m$ that are not shown to improve the representation.

Table 2. Root mean squared estimation error for wave power in kW/m

| Neighbourhood radius profile | $\gamma = 2$ | | $\gamma = 20$ | | $\gamma = 90$ | |
|---------------------------------|--------------|---------|---------------|---------|---------------|---------|
| | Apparent | K-folds | Apparent | K-folds | Apparent | K-folds |
| High | 5.97 | 5.95 | 6.00 | 5.97 | 6.01 | 6.00 |
| Low | 3.506 | 6.578 | 5.35 | 5.71 | 3.52 | 6.28 |

**Fig. 5.** RMSE box plot for different models

Apparent errors can be easily detected as they have very low standard deviation for all the models, as well as cross-validated errors have higher deviations. The main difference between models is the fact that the extremely competitive or cooperative models have lower apparent errors because of fitting the training set but they have higher cross-validated errors. Nevertheless, more balanced models offer more similar estimation errors with training and test data sets.

6 Conclusions

A whole data mining process was done. The variable selection phase was done using topography preserving conditions in order to facilitate visual inspection. Using visual inspection component maps can reveal correlations, even local ones. Additionally gradients also provide a lot of useful information about influences. Gradient component planes can be represented to detect locally relevant information, as well as box plots can reveal the significance of all the variables in one figure. Irrelevant variables usually have very low gradient values and relevant ones have high positive, negative or both values. Using the box plot it is important to take in care the quartiles, not only the median or mean values, as some variables are highly positive in some conditions and highly negative in others.

About the energy problem it was shown how local models can help to estimate complex variables. In this work common atmospheric data is used to estimate the wave power so, as temperatures and winds are usually available and predictable, wave power can be estimated too. Only one year of information was used knowledge extraction was done but using several years data will give more accurate estimators. Using several years data allow to have different local models for each season, such as very cold winters or windy ones instead of having only one kind. The main drawback of these models is that they are also local in a geographic sense, probably being inaccurate or even not valid for other places.

Acknowledgements. This research was funded by the Spanish Economy Ministry with the grant BES-2008-002664. Data was provided by the Campus of International Excellence in Universidad de Oviedo, whose members kindly advised us for the best data use.

References

- Chappell, G.J., Taylor, J.G.: The Temporal Kohonen Map. *Neural Networks* 6, 441–445 (1993)
- Crespo Ramos, M.J., Machón González, I., López García, H., Calvo Rolle, J.L.: Supervised hybrid SOM-NG algorithm. In: ADVCOMP 2011, The Fifth International Conference on Advanced Engineering Computing and Applications in Sciences, pp. 113–118 (2011)
- Fraser, A., Swinney, H.: Independent coordinates for strange attractors from mutual information. *Physical Review A* 33, 1134–1140 (1986)
- Göppert, J., Rosenstiel, W.: The Continuous Interpolating Self-Organizing Map. *Neural Processing Letters* 5, 185–192 (1997)
- Holland, J.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press (1975)
- Khazaee, P., Mozayani, N., Motlagh, M.: A genetic-based input variable selection algorithm using mutual information and wavelet network for time series prediction. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2008, pp. 2133–2137 (2008)

7. Kohonen, T.: Self Organizing Maps. Springer (2001)
8. Lloyd, S.P.: Least squares quantization in pcm. *IEEE Transactions on Information Theory* 28, 129–137 (1982)
9. Machón-González, I., López-García, H., Calvo-Rolle, J.L.: A hybrid batch SOM-NG algorithm. In: The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 198–202 (2010)
10. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, California, USA, vol. 1, p. 14 (1967)
11. Maier, H., Jain, A., Dandy, G., Sudheer, K.: Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. *Environmental Modelling & Software* 25, 891–909 (2010)
12. Martinetz, T., Berkovich, S., Schulten, K.: ‘Neural Gas’ network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks* 4, 558–569 (1993)
13. Nabhani, F., Shaw, T.: Performance analysis and optimisation of shape recognition and classification using ann. *Robotics and Computer-Integrated Manufacturing* 18, 177–185 (2002)
14. Narendra, P.M., Fukunaga, K.: A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers* C-26, 917–922 (1977)
15. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 1226–1238 (2005)
16. Varsta, M., Heikkonen, J., Millan, J.: Context learning with the Self Organizing Map, tech. report, Laboratory of Computational Engineering Helsinki University of Technology (April 1997)
17. Voegtlin, T.: Recursive self-organizing maps. *Neural Networks* 15, 979–991 (2002)

Automatic Detection of Different Harvesting Stages in Lettuce Plants by Using Chlorophyll Fluorescence Kinetics and Supervised Self Organizing Maps (SOMs)

Xanthoula Eirini Pantazi¹, Dimitrios Moshou¹, Dimitrios Kasampalis², Pavlos Tsouvaltzis², and Dimitrios Kateris¹

¹ Aristotle University of Thessaloniki, School of Agriculture, Department of Hydraulics, Soil Science and Agriculture Engineering, Laboratory of Agricultural Engineering, P.O 275 54 124, Thessaloniki, Greece

² Aristotle University of Thessaloniki, School of Agriculture Department of Horticulture and Viticulture, Laboratory of Vegetable Crop Production, P.O 275 54 124, Thessaloniki, Greece
renepantazi@gmail.com, dmoshou@agro.auth.gr

Abstract. Agriculture aims at increasing production and provision of high quality products to the market. Most of the times, quality is strongly correlated with harvesting stage of each product. Specifically, lettuce qualitative characteristics and nutrients appear to vary strongly in different development stages. In 46, 60 and 70 days of growth, the plants were harvested at baby, immature and mature stage. Then, the parameters of chlorophyll fluorescence were determined in two middle leaves of 3 plants of each hybrid at different harvest stage by using chlorophyll fluorescence kinetics. The measurements revealed significant differences between harvesting stages. The fluorescence parameters were utilized as inputs for training different models of supervised Self Organizing Maps (SOMs) aiming at the prediction of harvesting stage. It was shown that the prediction of different harvesting stages is h by supervised SOMs due to non-linearity nature of the problem which is owned to the heterogeneity of the fluorescence kinetics parameters.

Keywords: neural networks, data mining, clustering, horticulture, postharvest quality.

1 Introduction

An important issue for vegetable production concerns their harvesting stage which is highly associated with their nutrient content. The content of certain nutrients increases with age, due to senescence. In the same paper, it is concluded that the same trend appeared in the content of potassium, magnesium, manganese, iron, and zinc, which decreased progressively during the seven-day harvest period. Currently, there is no objective method to characterize the senescence of lettuce plants and therefore there is no existent standard according to which the growth stage can be characterized except of the size of the lettuce head. But between different hybrids the lettuce head size can vary so it is not reliable criterion regarding the determination of the growth stage. For

this reason a more objective criterion is sought regarding the determination of the growth stage as related to the level of senescence. A frequently applied technique to determine lettuce maturity concerns optical remote sensing either spectroscopic or by using fluorescence. Chlorophyll fluorescence has been routinely used for many years to monitor the photosynthetic performance of plants non-invasively. Possible specific applications of chlorophyll fluorescence include the screening of plants for tolerance to environmental stresses and for improvements in glasshouse production and post-harvest handling of crop. It is already known that when a dark-adapted leaf is exposed to light, large changes in chlorophyll fluorescence occur.

The rapid changes in fluorescence that occur during the rapid induction to a peak have long been attractive for detecting differences in photosynthetic performance between plants. The light energy absorbed by plants is converted into chemical energy (photosynthesis), heat and fluorescence.

Self-Organizing Maps (SOMs) are one of the most well-known among the several Artificial Neural Networks architectures proposed in literature [1]. Their applications have increased during the last decade and they have been applied in several different fields and nowadays they are considered as one of the foremost machine learning tools and an important tool for multivariate statistics [2]. Self-Organizing Maps (SOMs) are self-organizing systems able to solve problems in an unsupervised way, without needing target data. In order to cover certain needs, unsupervised models have been extended in order to be able to work in a supervised framework. To this end, methods like counterpropagation Artificial Neural Networks (CP-ANNs), which are very similar to SOMs, since an output layer is added to the SOM layer [3], have been introduced.

When dealing with classification issues, CP-ANNs are generally efficient methods for achieving class separation in non-linear boundaries. Recent modifications to CP-ANNs have led to the introduction of new supervised neural network architectures and relevant learning algorithms such as Supervised Kohonen Networks (SKNs) and XY-fused Networks (XY-Fs) [4]. In the current paper several Self Organizing Map using supervised learning approach and algorithm are used to classify fluorescence kinetics data in order to determine the harvesting stages of lettuce. To achieve this, two different parameterizations of fluorescence kinetics curves are developed; one is corresponding to fluorometer variables and the other resulting from polynomial fitting of Kautsky Curves.

2 Materials and Methods

One of the main objectives in agriculture is to identify ways in which chlorophyll fluorescence may be used effectively to improve plant selection processes and rapidly evaluate plant performance in agricultural and horticultural crop improvement programmes. Specifically, in the case of lettuce qualitative characteristics and nutrients appear to vary strongly in different development stages. In the current research, lettuce plants belonging to the hybrids Mastamar, Atoll and Starfighter of Batavia type as well as hybrids of Bacio, Picos CLX and Picos FM of Romana type

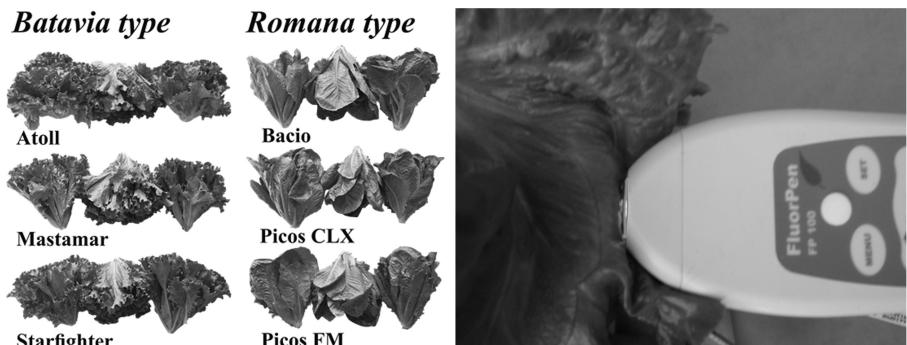


Fig. 1. (a) Hybrids Batavia type (Atoll, Mastamar and Starfighter) and Romana Type (Bacio, Picos CLX, Picos FM). (b) FluorPen FP 100-MAX-LM.

(Fig 1.a) were cultivated in a heated greenhouse constructed from glass during the period of 15/10-27/12/2012. In 46, 60 and 70 days of growth, the plants were harvested at baby, immature and mature stage.

Then, the parameters of chlorophyll fluorescence were determined in two middle leaves of 3 plants of each hybrid at different harvest stage by using chlorophyll fluorescence kinetics. The measurements revealed significant differences between harvesting stages by utilizing FluorPen FP 100-MAX-LM of SCI (Fig. 1. (b)) which is capable of measuring chlorophyll fluorescence kinetics through the OJIP method which concerns the fluorescence transient [7].

The fluorescence parameters were utilized as inputs for training different models of supervised Self Organizing Maps (SOMs) aiming at the prediction of harvesting stage. Fluorescence kinetics have been already used for the detection of mealiness in apples [5] by using self-organizing maps to discriminate between different mealiness severity levels. Multisensor fusion of fluorescence kinetics and hyperspectral imaging has been applied for the detection of plant diseases [6]. Self-Organizing Maps have been used for the monitoring and classification of pea varieties (*Pisum sativum*) according to their degree of resistance against drought stress [8].

2.1 Fluorescence Parameters

Many fluorescence parameters have already been proposed. The rapid changes in fluorescence that occur during the induction of photosynthesis when a dark-adapted leaf is exposed to light, have long been attractive for detecting differences in photosynthetic performance between plants (Fig. 2).

With this instrument, the fluorescence is excited by ultra-bright light emitting diodes (LED) with a peak wavelength of 650 nm. Chlorophyll fluorescent signals were detected using a photocell after passing through a high-pass filter (50% transmission at 720 nm). The recording time during the experiments was 1 s with a resolution of 10 μ s during the first 2 μ s and after that with a resolution of 1 μ s, resulting in 1200 values per measurement. The lettuce plants were not dark-adapted

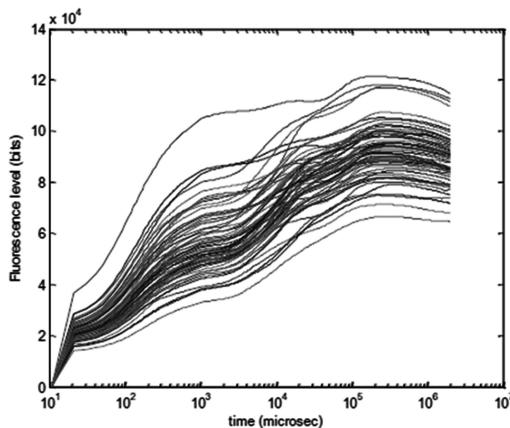


Fig. 2. Kautsky Curves resulting from the fluorometer

and were held under normal artificial lighting before measuring. The induction curves are shown in Fig.2.

2.2 Data Analysis

2.2.1 Fluorescence Parameters of the FP100

The fluorometer calculates automatically certain geometric parameters of the Kautsky curves. The obtained parameters are shown in Table 1. The fluorescence data were normalised by dividing by F_0 (or F_{50} ms), the fluorescence value measured after 50 ms. This value corresponds to the initial fluorescence, F_0 and is the fluorescence intensity when the electron acceptors are in their oxidised form. By normalisation, differences between the measurements due to factors, which are not characteristic of the internal properties of the sample as for example fluctuations in excitation light intensity caused by the diminishing power of the battery, and variations in extent of the excited surface due to variations in the curvature of the surface and in homogeneities in the tissue (e.g. lenticels), were minimised. Fluorescence was measured with high time resolution, resulting in a large number of data points for each measurement. In this case, it would be desirable to select a relatively small subset of variables that contain as much information for classification as the original curves. The fluorescent signals F in bits were fitted against $\log(\text{time})$ by a polynomial model of tenth order

$$(F = (\beta_0 + \beta_1 x + \beta_{10} x^{10}) \text{ with } x = \log(\text{time})) \quad (1)$$

The regression coefficients β_0 to β_{10} formed an 11-dimensional feature vector that was used in the classification algorithms. The tenth order polynomial model was chosen because it had demonstrated the best fitting accuracy.

Table 1. Fluorescence parameters that are calculated automatically by the fluorometer

| Formula abbreviation | Formula explanation |
|---|--|
| 1. Bckg | Background |
| 2. F_0 | $F_{50\mu s}$, fluorescence intensity at 50 μs |
| 3. F_J | fluorescence intensity at J-step (at 2 μs) |
| 4. F_i | fluorescence intensity at i-step (at 60 μs) |
| 5. F_M | maximal fluorescence intensity |
| 6. F_v | $F_M - F_0$ (maximal variable fluorescence) |
| 7. V_J | $(F_J - F_0) / (F_M - F_0)$ |
| 8. V_i | $(F_i - F_0) / (F_M - F_0)$ |
| 9. F_M / F_0 | |
| 10. F_v / F_0 | |
| 11. F_v / F_M | |
| 12. M_0 or $(dV/dt)_0$ | $TR_0 / RC - ET_0 / RC = 4 (F_{300} - F_0) / (F_M - F_0)$ |
| 13. Area | Area between fluorescence curve and F_M (background subtracted) |
| 14. Fix Area | Area below the fluorescence curve between $F_{40\mu s}$ and F_{1s} (background subtracted) |
| 15. S_m | Area / ($F_M - F_0$) (multiple turn-over) |
| 16. S_s | the smallest SM turn-over (single turn-over) |
| 17. N | $S_M, M_0 \cdot (1 / V_J)$ turn-over number Q_A |

Table 1. (*continued*)

| | |
|--|---|
| 18. ϕ_{Po} or TR_0 / ABS | $(1 - F_0) / F_M$ (or F_V / F_M) |
| 19. ψ_0 or ET_0/TR_0 | $1 - V_J$ |
| 20. ϕ_{Eo} or ET_0/ABS | $(1 - (F_0 / F_M)) \cdot \psi_0$ |
| 21. ϕ_{Dlo} | $1 - \phi_{Po} - (F_0 / F_M)$ where $Dl_0 = ABS - TR_0$ |
| 22. ϕ_{Pav} | $\phi_{Po} (S_M / t_{FM})$, t_{FM} = time to reach F_M (in μs) |
| 23. PI_{ABS} | $(RC/ABS) \cdot (\phi_{Po}/(1-\phi_{Po})) \cdot (\psi_0/(1-\psi_0))$. |
| 24. ABS / RC | $M_0 \cdot (1 / V_J) \cdot (1 / \phi_{Po})$ |
| 25. TR_0 / RC | $M_0 \cdot (1 / V_J)$ |
| 26. ET_0 / RC | $M_0 \cdot (1 / V_J) \cdot \psi_0$ |
| 27. DI_0 / RC | $(ABS / RC) - (TR_0 / RC)$ |

2.2.2 Counterpropagation Artificial Neural Networks

Counterpropagation Artificial Neural Networks (CP-ANNs) are modeling methods which combine features from both supervised and unsupervised learning [3]. CP-ANNs consist of two layers, a Kohonen layer and an output layer, whose neurons have as many weights as the number of classes to be modelled. The class vector is used to define a matrix C, with I rows and G columns, where I is the number of samples and G the total number of classes; each entry c_{ig} of C represents the membership of the i-th sample to the g-th class expressed with a binary code (0 or 1). When the sequential training is adopted, the weights of the rth neuron in the output layer (y_r) are updated in a supervised manner on the basis of the winning neuron selected in the Kohonen layer. Considering the class of each sample i, the update is calculated as follows:

$$\Delta y_r = \eta \left(1 - \frac{d_n}{d_{\max} + 1} \right) (c_i - y_r^{old}) \quad (2)$$

where d_{ri} is the topological distance between the considered neuron r and the winning neuron selected in the Kohonen layer; c_i is the ith row of the unfolded class matrix C, that is, a G-dimensional binary vector representing the class membership of the ith

sample. At the end of the network training, each neuron of the Kohonen layer can be assigned to a class on the basis of the output weights and all the samples placed in that neuron are automatically assigned to the corresponding class.

2.2.3 XY-Fused Networks

XY-fused Networks (XY-Fs) [4] are supervised neural networks for building classification models derived from Self- Organizing Maps (SOMs). In XY-fused Networks, the winning neuron is selected by calculating Euclidean distances between a) sample (x_i) and weights of the Kohonen layer, b) class membership vector (c_i) and weights of the output layer. These two Euclidean distances are then combined together to form a fused similarity, that is used to find the winning neuron. The influence of distances calculated on the Kohonen layer decreases linearly during the training epochs, while the influence of distances calculated on the output layer increases.

2.2.4 Supervised Kohonen Networks (SKNs)

As in the case for CP-ANNs and XY-Fs, Supervised Kohonen Networks (SKNs) [4] are supervised neural networks derived from Self-Organizing Maps (SOMs) and used to calculate classification models. In Supervised Kohonen Networks, Kohonen and output layers are glued together to give a combined layer that is updated according to the training scheme of Self-Organizing Maps. Each sample (x_i) and its corresponding class vector (c_i) are combined together and act as input for the network. In order to achieve classification models with good predictive performances, x_i and c_i must be scaled properly. Therefore, a scaling coefficient for c_i is introduced for tuning the influence of class vector in the model calculation.

3 Results and Discussion

The CPANN, SKN and XYF Networks were trained with the fluorescence based parameters and also the polynomial approximation of the Kautsky Curves in order to recognize discriminate the 3 stages at which the lettuce was harvested. The data consisted of 210 lettuce samples of which 60 belonged to the first stage, another 60 was at the second stage while the third stage was presented by 90 samples. In order to test the generalization capability of the networks cross validation was performed by splitting randomly the training data in four groups and using the three groups for training and the fourth group for testing. This process was repeated for all possible combinations of three groups and the average result was recorded. The results of different architectures regarding the size of the Self-Organizing Map are presented in Tables 2 and 3. Different Map sizes that were tested included 3x3, 5x5, 8x8, 10x10 and up to 30x30 with step 5. The shape of SOMs was rectangular. A common observation is that with increasing size the results tend to improve. Another important observation concerns the use of polynomial features which improve the generalization

capacity of smaller architectures. This proves that the polynomials can describe properties of the Kautsky Curves that were not determined by the fluorometer parameters. In order to assess the importance of the variables that constitute the fluorescence based features that are used by the Self-Organizing Map, the mean values of the weights of the SOM units that correspond to each class have been calculated. The variables that are more important will tend to have higher average weight values since more training samples would positively contribute to obtain this specific class in correlation with this specific variable. The average weights per class are plotted in Fig.3.

Table 2. Successful recognition of actual growth at stages 1, 2 and 3 from different network architecture based on the fluorescence features provided in Table 1

| Type of used network | Successful recognition of actual growth at stage 1 | Successful recognition of actual growth at stage 2 | Successful recognition of actual growth at stage 3 |
|----------------------|--|--|--|
| Xyf 30x30 | 100 % | 95 % | 100 % |
| Cpann 30x30 | 100 % | 95 % | 100 % |
| Skn 30x30 | 100 % | 96,7 % | 100 % |

Table 3. Successful recognition of actual growth at stages 1, 2 and 3 from different network architecture based on the features derived from the coefficients of polynomials fitted to the Kautsky Curves

| Type of used network | Successful recognition of actual growth at stage 1 | Successful recognition of actual growth at stage 2 | Successful recognition of actual growth at stage 3 |
|----------------------|--|--|--|
| Xyf 30x30 | 100 % | 98,3 % | 97,8 % |
| Cpann 30x30 | 100 % | 96,7 % | 96,7 % |
| Skn 30x30 | 98,3 % | 98,3 % | 98,9 % |

Looking at the results of the Tables 2 and 3, one could, in principle, utilize the classifiers that have been resulted taking into account both representations. Specifically, more weight could be given in the classifier used fluorescence features as far as classes 1 and 3 are concerned and vice versa for class 2. For sizes 10x10 and 20x20 the results of both classifiers based on fluorescence or polynomial features are worse. For example in the case of XY-F the diagonal elements were: 68, 3%, 76.7% and 87,8% respectively.

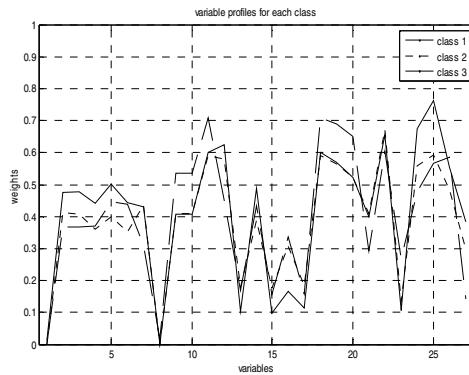


Fig. 3. The mean input weight values of the SOM for each class (corresponding to one of the three harvesting stages)

From Fig.3 we can observe that the variables that are more important with respect to the discrimination in three harvesting stages of lettuce are variable 25(TR_0 / RC) for the first harvesting stage (class 1), variable 7(V_J) for second harvesting stage (class 2) and variables 9(F_M/F_0), 10(F_V/F_0) and 18(TR_0 / ABS) for third harvesting stage (class 3). In order to visualize the contribution of the variables the two first principal components (explaining the 95.65% of the total information) of the weights of the SOM are plotted in Fig.4. The corresponding variables demonstrate a clear separation of classes since they are placed in different quarters of the plane. The correct classification percentages resulting from Supervised Kohonen Network with 30 neurons are 100%- 96, 67%-100% for classes 1-2-3 respectively. By comparing these results with Tables 2 and 3 it is obvious that the selected variables give a superior result and there is no need to use all the fluorescence variables.

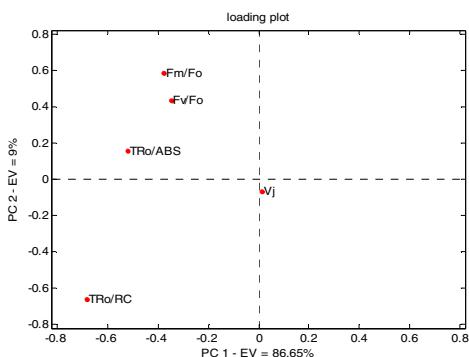


Fig. 4. Loading plot of the first two principal components calculated on the SOM weights. Each variable is labeled according to the fluorescence feature which is presented.

4 Conclusion

It is already known that lettuce qualitative characteristics and nutrients appear to vary strongly in different development stages. Fluorescence kinetics screening can be applied to determine different harvesting stages of lettuce based on the response of the plants to light excitation. In this paper, it was shown that the prediction of different harvesting stages can be achieved with supervised SOMs due to non-linearity problem which is owned to the heterogeneity of the fluorescence kinetics parameters. A parameterization using polynomial fitting of Kautsky Curves gives satisfactory results. The selection of the most important fluorescence features based on the SOM weight profiles per class provide an additional advantage which results in a better classification with less fluorescence features.

References

- [1] Kohonen, T.: *Self-Organization and Associative Memory*. Springer, Berlin (1988)
- [2] Marini, F.: Artificial neural networks in food analysis: trends and perspectives. *Analytica Chimica Acta* 635, 121–131 (2009)
- [3] Zupan, J., Novic, M., Gasteiger, J.: Neural networks with counter-propagation learning strategy used for modelling. *Chemometrics and Intelligent Laboratory Systems* 27, 175–187 (1995)
- [4] Melssen, W., Wehrens, R., Buydens, L.: Supervised Kohonen networks for classification problems. *Chemometrics and Intelligent Laboratory Systems* 83, 99–113 (2006)
- [5] Moshou, D., Wahlen, S., Strasser, R., Schenk, A., Ramon, H.: Apple Mealiness. Detection using Fluorescence and self- organizing maps. *Computers and Electronics in Agriculture* 40, 103–114 (2003)
- [6] Moshou, D., Gravalos, I., Kateris, D., Pantazi, X.E.: Water stress detection based on optical multisensor fusion with a least squares support vector machine classifier. In: Oral and Proceedings of IV International Workshop on Computer Image Analysis in Agriculture, 3rd CIGR International Conference of Agricultural Engineering (CIGR-AgEng 2012), July 8-12, Valencia, Spain (2012)
- [7] Strasser, R., Srivastava, A., Tsimilli-Michael, M.: The fluorescence transient as a tool to characterise and screen photosynthetic samples. In: Yunus, M., Pathre, U., Mohanty, P. (eds.) *Probing Photosynthesis: Mechanisms, Regulation and Adaptation*, pp. 445–483. Taylor & Francis, London (2000)
- [8] Ronald, M.-R., Stancho, P., Alberto, G., Abdallah, O., Strasser Reto, J.: Can machines recognize stress in plants? *Environmental Chemistry Letters* 1, 201–205

Analysis of Heating Systems in Buildings Using Self-Organizing Maps

Pablo Barrientos, Carlos J. del Canto, Antonio Morán, Serafín Alonso,
Miguel A. Prada, Juan J. Fuertes, and Manuel Domínguez*

SUPPRESS Research Group, Esc. de Ing. Industrial e Informática, Campus de Vegazana s/n,
24071, León, Spain

{pbarf,cdelm,a.moran,saloc,ma.prada,jj.fuertes,
manuel.dominguez}@unileon.es
<http://suppress.unileon.es>

Abstract. The highest cause of energy consumption in buildings is due to 'Heating, Ventilation, and Air Conditioning' (HVAC) systems. However, a large number of interconnected variables are involved in the control of these systems, so conventional analysis approaches are difficult. For that reason, data analysis by means of dimensionality reduction techniques can be a useful approach to address energy efficiency in buildings. In this paper, a method is proposed to visualize the relevant features of a heating system and its behavior and to help finding correlations between temporal, production and distribution variables. It uses a modification of the self-organizing map. The proposed approach is applied to a real building at the University of León.

Keywords: Self-Organizing Maps, Heating systems, Data analysis.

1 Introduction

One of the goals of the European Union (EU) is to reduce the CO_2 emissions in 20% by 2020 with the 20-20-20 plan. It is a fact that 40% of these emissions are caused by buildings, and 40% of emissions in buildings are due to HVAC (Heating, Ventilation and Air conditioning) systems [1]. For that reason, studies oriented to reduce CO_2 emissions in HVAC systems are required to meet this target. Furthermore, public buildings should be an example of the savings discussed above. A possible action is to ensure that HVAC control systems meet the requirements efficiently. It must be noted that each building presents a different behavior and its control system must be programmed considering its particularities [2].

Nevertheless, even the HVAC system of a medium-size public building may include several circuits and a large number of variables. A comprehensive theoretical study might therefore be economically unfeasible. Alternatively, as addressed in this work, the engineer can rely on visual inspection of the system [3]. However, the space of potentially informative variables is high-dimensional and the visual analysis of those variables becomes difficult. For that reason, the application of a dimensionality reduction

* This work was supported in part by the Spanish *Ministerio de Ciencia e Innovación* (MICINN) and the European FEDER funds under grant DPI2009-13398-C02-02.

technique is proposed to generate a 2D visualization space that aims to preserve most of the structure and patterns of the original data [4]. The self-organizing map and its visualization tools have been proposed in the literature to visually find correlations and clusters [5,6]. The variables associated to a heating system can be classified in groups such as temporal, production or distribution variables. An ordered analysis with respect to the time of the day or the day of the week can help to identify trends or highlight anomalous behaviors in the production or distribution variables. For that reason, the approach proposed in this paper uses a variant of the self-organizing map that allows to make the organization hierarchically conditional on some common variables.

The method presented in this work is tested on the heating system of the School of Education at the University of León. The aim is twofold. It is interesting to study the relationship among production and distribution variables, as well as with time, and the analysis of these consumption patterns can lead to the detection of anomalies in the system elements.

This paper is structured as follows: The proposed approach is presented in Section 2. In Section 3, a heating system is described in detail. The experimental results obtained are analyzed in Section 4. And finally, conclusions are drawn in Section 5.

2 Visual Analysis of Heating Systems

In order to provide an ordered visual analysis of a heating system, a dimensionality reduction technique needs to be used to preserve the main characteristics of the input space in a low-dimensional space. To organize the projected data according to some common variables, a variant of the self-organizing map, known as envSOM [7], has been used in this work. This algorithm is appropriate for visualization and comparison of large data sets from several processes, given common environmental information such as time variables.

2.1 Self-Organizing Maps

The Self Organizing Map [8] is a neural network based on unsupervised learning which produces a low-dimensional (generally two-dimensional) grid of the input space preserving the topological properties. Each neuron or unit is associated with a weight vector (codebook vector) \mathbf{m}_i with the same dimension as the data vectors in the input space, and a position vector \mathbf{g}_i in the low-dimensional grid of the output space. The units are related with adjacent ones according to a neighborhood function that is defined by the map topology.

In the sequential training, two stages are repeated for each sample of the input data set for a number of cycles. In the first stage, the best matching unit (BMU) is selected as the unit whose codebook vector \mathbf{m}_c minimizes the Euclidean distance to a given input vector $\mathbf{x}(t)$:

$$c(t) = \arg \min_i \|\mathbf{x}(t) - \mathbf{m}_i\|. \quad (1)$$

After that, the codebook vectors of the BMU and its neighbors are adapted according to the next equation:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t)h_{ci}(t)[\mathbf{x}(t) - \mathbf{m}_i(t)]. \quad (2)$$

where $\alpha(t)$ is the learning rate (typically monotonically decreasing with time), t denotes time and $h_{ci}(t)$ is the neighborhood function around the winner unit (BMU), which is usually implemented as Gaussian.

As a result of the training, the units of the grid end up placed in areas with high density of data dividing the space in a set of Voronoi regions. Therefore, the resulting low-dimensional grid preserves the topology of the input data set, effectively performing a dimensionality reduction, and compresses it, performing vector quantization. The properties of this grid enable low-dimensional, compact and ordered visualizations. For that reason, many visualization have been proposed [9]. In this work, the component planes are used, which associate the values of a variable in the codebook vectors \mathbf{m}_i , to their corresponding coordinates in the lattice \mathbf{g}_i and show them using color.

2.2 Proposed Approach

The analysis of the heating systems requires a detailed understanding of the main variables which define the system behavior. Generally, the main variables of heating systems can be classified into different groups according to the structure of the whole system. It can be easily distinguished production, distribution and time variables among others. The study and comparison of distribution circuits which provide heat to different building zones is vital to verify and optimize the operation of the heating system.

The envSOM algorithm extended to n phases [10] can be useful for this purpose since it captures relevant behavior patterns in real past data from several processes and build models conditioned hierarchically on all classes of common variables. The envSOM [7] allows us to analyze and compare different processes influenced by a common environment representing, to a large degree, the probability density function of the input data, given the environmental conditions. The main difference with a traditional SOM is that this algorithm introduces binary masks (ω and Ω) in the BMU computation and adaptation phase.

For that reason, in this work, we propose a method for monitoring the main variables involved in a heating system using a three-phase envSOM (see Fig. 1):

- Phase 1.** In the first phase, a SOM-based model of the time variables (TV_t , $t = 1, 2 \dots$) is achieved. The input vector given by Eq. 3 is used for the training.

$$\mathbf{x} = [TV_1 \dots TV_t \ddot{\vdots} PV_1 \dots PV_p \ddot{\vdots} DV_1^1 \dots DV_s^1 \ddot{\vdots} \dots \ddot{\vdots} DV_1^k \dots DV_s^k]. \quad (3)$$

In this vector, TV_t represents the temporal variables, PV_p represents the production variables and DV_s^k represents the distribution variables for each of the k circuits.

In this phase, a basic SOM is training using all variables of the input vector, but only the temporal variables are used to obtain the BMU. For this purpose the binary mask defined in the Eq. 4 is used in the winner computation according to the Eq. 5.

$$\omega^{(1)} = [1 \dots 1 \ddot{\vdots} 0 \dots 0 \ddot{\vdots} 0 \dots 0 \ddot{\vdots} \dots \ddot{\vdots} 0 \dots 0]. \quad (4)$$

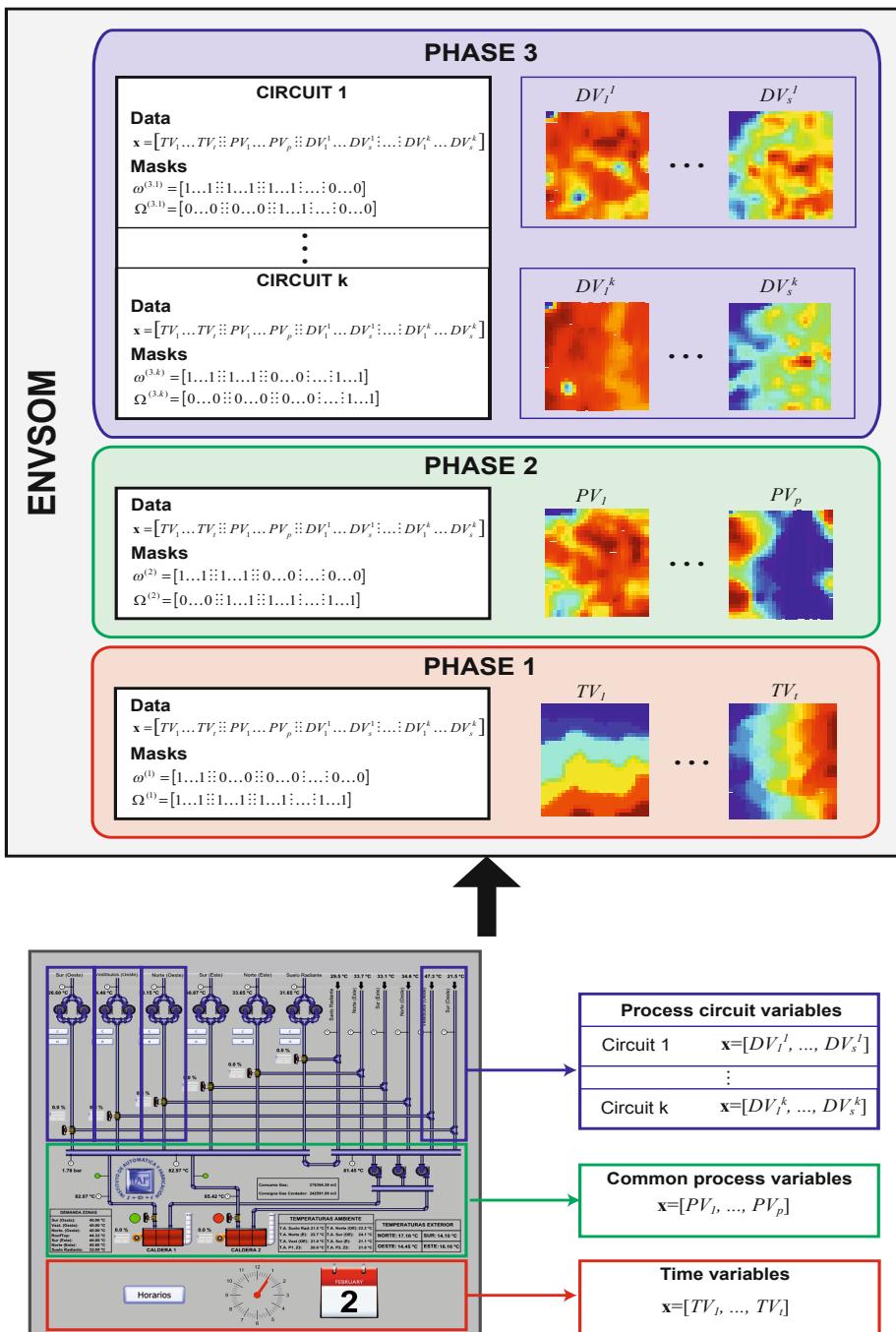


Fig. 1. Proposed approach for the analysis of heating systems

$$\begin{aligned} c(t) &= \arg \min_i \|\mathbf{x}(t) - \mathbf{m}_i(t)\|_{\omega^{(n)}} , \quad i = 1, 2, \dots, M , \\ \|\mathbf{x}(t) - \mathbf{m}_i(t)\|_{\omega^{(n)}}^2 &= \sum_k \omega_k^{(n)} [x_k(t) - m_{ik}(t)]^2 , \end{aligned} \quad (5)$$

where \mathbf{x} represents the current input, \mathbf{m} indicates the codebook vectors and $\|\cdot\|$ denotes the Euclidean norm. M and t are the number of units or neurons and the time, respectively. The SOM adaptation rule is not modified in this phase.

2. **Phase 2.** In the second phase, a new SOM is trained using only the production variables for computing the BMU according to the Eq. 5. The initialization is made using the codebook vectors from the phase 1 and the input vector is the same as the phase 1 (Eq. 3). The winner searching mask $\omega^{(2)}$ is as follows:

$$[0 \dots 0 \ddot{\vdots} 1 \dots 1 \ddot{\vdots} 0 \dots 0 \ddot{\vdots} \dots \ddot{\vdots} 0 \dots 0]. \quad (6)$$

In this case, a binary mask is introduced in the adaptation phase of a traditional SOM (Eq. 7), so that all variables are updated, except the temporal ones since they have already been well organized in the previous phase.

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) h_{ci}(t) \Omega^{(n)} [\mathbf{x}(t) - \mathbf{m}_i(t)]. \quad (7)$$

Therefore, in this case the adaptation mask is:

$$\Omega^{(2)} = [0 \dots 0 \ddot{\vdots} 1 \dots 1 \ddot{\vdots} 1 \dots 1 \ddot{\vdots} \dots \ddot{\vdots} 1 \dots 1]. \quad (8)$$

3. **Phase 3.** In the third phase, a new SOM is trained for each distribution circuit using its corresponding masks. The codebook vectors from the phase 2 are used in the initialization of this phase and the input vector is the same as the phase 1 (Eq. 3). The masks used in this phase are defined in the following:

– Circuit 1

$$\omega^{(3,1)} = [1 \dots 1 \ddot{\vdots} 1 \dots 1 \ddot{\vdots} 1 \dots 1 \ddot{\vdots} \dots \ddot{\vdots} 0 \dots 0]. \quad (9)$$

$$\Omega^{(3,1)} = [0 \dots 0 \ddot{\vdots} 0 \dots 0 \ddot{\vdots} 1 \dots 1 \ddot{\vdots} \dots \ddot{\vdots} 0 \dots 0]. \quad (10)$$

– Circuit k

$$\omega^{(3,k)} = [1 \dots 1 \ddot{\vdots} 1 \dots 1 \ddot{\vdots} 0 \dots 0 \ddot{\vdots} \dots \ddot{\vdots} 1 \dots 1]. \quad (11)$$

$$\Omega^{(3,k)} = [0 \dots 0 \ddot{\vdots} 0 \dots 0 \ddot{\vdots} 0 \dots 0 \ddot{\vdots} \dots \ddot{\vdots} 1 \dots 1]. \quad (12)$$

This phase is used for updating only the distribution variables, since the temporal and production variables are already hierarchically well organized. This way, the distribution variables are organized with regard to the temporal and production variables. That enables the analysis of heating system variables through the comparison among the corresponding component planes.

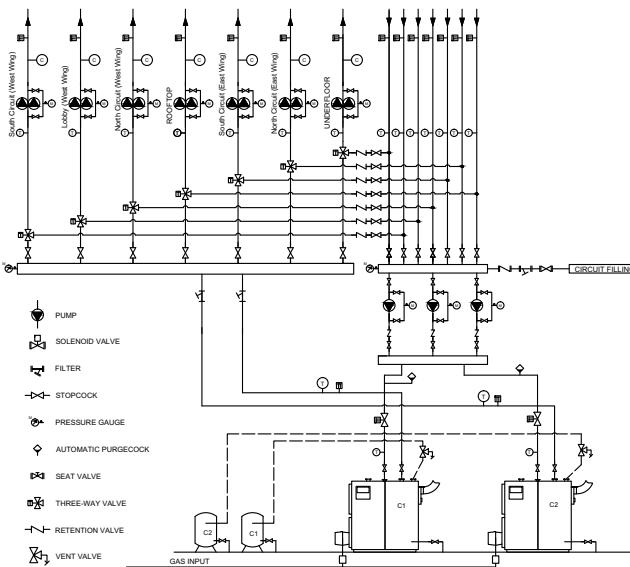


Fig. 2. Heating system of the School of Education

3 Experimentation System: A Heating System in an University Building

The heating system of a building in the tertiary sector, as is the case of our experiment, is composed of two main parts, the production and the distribution circuits. Both parts are linked by the delivery and the return manifolds, considering that these elements belong to the production circuit.

The production circuit is responsible for the generation of heat, and it usually consists of the following elements:

- **Boilers:** They produce heat through combustion of fuel, normally natural gas.
- **Isolation valves:** These valves, located at the water outlet of the boilers, are designed to avoid water flow into a boiler that is not working. To carry out this task, these valves open before turning on the boiler.
- **Production pumps:** These elements ensure water flow through the boilers when they are in operation. It is necessary to have a number of production pumps in operation equal to the number of boilers in operation.
- **Flow detectors:** These security elements indicate whether boilers have water flow. When boilers do not have water flow, they cannot be turned on to prevent damages.
- **Production temperature probes:** They measure the water temperature and are usually located in the delivery and return manifolds and the water outlet of the boilers.
- **Pressure sensor:** This element indicates the water pressure of the installation.

Table 1. Production circuit elements

| Id | Element | Units |
|----|--------------------|-------|
| 1 | Delivery manifold | 1 |
| 2 | Return manifold | 2 |
| 3 | Boilers | 2 |
| 4 | Isolation valves | 2 |
| 5 | Production Pumps | 3 |
| 6 | Flow detectors | 2 |
| 7 | Temperature probes | 4 |
| 8 | Gas meter | 1 |
| 9 | Pressure meter | 1 |

Table 2. Distribution circuit elements

| Id | Element | Units |
|----|---------------------------------|-------|
| 10 | Three-way valves | 1 |
| 11 | Drive pumps | 2 |
| 12 | Distribution temperature probes | 2 |
| 13 | Room temperature sensors | 1 |
| 14 | Outside temperature sensors | 1 |

The distribution circuit is in charge of the water distribution throughout the building to get the desired room temperature. It can be divided into several circuits depending on the magnitude of the building. Each of these circuits is constituted by the next elements:

- **Three way valves:** These valves blend the water from the return circuit with the water from the delivery manifold in order to get the temperature set point.
- **Drive pumps:** They are responsible to move the water through each of the circuits. Normally, two twin pumps are used for each circuit to ensure the heat distribution.
- **Distribution temperature probes:** Again, they measure the water temperature in the delivery and the return of each circuit.
- **Room temperature sensors:** They measure the room temperature in the areas associated to each circuit.
- **Outside temperature sensors:** They measure the outside temperature in each wall of the building.

The heating system of the School of Education at the University of León is used to perform the experiment. This building has been chosen because it has all the representative elements of a heating system with gas boilers. The technical scheme of this system is shown graphically in the Fig. 2. The heating system of this building is centralized with a production of heat by two boilers (Viessmann Vitrond 200) with a power of 350 KW and 94 % rated efficiency. The goal of the production circuit is to ensure that the temperature in the delivery manifold is slightly higher than the highest outlet temperature in the distribution circuit (4-6 °C). For this purpose, the two boilers are combined modulating the burners. To produce the boilers start-up, the production schedule must be enabled and the average outside temperature must be less than a value usually set around 18 °C. Furthermore, the circuit pressure must exceed a minimum which is usually set at 1 bar, and the flow detector must indicate flow through the boiler.

The heat distribution is performed by means of seven distribution circuits that supply heat to each of the parts of the building: South circuit (west wing), Lobby (west wing), North circuit (west wing), Rooftop, South circuit (east wing), North circuit (east wing), and Underfloor heating. The distribution system has a two-pipe installation through the delivery and return manifolds that supply heat to the radiators located in the departments

Table 3. Temporal and production variables

| Id | Variable | Unit |
|----|-------------------------------|-------------------|
| 1 | Day | Days |
| 2 | Hour | Hours |
| 3 | Gas counter | m ³ /h |
| 4 | Boiler 1 modulation | % |
| 5 | Boiler 2 modulation | % |
| 6 | Pressure | bar |
| 7 | Delivery manifold temperature | °C |
| 8 | Return manifold temperature | °C |
| 9 | East outside temperature | °C |
| 10 | North outside temperature | °C |
| 11 | West outside temperature | °C |
| 12 | South outside temperature | °C |
| 13 | Outlet temperature, Boiler 1 | °C |
| 14 | Outlet temperature, Boiler 2 | °C |

Table 4. Distribution variables

| Id | Variable | Unit |
|---------|--|------|
| 15 - 20 | Room temperature. Circuits 1 to 6 | °C |
| 21 - 26 | Outlet temperature setpoint. Circuits 1 to 6 | °C |
| 27 - 32 | Outlet temperature. Circuits 1 to 6 | °C |
| 33 - 38 | Return temperature. Circuits 1 to 6 | °C |
| 39 - 44 | Three-way valve. Circuits 1 to 6 | % |

and public areas. The halls and the library are heated through a network of underfloor heating. Tables 1 and 2 list the existing elements of the heating system under analysis.

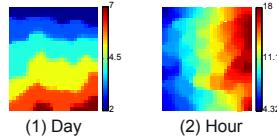
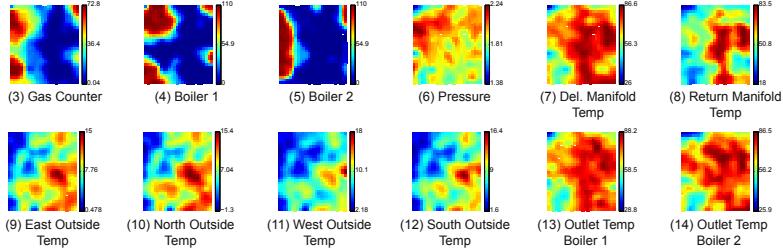
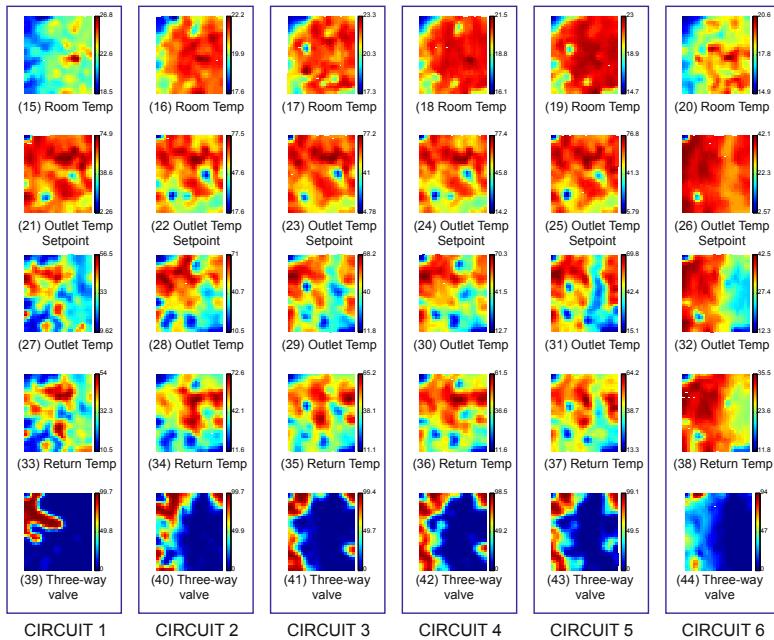
As explained above, the purpose of the distribution circuit is to maintain the desired temperature in all zones of the building providing heat supply to the radiators. To calculate the outlet temperature setpoint in each circuit, the system uses a compensation curve whose parameters are defined by the operator. This temperature is obtained from the outside temperature in each building facade. In addition, the system operator has to define an operation schedule and a comfort temperature for each circuit.

4 Experimental Results

To perform the experiments, 44 variables have been chosen to be used in the training of the proposed algorithm. The variables are classified into three groups: temporal variables (TV), production variables (PV) and distribution variables (DV). Tables 3 and 4 show these variables. The data set comprises 6849 samples corresponding to the variables acquired in a year of heating operation. Figures 3, 4 and 5 show the component planes obtained by applying the method described in section 2 to the data set. The maps have been gotten using a rectangular grid and sheet shapes, all maps have the same size 30x26. The maps are the results obtained in each phase of the training.

The planes from Phase 1 represent the temporal variables, i.e., the day of the week (1) and the hour of the day (2) of the system operation schedule. The component planes from Phase 2 show the most important variables in the production heating circuit, and the result of Phase 3 are the component planes of the relevant variables of each distribution circuit. The visual analysis of these component planes leads to several findings. The most relevant ones are discussed below:

1. Visualizing the components planes 4 and 5, which give information about the boilers modulation, and comparing them with plane 2, it can be observed that the boilers are turned on in the last hour of the operation schedule. This produces an unnecessary heat generation that it is not going to be distributed through the building.
2. Through the analysis of room temperatures (planes 15-20), it can be observed that plane 20, corresponding to circuit 6, has the most uniform temperature. In addition,

**Fig. 3.** Experimental results. Temporal variables.**Fig. 4.** Experimental results. Production variables.**Fig. 5.** Experimental results. Distribution variables.

this temperature is also the nearest one to the setpoint. Therefore, this circuit is the most efficient in providing comfort.

3. The plane 39 represents the setpoint of the three-way valve in circuit 1. It can be observed that there are only two states, totally closed or totally opened. This indicates that the control system opens completely the valve to try to reach the

outlet temperature setpoint. However, the desired outlet temperature was still not reached because the valve was broken and the production of the heating circuit was not enough to heat the distribution circuit.

5 Conclusions

In this work, we have proposed a dimensionality reduction method based on SOM to analyze heating systems. The proposed method has been tested in a real heating system of a building of the University of León. The application of this method enabled the joint monitoring of all the system variables, establishing correlations between temporal, production and distribution variables. The analysis focuses on the comparison of the behavior of the system circuits.

As a result of the analysis, faults in system elements and abnormal behaviors in some circuits have been found. These findings have been linked to the gas consumption of the system. Future work in this line should focus on on-line monitoring of the installations.

References

1. Communication from the Commission: Energy efficiency: delivering the 20% target, COM (2008) 772 final. Technical report, Commission of the European Communities, Brussels (November 2008)
2. Sane, H., Haugstetter, C., Bortoff, S.: Building HVAC control systems - role of controls and optimization. In: American Control Conference, pp. 1121–1126 (2006)
3. Meyers, S., Mills, E., Chen, A., Demsetz, L.: Building data visualization for diagnostics. *ASHRAE Journal* 38(6), 8 (1996)
4. Lee, J.A., Verleysen, M.: Nonlinear Dimensionality Reduction. Information Science and Statistics. Springer (2007)
5. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer-Verlag New York, Inc., Secaucus (2001)
6. Kohonen, T., Oja, E., Simula, O., Visa, A., Kangas, J.: Engineering applications of the self-organizing map. *Proceedings of the IEEE* 84(10), 1358–1384 (1996)
7. Alonso, S., Sulkava, M., Prada, M.A., Domínguez, M., Hollmén, J.: EnvSOM: A SOM algorithm conditioned on the environment for clustering and visualization. In: Laaksonen, J., Honkela, T. (eds.) *WSOM 2011. LNCS*, vol. 6731, pp. 61–70. Springer, Heidelberg (2011)
8. Kohonen, T.: The self-organizing map. *Proceedings of the IEEE* 78, 1464–1480 (1990)
9. Vesanto, J.: SOM-based data visualization methods. *Intelligent Data Analysis* 3(2), 111–126 (1999)
10. Alonso, S., Morán, A., Prada, M.A., Barrientos, P., Domínguez, M.: Monitoring power consumption using a generalized variant of self-organizing map (SOM). *International Journal of Modern Physics B* 26(25), 1246005 (2012)

IMMI: Interactive Segmentation Toolkit

Jan Masek, Radim Burget, and Vaclav Uher

Signal Processing Laboratory, Department of Telecommunications,

The Faculty of Electrical Engineering and Communication,

Brno University of Technology, Brno, Czech Republic

{masek.jan,vaclav.uher}@phd.feeec.vutbr.cz, burgetrm@feec.vutbr.cz

<http://splab.cz/en>

Abstract. General image segmentation is a non-trivial task, which requires significant computational power and huge amount of knowledge incorporated. Fortunately, it is not necessary in all the cases. In some specific cases, simpler non-supervised or supervised segmentation methods can be used giving even better results. In this paper, a novel trainable segmentation method based on RapidMiner data-mining platform is introduced, and its functionality is described. The method implementation was released under open-source license as a part of IMMI (IMage MINing) extension of the RapidMiner platform. When compared to other trainable segmentation algorithms, the platform provides flexibility connected with all the features of one of the most widely used data-mining platform today. The functionality has been verified on the satellite image use-case, accuracy achieving 78.3 % pixel error.

Keywords: Classification, image segmentation, interactive tool, IMMI, RapidMiner.

1 Introduction

Image segmentation is a process where the parts of an image are partitioned into significant regions. It is used in many applications such as biological or satellite data object detection and many others. Unfortunately, it is extremely difficult to obtain a reliable segmentation without any prior knowledge about the object that is being extracted from the scene. This is further complicated due to the lack of any clearly defined metrics for evaluating the quality of segmentation and consequently for comparing segmentation algorithms.

This paper does not focus on design of an universal segmentation algorithm for all the possible cases but rather on a trainable segmentation algorithm, which can be trained to user's demands. This approach has several advantages such as: 1) there is no need to integrate any prior knowledge base, 2) it adapts to user's needs, 3) it can often provide a higher accuracy than unsupervised or universal methods, and 4) it often has lower computational demands.

The algorithm was implemented and released under an open-source license as a part of Image Mining (IMMI) extension for RapidMiner¹ platform. The

¹ RapidMiner is available from: <http://rapid-i.com>

RapidMiner platform is today one of the most widely used data-mining platform worldwide. Video tutorials and related demo processes in RapidMiner have been released online².

The principle of the trainable segmentation algorithm can extract different features from the original image, and consequently, according to the user inputs can train different machine learning algorithms (e.g. support vector machines, k -nearest neighbors, neural networks, decision trees, random forests or any other from more than 250 other learning algorithms available in the RapidMiner) in order to segment the images as desired by a user's input.

Functionality of the trainable segmentation is further demonstrated on satellite image use-case, where output is visualized directly in Google Earth application. The overall achieved accuracy is 78.3 % pixel error. IMMI and RapidMiner tool were also successfully used in several previous works. In [1] temporomandibular joint disc in MRI images was detected by using object detection algorithms, another work [2] deals with artery transverse section detection and [3] deals with artery longitudinal section detection, multi-GPU implementation of face detector was described in [4].

When compared to another projects, this framework puts greater emphasis on the data-mining part. The advantage is that it can utilize more than 250 different learning algorithms (e.g. Random Forests, Decision Tree, k -Nearest Neighbors (k -NN), Support Vector Machines (SVM), Neural Network (NN), Bayes network and others) and meta learning algorithms (Bagging, Stacking, ADABoost, and others). It also can optimize the learned model using e.g. forward selection optimization, backward selection optimization, evolutionary selection optimization and others as well as parameters of a learned model.

The rest of the paper is introduced as follows: Section 2 deals with the related works. In section 3 the principle of trainable segmentation algorithm is described. Section 4 describes open-source IMMI project, which includes the proposed algorithm and in section 5 is an example the functionality of IMMI tool verification. Section 6 concludes this paper.

2 Related Work

There are several tools for the interactive segmentation e.g. interactive learning and segmentation toolkit (ilastik) [5], SIOX. ilastik supports segmentation into multiple classes, 3D segmentation and online prediction. In comparison with IMMI trainable segmentation, ilastik provides only limited number of algorithms, not allowing any apply data-mining optimization techniques. Plugin for simple interactive object extraction in still images (SIOX) was introduced in [6]. This plugin, which is easy utilized is included in GIMP³ image processing tool and it is very easy to use it. There are also not any extra features as training algorithm selection or parameters optimization. Another framework for fast and interactive image and video segmentation and matting was introduced in [7].

² Video tutorials are available online from: <http://splab.cz/ts>

³ GIMP is available from: <http://www.gimp.org/>

These segmentation tools are based on interactive segmentation methods, which are described in many works. In [8] a new method for interactive segmentation of n-dimensional images is proposed. This graph-cut segmentation method is very popular and has many variants [9][10][11]. In [12] segmentation framework, which unifies graph cuts and random walker [13] approaches, is described. Further, this work is extended by optional spanning forest algorithm in [14].

3 Trainable Segmentation

IMMI trainable segmentation is in general based on: 1) local-level feature extraction and 2) machine learning and 3) user input.

3.1 Feature Extraction

On the input of the algorithm, the original image that should be segmented and several selected transformations of the original image should be given. The transformations provide different information, i.e. different features. User inputs are some positive and negative training points. These points tell the algorithm how the user wants to have the image segmented. Thus for each training point a value of a pixel on its location of the original and transformed image is given, plus the information if the point chosen by the user is positive or negative is also provided (i.e. the label using the data-mining terminology). Using different transformation, variety of features can be extracted from the original image. The examples of these transformations are e.g. edge detection, median filtering, minimal and maximal value filtering or Gaussian filtering. So the value of each feature is acquired according to user labeled point that is located in relevant transformed image.

3.2 Training Process

When the feature values are obtained, the training process can start. The training process is applied to these training data. There are several algorithms of artificial intelligence, which can be used for model training (e.g. SVM, Random Forests, Decision Tree, k-NN and the others). The result of the trainable segmentation process is the trained model, which can be applied on testing image, in which each pixel is classified and the result of it is the image with the segmented areas. The parameters, which are set before the training process starts, can be optimized for example using genetic algorithms.

4 Graphical User Interface

The trainable segmentation was implemented and released under open-source license as a part of IMMI extension for the RapidMiner platform. Functional

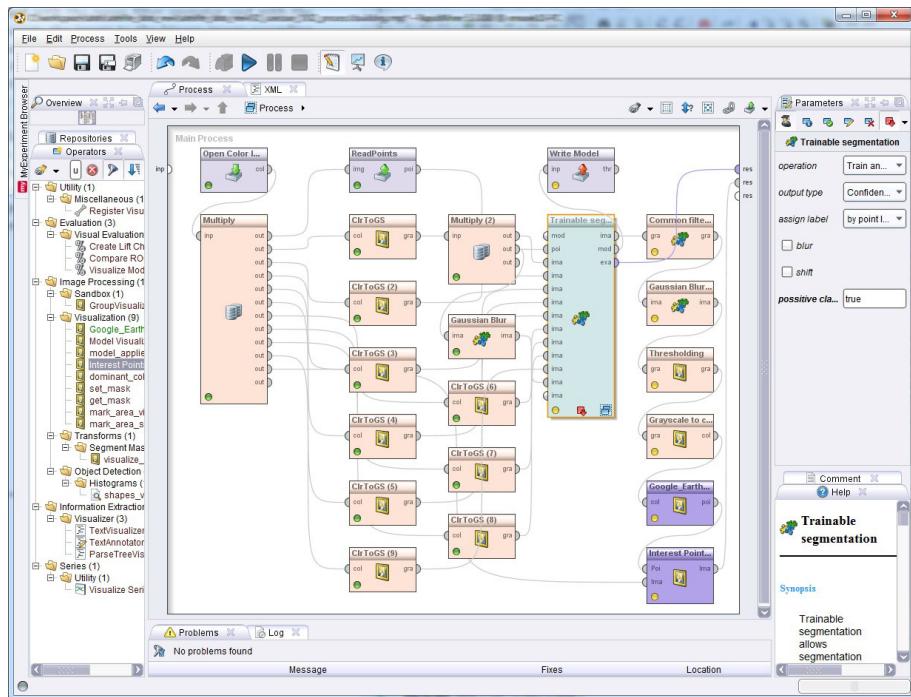


Fig. 1. Training process in RapidMiner platform

blocks in RapidMiner are called operators and they have inputs, outputs and some adjustable parameters. The IMMI extension consists of about 200 operators usable for image processing.

To illustrate how trainable segmentation works, we have selected house detection problem as an example. After training the model, it is applied on the testing image. The RapidMiner graphical user interface and training process are shown in Fig. 1. The first operator loads an input image. Then the training points are selected in that image (see Fig. 2a). Green points (positive label) mark houses and red points (negative label) mark background. Then, some transformations with the input image are carried out e.g. saturation process (Fig. 2b) or Hue value extraction (Fig. 2c) with Gaussian blur. New transformed image is created after each transformation. Then the pixel values on defined green and red points are obtained from these images. Having now several attributes for each training labeled point and we can train the model with these data using the algorithms of artificial intelligence. We select for example Random Forest operator that has to be inserted as sub-process of the trainable segmentation operator, which is set to training mode. Parameters of Random Forest operator was set to default according to RapidMiner. The outputs of training process are: the segmented input image and the trained model, which is stored on a disk. The segmented image contains segmented areas that are shown in gray scale colors. The darker areas



(a) Selected training points. **(b)** The input image after the saturation value extraction. **(c)** The input image after the hue value extraction.

Fig. 2. The input training image and its transformations

determine in this case houses (positive class) and the lighter areas determine background (negative class).

4.1 Threshold Settings

The segmented image is then post-processed with the mean filter operator and with the thresholding operator, which converts the image to the binary format according to its threshold. This threshold was set for all cases to 100 in 0–255 value scale, where 255 is black value and 0 is white value.

4.2 Testing Process

The Google earth annotation operator is used for the finding houses coordinates and also it saves these coordinates into *kml* file, which is executable in the



(a) The original testing image. **(b)** The image after the trainable segmentation process. **(c)** The image after thresholding.

Fig. 3. The results of testing process

Google earth application. The input image is then shown with the detected houses bounds.

For the testing of trained model, the trainable segmentation operator is set to applied mode. Its inputs are: the testing image (see Fig. 3a), which is pre-processed using same transformations as it was in the training process and the second input of this operator takes the saved trained model. The output segmented image (see Fig. 3b) is post-processed with mean filter and then with thresholding operator (see Fig. 3c). Then, the Google annotation operator creates *kml* file, which can be opened in the Google earth application (see Fig. 4). In this picture, the area that was processed is marked with red lines.

5 Results

The functionality of IMMI tool was verified by a use-case, which focuses on house detection in satellite images. The images, which were used in our experiments, were obtained from Google earth application. We selected only five differently scaled images containing cities, because this is only the simple example of IMMI functionality and the aim is not to train a model, which is very accurate on all types of images. Each image was divided into two halves. The first half was used for training and the second half was used for testing. The testing images were manually labeled and then compared with the output thresholded images (see Fig. 3c) to compute accuracy A_{CC} . The results of house detection are described in Table 1, where F_{NR} denotes false negative rate and F_{PR} denotes false positive rate. The approximate accuracy is 78.3% and sometimes it is hard to determine whether processed object is a house or not. Since the roof of house can

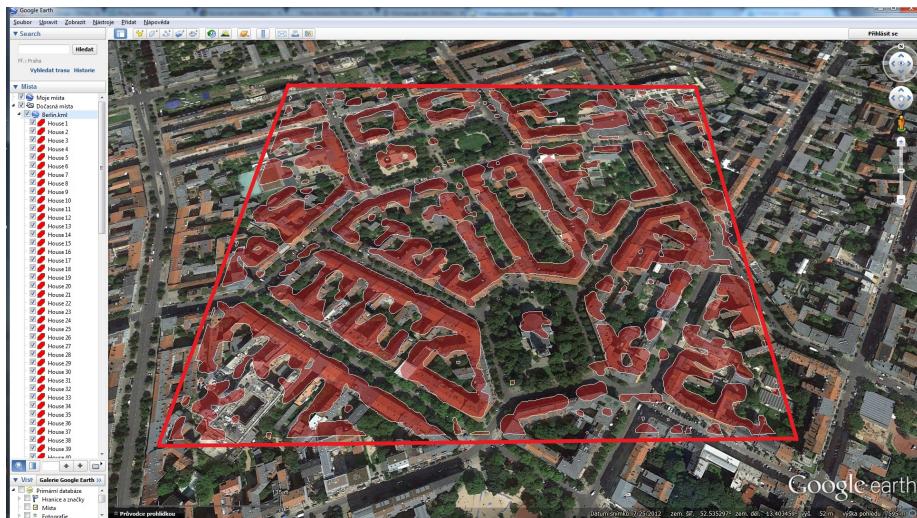


Fig. 4. Google earth application with the resulting image

Table 1. Results of house detection - selected cities

| City | A_{CC} [%] | F_{NR} [%] | F_{PR} [%] |
|----------|--------------|--------------|--------------|
| Berlin | 79.2 | 18.7 | 24.3 |
| Budapest | 76.7 | 21.4 | 29.7 |
| Lisbon | 80.3 | 21.5 | 18.2 |
| Paris | 76.1 | 40.1 | 12.6 |
| Warsaw | 79.2 | 21.6 | 18.6 |

be gray and also the road can have similar shape and color. When the trained model of Paris is used to classify the image of Berlin, the accuracy of detection is 72.2 %.

As seen from the previous results, the house detection from satellite images is still problem today, but these results are only informative. With using different satellite images, the results can be different.

6 Conclusion

In this paper a trainable segmentation algorithm has been introduced. The algorithm has been implemented and released under open-source license online as a part of IMMI extension. Our toolkit consists of many image processing algorithms and it is used as RapidMiner plugin.

The IMMI trainable segmentation algorithm has been described and presented on satellite data use-case.

References

1. Burget, R., Cika, P., Zukal, M., Masek, J.: Automated Localization of Temporomandibular Joint Disc in MRI Images. In: International Conference on Telecommunications and Signal Processing (TSP), pp. 413–416 (2011) ISBN: 978-1-4577-1409-2
2. Riha, K., Masek, J., Burget, R., Benes, R., Zavodna, E.: Novel Method for Localization of Common Carotid Artery Transverse Section in Ultrasound Images Using Modified Viola–Jones Detector. Ultrasound in Medicine and Biology (2013)
3. Masek, J., Burget, R., Karasek, J., Uher, V., Güney, S.: Evolutionary Improved Object Detector for Ultrasound Images. In: International Conference on Telecommunications and Signal Processing, TSP (2013)
4. Masek, J., Burget, R., Uher, V., Güney, S.: Speeding up Viola–Jones Algorithm Using Multi–Core GPU Implementation. In: International Conference on Telecommunications and Signal Processing, TSP (2013)
5. Sommer, C., Straehle, C., Kothe, U., Hamprecht, F.: Ilastik: Interactive Learning and Segmentation Toolkit. In: Proceedings of the 8th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Chicago, Illinois, USA, pp. 230–233. IEEE, ISBI (2011)
6. Friedland, G., Jantz, K., Rojas, R.: SIOX: Simple Interactive Object Extraction in Still Images. In: Seventh IEEE International Symposium on Multimedia (2005)

7. Bai, X., Sapiro, G.: A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting. In: IEEE ICCV, pp. 1–8 (2007)
8. Boykov, Y., Jolly, M.: Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In: IEEE ICCV, pp. 105–112 (2001)
9. Kolmogorov, V., Boykov, Y.: What Metrics Can Be Approximated by Geo-Cuts, or Global Optimization of Length/Area and Flux. In: IEEE ICCV, vol. 1, pp. 564–571 (2005)
10. Price, B., Morse, B., Cohen, S.: Geodesic Graph Cut for Interactive Image Segmentation. In: IEEE CVPR, pp. 3161–3168 (2010)
11. Vicente, S., Kolmogorov, V., Rother, C.: Graph Cut Based Image Segmentation with Connectivity Priors. In: IEEE CVPR (2008)
12. Sinop, A., Grady, L.: A Seeded Image Segmentation Framework Unifying Graph Cuts and Random Walker which Yields a New Algorithm. In: IEEE ICCV (2007)
13. Grady, L.: Random Walks for Image Segmentation. IEEE Trans. PAMI 28(11), 1768–1783 (2006)
14. Couprie, C., Grady, L., Najman, L., Talbot, H.: Power watersheds: A New Image Segmentation Framework Extending Graph Cuts, Random Walker and Optimal Spanning Forest. In: IEEE ICCV (2009)

Local Binary Patterns and Neural Networks for No-Reference Image and Video Quality Assessment

Marko Panić, Dubravko Ćulibrk, Srdjan Sladojević, and Vladimir Crnojević

University of Novi Sad, Faculty of Technical Sciences,

Trg Dositeja Obradovica 6, 21000 Novi Sad, Serbia

{mpanic, dculibrk, sladojevic, crnojevic}@uns.ac.rs

<http://www.ftn.uns.ac.rs>

Abstract. In the modern world, where multimedia is predicted to form 86% of traffic transmitted over the telecommunication networks in the near future, content providers are looking to shift towards Quality of Experience, rather than Quality of Service in multimedia delivery. Thus, no-reference image quality assessment and the related video quality assessment remaining open research problem, with significant market potential. In this paper we describe a study focused on evaluating the applicability of Local Binary Patterns (LBP) as features and neural networks as estimators for image quality assessment. We focus on blockiness artifacts, as a prominent effect in all block-based coding approaches and the dominant artifact in occurring in videos coded with state-of-the-art video codecs (MPEG-4, H.264, HVEC). In this initial study we show how an LBP-inspired approach, tuned to this particular effect, can be efficiently used to predict the MOS of JPEG coded images. The proposed approach is evaluated on a well-known public database and against widely-used features. The results presented in the paper show that the approach achieves superior performance, which forms a sound basis for future research aimed at video quality assessment and precise blocking artifact detection with sub-frame precision.

Keywords: Local Binary Patterns, Neural Networks, Multimedia Quality, Image Quality, Assessment.

1 Introduction

Video traffic is predicted to form approximately 86 percent of global consumer traffic by 2016 [7]. Every second, 1.2 million minutes of video content will cross the network in 2016. Video traffic is becoming the dominant application over 3G/4G mobile systems. Fuelled by proliferation of smartphones, netbooks and tablets, mobile data traffic is estimated to grow 18 times from 2011-2016, with the surge caused mainly by mobile video services. Within this landscape an

important question for service providers is the quality assurance of the service provided to the consumers of multimedia content.

Video Quality Assessment (VQA) algorithms attempt to automatically assess perceptual degradations introduced by signal processing and transmission operations performed on video sequences. Unfortunately, despite recent advances in video processing and communication technology, their performance leaves something to be desired and there is considerable room for improvement [12]. This is especially true for no-reference methods, which have no notion of the original, non-compressed multimedia content.

To estimate video quality one usually first needs to be able to derive metrics of impairments at the level of single frames. Once the quality is estimated at frame level, diverse methods exist to aggregate this information to create a quality score for the whole video. Overall degradation of the quality of frames(images) is a compound effect of different coding artifacts. Three types of artifacts are typically considered pertinent to DCT block (JPEG, MPEG and H.264) coded data: blocking, ringing and blurring. Blocking appears in all block-based compression techniques due to coarse quantization of frequency components [16]. It can be observed as surface discontinuity (edge) at block boundaries. These edges are perceived as abnormal high-frequency components in the spectrum. Ringing is observed as periodic pseudo edges around original edges [9]. It is due to improper truncation of high frequency components. This artifact is also known as the Gibbs phenomenon or Gibbs effect. In the worst case, the edges can be shifted far away from the original edge locations, observed as false edge. Blurring, which appears as edge smoothness or texture blur, is due to the loss of high frequency components when compared with the original image. Blurring causes the received image to be smoother than the original one [5].

Broadly, the different artifacts can be considered to add texture not present in original content, to the images. However, general-purpose texture descriptors are rarely, if ever, used as basis for deriving the metrics of the level of artifacts introduced into multimedia content. In the study presented here, we explore the applicability of Local Binary Patterns (LBP) to the problem of creating quality metrics. LBPs represent texture descriptors which have been successfully used in a number of computer vision applications. They have, so far as we know, not been considered for no-reference image and video quality assessment.

The goal of each no-reference approach is to create an estimator based on the proposed features that would predict the Mean Opinion Score (MOS)[8] of human observers, without using the original (not-degraded) image or sequence data. In the study presented here, we evaluated the applicability of LBPs as features on which to base no-reference MOS estimation and the Multilayer Perceptron (MLP) [6] neural network as an estimator. To evaluate both the proposed approach images from a public and commonly LIVE Image Quality Assessment Database [13] were used. The experiments conducted show that the proposed approach is able to achieve no-reference image quality assessment beyond that of a widely used state-of-the art approach.

2 Related Work

When objective image and video quality is concerned, most studies focus on full-reference approaches, which assume the availability of pristine, uncompressed original content at the time of assessment. This is a fairly well researched and solved problem. Evaluation of a number of such approaches on the LIVE database is presented in [14].

In a realistic multimedia content delivery over telecommunication network scenario, these algorithms can only be used at server side, or in specific cases where the pristine content has been previously delivered to client side. The problem of no-reference quality assessment is more complex and the need for better solutions still exists.

Wang *et al.*[16] proposed an early no-reference approach to quality assessment in JPEG coded images. Their final measure is derived as a non-linear combination of a blockiness, local activity and a so-called zero-crossing measure. The combination is supposed to provide information regarding both blockiness and blurring (via the two latter measures) in JPEG coded images. Their approach remains to this date the one usually compared against, when no-reference quality assessment is concerned. Therefore, we evaluated the results of the approach proposed here against that of the approach proposed in [16].

More recently Culibrk *et al.* proposed a VQA approach that used different previously proposed artifact and quality metrics as basic features and machine learning algorithms to create a no-reference video quality (MOS) estimator [3]. The study demonstrated the viability of a machine learning approach and the ability to achieve superior quality estimation this way. In fact, one of the algorithms evaluated was MLP. In addition, the authors provided evidence that the basic features used by Wang *et al.* are among best predictors selected from the set of 35 classic measures and saliency-related features evaluated.

LBP_s were originally proposed as a texture descriptor [11]. They have since then been employed as dynamic texture descriptors [18], as well as found a number of successful applications such as face recognition [1] and object detection [15]. To the best of our knowledge it has never been applied to the problem of image and video quality assessment.

3 LBP-VQA Approach

The LBP operator is primarily used as an unifying approach to structural and statistical texture analysis. A detailed explanation of LBP derivation and its extensions is provided in [10]. Equation (1) describes the process of obtaining LBP code for a local area in a image, defined by the location of the central pixel (x_c, y_c), radius from it R and number of pixels that surround it P at the distance of the radius R .

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^P s(g_p - g_c)2^p \quad (1)$$



Fig. 1. Referent image (left) and images compressed with bitrates 0.45313, 0.3263, and 0.15771, respectively (right)

If the location of surrounding pixels (x_p, y_p) , where p denotes the index of neighboring pixel does not match the pixel grid exactly, the intensity of that pixel is obtained through bilinear interpolation. The differences in the values between the central pixel and those surrounding it, as defined by a local area (mask) with parameters R and P , are mapped to zero or one using the function $s(g_p - g_c)$. Where g_c and g_p denote gray value of central pixel and surrounding pixel respectively:

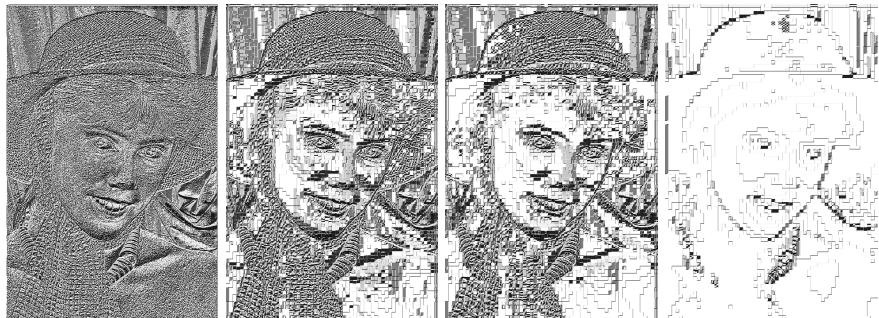
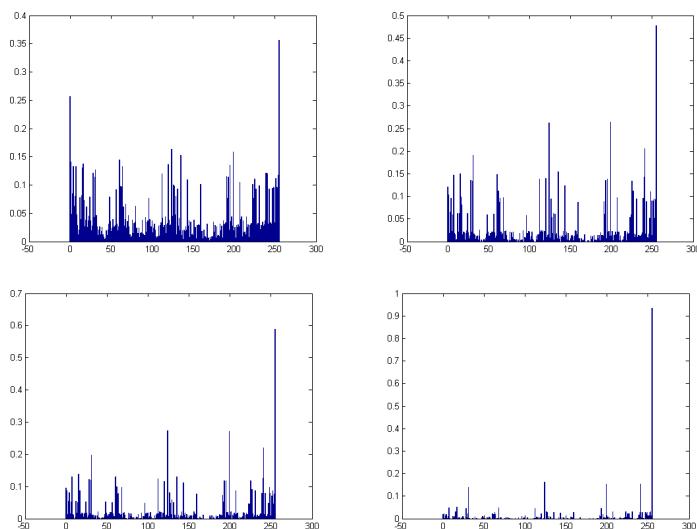
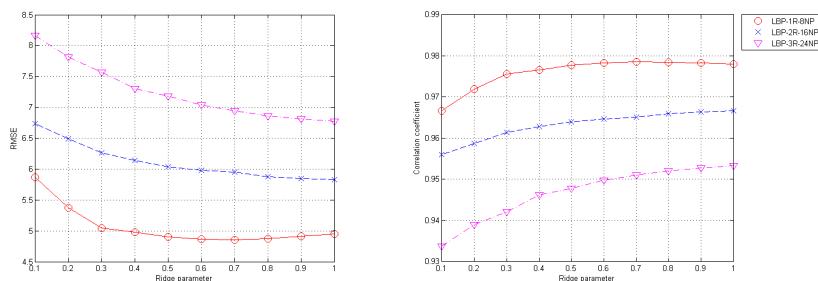
$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (2)$$

The LBP value is obtained as a sum of ones weighted by 2^p , where p corresponds to the index of the pixel in the mask. Thus, the LBP value is a decimal representation of the binary number extracted from the predefined mask. The start of the pixel counting sequence within the mask is chosen arbitrarily the first time, and then must remain unchanged.

Finally, the feature vector is a histogram of values of LBP codes calculated for the whole image. Figure 1 shows referent and images compressed with different bitrates. Figure 2 shows the LBP codes for the same images. As artifacts introduced by JPEG compression become more visible, the distribution of LBP codes changes in such a way that only a few codes remain, representing blockiness and uniform regions. Therefore, histograms of LBP codes seem to be a good representation of image quality, as one can easily distinguish between those pertaining to compressed and non-compressed images (Figure 3). It should be noted that all histograms were normalized using $\sqrt{\|h\|_1}$ norm, where h represents the histogram.

4 Experiments and Results

The evaluation was performed using images from the LIVE Image Quality Assessment Database [13]. The images were compressed using JPEG and form the release

**Fig. 2.** LBP codes for images in Figure 1**Fig. 3.** Histograms of LBP codes for images in Figure 1**Fig. 4.** Correlation coefficients and RMSE values for different MLP ridge parameter values and LBP codes generated for diverse neighborhoods: $R = 1$, $N = 8$ - circles, $R = 2$, $N = 16$ - crosses, and $R = 3$, $N = 24$ - triangles

1 of the database. The database was created from twenty-nine high-resolution 24-bits/pixel RGB color images (typically 768 x 512) using different compression ratios. This yielded 204 images. For all 233 images, subjective MOS is available.

Feature extraction was implemented in C++ using OpenCV library [2]. We evaluated the MLP neural network as a MOS predictor. The network was trained and its performance tested using the machine learning package Weka [17]. The neural network had one hidden layer and was trained by minimizing the root mean squared error plus a quadratic penalty with the BroydenFletcherGoldfarb-Shanno (BFGS) method [4].

To provide for better evaluation we considered different values for the key parameters. The estimator proved to be most sensitive to the value of the ridge parameter, which controls the overfitting. Figure 4 shows how the performance

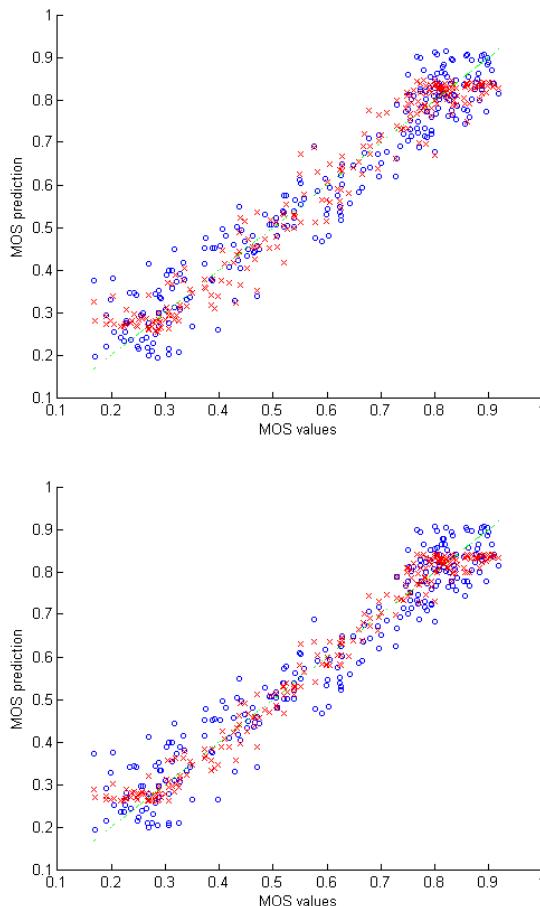


Fig. 5. MOS predictions for the proposed approach (crosses) and Wang and Bovik model (circles), when cross-validated (top) and evaluated on the training set (bottom)

of model changes for different ridge parameter values. We considered three sizes for the neighborhoods used to extract the LBP codes. The results shown were obtained using 10-fold cross-validation and the network that had just 2 neurons in the hidden layer. Due to the complexity of the BFGS model, 2 neurons in the hidden layer are the default in Weka.

As Figure 4 shows, good correlation and low RMSE can be achieved using the proposed approach. For the ridge parameter value of 0.7, correlation (R) of 0.9767 and a RMSE of 0.4890 is achieved between the predicted and subjective MOS. When tested on the train set, slightly better result are obtained, 0.9867 for the correlation coefficient and 0.3740 for RMSE, indicating that the model generalizes well and is stable.

Figure 5 shows the scatter plots of predictions of the proposed approach and the well-known model proposed Wang and Bovik [16]. The scatters show the comparison of model predictions, using both 10-fold cross-validation and when tested on the train set. The evaluation on the training set was included as the final model of Wang and Bovik was fitted using all the images in the dataset, making the evaluation of their model, as conducted here, equivalent to testing on the training set.

As the scatters show, the performance of the proposed approach is better than that reported by Wang *et al.* in [16], as their model achieved an average RMSE of 0.7256. In addition, proposed model correlates with MOS very well.

5 Conclusion

The paper proposes an approach to image and video quality assessment based on LBPs as features and an MLP neural network estimator.

The proposed approach was evaluated on a standard database of JPEG images and achieved superior performance when compared to a classic no-reference image quality assessment methodology.

The study demonstrated the usefulness of LBPs and neural networks for quality assessment. Something that has not been explored before. In the future the study will be extended to evaluate the proposed approach on H.264 videos and to create a blockiness artifact detector which will be able to detect isolated artifacts that appear due to network-introduced errors in multimedia transmission.

Acknowledgment. This research is supported in part by the FP7 project QoSTREAM.

References

1. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(12), 2037–2041 (2006)
2. Bradski, G.: The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000)

3. Culibrk, D., Mirkovic, M., Zlokolica, V., Pokric, M., Crnojevic, V., Kukolj, D.: Salient Motion Features for Video Quality Assessment. *IEEE Trans. on Image Processing* 20, 948–958 (2010)
4. Dennis, J.E., Schnabel, R.B.: Numerical methods for unconstrained optimization and nonlinear equations. Society for Industrial and Applied Mathematics, vol. 16 (1987)
5. Ferzli, R., Karam, L.: A no-reference objective image sharpness metric based on just-noticeable blur and probability summation. In: Proceedings of IEEE 2007 International Conference on Image Processing 3, October 16-19, pp. III–445–III–448 (2007)
6. Haykin, S.: Neural Networks: A Comprehensive Foundation. Macmillan, New York (1994)
7. Index, C.: Forecast and methodology, 2011-2016. White paper, CISCO (December 2012)
8. ITU-R BT.500: Methodology for the Subjective Assessment of the Quality of Television Pictures. Video Quality Experts Group (2002)
9. Kirenko, I.: Reduction of coding artifacts using chrominance and luminance spatial analysis. In: International Conference on Consumer Electronics, ICCE 2006. 2006 Digest of Technical Papers, pp. 209–210 (January 2006)
10. Mäenpää, T.: The Local Binary Pattern Approach to Texture Analysis: Extenxions and Applications. Oulun yliopisto (2003)
11. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(7), 971–987 (2002)
12. Seshadrinathan, K., Bovik, A.C.: An information theoretic video quality metric based on motion models. In: Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics, pp. 25–26. Citeseer (2007)
13. Seshadrinathan, K., Soundararajan, R., Bovik, A., Cormack, L.: Study of subjective and objective quality assessment of video. *IEEE Transactions on Image Processing* 19(6), 1427–1441 (2010)
14. Seshadrinathan, K., Soundararajan, R., Bovik, A., Cormack, L.: A subjective study to evaluate video quality assessment algorithms. In: SPIE Proceedings Human Vision and Electronic Imaging, vol. 7527. Citeseer (2010)
15. Wang, X., Han, T., Yan, S.: An hog-lbp human detector with partial occlusion handling. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 32–39. IEEE (2009)
16. Wang, Z., Sheikh, H.R., Bovik, A.C.: No-reference perceptual quality assessment of jpeg compressed images. In: Proceedings of IEEE 2002 International Conferencing on Image Processing, pp. 477–480 (2002)
17. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
18. Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(6), 915–928 (2007)

Learning Accurate Active Contours

Adas Gelzinis^{1,*}, Antanas Verikas^{1,2},
Marija Bacauskiene¹, and Evaldas Vaiciukynas¹

¹ Department of Electrical & Control Equipment,
Kaunas University of Technology,
Studentu 50, LT-51368, Kaunas, Lithuania
adas.gelzinis@ktu.lt

² CAISR, Halmstad University, Box 823, S-30118 Halmstad, Sweden

Abstract. Focus of research in Active contour models (ACM) area is mainly on development of various energy functions based on physical intuition. In this work, instead of designing a new energy function, we generate a multitude of contour candidates using various values of ACM parameters, assess their quality, and select the most suitable one for an object at hand. A random forest is trained to make contour quality assessments. We demonstrate experimentally superiority of the developed technique over three known algorithms in the *P. minimum* cells detection task solved via segmentation of phytoplankton images.

Keywords: Active contour models, Energy function, Object detection, Image segmentation, Learning, Phytoplankton images.

1 Introduction

Active contour models (ACM) is one of the most successful techniques for object detection in images and have been widely used in image segmentation [1–3], analysis of medical images [4], analysis of protein spots in two-dimensional gel electrophoresis images [5], and many other tasks. ACMs are based on an energy minimization approach and allow incorporating a priori knowledge. To extract the desired object, ACM aims at evolving a curve, under some constraints, to fit the object.

Thus, ACM is a curve $\mathbf{c}(s) = [x(s), y(s)]$, $s \in [0, 1]$ deforming on the image region, to attain desired properties. Curve deformation is achieved by minimizing an energy function, for example:

$$E(\mathbf{c}) = \int_0^1 \left[\frac{1}{2} (\alpha \|\mathbf{c}'(s)\|^2 + \beta \|\mathbf{c}''(s)\|^2) + E_{ext}[\mathbf{c}(s), f(I)] \right] ds \quad (1)$$

where $\mathbf{c}'(s)$ and $\mathbf{c}''(s)$ are the first and the second derivative of $\mathbf{c}(s)$ with respect to s , α and β are parameters, and E_{ext} stands for external energy linking the contour $\mathbf{c}(s)$ to specific features $f(I)$ of the image I . Both static (depending on

* Corresponding author.

the image) and dynamic external forces (depending on the contour) have been used [6]. Static forces can be evaluated using image region intensity and texture information [7]. The following are two typical external energy functionals used to place a contour at edges [1]:

$$E_{ext}(x, y) = -\|\nabla I(x, y)\|^2 \quad (2)$$

$$E_{ext}(x, y) = -\|\nabla[G_\sigma(x, y) * I(x, y)]\|^2 \quad (3)$$

where ∇ is the gradient operator, $G_\sigma(x, y)$ is a two-dimensional Gaussian of standard deviation σ , and $*$ stands for the convolution operation.

Edge-based [1, 8] and region-based [2, 9] models are distinguished, depending on constraints applied. Edge-based models suffer from nearby adjacent objects and strong edges inside an object being searched, since they act as attraction sources for the active contour. Region-based models use statistical information inside and outside the contour and, for images with weak edges, usually perform better than edge-based models. ACM proposed by Chan and Vese is a popular representative (C-V method) of region-based models [2]. The C-V model as well as some other popular ACMs assume constant intensity in various image regions [2]. Numerous techniques have been suggested to combine edge-based and region-based models [10].

The literature analysis indicates that focus of research in ACM area is on development of various energy functions based on physical intuition. Despite the vast number of energy functions used, all techniques experience difficulties in noisy environment. It is hard to succeed in designing a "universal" energy function, capable of dealing with large data variations.

In this work, instead of designing a new energy function, we generate a multitude of contour candidates, assess their quality, and select the most suitable one for an object at hand. Also, aiming to provide the edge-based ACM with rich and diverse information on image edges, we focus on image preprocessing. The energy function used in this work is given by equations (1), (2), and (3).

2 Data

Accurate detection of *Prorocentrum minimum* *P. minimum* cells in phytoplankton images was the main motivation to develop a new contour detection technique. Fig. 1 presents two phytoplankton images with *P. minimum* cells. Images for the analysis were obtained from a simple RGB colour camera of 3264×2448 pixels attached to an inverted microscope with magnification of 400x. Only G image component has been used in this study, since the other two components do not provide much new information for contour detection.

The *P. minimum* species is known to cause harmful blooms in many estuarine and coastal environments. There is a hypothesis that *P. minimum* cells gradually change their shape when adapting to adverse biotic (increased virus pressure) conditions. To study shape changes of *P. minimum* cells, accurate cell detection is required. The length of *P. minimum* cells varies from 14 to 22 μm , while the width range from 12 to 18 μm .

Presence of chlorophyll in phytoplankton cells makes them glow under UV light and this property becomes of immense help for detecting *P. minimum* cells. Two consecutive, with a delay of a few seconds, photos (using light and fluorescence microscopy) were obtained of the same location, see Figure 1.

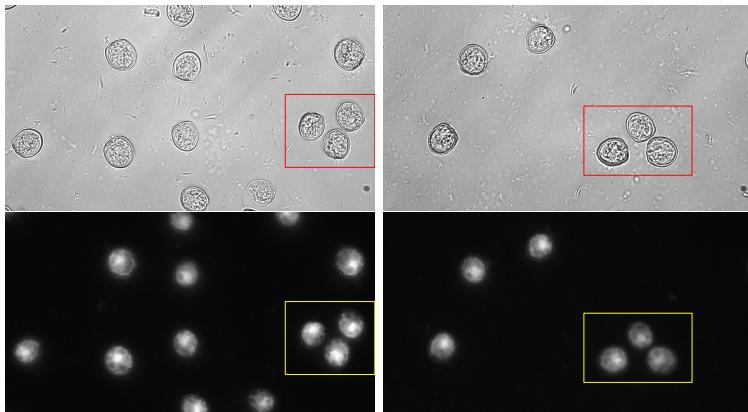


Fig. 1. Two examples of phytoplankton images with *P. minimum* cells (top); phytoplankton fluorescence images of the same location (bottom).

3 Contours of Cells

Images from both the light microscopy and fluorescence microscopy are used to find cells and their center coordinates. Regions of interest are found by simple thresholding [11] of images from the fluorescence microscopy, while the phase congruency-based technique [12] is used to find the center coordinates of objects (cells). Only cells detected in both fluorescence and light microscopy images are used for further analysis.

3.1 Generating a Multitude of Contour Candidates

A 2D shell is a characteristic feature of *P. minimum* cells. We rely on this fact by computing a local intensity gradient in the direction of cell center (see Section 3.2) and use this gradient image to generate a multitude of contour candidates (12 for each cell) by varying the parameters α and β of the ACM, see Eq. 1. A circle with a center point coinciding with the cell center and a radius exceeding the maximal possible cell radius is selected as initial contour. Parameters α and β control the stretch and stiffness of the contour. Optimal values of α and β parameters are different for different cells due to many factors, such as: presence of other nearby objects, clarity of cell contour, focussing accuracy, and others. We believe that the multitude of contour candidates generated using different values of the parameters α and β , contains a contour coinciding well with borders of the cell being studied.

3.2 Local Intensity Gradient in the Direction of Cell Center

To obtain the gradient, a cell image along with surrounding areas is transformed to the polar coordinate system and the local intensity gradient is computed along the direction of radius. The obtained gradient image is then transformed back to the original coordinate system and used by the ACM. Fig. 2 provides some insights into the gradient. Each of the three images shown in Fig. 2 presents results of gradient computation in the direction of a different cell center. The cell center used to compute the gradient is shown by a cross. As can be seen, the cell, in the direction of which center the gradient is computed, manifests itself by a much clear cell boundary if compared to the other cells. It is worth noting rather small influence of the nearby objects on the gradient computation results.

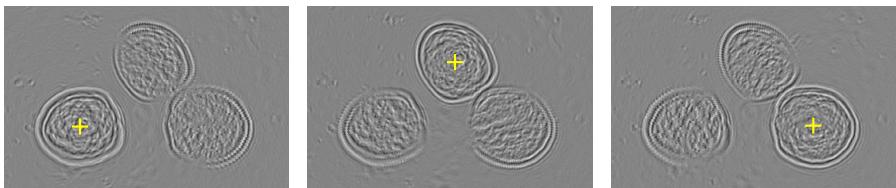


Fig. 2. Images of local intensity gradient in the direction of cell center

4 Assessing Quality of Cell Contours

A random forest (RF) [13] is trained to provide the quality assessment of cell contours. A training set of labelled data (contours) is required to build a random forest. An expert performs the labelling by assigning one of the following five labels to each contour from the training set: a) *unappropriate*, b) *bad*, c) *satisfactory*, d) *good*, and e) *excellent*.

Many factors influence the quality of a generated contour: clarity and complexity of the cell border, the image quality, accuracy of focussing, nearby objects, and cell texture. Therefore, properties of both the contour curve and image region where the curve is placed are considered, to assess the contour quality.

4.1 Features Used to Characterize Contours

The following four groups of features are used to characterize contours of cells: i) *Object geometry*; ii) *Fourier descriptors*; iii) *Contour curvature*; and iv) *Image properties in the vicinity of the contour*.

The object geometry features, Fourier descriptors, and the contour curvature features characterize object and contour curve properties, while features of the fourth type reflect image properties in the vicinity of the contour.

Object Geometry Features

We adopted the following features to characterize object geometry: i) *Area*; ii) *Perimeter*; iii) *MajorAxis*; iv) *Eccentricity*; v) *Circularity*; vi) *Roundness*.

Fourier Descriptors

A K -point digital boundary in the xy -plane can be represented as the sequence of coordinates $c(k) = [x(k), y(k)]$, for $k = 0, 1, \dots, K - 1$. Each coordinate pair can be treated as a complex number so that $c(k) = x(k) + jy(k)$. The discrete Fourier transform of $c(k)$ is

$$a(u) = \frac{1}{K} \sum_{k=0}^{K-1} c(k) e^{-j2\pi uk/K} \quad (4)$$

for $u = 0, 1, \dots, K - 1$. The complex coefficients $a(u)$ are called the Fourier descriptors of the boundary. Instead of using all the Fourier descriptors, only the absolute values of the first $P < K$ coefficients were used.

Curvature Features

For a plane curve given as $c(k) = [x(k), y(k)]$, the curvature is given by

$$\kappa(k) = \frac{|x'y'' - x''y'|}{(x'^2 + y'^2)^{3/2}} \quad (5)$$

where primes refer to derivatives with respect to k .

Based on the curvature assessment at each contour point, we compute four curvature related features: (1) mean(κ); (2) max(κ); (3) std(κ); and (4) upper quartile value of κ .

Image Features

For each contour $c(k)$ point $k = 1, 2, \dots, K$, one interior $c_{int}(k)$ and one exterior $c_{ext}(k)$ point are chosen perpendicular to the contour at a distance δ . Various image intensity characteristics on both sides of the contour are computed based on positions of $c_{int}(k)$ and $c_{ext}(k)$. Average image gradient as well as average image gradient in the vicinity of the contour are also used as image features: (1) standard deviation of the intensity difference between the interior and exterior points: $\text{std}(I_{c_{int}} - I_{c_{ext}})$; (2-3) mean and standard deviation of intensity of the interior and exterior points: $\text{mean}(I_{c_{int}})$, $\text{std}(I_{c_{int}})$; (4-5) mean intensity and standard deviation of contour points: $\text{mean}(I_c)$, $\text{std}(I_c)$; (6-7) intensity mean and standard deviation of the exterior points: $\text{mean}(I_{c_{ext}})$, $\text{std}(I_{c_{ext}})$; (8) average gradient magnitude: $\text{mean}\|\nabla I\|$; (9) average gradient magnitude in the vicinity of the contour: $\text{mean}(\|\nabla I_{c_{int}, c, c_{ext}}\|)$.

4.2 Assessing Contour Quality by Experts – Labelling Contours

Twelve contours were generated for each cell using different parameters of the ACM algorithm. Different background, varying clarity of cell borders, other objects inside and outside cells are the factors strongly affecting results of automatic contour detection. It is a rather hard task for the experts to decide where

exactly the actual cell contour should be drawn. To draw exact contours for a large number of cells is a very tedious task. Therefore, instead of drawing exact cell contours, three experts were asked to assess quality of automatically generated contours. We used five contour quality grades: *Unappropriate* (1), *Bad* (2), *Satisfactory* (3), *Good* (4), and *Excellent* (5).

Fig. 3 presents four examples of contours of different quality. It is worth noting that, for some cells, amongst 12 contours generated by the algorithm for one cell, non the contours was deemed by the experts as being *excellent*, *good* or *satisfactory*, meaning that all 12 contours were deemed as being *bad* or *unappropriate*. Table 1 presents statistics of contour assessment, where the "All contours" columns show the percentage of all contours in different categories, as deemed by the experts. The rightmost column shows how the best contours, only one (best) contour for each cell, were assessed by the experts.

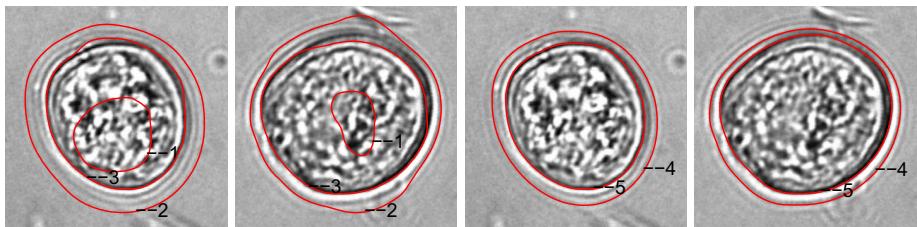


Fig. 3. Contours graded as *Unappropriate*, *Bad* and *Satisfactory* on the *left*, and *Good* and *Excellent* on the *right*

Table 1. Statistics of contour assessment by experts

| Grade | All contours | | Best contours | |
|---------------|---------------------|------|----------------------|------|
| | #N | % | # N | % |
| Excellent | 710 | 22.9 | 200 | 77.5 |
| Good | 324 | 10.5 | 35 | 13.6 |
| Satisfactory | 487 | 15.7 | 21 | 8.1 |
| Bad | 358 | 11.6 | 2 | 0.8 |
| Unappropriate | 1217 | 39.3 | 0 | 0.0 |

5 Experimental Investigations

The following issues, which are of key importance for the success of the algorithm, have been studied experimentally: i) accuracy of automatic contour quality assessment; ii) quality of selected ("best") contour, ability to detect a selected contour of bad quality; iii) optimal values of ACM parameter. The study confirms that to generate a contour of good quality, different ACM parameters are needed for different cells even in images of the same type; iv) comparison of the proposed contour detection technique with other methods.

5.1 Accuracy of Automatic Contour Quality Assessment

For each of 258 cells used in the experiments, 12 contours were generated using different values of α and β parameters, see Eq.(1). In this experiment, we set $\alpha = \beta$ and varied their values in the interval [0.02 0.25]. Quality of all generated contours was assessed by three experts, giving grades from 1 to 5, as it has been discussed in Section 4.2. These contour quality grades together with equally weighted contour features were used to train RF for automatic contour quality assessment.

To evaluate accuracy of the automatic contour quality assessment, the out-of-bag set of contours was used. The average mean squared error of contour quality evaluations given by the experts and the model was equal to $MSE = 0.16$. When quality prediction values were rounded off to the five discrete grades, discrepancy between grades given by the experts and the model were observed for 17.9% of contours. Bearing in mind the fact that the experts often disagreed when giving grades *Good*, *Satisfactory* and *Bad*, the obtained results are very encouraging.

5.2 Quality of Selected Contours

The following are important issues to consider when evaluating robustness of the developed model for automatic contour quality assessment. Is the automatically selected contour the best amongst all generated? How often the automatically selected contour is the best amongst all generated? To address these issues, an automatic assessment was deemed to be erroneous, if an automatically selected “best” contour was different from the one that obtained a highest grade from the experts for that particular cell. There were only 13 such cells out of 258 available (about 5.0% of errors).

The experiment was repeated using new contours generated by varying α and β . To increase the variety of new unseen contours, 36 contours were generated for each cell. To assess quality of the contours, RF trained in the previous experiment was used. Since the number of contours used in this experiment was rather large, the experts assessed only if the RF failed to select the best available contour for each cell. There were 7 such errors, out of 258 possible, i.e., for 7 cells, a contour selected by the RF, was not the best according to the experts.

Table 2 presents quality of best cell contours according the experts. In this table, the contour categories *Bad* and *Unappropriate* were combined into one, *Unappropriate*. The results presented in Table 2 slightly differ from those given in Table 1. The discrepancy is due to the fact that, the best contour for each cell was selected from a larger set in this experiment (36 instead of 12). Moreover, since in this experiment the experts were asked to select and assess only one, best, contour for each cell, the assessments happened to be more rigorous compared to the assessments used to train the RF. As can be seen from Table 2, more than 92% of all contours are of excellent and good quality.

It is important to detect contours of bad quality and eliminate them from further analysis. The experiments have shown that most of automatically selected contours deemed by the experts as being *unappropriate* or *satisfactory*,

Table 2. Assessment of the best cell contours, one for each cell

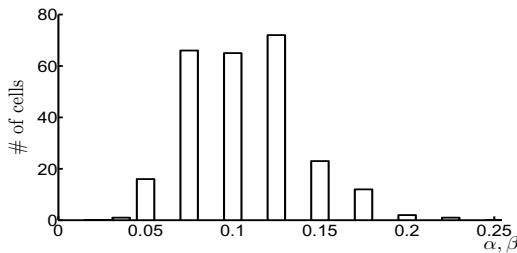
| Grade | # N | % |
|---------------|-----|------|
| Excellent | 198 | 76.8 |
| Good | 40 | 15.5 |
| Satisfactory | 13 | 5.0 |
| Unappropriate | 7 | 2.7 |

obtained low grades from the automatic evaluation too. To eliminate contours of low quality, an experimentally selected threshold value equal to 3.2 was applied. The thresholding eliminated 9% of lowest quality contours. To this group were automatically attributed all contours deemed by the experts as being of *unappropriate* or *satisfactory* quality. Only one contour of *satisfactory* quality according to the experts obtained rather high quality value, equal to 4.5, from the RF. Four contours of *good* quality according to the experts were automatically assigned to the group of bad quality contours.

The experimental investigations show that the best contour is selected from the set of contours generated for one cell with 95% accuracy. Detection accuracy of contours of satisfactory and bad quality exceeds 98% (5 errors). After the automatic elimination of bad quality contours (about 9%), only one contour of *satisfactory* quality was found in the group of good quality contours.

5.3 Optimal Values of ACM Parameters

For a given cell, ACM parameter values leading to contours of the highest grade are called optimal. In this experiment, the assessment was done automatically using the trained RF. Fig. 4 displays the histogram of α and β values (in this case, $\alpha = \beta$) leading to the best cell contours. Observe that values of the ACM parameters were varied in a broad range: from values hindering the ACM from placing a contour sufficiently close to an actual cell contour to values placing the contour inside the cell. The distribution confirms that optimal α and β vary in a broad range, and different ACM parameter values are optimal for different cells.

**Fig. 4.** The histogram of α and β values leading to the best cell contours

5.4 Comparison with Other Methods

The technique was compared with the C-V method [2], and two recently developed algorithms [3, 14]. All the algorithms have several adjustable parameters. Values of the parameters were carefully selected separately for each algorithm.

In total, 49 phytoplankton images containing 258 *P. minimum* cells were processed to test the algorithms. The segmentation results were evaluated by the experts and are summarized in Table 3. Observe that in this experiment, only contours assigned to the *Excellent* and *Good* groups were considered as being correct. As can be seen from Table 3, the proposed technique provided the best performance. Segmentation by the algorithms used for the comparison often resulted either in over-segmentation or under-segmentation (leakages of contour and inability to separate several nearby cells).

Table 3. Summary of cell contour detection results

| Algorithm | Correct | Accuracy, % |
|-----------------|---------|-------------|
| Bernard [3] | 42 | 16.28 |
| Chan & Vese [2] | 52 | 20.16 |
| Shi [14] | 152 | 58.91 |
| Proposed | 238 | 92.25 |

6 Discussion and Conclusions

Rather than focusing on constructing an energy function based on physical intuition, as many recent studies of ACMs do, we advocate the idea of generating a multitude of contour candidates, assessing their quality, and selecting the most suitable one for an object at hand. We experimentally demonstrate the superiority of the developed approach over three well known ACM algorithms applied to cell segmentation in the microscopy imagery task. Sensitivity to nearby objects and strong edges inside a target object is highly reduced and segmentation becomes more noise-robust using the proposed approach.

The experimental investigations have shown that in 92.25% of cases, contours of *P. minimum* cells were accurately determined. For each cell tested, the proposed algorithm was able to select the best contour from a set of generated in 95% of cases. In the remaining 5% of cases, contours selected by the algorithm were only slightly worse than the best possible. Cases where none of generated contours was appropriate for a given cell were automatically detected with higher than 98% accuracy. By setting an appropriate, experimentally selected, threshold value on predicted contour quality, 9% of contours were categorized as being of unappropriate quality and eliminated from further analysis. Amongst the remaining contours only one contour attributed by the experts to the *Satisfactory* group, was assigned to the *Good* group by the algorithm. The investigations have shown that the developed technique provides contour quality assessment with high accuracy and can be used for automatic contour quality evaluation and contour selection.

Acknowledgement. This research was funded by a grant (No. LEK-09/2012) from the Research Council of Lithuania.

References

1. Kass, M., Witkin, A.P., Terzopoulos, D.: Snakes: Active contour models. *International Journal of Computer Vision* 1(4), 321–331 (1988)
2. Chan, T., Vese, L.: Active contours without edges. *IEEE Transactions Image Processing* 10(2), 266–277 (2001)
3. Bernard, O., Fribois, D., Thevenaz, P., Unser, M.: Variational B-spline level-set: A linear filtering approach for fast deformable model evolution. *IEEE Transactions on Image Processing* 18(6), 1179–1191 (2009)
4. Shang, Y., Yang, X., Zhu, L., Deklerck, R., Nyssen, E.: Region competition based active contour for medical object extraction. *Computerized Medical Imaging and Graphics* 32(2), 109–117 (2008)
5. Savelonas, M.A., Mylona, E.A., Maroulis, D.: Unsupervised 2D gel electrophoresis image segmentation based on active contours. *Pattern Recognition* 45(2), 720–731 (2012)
6. Veronese, E., Stramare, R., Campion, A., Raffeiner, B., Beltrame, V., Scagliori, E., Coran, A., Ciprian, L., Fiocco, U., Grisan, E.: Improved detection of synovial boundaries in ultrasound examination by using a cascade of active-contours. *Medical Engineering & Physics* (2012), doi.org/10.1016/j.medengphy.2012.04.014
7. Chakraborty, A., Staib, L., Duncan, J.: Deformable boundary finding in medical images by integrating gradient and region information. *IEEE Transactions Medical Imaging* 15(6), 859–870 (1996)
8. Caselles, V., Kimbel, R., Sapiro, G.: Geodesic active contours. *International Journal of Computer Vision* 22(1), 61–79 (1997)
9. Ronfard, R.: Region-based strategies for active contour models. *International Journal of Computer Vision* 13(2), 229–251 (1994)
10. Tao, W., Tai, X.C.: Multiple piecewise constant with geodesic active contours (MPC-GAC) framework for interactive image segmentation using graph cut optimization. *Image and Vision Computing* 29, 499–508 (2011)
11. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Trans Systems Man & Cybernetics* 9(1), 62–66 (1979)
12. Verikas, A., Gelzinis, A., Bacauskiene, M.: Phase congruency-based detection of circular objects applied to analysis of phytoplankton images. *Pattern Recognition* 45(4), 1659–1670 (2012)
13. Breiman, L.: Random forests. *Machine Learning* 45, 5–32 (2001)
14. Shi, Y., Karl, W.C.: A real-time algorithm for the approximation of level-set-based curve evolution. *IEEE Transactions on Image Processing* 17(5), 645–656 (2008)

Pattern Recognition in Thermal Images of Plants Pine Using Artificial Neural Networks

Adimara Bentivoglio Colturato¹, André Benjamin Gomes¹, Daniel Fernando Pigatto¹,
Danielle Bentivoglio Colturato², Alex Sandro Roschmidt Pinto⁴,
Luiz Henrique Castelo Branco⁵, Edson Luiz Furtado³,
and Kalinka Regina Lucas Jaquie Castelo Branco¹

¹ Institute of Mathematics and Computing Sciences (ICMC),
University of São Paulo (USP), São Carlos, São Paulo, Brazil

² Universidade Paulista (UNIP), Araraquara, São Paulo, Brazil

³ Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP),
Botucatu, São Paulo, Brazil

⁴ Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP),
São José do Rio Preto, São Paulo, Brazil

⁵ Instituto Federal de Ciência e Tecnologia de São Paulo, Araraquara, São Paulo, Brazil
{adimara, andregomesb, dbcolturato}@gmail.com,
{pigatto, kalinka}@icmc.usp.br, arpinto@ibilce.unesp.br,
elfurtado@fca.unesp.br, luiz.branco@ifsp.edu.br}

Abstract. Pine is used primarily as a source of raw materials for the industries of lumber and laminated plates, resin, pulp and paper. Pine may be affected, from the nursery to adults, in plantations by pathogens such as fungi and/or pests. The aim of this work was to recognize patterns in images obtained from a thermal plants camera in pine. An Unmanned Aerial Vehicle with a thermal camera embedded was used to take video images of pine trees. The video was segmented in pictures and all the pictures were standardized to the same size 240 x 350px. The images were segmented and a two-layer neural network feed-forward and the Scaled Conjugate Gradient (SCG) algorithm were used. The results proved to be satisfactory, with most errors near zero.

Keywords: Artificial neural networks, thermal images, Pine tree and UAVs.

1 Introduction

The genus *Pinus* can suffer various types of damage, such as pathogenic agents that cause root rot (*Armillaria*, *Cylindrocladium*, *Fusarium*), burning needles (*Cylindrocladium*, *Mycosphaerella*, *Cercospora pini-densiflorae*) dry hands (*Mycosphaerella pini*), blemishes of needles (*Davisonomyces* and *Lophodermium*) and death of trees caused by *Hendersonula*, and by pests like aphids (*Cinara* spp.), Beetles (*Migdolus fryanus*), rodents and other insects [1, 2, 14, 18]. These kind of pests and damages might weaken their growth and development, as well as causing direct damage on wood quality.

Methods for detection of damage in plantations may be done through interpretation of the sound produced by percussion of the trunk with a hammer; visual assessment of the tissue collected with an auger; insertion of measuring probes in the tissue and use of radioisotopes, radiographic systems, or radar systems which takes time and requests a physical contact with the area to be investigated. In order to improve the detection and save time, remote sensing coupled with field data is presented as an important tool.

The thermal remote sensing is based on estimates of the land surface radiometric temperature, determined by the radiation intensity emitted by Earth's surface, in the range of 8-14 microns, obtained via satellite or airborne sensors [3].

The early work relative to the relationship between water temperature in soil and plant physiological factors, atmospheric and plant response, were from the 60s, with the knowledge of thermal remote sensing as applied in several studies: evaluation of evapotranspiration, humidity ground control, irrigation, agricultural productivity, water stress and damage detection studies on tree trunks [6, 8].

The images obtained from the infrared camera allow early detection of various types of changes in trees including bark necrosis, decomposition and early growth response to damage or mechanical stress. It has the advantage of being non-invasive, possesses rapid use and provides information in real time and ability to work at a distance of up to 25 m. In order to assess the trees, however, all surfaces should be out of direct sunlight, and there must be no rain or damp wood. The images do not distinguish between different types of changes, but usually can be correctly interpreted. The technique does not allow a quantitative assessment of the extent of decaying wood, but seems to be accurate enough to identify trees that deserve control measures or a more accurate assessment [6].

In agriculture the pattern recognition can be done using some approaches as statistical methods, analysis features and artificial neural networks [9]. Artificial Neural Networks (ANN) are Artificial Intelligence tools that have the ability to adapt and learn how to perform a certain task, or behavior, from a set of sample data. It is used for recognizing and classifying patterns, and for imaging, due to its characteristics such as strength, generalization, parallelism and noise tolerance [13].

A neural network consists of one or several interlinked basic processing units arranged in one or more layers. Each layer may contain various processing elements. The connection between the processing units is done via the synaptic weight, which defines the influence among the processing units interconnected [5].

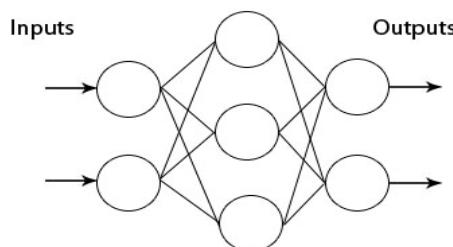


Fig. 1. Representation of Multilayer Perceptron Neural network (MLP)

Multilayer perceptron are networks in which the neurons are arranged in multiple layers and can represent functions not linearly separable (see Fig. 1). Every limited piecewise continuous function can be approximated with an error arbitrarily small by a network with one hidden layer [12].

In the literature, there are applications using artificial neural networks for extracting attributes of an image with good results by using color segmentation, which consists in subdividing an image into its basic components with the most relevant characteristics and these characteristics depend on the object of interest [15, 16].

The Scaled Conjugate Gradient (SCG) is a training algorithm based on optimization technique, which is called conjugate gradient method. Unlike other backpropagation algorithms, this one does not require any parameters supplied by the user, and its implementation has a low computational cost [10].

Devices with specific functions, such as those present in a microwave oven, are called embedded systems. There are several definitions of embedded systems: Netrino [11] defines them as a combination of hardware and software, and maybe a few mechanical parts, designed to perform a specific function. In some cases, they are part of a bigger system or a product.

These systems can be found in many everyday situations. Vahid and Givargis [17] have listed some applications including portable electronics (cellphones, pagers, digital cameras, calculators and PDAs), appliances (microwave ovens, thermostats, security systems, washing machines and lighting systems), office automation (fax machines, printers and scanners), business equipment (cash registers, alarm systems and card readers) and automobiles (electronic fuel injection, antilock brakes and active suspension).

Examples of critical embedded systems are UGVs (Unmanned ground vehicles), which are devices used mainly in military, environmental and agricultural applications to automate processes.

In most cases, general-purpose programmers do not concern about the performance of their applications since they can use available resources without limitations and need their programs to run "fast enough", not "as fast as possible".

In embedded systems, on the other hand, performance is a clear goal of every developer, i.e., developed programs must consider specific short response times. At the heart of embedded computing there is real-time computing, which is the science and art of programming respecting specific times [19]. The program receives input data and has a deadline for the completion of the necessary computational operations. If the program does not produce the required result on time, it is not useful, even if the eventual output produced is correct.

Embedded systems are such systems that have a dedicated and customized processing among software and hardware, usually geared toward a specific goal, thus improving the investment, lodging, performance and power consumption [20]. Unmanned aerial vehicles (UAV) are examples of critical embedded systems that can be used for many applications, including in agricultural monitoring. For instance, a drone is a well-employed type of UAV which can be used for capturing images of plantations, pesticides spraying, counting trees etc. Moreover, they can also fly lower,

collect imagery at a much higher resolution, monitor highly dynamic vegetation and re-visit the area many times, unlike the conventional platforms such as satellites and manned aircrafts [4].

2 Material and Methods

2.1 Location

The survey was conducted in an area planted with Pinus located in Campus 2 of USP (University of São Paulo) in São Carlos. A drone was used with an embedded FLIR TAU thermal camera (with lens 13"). Video images of plants were also captured.

2.2 Neural Networks

The video was segmented in pictures and the coding was done from colored images (RGB) with dimensions of 240 x 350 pixels, where in each 10x10 segment a predominant color was selected. In order to simplify the images, we used the segmentation process and chose to leave the three colors in the image with the same tone, then converting the pixel values into the predominant color in order to facilitate data input (Fig. 2). Moreover, the pixels of the image backgrounds were grouped and converted to the predominant color in that region, with the aim of reducing the number of inputs of the neural network. The RGB image was encoded, so each pixel contains a integer value between 0 and 2. These values represent blue, green and red, respectively.

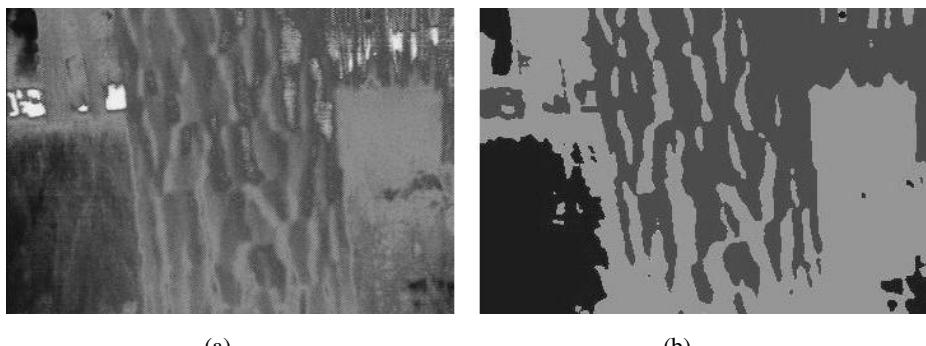


Fig. 2. Demonstration of segmentation process, where (a) represents a normal picture and (b) a segmented image

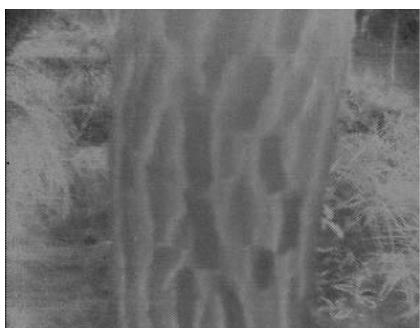
The neural network is a network of two-layer feed-forward, having as input an image with a total of 840 pixels, a hidden layer of 1500 neurons and a binary output indicating whether the image has a sick tree (0) or a healthy tree (1). The algorithm used was the Scaled Conjugate Gradient (SCG) using the method of Levenberg-Marquardt algorithm to find the global minimum of the function [10].

3 Results and Discussion

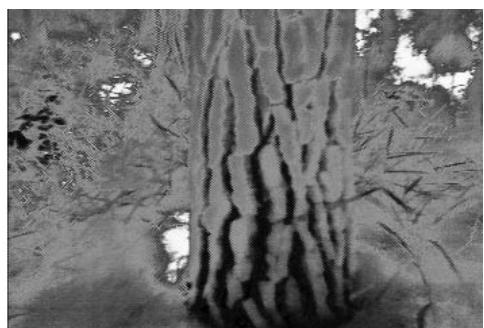
The application of a thermal camera enables the observation of trees with apparent damage on the trunk of wood: Fig. 3 shows a plant with black streaks and apparently healthy when analyzed by naked eye; but when we used the thermal camera it was possible to see that the plant have had a dysfunction in the translocation of sap. It helps us to conclude that these plants suffered some kind of damage that was affecting the plant inside.

The results proved to be satisfactory with most errors near zero. The neural network had 840 neurons inputs representing each 10x10 square values between 0 and 2, 1500 neurons in one hidden layer and 1 output neuron stating that the tree is healthy (1) or damaged (0). The base is comprised of 660 samples, 70% of the data for training, 15% for validation and 15% for testing. The best result was a total error of 7.4% (Tab.1). The training, validation and the test showed error rate of 3.67%, 15.15% and 17.17%, respectively (Tab. 2).

Regarding the training set, testing and validation, the network hit a total of 92.6% (Tab. 1), and the Mean Square Error (MSE) and the percentage of error can be seen in Tab. 2.



(a)



(b)

Fig. 3. A healthy plant (a), and a plant with interior damage, invisible by naked eye, indicated by the black color (b)

The use of a thermal camera proved to be an effective and fast method for the detection of trunk cavities and damages even in early stage while it still shows no visible signs, which enables preventive measures before the damage becomes greater, affecting the quality of wood. It is a method that allows a diagnosis without the risk of damaging the plant, agreeing with Catena [8], which used a thermal camera to detect cavities in the trunk of urban trees.

These recorded images can be used to survey the damage evolution over time, if the damage was caused by a pathogen. And this neural network will help identify diseased and healthy plants, providing faster and more accurate results.

Table 1. Percentage of correct answers in the training set, validation and testing

| | | Actual class | | |
|-----------------|---|--------------|--------|--------|
| | | 0 | 1 | Total |
| Predicted class | 0 | 282 | 35 | 89,00% |
| | 1 | 14 | 329 | 95,90% |
| Total | | 95,30% | 90,40% | 92,60% |

Table 2. MSE and percentage error

| | MSE | E% |
|------------|------------|-------------|
| Training | 4.44482e-2 | 3.67965e-0 |
| Validation | 1.16879e-1 | 15.15151e-0 |
| Testing | 1.18542e-1 | 17.17171e-0 |

4 Conclusion

Areas of eucalyptus plantations occupy large areas and are targets for several diseases that attack the wood causing economic damages. Techniques for identification and monitoring of pests and diseases are needed to maximize production.

This paper proposed the use of a FLIR TAU thermal camera embedded on a drone to capture images of an entire planting area. The use of critical embedded systems such as drones for image processing in hard to reach areas enables identification of damage due to the flexibility of the aircraft when compared to ground methods of acquiring images.

Equipped with a thermal camera, drones proved to be a good alternative to obtain aerial images for remote sensor tasks, mainly in pine diseases. UAVs with cameras provide several tools for aerial image acquisition, including mission planning, automatic image acquisition and geo-referencing of photographic and video images. The use of drones was evaluated and proved to be a possible way to collect videos and provide a good quality images to the neural network.

The images acquired by the drone with a thermal camera handled by a MLP network identified the damaged parts of the trunk, even when not visible by naked eye. In addition, the agility of acquisition of these images by air and the good accuracy of the network with most errors near zero should facilitate future work in image processing in real time applications, because this work is part of a major project which will need online image analyzing.

Acknowledgments. The authors acknowledge the support granted by CAPES, CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology - safety-critical Embedded Systems - Brazil), processes 573963/2008-9 and 08/57870-9 and FAPESP process number 2012/08498-5.

References

1. Andreiv, J.: Danos causados por roedores em povoamentos de pinus e técnicas de redução de danos. Dissertação de mestrado. Universidade Federal do Paraná, 74 p. (2002), <http://dspace.c3sl.ufpr.br/dspace/bitstream/handle/1884/25950/D%20-%20ANDREIV,%20JUARES.pdf?sequence=1>
2. Auer, C.G., Gricoletti Jr., A., Santos, A.F.: Doenças de pinus: identificação e controle. Embrapa Florestas, Colombo (2001), <http://www.infoteca.cnptia.embrapa.br/bitstream/doc/289928/1/circtec48.pdf>
3. Bernardi, A.C.: Sensoriamento remoto no termal e infravermelho próximo no estudo de depósitos de turfa no vale do rio Parafba do Sul. Dissertação, Instituto Nacional de Pesquisas Espaciais, 122 p. (1986)
4. Berni, J.A.J., Zarco-Tejada, P.J., Suarez, L., Fereres, E.: Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle. IEE Transactions on Geoscience and Remote Sensing 47(3), 722–738 (2009)
5. Biondo, L.N., Pacheco, M.A.C., Vellasco, M.M.B.R., Passos, E.P.L., Chiganer, L.: Sistema Híbrido de Apoio à Decisão para Detecção e Diagnóstico de Falhas em Redes Elétricas. In: Anais do III Simpósio Brasileiro de Redes Neurais, Recife, pp. 197–204 (1996)
6. Catena, A., Catena, G.: Overview of thermal imaging for tree assessment. Arboricultural Journal 30, 259–270 (2008)
7. Catena, A., Catena, G.: Use of a hand-held thermal imager to detect cavities and rotten tissue in trees. Geomatics, earth observation and the information society, pp. 260–267 (2001)
8. Epiphanio, J.C.N.: Sensoriamento remoto termal para avaliação de produtividade e deficiência hídrica de milho na região dos cerrados, Dissertação, 123 p. Instituto Nacional de pesquisas espaciais, São José do Campo/SP (1983)
9. Kondo, N., Ahmad, U., Monta, M., Murase, H.: Machine vision based quality evaluation of Iokan orange fruit using neural networks. Computers and Electronics in Agriculture 29 (2000)
10. Moller, M.F.: A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6(4), 525–533 (1993)
11. Netrino. Embedded Systems Glossary (2011), <http://www.netrino.com/Embedded-Systems/Glossary-E>
12. Oliveira, M.A.A.: Desenvolvimento de um medidor de vazão termal inteligente. Dissertação. Universidade do Estado do Rio de Janeiro, 113 p. (2010)
13. Osorio, F., Bittencourt, J.R.: Sistemas inteligentes baseados em Redes Neurais Artificiais aplicados ao processamento de imagens. In: I Workshop de Inteligência Artificial, Universidade de Santa Cruz do Sul, p. 30 (2000)
14. Rodigheri, H.R., Lede, E.T.: Avaliação ambiental, econômica e social dos danos causados pelos pulgões-gigantes-do-pinus *Cinara* spp. em plantios de *Pinus* no Sul do Brasil. Embrapa Florestas, Comunicado Técnico n. 110, Colombo (2004), http://www.cnptia.embrapa.br/publica/comunitec/edicoes/com_tec110.pdf

15. Simões, A.S.: Segmentação de imagens por classificação de cores: uma abordagem neural. Dissertação de mestrado em engenharia, Escola Politécnica da Universidade de São Paulo, São Paulo, SP (2000)
16. Simões, A.S., Reali Costa, A.H.: Using neural color classification in robotic soccer domain. In: Barros, L.N., Cesar Jr., R.M., Cozman, F.G., Reali Costa, A.H. (eds.) Proceedings of International Joint Conference IBERAMIA-SBIA- Workshop Meeting on Multi-Agent Collaborative and Adversarial Perception, Planning, Execution, and Learning, Atibaia, SP, pp. 208–213 (2000)
17. Vahid, F., Givargis, T.: Embedded System Design: A Unified Hardware/Software Introduction. John Wiley & Sons, United States of America (2002)
18. Wilcken, C.F., Orlato, C., Ottati, A.L.T.: Ocorrência de *Migdolus fryanus* (Coleoptera: Cerambycidae) em plantios de *Pinus caribaea* var. *hondurensis*, Revista Árvore, Viçosa 29(1), 171–173 (2005)
19. Wolf, W.: Computers as Components: Principles of Embedded Computing System Design. Morgan Kaufmann Publishers, Burlington (2008)
20. Yu, Y., et al.: The Practice and exploration on the education mode for embedded systems major. In: 2010 International Conference on Education and Management Technology, pp. 367–370 (2010)

Direct Multi-label Linear Discriminant Analysis

Maria Oikonomou and Anastasios Tefas

Aristotle University of Thessaloniki, Department of Informatics,
Box 451, 54124 Thessaloniki, Greece
tefas@aiai.csd.auth.gr

Abstract. Multi-label problems arise in different domains such as digital media analysis and description, text categorization, multi-topic web page categorization, image and video annotation etc. Such a situation arises when the data are associated with multiple labels simultaneously. Similar to single label problems, multi label problems also suffer from high dimensionality as multi label data often happens to have large number of features. In this paper, the Direct Multi-label Linear Discriminant Analysis method is proposed for dimensionality reduction of multilabel data. In particular we extend Multi-label Discriminant Analysis (MLDA) and modify the between-class scatter matrix in order to improve classification accuracy. The problem that Direct MLDA overcomes is the limitation of the produced projections that in MLDA are defined as $K - 1$ for a K class problem. Experimental results on video based human activity recognition for digital media analysis and description as well as on other challenging problems indicate the superiority of the proposed method.

Keywords: Direct Multi-label Discriminant Analysis, Dimensionality Reduction, Multi-label classification, Activity recognition, Video analysis.

1 Introduction

In recent years, with the development of the internet and the technologies used for media production, the amount of available information has been increased dramatically. The main problem of the huge amount of information is how the user can interpret the content of the information and how he can retrieve successfully the information he is interested in. Moreover, big data problems arise in several steps of the media production chain. Pictures, videos, texts, music can contain information on various topics. The user can not be sure that the correct information will be retrieved. This is the reason why it is necessary to classify the information into categories. The most popular way of classification is label classification. The label is attached to the item, indicating the category that the item belongs to. In label classification there are two categories, single-label classification and multi-label classification. The first refers to problems when each instance is associated with one label and the second when each instance is associated with multiple different categories. In most cases, items belong to the second category. For example in music information retrieval, a song could

belong to categories such as *piano*, *classical music* and *Mozart*. The interest of researchers on multi-label learning has increased due to the large number of applications, multi label data are associated with. Multi-label learning algorithms are examined in [1], [7], multilabel semantic image annotation methods are presented in [2], [3], [5]. In [22] music is categorized into emotions. Gene and protein function prediction are proposed in [4].

The *curse of dimensionality* often causes serious problems when high dimensional data are used for learning, and thus a lot of dimensionality reduction methods have been developed. Depending on whether the label information is used, those methods can be classified into two categories, *supervised* and *unsupervised*. In *unsupervised* learning, label information is not provided. There are two options available in order to reduce the dimensionality of multilabel data.

In the first approach, any *unsupervised* dimensionality reduction method built for single label problems can be used. A representative of *unsupervised* dimensionality reduction methods is *Principal Component Analysis (PCA)* that tries to find those projections that maximize the variance among data. *Random Projection* [17] is another method that projects the data on a random lower-dimensional orthogonal subspace that captures as much of the variation of the data as possible. *Latent semantic indexing (LSI)* [18] can also be used directly in multilabel data. *LSI* is widely applied to documents analysis and information retrieval. To apply *LSI*, documents are represented in a vector space model, and *Singular Value Decomposition (SVD)* is performed to find the sub-eigenspace with large eigenvalues. *Partial least squares (PLS)* [12] can also be applied directly to multi-label data by ignoring label correlation.

The second approach uses the provided labels in order to find a projection that enhances discriminability between labels. *Multi-Label Dimensionality Reduction via Dependence Maximization (MDDM)* [14] projects the original data into a lower-dimensional feature space maximizing the dependence between the original feature description and the associated class labels. *Multi-label latent semantic indexing (MLSI)* [15] is an extend of *LSI* so that it can properly manage multilabel data. A representative of supervised dimensionality reduction method is *Linear Discriminant Analysis (LDA)*, which aims at identifying a lower-dimensional space minimizing the inter-class similarity while maximizing the intra-class similarity. *LDA* cannot be directly applied to multilabel data, thus an extend of *LDA* is proposed in [13], named *Multi label Discriminant Analysis MLDA* that takes advantage of label correlation between label sets of multi label data.

Multilabel learning considers both *multi-label classification (MLC)* and *label ranking (LR)*. *MLC* is concerned with learning a model that outputs a bipartition of the set of labels into relevant and irrelevant with respect to a query instance. *LR* extends conventional multiclass classification in the sense that it gives an ordering of all class labels. We can group the existing methods for multi-label learning problems into two main categories a) *transformation methods* and b) *adaption methods* [7]. The first approach transforms the multilabel classification problem into one or more single label classification, regression or ranking task.

The second approach extends specific learning algorithms in order to handle multilabel data directly. The most popular approach as a transformation method is *Binary Relevance (BR)* [7]. *BR* creates k datasets each for one class label and trains a binary classifier, one for each different dataset. In [8] *Calibrated Label Ranking (CLR)* is proposed. A label ranking method is able to predict a ranking of all topics in decreasing order of relevance to a specific instance but it is not able to distinguish between the sets of relevant and non-relevant topics. To overcome this problem *CLR* adds an extra label to the original label set which is interpreted as "neutral element". *ML-kNN* [9] adapts lazy learning techniques to solve multi-label problems. To identify the label set for a given instance it uses *maximum a posteriori (MAP)*, based on prior and posterior probabilities for each k nearest neighbor label. *BP-MLL* [19] is an adaptation of *back-propagation* algorithm for multi-label learning. The algorithm introduces a new error function that takes multiple labels into account. Multi-label perceptron based algorithms have also been extended for multi-label learning. *Multi-class Multi-layer Perceptron (MMP)* is proposed in [10] where the perceptron algorithms weight update is performed in such a way that it leads to correct label ranking. *Support Vector Machine (SVM)* is used in [11] where *RankSVM* is proposed. *RankSVM* defines a specific cost function and the corresponding margin in order to solve multilabel problems. In this paper we extend MLDA and propose the Direct Multi-label Discriminant Analysis method for dimensionality reduction of multi-label data. MLDA cannot find a space with a larger dimensionality than the number of the labels. The proposed method overcomes this limitation and projects data onto a subspace that gives more than $K - 1$ dimensions. This way we can find a reduced space that improves classification accuracy.

2 Direct Multi-label Discriminant Analysis

The proposed Direct MLDA extends MLDA by modifying the between class scatter matrix in order to improve classification accuracy. Given a multi label data set with n samples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ and K classes where $\mathbf{x}_i \in \mathcal{R}^p$ and $\mathbf{y}_i \in \{0, 1\}^K$, $y_i(k) = 1$ if \mathbf{x}_i belongs to the k -th class, and 0 otherwise. Let the input data be partitioned into K groups as $\{\pi_k\}_{k=1}^K$ where π_k denotes the sample set of the k -th class with n_k data points. We write $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T$ and $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n]^T = [\mathbf{y}_{(1)}, \dots, \mathbf{y}_{(K)}]$ where $\mathbf{y}_i \in \{0, 1\}^n$ is the class-wise label indication vector for the k -th class.

In order to improve classification accuracy multi-label learning takes into account label correlation that takes advantages of label interactions. The label correlation between two classes is formulated as following [16]:

$$C_{kl} = \cos(\mathbf{y}_{(k)}, \mathbf{y}_{(l)}) = \frac{\langle \mathbf{y}_{(k)}, \mathbf{y}_{(l)} \rangle}{\|\mathbf{y}_{(k)}\| \|\mathbf{y}_{(l)}\|}. \quad (1)$$

In this Section, we discuss *MLDA* in Section 2.1 and in Section 2.2 we propose *Direct MLDA* for dimensionality reduction of multilabel data.

2.1 Multi-label Linear Discriminant Analysis

Multi label Linear Discriminant Analysis (MLDA) is a multi-label version of *Linear Discriminant Analysis (LDA)* that has been adapted for multilabel data. The proposed scatter matrices are calculated class-wise as:

$$\mathbf{S}_b = \sum_{k=1}^K \mathbf{S}_b^{(k)}, \quad \mathbf{S}_b^{(k)} = \left(\sum_{i=1}^n Y_{ik} \right) (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T. \quad (2)$$

$$\mathbf{S}_w = \sum_{k=1}^K \mathbf{S}_w^{(k)}, \quad \mathbf{S}_w^{(k)} = \sum_{i=1}^n Y_{ik}(\mathbf{x}_i - \mathbf{m}_k)(\mathbf{x}_i - \mathbf{m}_k)^T. \quad (3)$$

where \mathbf{m}_k is the mean vector of class k and \mathbf{m} is the multi-label global mean vector. *MLDA* takes advantage of label correlation (1), and constructs the correlation matrix $\mathbf{C} \in \mathcal{R}^{K \times K}$ to define the correlations between labels. Moreover solves the over-counting problem, when a datapoint \mathbf{x}_i is used more than once on the calculation of the scatter matrices, by the following normalized matrix $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_n]^T \in \mathcal{R}^{n \times K}$:

$$\mathbf{z}_i = \frac{\mathbf{C}\mathbf{y}_i}{\|\mathbf{y}_i\|_{\ell_1}}. \quad (4)$$

where $\|\cdot\|_{\ell_1}$ is the ℓ_1 -norm of a vector. In equations (2) and (3) \mathbf{Y} is replaced by \mathbf{Z} .

MLDA projects the original data \mathbf{X} onto a lower q -dimensional feature space by taking into account both the within-class scatter matrix and the between-class scatter matrix. *MLDA* tries to minimize \mathbf{S}_w in order to keep each class compact and maximize \mathbf{S}_b in order to separate classes as much as possible. Thus the following criterion is maximized:

$$J = \frac{\text{tr}\{\mathbf{G}^T \mathbf{S}_b \mathbf{G}\}}{\text{tr}\{\mathbf{G}^T \mathbf{S}_w \mathbf{G}\}}. \quad (5)$$

where \mathbf{G} the transformation matrix that consists of q eigenvectors \mathbf{g} that correspond to the q largest eigenvalues of the eigenanalysis problem, $\mathbf{S}_w^{-1} \mathbf{S}_b \mathbf{g} = \lambda \mathbf{g}$.

2.2 Direct Multi-label Discriminant Analysis

A very strong limitation that *MLDA* suffers from, is the number of the produced projections that are defined as $K - 1$ for a K class problem. Indeed, the between scatter matrix as defined in 2 is comprised of K rank 1 matrices that consider only the mean vectors for each label and the global mean vector. This can be interpreted as information less in the definition of \mathbf{S}_b since the data samples are represented by their mean vectors. This representation results to a low rank matrix \mathbf{S}_b that can produce at most $K - 1$ projections for dimensionality reduction. *Direct MLDA* overcomes this problem and searches for a subspace that gives more than $K - 1$ dimensions and better classification accuracy. In *Direct MLDA* the definition of the between-class scatter matrix \mathbf{S}_b changes in order to

distinguish data that do not have a specific label from the mean vector of the data belong to this label. \mathbf{S}_b is defined as:

$$\mathbf{S}_b = \sum_{k=1}^K \mathbf{S}_b^{(k)}, \mathbf{S}_b^{(k)} = \sum_{i=1}^n (1 - Y_{ik})(\mathbf{x}_i - \mathbf{m}_k)(\mathbf{x}_i - \mathbf{m}_k)^T. \quad (6)$$

The definition of the within-class scatter matrix remains the same as we try to minimize the covariance of data that belong to the same class.

$$\mathbf{S}_w = \sum_{k=1}^K \mathbf{S}_w^{(k)}, \mathbf{S}_w^{(k)} = \sum_{i=1}^n Y_{ik}(\mathbf{x}_i - \mathbf{m}_k)(\mathbf{x}_i - \mathbf{m}_k)^T. \quad (7)$$

where \mathbf{m}_k is the mean vector of class k which is defined as:

$$\mathbf{m}_k = \frac{\sum_{i=1}^n Y_{ik} \mathbf{x}_i}{\sum_{i=1}^n Y_{ik}}. \quad (8)$$

Direct MLDA projects the original p -dimensional feature vectors into a new reduced q -dimensional feature space while keeping the discrimination information between classes. We wish to determine a transform $\mathbf{w}_i = \mathbf{G}^T \mathbf{x}_i$, where $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_q]$ the projection matrix, such that the projected classes are well separated. After the projection onto \mathbf{g}_t the between-class scatter matrix is defined as $\mathbf{S}_b^{gt} = \mathbf{g}_t^T \mathbf{S}_b \mathbf{g}_t$ and the within-class scatter matrix is defined as $\mathbf{S}_w^{gt} = \mathbf{g}_t^T \mathbf{S}_w \mathbf{g}_t$. To enhance the separability of the classes we wish to maximize \mathbf{S}_b^{gt} and keep each cluster compact by minimizing \mathbf{S}_w^{gt} , thus the following criterion function should be maximized:

$$J = \frac{\mathbf{g}_t^T \mathbf{S}_b \mathbf{g}_t}{\mathbf{g}_t^T \mathbf{S}_w \mathbf{g}_t}. \quad (9)$$

We wish to maximize criterion J for each \mathbf{g}_t of matrix \mathbf{G} , thus we define the following eigenanalysis problem:

$$\mathbf{S}_w^{-1} \mathbf{S}_b \mathbf{g} = \lambda \mathbf{g}. \quad (10)$$

The projections vectors of *Direct MLDA* are the q eigenvectors \mathbf{g}_t corresponding to the q largest eigenvalues of $\mathbf{S}_w^{-1} \mathbf{S}_b$. Label correlation was also examined in *Direct MLDA* according to (1) and the over-counting problem was solved similar to *MLDA* by the following normalized matrix $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_n]^T \in \mathcal{R}^{n \times K}$:

$$\mathbf{z}_i = \frac{\mathbf{C} \mathbf{y}_i}{\|\mathbf{y}_i\|_{\ell 1}}. \quad (11)$$

where $\|\cdot\|_{\ell 1}$ is the $\ell 1$ -norm of a vector. In equations (6) and (7) \mathbf{Y} is replaced by \mathbf{Z} .

3 Experimental Results on Video Analysis

The proposed approach has been applied to several challenging problems that arise in digital media analysis and description. That is, in digital video analysis one objective is to annotate the video according to the appearances of the actors and to automatically recognize their faces [25], their activities [26] or even their facial expression [27]. All these recognition problems refer to the same multi-label video data and thus, it would be useful to reduce the video dimensionality keeping the discriminability for each label.

Performance evaluation in multi-label learning is more complicated than traditional single label classification as multi-label data are associated with more than one label simultaneously. The multi-label evaluation metrics that are used in this paper consider the performance of label set prediction as well as the performance of label ranking. For the evaluation of label prediction *macro* (Precision, Recall, F-measure, Accuracy) and *micro* (Recall, F-measure, Accuracy) measures were used that average difference of the actual and the predicted sets of labels over all labels (*micro* measures) or for each label and subsequently average over all labels (*macro* measures). Hamming Loss was also used to evaluate the fraction of misclassified instance-label pairs. For the evaluation of label ranking One Error, Coverage, Ranking Loss and Average Precision were used. One error evaluates the fraction of examples whose top-ranked label is not in the relevant label set. Coverage evaluates how far on average a learning algorithm needs to move down in the ranked label list in order to cover all the relevant labels of the example. Ranking Loss evaluates the fraction of reversely ordered label pairs and Average Precision computes for each relevant label the fraction of relevant labels ranked higher than it, and finally averages over all relevant labels.

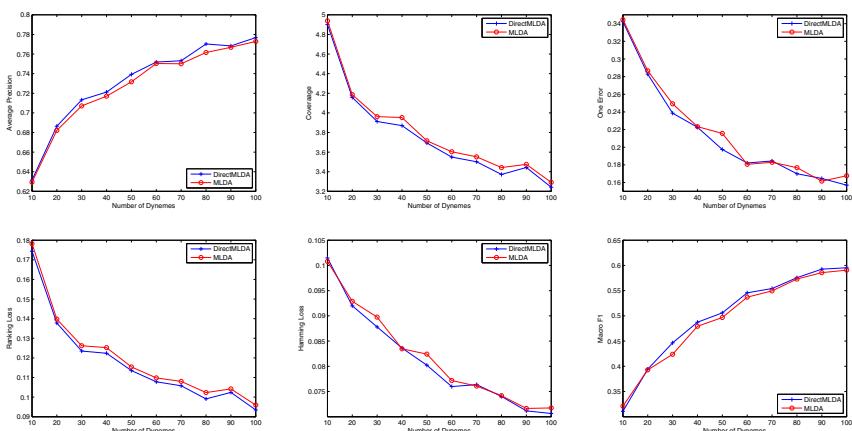


Fig. 1. Performance of *MLDA* and *Direct MLDA* for different evaluation metrics and number of dynemes for the database *i3DPost mask*

To examine *Direct MLDA* performance we used the following databases that refer mostly to multi-label activity recognition for video analysis:

i3DPost [20] is an image sequence database that contains 64 high-resolution image sequence of eight persons performing eight actions and two person interaction. Eight cameras having a wide 45° viewing angle difference was used to provide 360° coverage of the capture volume. *Mobiserv*[24] is an image sequence database, that depicts twelve persons performing three actions activities "eat", "drink" and "apraxia". The total number od activity video is 954.

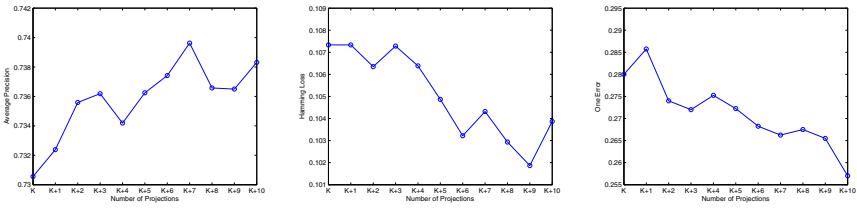


Fig. 2. Performance of *Direct MLDA* for different number of projections and metrics

IXMAS[21] database contains 330 low resolution image sequences of ten persons performing eleven activities. Each sequence has been captured by five cameras. The persons freely change position and orientation.

The activity representation of those databases was formed with the use of dynemes that refer to sequences of movement primitives. By calculating the similarity from multi-view dynemes the final representation of human activity was formed. We used 5-fold cross validation to evaluate the performance of the proposed approach to those databases for different number of dynemes [24]. In all these activity video datasets there are multiple labels per sample (i.e., video) that correspond to the identity of the person performing the activity, the different activity type and the different camera view. That is, the proposed approach is tested on recognizing simultaneously the correct person id, the correct activity and the correct camera view using the same dyneme based representation and the direct MLDA for dimensionality reduction.

Emotion [22] categorizes 593 songs into 6 emotions. A 5-fold cross validation was also used on *Emotion* database.

Reference [23] database, from Yahoo dataset refers to multi-label web pages. In *Reference*'s label set we removed topics with less than 100 web pages. The high dimensionality was reduced using *Random Projections*.

Multi-label K-Nearest Neighbor (ML-KNN) was used for classification after dimensionality reduction by *MLDA* and *Direct MLDA*. The selected number of K-Nearest Neighbors was defined to 5.

Evaluation of *MLDA* and the proposed *Direct MLDA* are depicted in Tables 1 and 2. The results show that *Direct MLDA* is superior to *MLDA* in terms of both label prediction and label ranking. Although *Direct MLDA* can

project data to more than $K - 1$ projections to improve classification accuracy, we prove that *Direct MLDA* is superior to *MLDA* also for $K - 1$ projections. In Figure 1 we project the original database onto a $K - 1$ -dimensional space in both cases (*MLDA*, *Direct MLDA*) and show that *Direct MLDA* gives better results for the database *i3DPost mask* for different number of dynemes for different evaluation metrics. In Figure 2 performance of *Direct MLDA* for *Reference* is presented for different number of projections for the evaluation metrics Average Precision, OneError and Hamming Loss.

Table 1. Performance evaluations of *MLDA* and *Direct MLDA* for *i3DPost mask*, *i3DPost stips* and *mobiliserv* databases

| | i3DPost mask (180) | | i3DPost STIPs (30) | | mobiliserv (50) | |
|---------------------|--------------------|---------------|--------------------|---------------|-----------------|---------------|
| | MLDA | DMLDA(K-1) | MLDA | DMLDA(K+5) | MLDA | DMLDA(K+1) |
| F1-macro ↑ | 0.4940 | 0.6215 | 0.4324 | 0.4510 | 0.4183 | 0.4237 |
| Macro Precision ↑ | 0.6812 | 0.7985 | 0.6477 | 0.6648 | 0.5881 | 0.5895 |
| Macro Recall ↑ | 0.4328 | 0.5498 | 0.3756 | 0.3887 | 0.3747 | 0.3833 |
| F1-micro ↑ | 0.5697 | 0.6788 | 0.5164 | 0.5292 | 0.6619 | 0.6661 |
| Micro Recall ↑ | 0.4659 | 0.5778 | 0.3783 | 0.3887 | 0.5368 | 0.5484 |
| Hamming Loss ↓ | 0.0810 | 0.0684 | 0.0883 | 0.0863 | 0.0715 | 0.0717 |
| Accuracy ↑ | 0.9190 | 0.9316 | 0.9116 | 0.9137 | 0.9285 | 0.9283 |
| Average Precision ↑ | 0.6822 | 0.6864 | 0.6822 | 0.6848 | 0.7874 | 0.7879 |
| Coverage ↓ | 4.0871 | 3.3483 | 8.5161 | 8.3545 | 6.0526 | 5.9573 |
| One Error ↓ | 0.2433 | 0.1517 | 0.1093 | 0.1054 | 0.1094 | 0.1126 |
| Ranking Loss ↓ | 0.1387 | 0.0982 | 0.1312 | 0.1325 | 0.0907 | 0.0898 |

Table 2. Performance evaluations of *MLDA* and *Direct MLDA* for *IXMAS mask*, *Emotion* and *Reference* databases

| | IXMAS mask (10) | | Emotion | | Reference | |
|---------------------|-----------------|----------------|---------------|---------------|---------------|---------------|
| | MLDA | DMLDA(K-1) | MLDA | DMLDA(K+2) | MLDA | DMLDA(K+16) |
| F1-macro ↑ | 0.0505 | 0.0624 | 0.5794 | 0.6009 | 0.3221 | 0.3105 |
| Macro Precision ↑ | 0.1628 | 0.1841 | 0.6752 | 0.6519 | 0.3014 | 0.3303 |
| Macro Recall ↑ | 0.0335 | 0.0439 | 0.5334 | 0.5802 | 0.3644 | 0.3177 |
| F1-micro ↑ | 0.0628 | 0.0794 | 0.6093 | 0.6248 | 0.5374 | 0.6100 |
| Micro Recall ↑ | 0.0333 | 0.0431 | 0.5509 | 0.5964 | 0.5543 | 0.5775 |
| Hamming Loss ↓ | 0.0826 | 0.0827 | 0.2198 | 0.2230 | 0.1331 | 0.1026 |
| Accuracy ↑ | 0.9174 | 0.9173 | 0.7802 | 0.7761 | 0.8669 | 0.8964 |
| Average Precision ↑ | 0.3518 | 0.3579 | 0.7695 | 0.7758 | 0.6808 | 0.7479 |
| Coverage ↓ | 11.4778 | 11.3227 | 1.9360 | 1.9044 | 4.0580 | 3.8367 |
| One Error ↓ | 0.7069 | 0.7009 | 0.3170 | 0.3256 | 0.3765 | 0.2485 |
| Ranking Loss ↓ | 0.3164 | 0.3102 | 0.1948 | 0.1903 | 0.1410 | 0.1187 |

4 Conclusions

In this paper we have proposed the *Direct Multi-label Linear Discriminant Analysis* method, as an extend of *Multi-label Linear Discriminant Analysis*. We reformulated the between-class scatter matrix in order to distinguish data that do not have a specific label. Direct MLDA gives more than $K - 1$ projections and search this way for the better reduced space that improves classification accuracy to the maximum. The theoretical advantages of our method are confirmed in experimental evaluations on multi-label video analysis that show that *Direct MLDA* performs better than *MLDA* in terms of both label prediction and label ranking.

Acknowledgment. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 316564 (IMPART). This publication reflects only the authors views. The European Union is not liable for any use that may be made of the information contained therein.

References

1. Zhang, M., Zhou, Z.: A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering* (2013)
2. Yang, S., Kim, S.-K., Ro, Y.-M.: Semantic Home Photo Categorization. *IEEE Transactions on Circuits and Systems for Video Technology*, 324–335 (2007)
3. Tang, J., Hua, X.-S., Wang, M., Gu, Z., Qi, G.-J., Wu, X.: Correlative Linear Neighborhood Propagation for Video Annotation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 409–416 (2008)
4. Borges, H.B., Nievola, J.C.: Multi-Label Hierarchical Classification using a Competitive Neural Network for protein function prediction. In: The 2012 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2012)
5. Wang, H., Hu, J.: Multi-label image annotation via Maximum Consistency. In: 2010 17th IEEE International Conference on Image Processing (ICIP), pp. 2337–2340 (2010)
6. Huang, S., Jin, L.: A PLSA-Based Semantic Bag Generator with Application to Natural Scene Classification under Multi-instance Multi-label Learning Framework. In: Fifth International Conference on Image and Graphics, pp. 331–335 (2009)
7. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data-instance Multi-label Learning Framework. In: Data Mining and Knowledge Discovery Handbook, pp. 667–685 (2010)
8. Fürnkranz, J., Hüllermeier, E., Loza Mencía, E., Brinker, K.: Multilabel classification via calibrated label ranking, pp. 133–153 (2008)
9. Zhang, M.-L. and Zhou, Z.-H.: A k-nearest neighbor based algorithm for multi-label classification. In: 2005 IEEE International Conference on Granular Computing, pp. 718–721 (2005)
10. Crammer, K., Singer, Y., Jaz, K., Hofmann, T., Poggio, T., Shawe-taylor, J.: A Family of Additive Online Algorithms for Category Ranking. *Journal of Machine Learning Research* (2003)

11. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Advances in Neural Information Processing Systems, pp. 681–687 (2001)
12. Arenas-garcia, J., Petersen, K.B., Hansen, L.K.: Sparse Kernel Orthonormalized PLS for feature extraction in large data set. In: Advances in Neural Information Processing Systems (2007)
13. Wang, H., Ding, C., Huang, H.: Multi-label linear discriminant analysis. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 126–139. Springer, Heidelberg (2010)
14. Zhang, Y., Zhou, Z.-H.: Multilabel dimensionality reduction via dependence maximization. ACM Transactions on Knowledge Discovery from Data (TKDD), 1–14 (2010)
15. Yu, K., Yu, S., Tresp, V.: Multi-label informed latent semantic indexing. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 258–265 (2005)
16. Wang, H., Huang, H., Ding, C.: Image annotation using multi-label correlated Green’s function. In: IEEE 12th International Conference on Computer Vision, pp. 2029–2034 (2009)
17. Bingham, E., Mannila, H.: Random projection in dimensionality reduction: applications to image and text data. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 245–250 (2001)
18. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science, 391–407 (1990)
19. Zhang, M.L., Zhou, Z.H.: Multi-label neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering, 1338–1351 (2006)
20. Gkalelis, N., Kim, H., Hilton, A., Nikolaidis, N., Pitas, I.: The i3DPost multiview and 3D human action/interaction database. In: 6th Conference on Visual Media Production, pp. 159–168 (2009)
21. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. Computer Vision and Image Understanding, 249–257 (2006)
22. Trohidis, K., Tsoumacas, G., Kalliris, G., Vlahavas, I.: Multilabel classification of music into emotions. In: Proc. of ISMIR (2008)
23. Ueda, N., Saito, K.: Single-shot detection of multiple categories of text using parametric mixture models. In: Proc. of SIGKDD, pp. 626–631 (2002)
24. Iosifidis, A., Tefas, A., Pitas, I.: Activity-Based Person Identification Using Fuzzy Representation and Discriminant Learning. IEEE Trans. on Information Forensics and Security, 530–542 (2012)
25. Kyperountas, M., Tefas, A., Pitas, I.: Dynamic training using multistage clustering for face recognition. Pattern Recognition, 894–905 (2008)
26. Gkalelis, N., Tefas, A., Pitas, I.: Combining fuzzy vector quantization with linear discriminant analysis for continuous human movement recognition. IEEE Transactions on Circuits and Systems for Video Technology, 1511–1521 (2008)
27. Kyperountas, M., Tefas, A., Pitas, I.: Salient feature and reliable classifier selection for facial expression classification. Pattern Recognition, 972–986 (2010)

Image Restoration Method by Total Variation Minimization Using Multilayer Neural Networks Approach

Mohammed Debakla¹, Khalifa Djemal², and Mohamed Benyettou³

¹ Laboratoire IRBG Université de Mascara Algérie

² Laboratoire IBISC Université d'Évry Val d'Essonne France

³ Laboratoire MOSIM USTOran Algérie

{Debakla_med,med_benyettou}@yahoo.fr,
djemal@iup.univ-evry.fr

Abstract. Neural network have seen an explosion of interest over the last years and have been successfully applied across an extraordinary range of problem domains such as medicine, engineering, geology, physique, biology and especially image processing field. In image processing domain, the noise reduction is a very important task. Indeed, many approaches and methods have been developed and proposed in the literature. In this paper, we present a new restoration method for noisy images by minimizing the Total Variation (TV) under constraints using a multilayer neural network (MLP). The proposed method can restore degraded images and preserves the discontinuities. Effectiveness of our proposed approach is showed through the obtained results on different noisy images.

Keywords: Image restoration, Partial Differential Equations, Total variation, Multilayer neural network.

1 Introduction

Image restoration refers to the recovery of an original image from degraded observations. The purpose of image restoration is to "compensate for" or "undo" defects which degrade an image. The degraded image restoration problems are largely treated in literature in many applications [21, 9, 19, 11, 1]. It seeks to correct the distortions that result in the emergence of many degradations, they result in a reduction of the contrast, blur and random or due to noise occurring during image formation.

In recent years, a new emerging technique has grown considerably. It helps to process information more efficiently than conventional systems, to remove data previously unavailable to assist in decision making, it is the neural network. These methods have been successfully introduced in image processing and computer vision. Their applications are numerous, such as edge detection [7, 16, 2], segmentation [13], stereovision [14], restoration [22, 17, 20, 2].

In this paper, we present a new image restoration method based on solving PDE with nonlinear model by multilayer neural network. We use the TV model proposed by Rudin et al [19]. This approach rely on the function approximation capabilities of feedforward neural networks and results in the construction of a solution written in a differentiable, closed analytic form. The multilayer neural network considered as the basis of approximation, whose parameters (weights) are adjusted to minimize an appropriate error function. To train the network we use optimization techniques, which require the calculation of the gradient of the error to set the network settings with respect its parameters.

The paper is organized as follows: section 2 we provide the problem formulation, and the error function to be minimized. In Section 3 we present the architecture of the neural net under analysis and the error function minimization based MLP approach. We illustrate the method by presenting some examples of synthetic and real image restoration in Section 4.

2 Restoration Problem Formulation

TV regularization has been extremely successful in a wide variety of restoration problems, and remains one of the most active areas of research in mathematical image processing and computer vision. By now, their scope encompasses not only the fundamental problem of image denoising, but also other restoration tasks such as deblurring, blind deconvolution, and inpainting [19, 6, 10].

In all these approaches, a TV model is minimized in different way. The typical problem in image restoration case were introduced by Rudin et al. in the pioneering work [19] on edge pre-serving image denoising with the minimization of the following functional:

$$F(u) = \int_{\Omega} |Du| + \lambda \|u_0 - u\|^2 dx dy \quad (1)$$

where $\int_{\Omega} |Du|$ represents the TV model of the image u . If image u is regular, the equation (1) becomes only $\int_{\Omega} |\nabla u| dx$. In [19] the authors considered that the noise witch corrupted image is distinguished from noiseless one by the size of total variation, which is defined as $\int_{\Omega} \sqrt{u_x^2 + u_y^2} dx dy$, where Ω denotes the image domain u_x and u_y denote the corresponding partial differentiation. Consequently, they propose to restore a noisy and blurred image by minimizing total variation:

$$\min_u \int_{\Omega} \sqrt{u_x^2 + u_y^2} dx dy \quad (2)$$

under constraints:
$$\begin{cases} \int_{\Omega} \frac{1}{2} (u(x, y) - u_0(x, y))^2 dx dy = \sigma^2 \\ \int_{\Omega} (u(x, y) - u_0(x, y)) dx dy = 0 \end{cases} \quad (3)$$

Where $u_0(x, y)$ represents the given observed image, which considered to be corrupted by a Gaussian noise of variance σ^2 and $u(x, y)$ denote the desired clean image. To minimise (2) Rudin et al. [19] have applied the Euler-Lagrange equation under the two constraints (3) they obtain the following equation:

$$\frac{\partial}{\partial x} \left(\frac{u_x}{\sqrt{u_x^2 + u_y^2}} \right) + \frac{\partial}{\partial y} \left(\frac{u_y}{\sqrt{u_x^2 + u_y^2}} \right) - \lambda (u - u_0) = 0 \quad (4)$$

Where λ the Lagrange multiplier is given by:

$$\lambda = \frac{1}{2\sigma^2} \int \left[\sqrt{u_x^2 + u_y^2} - \left(\frac{(u_0)_x u_x}{\sqrt{u_x^2 + u_y^2}} \right) + \left(\frac{(u_0)_y u_y}{\sqrt{u_x^2 + u_y^2}} \right) \right] \quad (5)$$

Over the years, the TV model [19], has been extended to many other image restoration tasks, and has been modified in a variety of ways to improve its performance. The classical approach is then to use the associated Euler-Lagrange equation to compute the solution. Fixed step gradient descent [19], or later quasi-Newton methods [7, 15, 4]. Iterative methods have proved successful [8]. Ideas from duality can be found in [5, 3]. Recently, it has been shown that a combination of the primal and dual problems has been introduced in [23]. Notice that all these works use an approximate numerical method for solving the associated PDE with the restoration problem.

An image restoration problem can be transformed to an optimization problem. In our assumption and from (4) we can formulate the image restoration problem as minimize the following error function:

$$E(x, y) = \frac{\partial}{\partial x} \left(\frac{u_x}{\sqrt{u_x^2 + u_y^2}} \right) + \frac{\partial}{\partial y} \left(\frac{u_y}{\sqrt{u_x^2 + u_y^2}} \right) - \lambda (u - u_0) \quad (6)$$

We propose to minimize the error function E in equation (6) using an MLP neural network approach. We present a generalization of the problem and then we introduce a weighting technique of weight for a term based on the data fidelity.

3 Error Function Minimization Based MLP Approach

To minimize (6), the used MLP is based on three layers: an input layer contains one neuron, one hidden layer consists of a varying number of neurons at the discretion of the programmer to be determined after several tests and an output layer contains only

one neuron. The sigmoid function δ is applied to each neuron in the hidden layer and output neuron (Figure 1). The general idea of our method is based on the work proposed by Lagaris et al. [12], where the authors were able to solve partial differential equations by neural networks.

Recall that in all previous work using neural networks in the field of restoration, the chosen network type is related to the assumptions given by the authors in the restoration process.

In our case we take the assumption that the output of the MLP is an image corresponding to the desired image $u(x, y)$. So we provided as inputs to the multilayer neural network the degraded image $u_0(x, y)$ and at the output of the network, we have:

$$u(x, y) = N(u_0(x, y), w) \quad (7)$$

Where $u_o(x, y)$ is the intensity of pixel (x, y) of the noisy image and w is the parameters vector (weights) of the MLP.

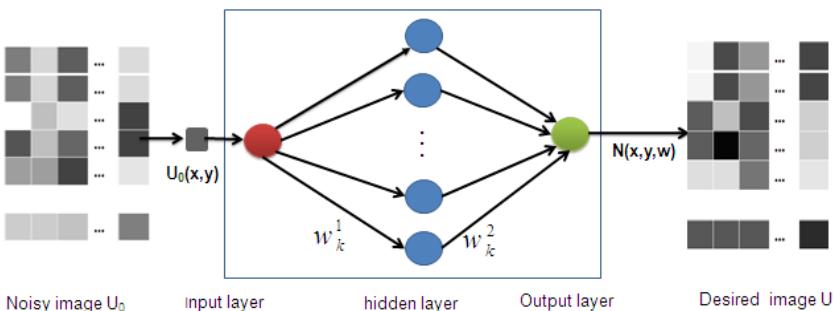


Fig. 1. The used multilayer neural network structure

We consider that all values of the sigmoid function are taken between 0 and 1; we must normalize the input and output values of the network. Indeed, each pixel of noisy image is coded by one byte gray-level, then all input values will be divided by 255, and the output results will be multiplied by 255.

After normalization, the gray level of the output of each pixel is calculated by:

$$N(x, y) = 255 \delta \left(\sum_{k=1}^n w_k^2 \delta(w_k^1 u_0(x, y)) \right) \quad (8)$$

Once all pixels of the noisy image are presented to the network, we get an image U aimed that is satisfied the equation (4). In fact, we calculate the error function E caused by the network from the equation (6). Because of the update rule in the MLP algorithm, these error decreases gradually. To make change the weights, we use steepest gradient descent method. The adaptation of weights is done by the following formula:

$$w_k^i(t+1) = w_k^i(t) - \eta \Delta w_k^i(t) \quad (9)$$

Where $w_k^i(t+1)$ and $w_k^i(t)$ represents respectively the new and the last values of weights, with $i=1$ (weights between input layer and hidden layer) or 2 (weights between hidden layer and output layer), k is the number of neurons in the hidden layer and η learning positive constant.

The weight variation Δw_k^i is obtained by minimization of error (6) using the following equation:

$$\Delta w_k^i = \sum_{x,y} \frac{\partial E(x,y)}{\partial w_k^i} \quad (10)$$

Then

$$\frac{\partial E(x,y)}{\partial w_k^i} = \frac{\partial}{\partial x} \left(\frac{u_x}{\sqrt{u_x^2 + u_y^2}} \right) + \frac{\partial}{\partial y} \left(\frac{u_y}{\sqrt{u_x^2 + u_y^2}} \right) - \frac{\partial \lambda}{\partial w_k^i} (u - u_0) - \lambda \frac{\partial u}{\partial w_k^i} \quad (11)$$

4 Experimental Results

In this section, we present some experimental results that evaluate the performance of our approach. We also chose to compare the denoising performance of our approach (10 neurons at hidden layer) with other methods using their optimal parameters: Tichonov regularization [21], minimizing TV model of ROF [19] and Multiscale Neural Network (MNN) [2] for synthetic and real noisy images. For the purpose of objectively testing the performance of image restoration algorithm, the improvement ISNR is often used. This metric using the restored image is given by:

$$ISNR(f,u) = 10 \log_{10} \frac{\sum_{i,j} [f(i,j) - u_0(i,j)]^2}{\sum_{i,j} [f(i,j) - u(i,j)]^2} \quad (12)$$

Where $f(i,j)$, $u_0(i,j)$ and $u(i,j)$ denote the original, degraded and restored images, respectively.

We also used the Normalized Mean Square Error (NMSE) as another measure of quality. If the value of NMSE decreases, the restoration is better. NMSE is given by:

$$NMSE(f,u) = \frac{\sum_{i,j} [f(i,j) - u(i,j)]^2}{\sum_{i,j} [f(i,j)]^2} \quad (13)$$

The first denoising experiment is shown in Figure 2. For this experiment, using synthetic image, we added white Gaussian noise with standard deviations $\sigma = 25$.

We can see clearly that the Tikhonov method tends to blur the image and can't preserve details. The restored image obtained by the ROF method, suppress some details and blur is still apparent. For the MNN method we can observe an increase in brightness and contrast for the restored image. The proposed method gives a good visual quality with strong noise suppression and also more details are preserved. The corresponding ISNR and NMSE values are shown in Table I.

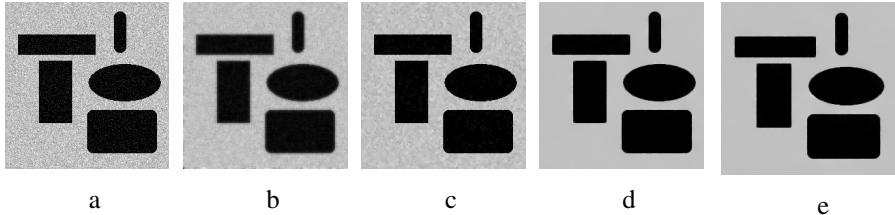


Fig. 2. Examples of white Gaussian noise reduction: (a) Gaussian degraded image with $\sigma = 25$, restoration result using: (b) Tikhonov's algorithm, (c) (ROF) total variation model, (d) Multiscale neural network, (e) Our approach

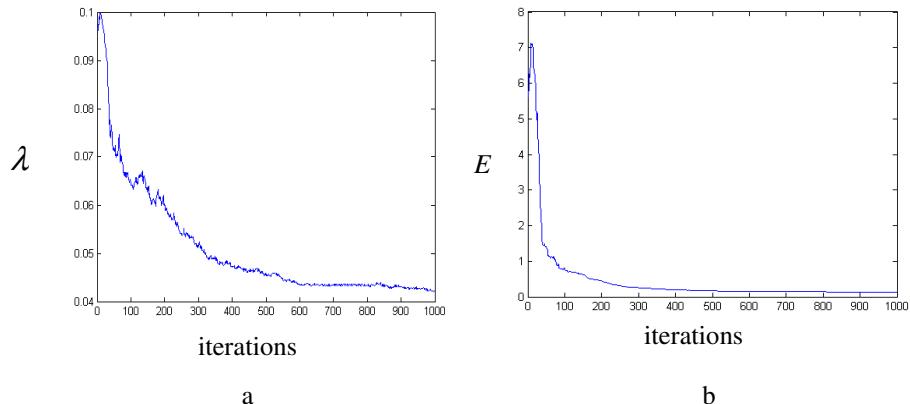


Fig. 3. Evolution of λ and E (3.a and 3.b respectively) according to the iterations number

Figure 3, illustrates the evolution of the Lagrange multiplier λ and the error function E according to the iterations number.

We applied our approach on real known noisy images (Lena image). The comparisons with other methods are given in figure 4 with the optimal parameters of each method.

The image is disturbed by a Gaussian additive noise with zero mean and standard deviations $\sigma = 15$, restored images, obtained after 400 iterations of our approach are presented Figure 4f.



Fig. 4. Lena image restoration: (a) original image, (b) Gaussian degraded image with $\sigma = 15$, restoration result using: (c) Tikhonov's algorithm, (d) (ROF) total variation model (e) Multiscale neural network method, (f) Our approach with 400 iteration

The results presented show the good performance of our algorithm, especially the preservation of discontinuities. Moreover the geometric characteristics such as corners and edges and originals contrast are well restored. For purposes of comparison, the results of restoration by the four methods chosen are summarized in Table 1.

Table 1. The ISNR and NMSE values of white Gaussian noise reduction

| Image | Tikhonov algorithm | | (ROF) TV model | | Multiscale Neural Network | | MLP algorithm (400 iteration) | |
|-----------|--------------------|--------|----------------|--------|---------------------------|--------|-------------------------------|---------------|
| | ISNR | NMSE | ISNR | NMSE | ISNR | NMSE | ISNR | NMSE |
| Synthetic | 2.21 | 0.0124 | 9.38 | 0.0024 | 12.57 | 0.0011 | 13.81 | 0.0011 |
| Lena | 1.26 | 0.0130 | 5.12 | 0.0053 | 4.79 | 0.0058 | 6.79 | 0.0036 |

5 Conclusion

In this paper, from image modelling and TV model minimization under constraints, we have showed that the obtained Euler-Lagrange functional can be resolved by minimizing an error functional using MLP approach. The proposed image restoration method which based on multi-layer neural network (MLP) has provides interesting results. The performance of our method is presented through comparison results.

References

- [1] Banham, M.R., Katsaggelos, A.K.: Digital image restoration. *IEEE Signal Process. Mag.* 14(2), 24–41 (1997)
- [2] Castro, A.P.A., Silva, J.D.S.: Neural Network-Based Multiscale Image Restoration Approach. In: *Proceeding on Electronic Imaging*, San Jose, vol. 6497, pp. 3854–3859 (2007)
- [3] Chambolle, A.: An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.* 20, 89–97 (2004)
- [4] Chan, T.F., Esedoglu, S., Park, F., Yip, A.: Total variation image restoration: overview and recent developments. In: *Handbook of Mathematical Models in Computer Vision*, pp. 17–31. Springer, New York (2006)
- [5] Chan, T., Golub, G., Mulet, P.: A nonlinear primal-dual method for total variation-based image restoration. *SIAM J. Sci. Comput.* 20(6), 1964–1977 (1999)
- [6] Chan, T., Shen, J.: *Image Processing and Analysis Variational, PDE, Wavelet, and Stochastic Methods*. SIAM, Philadelphia (2005)
- [7] Charbonnier, P., Blanc-Féraud, L., Aubert, G., Barlaud, M.: Deterministic edge-preserving regularization in computer imaging. *IEEE Transactions on Image Processing* 6(2), 298–311 (1997)
- [8] Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.* 57, 1413–1457 (2004)
- [9] Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6(6), 721–741 (1984)
- [10] Djemal, K.: Speckle Reduction in Ultrasound Images by Minimization of Total Variation. In: *IEEE International Conference on Image Processing*, Genova, Italya, vol. 3, pp. 357–360 (2005)
- [11] Katsaggelos, A.K., Biemond, J., Schafer, R.W., Mersereau, R.M.: A regularized iterative image restoration algorithm. *IEEE Trans. Signal Processing* 39, 914–929 (1999)
- [12] Lagaris, I., Aristidis, L., Fotiadis, D.: Artificial Neural Networks for Solving Ordinary and Partial Differential Equation. *IEEE Transaction on Neural Network* 9(5) (1998)
- [13] Meftah, B., Benyettou, A., Lzoray, O., Debakla, M.: Image Segmentation with Spiking Neuron Network. In: *1st Mediterranean Conference on Intelligent Systems and Automation (CISA 2008)*, Annaba, Algeria, AIP Conf. Proc., vol. 1019, pp. 15–19 (2008)
- [14] Nasrabadi, N., Choo, C.: Hopfield network for stereo vision correspondence. *IEEE Trans. Neural Network* 3, 5–13 (1992)
- [15] Ng, M.K., Qi, L., Yang, Y.F., Huang, Y.: On semismooth Newton methods for total variation minimization. *J. Math. Imaging Vis.* 27, 265–276 (2007)
- [16] Paik, J.K.: Image restoration and edge detection using neural networks. Ph.D.Dissertation, Dep. Elec. Eng., Computer Sci., North-Western Univ. (1990)
- [17] Paik, J.K., Katsaggelos, A.: Image restoration using a modified Hopfield Network. *IEEE Transactions on Image Processing* 1(1), 49–63 (1992)
- [18] Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12(7), 629–239 (1990)
- [19] Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* 60, 259–268 (1992)

- [20] Sun, Y.: Hopfield neural network based algorithms for image restoration and reconstruction Part II: Algorithms and simulation. *IEEE Trans. Signal Processing* 48, 2105–2118(2000)
- [21] Tikhonov, A.N., Arsenin, V.Y.: *Solutions of Ill-Posed Problems*. Winston and Sons, Washington (1977)
- [22] Zhou, Y.T., Vaid, A., Jenkins, B.K.: Image restoration using a neural network. *IEEE Trans. Acoust. Speech Signal Processing* 36, 1141–1151 (1988)
- [23] Zhu, M., Chan, T.F.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report 08-34* (2008)

Algorithmic Problem Solving Using Interactive Virtual Environment: A Case Study

Plerou P. Antonia and Panayiotis M. Vlamos

Department of Informatics,
Ionian University, Corfu, Greece
{tpplerou,vlamos}@ionio.gr

Abstract. The evaluation of basic arithmetic algorithms has been until recently the core of mathematical tests in elementary and secondary education. However, it is necessary that students are able to understand, analyze and improve more complex algorithms in order to support further the study of mathematics and science. In this paper, a number of issues concerning algorithmic thinking are explored. In particular, a case study is proposed in order to compare the efficiency of the traditional algorithmic problem solving in relation to problem solving using interactive virtual environment. The findings suggest that when problem solving using interactive interface is used under conditions the results are more efficient comparing to the traditional way of algorithmic problem solving.

Keywords: Algorithmic thinking, problem solving, interactive virtual environment.

1 Introduction

In this paper a method is proposed in order to check fifteen – sixteen year old students' performance in algorithmic problems presented in an interactive way. Besides generalized test concerning dyscalculia, there are several skill games, mainly commercial releases that are related to creativity and algorithmic thinking. However, the motivation of this case study is the evaluation of users' efficiency in algorithmic problem solving in interactive environments. In particular a group of high school students had to deal with three algorithmic problems of escalating difficulty and the results collected were subsequently studied and statistically analyzed with a critical and evaluation aim. The key contribution of this study is the resulting finding that students appear to be more efficient while dealing algorithmic problems in an interactive interface and that interactivity appears to affect students' performance, enhancing perception in algorithmic issues.

2 Related Work

There are quite many cases where several methods or games have been used to teach or improve logic skills and research dyscalculia features. In particular, according

Manuela Piazza et al. [5] a clear association between dyscalculia and impaired "number sense", is established using a psychophysical test to measure the numerosity of sets of dots using as a sample a group of 10-year-old dyscalculics. On the other hand according to Teressa Lucucano et al research conducted in 8–9-year-olds indicated no relationship between children with low numeracy and children with developmental dyscalculia. Therein, low numeracy was related not to a poor grasp of exact numerosities but to a poor understanding of symbolic numerals. Additionally, according to Karin Landerla et al. [7]. research in 8- and 9-year-old children selected for dyscalculia, reading difficulties or both, led to the conclusion that dyscalculia is the result of specific disabilities in basic numerical processing, rather than the consequence of deficits in other cognitive abilities.

There are also several works concerning teaching methods like the one proposed from Praveen Kumar et al. [8] where in three learning styles were proposed: visual, auditory and tactile/kinesthetic. In the same work the authors additionally presented methods of how these styles can be adopted to overcome Dyscalculia.

Furthermore according to Nagavalli [9] children with dyscalculia lack a "number sense" i.e. they have problems relating number symbols to real-world objects and situations. Software such as "Mighty Math" incorporate math learning into interactive video, using stories, music and visual cues to help students relate math concepts to everyday life. Electronic Math Worksheets and Math Software-Number Race are also used for the same reason.

On the other hand the creative and building aspects of Minecraft [12] allow players to build constructs out of textured cubes in a 3D procedurally generated world. Other activities in the game include exploration, gathering resources, crafting, and combat and although the game is about logical skills it has not actually diagnostic properties.

The proposed study is not actually about teaching or improving logical skills and algorithmic thinking or dyscalculia related issues. On the contrary, is about identifying students' performance while they are evaluated to algorithmic thinking abilities in traditional environment comparing to interactive environment. A strong aspect of this work is that assessment of algorithmic thinking is an area not sufficiently explored as previous work moves in a different aspect. Moreover, the additional focus of this work is on the evaluation and connection of ability of thinking algorithmically through different expression environments.

3 Methodology

The purpose of the case study that follows is to control and evaluate students' efficiency in algorithmic problem solving using interactive (virtual environment) interfaces and not actually proposing a method to improve or teach logic skills. As far as the notion of algorithm is concerned, a plethora of definitions exist, though most commonly the term algorithm is used in mathematics typically referring to a step by step problem solving procedure, especially an established recursive computational procedure for solving a problem in a finite number of steps. An important aspect of

algorithmic thinking concerns the way people conceive, analyze and solve problems. Additionally, algorithmic thinking has a strong creative aspect: the construction of new algorithms that solve given problems [1].

The results collected taken from proposed test were analyzed in order to explore whether algorithmic problem solving processes using computers is efficient enough comparing to traditional problem solving methods. The procedure proposed in this paper is based on the core principles of research methodology using pedagogical and technological rules. Initially, a pre test took place and a group of 46 students was divided in two equal groups, respective to performance level and test. The first group of students was asked to take a traditional test while in their classrooms. This test consisted of three algorithmic problems with escalating difficulty levels. The available time was 25 minutes including the time needed for comprehension of tasks and the time provided for students to write down their answers on the answer-sheet. The second group was asked to work on same algorithmic problems, though in this case the problems were presented as an interactive game using an interactive/virtual environment. The total time available to this group was 15 minutes in addition to the time provided in order to read instructions and comprehend game rules. Moreover, a questionnaire was given to second group in order to identify and analyze their feelings during testing process following the approach adopted in [2].

4 Experimentation

4.1 Sampling

This case study was conducted in the first Grade of a Vocational Educational School in Corfu, Greece and all students participated were around 15-16 years. Initially, the sample pool had to be divided into two equal groups. In order to do so, a weighted diagnostic test concerning the adequacy of mainly perceptual abilities and logical skills evaluation, and less than basic arithmetic skills, as provided by the Ministry of Education, Lifelong Learning and Religious was used as a pre-test. To achieve an equal and assimilated group separation, the one to one individual matching principal [10] was utilized herein. Matching refers to the selection of the reference or comparison group that is similar to the index group with respect to the distribution of one or more potentially confounding factors. That means that one student included to the traditional tested group was matched to other student who was included to the interactive evaluated group, with similar performance in pre test conducted.

Thus, according to the pre-test results, 2 groups of 23 students each with high similarity of performance rate, were assembled and evaluated in the same algorithmic tasks. The traditional method test group was hosted in classroom whereas the group tested using the interactive virtual interface was evaluated in a computer lab. A supervisor observed students while they were completing tasks without being actively involved. In the case of the interactive virtual environment test, an assistant was also present in lab in order to support students in case of technical issues.

4.2 Measurements

Some of the most widely known and common algorithms are the elementary school procedures of addition, subtraction, multiplication and division. In our test the same tasks were given to both groups and the idea was to measure students' creativity during this test, following the paradigm of [3]. Thus, problem statements were presented in an analytical-detailed descriptive way using visual representations making clear the data and objective of task procedure. Accordingly, abbreviations, acronyms, obscure words as well as unknown domain-specific terms were intentionally avoided. Therefore, potential issues of dyslexia may not be involved due to dyslexics show inaccurate and particularly slow word-recognition skills, and in some cases reading-level-appropriate word recognition processes [11]. So, in this case three types of escalated difficulty tasks based in Adobe Flash Player were used; in the first one, Figure 1, students' had to move objects following predetermined rules to get to a fast, precise and optimized solution. In this task a three step algorithm was needed to be completed correctly. In particular:

Task 1: Please help the man in boat move the wolf, the sheep and the box of cabbage to the other side of the lake. Notice wolf eats sheep; sheep eats cabbage when no man is around.



Fig. 1. Screenshot of the visualization of Task 1

The second procedure was built around the Hanoi Towers puzzle, where dragging discs from one peg to another was asked, following a set rules. In this task an algorithm of seven steps was needed. The title of the task as given to both groups was:

Task 2: Please move all rings to the third pile and stack them according to the original order. Click and drag a ring to the target pile then click again to place it.

Rules:

- 1) Only the topmost ring on a pile can be moved each time.
- 2) A ring with larger size and number cannot be placed on top of a smaller ring.



Fig. 2. Screenshot of the visualization of Task 2, The Hanoi Towers puzzle

The third exercise, Figure 3, was about moving items under temporal limitations and following appropriate algorithmic steps for optimizing the solution of the problem. In this task to achieve a solution a thirteen step algorithm was required.

Task 3: Please help this family cross to the other side of the bridge.

Notice that it is night so you must have a lamp. Each person crosses the bridge at a different speed: 1 sec, 3 sec, 6 sec, 8 sec and 12 sec respectively. The bridge can hold maximum of 2 persons and a pair must walk together at rate of the slower person. Watch out, the lamp's light is enough for 30 sec only.



Fig. 3. Screenshot of the visualization of Task 3

The two first exercises required simple comprehension, while third one required increased concentration in order to be solved. It should be noted that students could review each one of those tasks more than once within the available time.

4.3 Results

A Chi-Square Test was performed in order to compare the performance of the groups of students dealing with the problems on algorithmic thinking. A Chi-Squared Test, also referred to as χ^2 test or Pearson Chi Square Test, is a statistical hypothesis test used to determine whether there is a significant difference between the expected frequencies and the observed frequencies in one or more categories [4].

The Chi-Square statistic may be used to test the hypothesis of no association between two or more groups, populations, or criteria.

In this case, the null hypothesis (H_0) was that there is a difference in performance between the two groups (students tested using interactive virtual interface were more efficient than students tested traditionally). The research hypothesis (H_1) was that there is no difference in efficiency of two groups. Analyzing the groups' performance in each exercise separately, it could be thus inferred that efficiency was affected in a different way in each testing problem procedure.

More interestingly, using the Chi-Square Test, more significant statistical outcomes were obtained: comparing the results for first and second p-values for Pearson Chi-Square Test, 0.75 and 0.219 respectively, were greater than 0.05 which is the significance sated level. Checking the p-value of Pearson Chi-Square control on each occasion, the null hypothesis is confirmed, thus there exists considerable difference between the groups' tested efficiency. As far as the answers in third task, the p-value for Pearson Chi-Square Test was 0.008, far less than the 0.05 significance sated level. In this case null hypothesis is rejected, so there was no considerable difference of groups' efficiency in this task. So, as far as fairly easy topics were concerned, wherein developed algorithmic thinking is not required, the interactive virtual environment use was quite effective but on particularly complex issues students' efficiency was not affected.

In order to better depict the results collected Figure 4 shows the task number is at the horizontal axis and the rate of students that answered correctly on vertical axis. As evident, the group tested using the interactive virtual interface had significantly better performance comparing to the traditionally tested group.

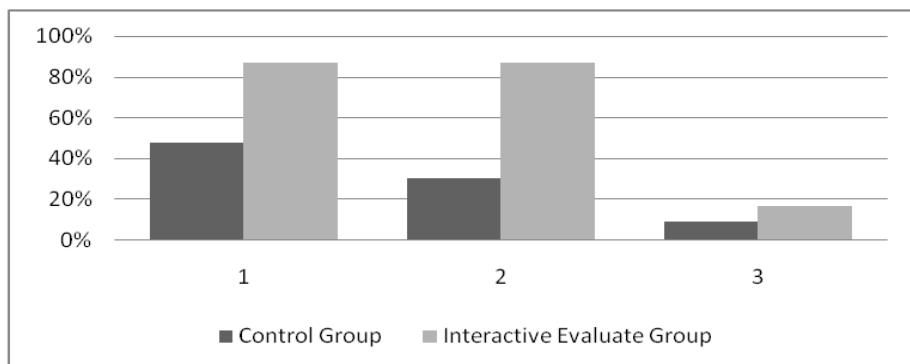


Fig. 4. Percentage of pupils that answered correctly in each task

Specifically, regarding the first task, only 48% of students tested by traditional means answered correctly in contrast to 87% of students of the group tested using interactive virtual interface that needed 2.35 minutes and 12.25 efforts on average. Respectively, for the second task only 30% of students of the first group answered properly while 87% of second group students answered correctly in 1.05 minutes and 1.75 efforts on average. Finally, in the third task 9% of the traditionally tested group

answered rightly whereas 15% of students completed the task properly using interactive virtual interface in 9.5 minutes and 6.5 efforts on average. Detailed results of the statistical analysis of test data are available in the Appendix.

In order to enhance Pearson chi-square results' reliability, Fisher's exact test [14] is additionally used as in this case the problem deals with small number of data. Fisher's exact test is a statistical significance test for categorical data that result from classifying objects and it is used to examine the significance of contingency. Initially, a control process concerning performance and causality was applied as in many cases there were only a few observations. According to Tables T7, T8 and T9 (found in the Appendix) at the first two tasks there is no statistically significant correlation between the examination method and the performance of students, while results are contrary for the third task. In detail the p collected in the first, second and third task were 0.011, 0.001 and 0.033, respectively.

In order to present an overall model test assessment herein Ordinal Logistic Regression [15] was additionally used assuming the number of correct answers given by each student as a dependent variable. The outcome of this test showed a firm statistically significant correlation between the examined way and the overall performance of students while observing that students evaluated with interactive way performed better than the traditionally tested group. As far as the dependent variable is concerned, the number of overall correct answers, the ordinal logistic regression model achieves a ratio count of 8.862 probability (Appendix Table T11), implying that students dealt with tasks using the interactive interface had 8.86 times higher probability of answering correctly comparing to students evaluated in the traditional way. This result is highly statistically significant.

Additionally, and in order to control the proper simulation of the two groups, a T-test was preformed [15]. Taking into consideration the results of the pre-test as a dependent variable, the T-test was applied in order to test if differences do exist in the grades/scores both teams achieved. The initial hypothesis H_0 was: The average scores are equal, while the H_1 hypothesis was: The average scores differ. According to Table T13 in Appendix the p value reached 0.612 showing that no rating difference between the two groups exists and that simulation was efficient.

The ordinal logistic regression was then repeated considering as a dependent variable the overall number of correct answers and as independent variables the groups (considered as qualitative variable with values: 0 = traditional evaluated group, 1= Interactively evaluated group) and pre test students' scores. According to the Table T14 of the Appendix it evident that probability of a student answering correctly in more tasks becomes thirty-six times greater when the student is evaluated interactively, keeping the pre-test performance the same. Pre-test's performance is statistically significant, i.e. students that scored higher per unit in pre-test have 2.2 times greater probability to respond correctly to more tasks in the diagnostic evaluation. However not only group separation remains statistically significant, but the interaction was also enhanced compared to the case that pre-test rating performance was not taken into consideration (the probability ratio was 8 in the previous analysis).

As this study suggests a comparison between homogenous groups formed without aiming to compare the results of earlier studies using control group was not necessary [13]. The statistical test aims at showing that the differences in the performance of the

students in the algorithmic problems was not due to factors such as the higher level of the group in contrast to the other since the division into groups was done using the pre-test.

As far as the students' feeling while completing tasks using interactive virtual environment, based on the supervisor's remark they were focused and not actually destructed by the electronic means. The results of the questionnaire given showed students mentioning positive attitude towards the testing process using interactive virtual interface. In detail, 70% of students tested consider that algorithmic thinking problem solving using interactive virtual environment was more interesting comparing to typical problem solving process. Furthermore, 83% of students stated that they felt joy and excitement during this procedure and about 65% would repeat the test, featuring this kind of extremely amusing testing process.

4.4 Discussion

One of the findings of this case study is that using a proper visualization of specific algorithmic problems can enhance students' comprehension of some basic concepts related to algorithms like correctness and efficiency of algorithms. Additionally, the interactive application method for dealing with algorithmic problems has been recorded to enhance the perspective of mathematics as a meaningful and creative subject, according to students' response to the questionnaire. Nevertheless, this does not mean that using an interactive environment can overall improve abilities concerning the algorithmic thinking but that may merely highlight existing capacities. Concluding, it is indicated that students using interactive interfaces, when faced with algorithmic problems, cannot always invent new algorithms for dealing with difficult problems or even issues of wider context.

Conclusion

This case study is utilizing escalating difficulty tasks to control (check) students' answers, abilities, reactions and feelings when attempting to deal problems with the use of new interactive technologies.

The case study attempts to compare a traditional group, evaluated to the classic manner by hand writing, to a group evaluated through interactive environments to check the efficiency of both methods. No intent is given to teach or introduce subject students to the principles of algorithmic thinking but merely to estimate the possible efficiency of students' algorithmic working through an interactive environment. The results of the experiment conducted indicated that students dealing with algorithmic problems using an interactive environment appear to perform better to less complex issues in comparison to less complex issues

Future work includes the design and implementation of a new innovative diagnostic screener that will indicate dyscalculia and other difficulties in algorithmic thinking as well as the connection among their features

Acknowledgments. Author would like to thank school director, supervisor and technical assistant for their help as well as students for their participation and excellent cooperation.

References

1. Vlamos, M.P.: Diagnostic Screener on Dyscalculia and algorithmic thinking. In: Workshop on Informatics in Education, WIE 2010, Tripolis (2010)
2. Giannakos, N.M, Vlamos, M.P.: Comparing a well designed webcast with traditional learning. In: Proceedings of the 2010 ACM Conference on Information Technology Education (2010)
3. Nisan, N., Roughgarden, T., Roughgarden, E., Vazirami, V.V.: Algorithmic Game Theory. Cambridge University Press (2007)
4. Howell, D.C.: Chi-Square Test - Analysis of Contingency Tables University of Vermont
5. Piazza, M., Facoetti, A., Noemi Trussardi, A., Berteletti, I., Conte, S., Lucangeli, D., Dehaene, S., Zorzi, M.: Developmental Trajectory of Number Acuity Reveals a Severe Impairment in Developmental Dyscalculia
6. Luculano, T., Tang, J., Hall, C.W.B., Butterworth, B.: Core information processing deficits in developmental dyscalculia and low numeracy (2008), Article first published online: doi: 10.1111/j.1467-7687.2008.00716
7. Landerla, K., Bevana, A., Butterworth, B.: Developmental Dyscalculia and Basic Numerical Capacities: A study of 8–9 year old students. Institute of Cognitive Neuroscience, University College London (2004)
8. Kumar, P., Raja, B.W.D.: Minimising Dyscalculic Problems Through Visual Learning
9. Nagavalli, T., Fidelis, P.J.: Technology for Dyscalculic Children, NCERT-ERIC
10. Mandrekar, J.N., Mandrekar, S.J.: An Introduction to Matching and its Application using SAS® Division of Biostatistics. Mayo Clinic, Rochester
11. Bruck, M.: Word-recognition skills of adults with childhood diagnoses of dyslexia. *Developmental Psychology* 26(3), 439–454 (1990)
12. Folmer, E.: Virtual World Accessibility: Directions for Research, Player-Game Interaction Lab, Department of Computer Science and Engineering, University of Nevada, Reno, Reno, NV 89557-0208
13. Canterbury Christ Church University, <http://www.canterbury.ac.uk/education/quality-in-study-support/docs/4%20-%20Sample%20size%20and%20control%20groups.pdf>
14. Lydersen, S., Fagerland, M.W., Laake, P.: Tutorial in Biostatistics, Recommended tests for association in 2×2 tables. *Statistics in Medicine* (2009), doi: 10.1002/sim.3531
15. Liu, I., Agresti, A.: The Analysis of Ordered Categorical Data: An Overview and a Survey of Recent Developments. *Sociedad de Estadistica e Ivestigacion Operativa Test* 14 (2005)

Appendix - (Statistical Data Analysis)

Interactively Evaluated Group- Task 1 * Traditionally Evaluated Group-Task 1
 Crosstab – Count [T1]

| | | Traditionally Evaluated Group Task 1 | | Total |
|--|---------|--------------------------------------|-------|-------|
| | | Correct | False | |
| Interactively Evaluated Group - Task 1 | Correct | 11 | 9 | 20 |
| | False | 0 | 3 | 3 |
| | Total | 11 | 12 | 23 |

Chi-Square Tests [T2]

| | Value | Df | Asymp. (2-sided) | Sig. (2-sided) | Exact Sig. (2-sided) | Exact Sig. (1-sided) |
|------------------------------------|--------------------|----|------------------|----------------|----------------------|----------------------|
| Pearson Chi-Square | 3.163 ^a | 1 | .075 | | | |
| Continuity Correction ^b | 1.342 | 1 | .247 | | | |
| Likelihood Ratio | 4.316 | 1 | .038 | | | |
| Fisher's Exact Test | | | | .217 | | .124 |
| N of Valid Cases | 23 | | | | | |

a. 2 cells (50.0%) have expected count less than 5. The minimum expected count is 1.43.

b. Computed only for a 2x2 table

Interactively Evaluated Group - Task 2 * Traditionally Evaluated Group-Task 2
 Crosstab – Count [T3]

| | | Traditionally Evaluated Group- Task 2 | | Total |
|--------------------------------------|---------|---------------------------------------|-------|-------|
| | | Correct | False | |
| Interactively Evaluated Group-Task 2 | Correct | 7 | 13 | 20 |
| | False | 0 | 3 | 3 |
| | Total | 7 | 16 | 23 |

Chi-Square Tests [T4]

| | Value | Df | Asymp. (2-sided) | Sig. (2-sided) | Exact Sig. (2-sided) | Exact Sig. (1-sided) |
|------------------------------------|--------------------|----|------------------|----------------|----------------------|----------------------|
| Pearson Chi-Square | 1.509 ^a | 1 | .219 | | | |
| Continuity Correction ^b | .309 | 1 | .578 | | | |
| Likelihood Ratio | 2.369 | 1 | .124 | | | |
| Fisher's Exact Test | | | | .526 | | .316 |
| N of Valid Cases | 23 | | | | | |

| | Value | Df | Asymp. (2-sided) | Sig. (2-sided) | Exact (2-sided) | Sig. (1-sided) | Exact (1-sided) |
|------------------------------------|--------------------|----|---------------------|-------------------|--------------------|-------------------|--------------------|
| Pearson Chi-Square | 1.509 ^a | 1 | .219 | | | | |
| Continuity Correction ^b | .309 | 1 | .578 | | | | |
| Likelihood Ratio | 2.369 | 1 | .124 | | | | |
| Fisher's Exact Test | | | | .526 | | | .316 |
| N of Valid Cases | 23 | | | | | | |

a. 2 cells (50.0%) have expected count less than 5. The minimum expected count is .91.

b. Computed only for a 2x2 table

Interactively Evaluated Group- Task 3 * Traditionally Evaluated Group-Task 3
Crosstab – Count [T5]

| | | Traditionally Evaluated Group-Task 3 | | Total |
|-------------------------------|---------|--------------------------------------|-------|-------|
| | | Correct | False | |
| Interactively Evaluated Group | Correct | 1 | 2 | 3 |
| Task 3 | False | 0 | 20 | 20 |
| Total | | 1 | 22 | 23 |

Chi-Square Tests[T6]

| | Value | Df | Asymp. (2-sided) | Sig. (2-sided) | Exact (2-sided) | Sig. (1-sided) | Exact (1-sided) |
|------------------------------------|--------------------|----|---------------------|-------------------|--------------------|-------------------|--------------------|
| Pearson Chi-Square | 6.970 ^a | 1 | .008 | | | | |
| Continuity Correction ^b | 1.259 | 1 | .262 | | | | |
| Likelihood Ratio | 4.408 | 1 | .036 | | | | |
| Fisher's Exact Test | | | | .130 | | | .130 |
| N of Valid Cases | 23 | | | | | | |

a. 3 cells (75.0%) have expected count less than 5. The minimum expected count is .13.

b. Computed only for a 2x2 table

[T7]

| First Tasks | Group | | | p-value |
|-------------|-----------------------|-------------------------|-------|---------|
| | Traditional Evaluated | Interactively Evaluated | Total | |
| False | 12 | 3 | 15 | |
| Correct | 11 | 20 | 31 | p=0.011 |
| Total | 23 | 23 | 46 | |

[T8]

| | Group | | | p-value |
|-------------|-----------------------|-------------------------|-------|---------|
| Second Task | Traditional Evaluated | Interactively Evaluated | Total | P<0.001 |
| False | 16 | 3 | 19 | |
| Correct | 7 | 20 | 27 | |
| Total | 23 | 23 | 46 | |

[T9]

| | Group | | | p-value |
|------------|-----------------------|-------------------------|-------|---------|
| Third Task | Traditional Evaluated | Interactively Evaluated | Total | P=0.33 |
| False | 21 | 19 | 40 | |
| Correct | 2 | 4 | 6 | |
| Total | 23 | 23 | 46 | |

[T10]

| Correct Aswers in | Group | | Total | P-value |
|-------------------|-----------------------|-------------------------|-------|---------|
| | Traditional Evaluated | Interactively Evaluated | | |
| Total | | | | |
| 0 | 11 | 1 | 12 | p=0.001 |
| 1 | 6 | 4 | 10 | |
| 2 | 4 | 14 | 18 | |
| 3 | 2 | 4 | 6 | |
| Total | 23 | 23 | 46 | |

[T11]

Outcome variable: total, n = 46

| | | | |
|-----------|------------|-----------|--------------------------|
| Covariate | Odds Ratio | Std. Err. | P> z 95% Conf. Interval |
|-----------|------------|-----------|--------------------------|

Group

| | |
|-------------------------|-----------------------------------|
| Traditional evaluated * | 1 1 |
| Interactively evaluated | 8.862 5.591 0.001 2.573 to 30.518 |

* Baseline category

[T12]

| Group Statistics | | | | | |
|-------------------------|-------------------------|----|---------|----------------|-----------------|
| | GROUP | N | Mean | Std. Deviation | Std. Error Mean |
| PRE-TEST | Traditional Evaluated | 23 | 13,6522 | 26,47402 | 5,52021 |
| | Interactively Evaluated | 23 | 18,5652 | 37,77665 | 7,87698 |

[T13]

| | | Levene's t-test for Equality of Means | | | | | | | | |
|---------|-----------------------------|---------------------------------------|------|------|--------|-----------------|-----------------|-----------------------|---|----------|
| | | F | Sig. | t | Df | Sig. (2-tailed) | Mean Difference | Std. Error Difference | 95% Confidence Interval of the Difference | |
| | | | | | | | | | Lower | Upper |
| PRETEST | Equal variances assumed | ,997 | ,323 | ,511 | 44 | ,612 | -4,91304 | 9,61871 | 24,29827 | 14,47218 |
| | Equal variances not assumed | | | ,511 | 39,410 | ,612 | -4,91304 | 9,61871 | 24,36224 | 14,53615 |

[T14]

Outcome variable: total, n = 46

Covariate Odds Ratio Std. Err. P>|z| 95% Conf. Interval

Group

Traditional evaluated * 1

Interactively evaluated 36.103 31.288 <0.001 (6.605 to 197.346)

Pretest

per unit 2.251 0.427 <0.001 (1.552 to 3.264)

* Baseline category

No-Prop-*fast* - A High-Speed Multilayer Neural Network Learning Algorithm: MNIST Benchmark and Eye-Tracking Data Classification

André Frank Krause^{1,2}, Kai Essig^{1,2}, Martina Piefke³, and Thomas Schack^{1,2}

¹ Faculty of Sport Science, Dept. Neurocognition & Action

{andre_frank.krause, kai.essig, thomas.schack}@uni-bielefeld.de

² Cognitive Interaction Technology, Center of Excellence

Bielefeld University, D-33615 Bielefeld, Germany

³ Dept. for Psychology and Psychotherapy,

Witten/Herdecke University, D-58448 Witten, Germany

martina.piefke@uni-wh.de

Abstract. While the No-Prop (no back propagation) algorithm uses the delta rule to train the output layer of a feed-forward network, No-Prop-*fast* employs fast linear regression learning using the Hopf-Wiener solution. Ten times faster learning speeds can be achieved on large datasets like the MNIST benchmark, compared to one of the fastest backpropagation algorithm known. Additionally, the plain feed-forward network No-prop-*fast* can distinguish gaze movements on cartoons with and without text, as well as age-specific attention shifts between text and picture areas with minimal pre-processing.

Continuously learning mobile robots and adaptive intelligent systems require such fast learning algorithms. Almost real-time learning speeds enable lower turn-around cycles in product development and data analysis.

1 Introduction

Bernard Widrow, the inventor of the delta rule [1], recently introduced a simple and effective training algorithm for feed-forward neural networks [2], which he called "No-Prop", short for "no backpropagation". No-Prop avoids the two most common problems of backpropagation (BP) algorithms: First, there is an inherent risk of converging to local minima. Second, the error gradients during learning tend to vanish or explode while the error values are propagated backwards through hidden layers [3]. In summary, BP can be extremely slow for multilayer networks, because "the cost surface is typically non-quadratic, non-convex and high dimensional with many local minima and/or flat regions" [4].

The basic principle of the No-Prop algorithm is to train only a linear output layer while the hidden layers are fixed and randomly initialized [2]. The hidden layers provide rich, random features calculated from the input data, among those the output layer "chooses" suitable combinations. This method is well known for special recurrent networks called Echo State Networks [5]. The delta rule, used for training the output layer, is known to have a single, global minimum and has convergence properties which can be orders of magnitude faster than BP [6]. Here, we present an even faster implementation that avoids the iterative nature of the delta rule, coined No Prop-*fast*.

2 No-Prop-*fast* Method

No-Prop-*fast* replaces online delta rule learning by offline linear regression using the Wiener-Hopf solution to compute the output weight matrix. This results in a further speed-up of the learning process that now uses only few, highly optimized and cache friendly matrix operations.

For the No-Prop-*fast* network training input values are presented to the network and the activation of the hidden layer units are collected into a state collection matrix \mathbf{S} . Then the output layer weight matrix \mathbf{W}^{out} is calculated from the correlation matrix $\mathbf{R} = \mathbf{S}'\mathbf{S}$ and the cross-correlation matrix $\mathbf{P} = \mathbf{S}'\mathbf{D}$ of hidden layer states versus desired output values using the Wiener-Hopf solution:

$$\mathbf{W}^{out} = \mathbf{R}^{-1}\mathbf{P}$$

In Matlab, above equation is solved using the backslash matrix right division operator: $\mathbf{W}_{out} = \mathbf{R} \backslash \mathbf{P}$, which internally performs a Cholesky Decomposition if \mathbf{R} is positive semi-definite. For the given datasets, this was always the case. Cholesky decomposition was reported to be roughly twice as fast as LU decomposition [7].

To evaluate the No-Prop-*fast* method, we tested it with the MNIST [4] database of handwritten digits, widely applied as a standard benchmark for pattern classification algorithms. The MNIST database includes a training set of 60,000 samples (28x28 pixel images) and a test set of 10,000 samples, collected among high-school students and Census Bureau employees. It is freely available from <http://yann.lecun.com>.

Fig. 1 shows the general network structure used to learn the MNIST dataset. The 28x28 gray-scale pixel images were unfolded row-wise, rescaled to the range -1 to 1 and presented directly as input values to the network. The hidden layer weight matrix was initialized using uniformly distributed random numbers (see Fig. 3).

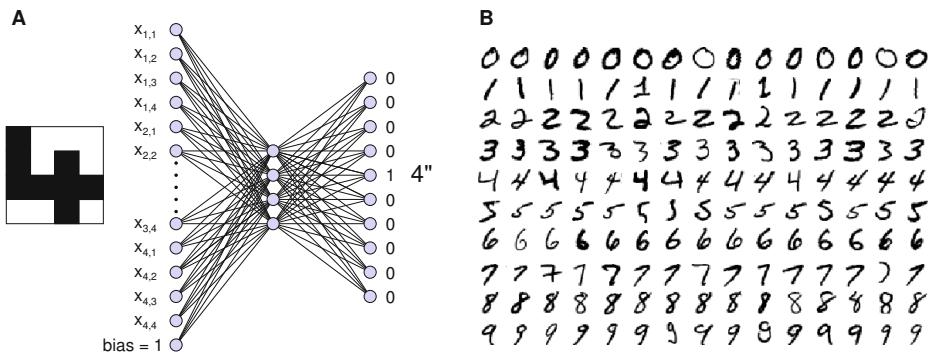


Fig. 1. A) General network structure used for handwritten digit recognition. A plain feed forward network with a large hidden layer is used. For example, a 4x4 pixel input image is unwrapped to the 16 network inputs and gray level values are rescaled to the range -1 to 1. Hidden layer weight values are initialized randomly and only the output layer is trained (see text). Class labels use a "one-out-of-n coding". B) A small sample selection of handwritten digits (28x28 pixels each) from the MNIST database.

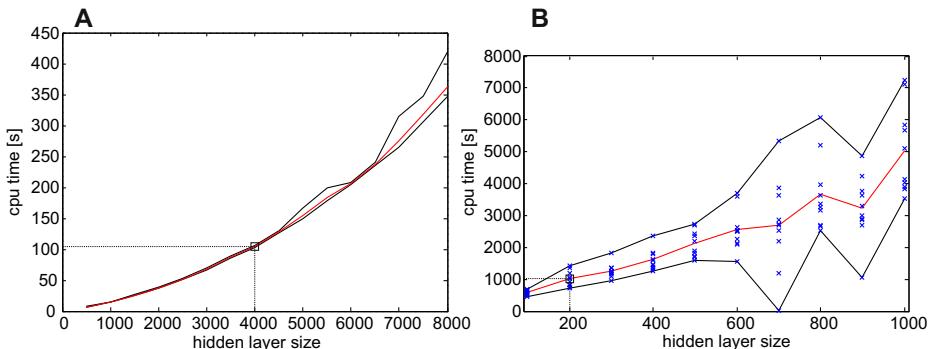


Fig. 2. Training times on a standard Laptop with 1.7 Ghz and 1 MByte second level cache. A) CPU time required to learn the MNIST dataset using No-Prop-*fast*. B) CPU time required with SCG-BP, using the Matlab neural network toolbox function "trainscg". No-Prop-*fast* is approximately ten times faster learning the MNIST dataset. Red curve - mean values, black curves - minimum and maximum values, N=10.

Training times of No-Prop-*fast* were compared with No-Prop and the default pattern classification training algorithm of the Matlab neural network toolbox, scaled conjugate gradient (SCG) BP [8]. The sizes of the hidden layers were chosen to result in a similar classification performance of 97%. The measurements were performed on a Laptop with an AMD A6-3420M CPU at 1.7 GHz with 1024 Kbyte second level cache.

Performance & Learning Speed. With a classification performance of 97.7%, the best No-Prop-*fast* network ($n=8000$ hidden layer units) was almost on-par with the best BP-network that achieved 98.1% (Matlab2012a neural network toolbox, $n=1000$ hidden layer units), see Fig. 5. No-Prop-*fast* with $n=4000$ hidden layer units required approximately 100 seconds of cpu time. For comparison, No-Prop[2] using the online delta rule implemented in Matlab ($n\text{-hidden}=4000$, initial learning rate $lr=0.001$, $lr\text{-decay}=0.7$ per epoch, stop-condition: $\Delta rmse<1e-3$) required on avg. 500 seconds. The Matlab neural network toolbox with the "trainscg" function required almost 1000 seconds to learn the dataset with similar performance, see Fig. 2. Hence, for the MNIST dataset, No-Prop-*fast* is approximately five times faster than No-Prop and ten times faster than the Matlab implementation of one of the fastest BP-algorithms.

Some advanced classification algorithms with sophisticated preprocessing gain classification rates of over 99.6% [9][10]. Yet, plain, but big feed-forward networks with large hidden layers can achieve excellent classification results using standard online BP [11], but would require extremely long training times of several hours or days on a standard, year 2012 personal computer. A fast graphics card GPU implementation of the classic BP algorithm was presented in [11] to achieve acceptable training times (2 hours for an error below 1%). No-Prop-*fast* - without any preprocessing - does not reach these high classification rates, yet is much easier to implement and incredibly fast.

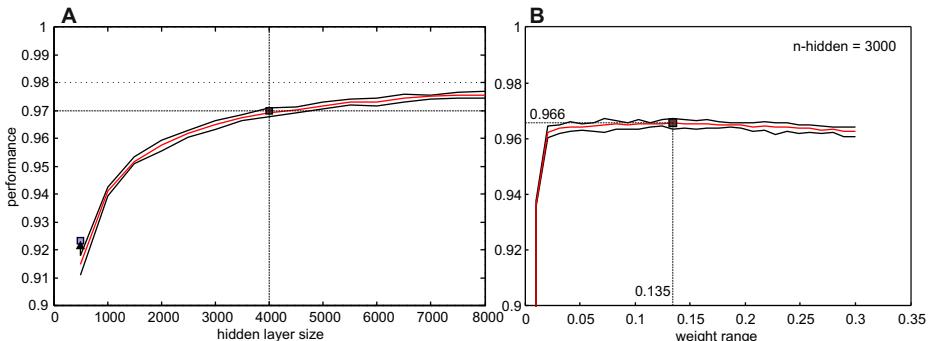


Fig. 3. The two relevant parameters of a No-PROP network are the hidden layer size and the weight range of the hidden layer. A) 1. Performance depending on hidden layer size. 2. Optimization of hidden layer weights with an evolutionary algorithm. For a hidden layer size of $n=500$ units, a small improvement in performance could be achieved. For $n=400$ hidden layer units, performance does not increase. $N=10$. B) Performance depends slightly on the hidden layer weight range. For a network with 3000 hidden layer units, performance peaks at 0.966. $N=10$ repetitions with uniformly distributed random hidden layer weight values.

Parameter Tuning. A No-Prop network has several parameters to tune: 1. hidden layer size, 2. hidden layer weight range, 3. hidden layer sparsity and 4. bias input scaling. The sparsity, that is the percentage of zero weight values in the hidden layer, had no positive effect on classification performance. The performance saturates as the sparsity approaches zero. For the MNIST dataset, the bias input was also not useful, because the borders of the training images are constant and can serve as a bias input to the network.

The remaining two parameters have a significant impact on the performance. For the MNIST database, the performance appears to saturate with increasing hidden layer sizes (Fig. 3). Because of memory limits, the hidden layer size could not be increased beyond 8000 units. For the eye-tracking dataset, an optimal number of hidden layer units exists (Fig. 7). Further, the hidden layer weight range slightly influences the classification performance, see Fig. 3.

Network Pruning & Optimization. Among the high number of hidden layer units, some units might not contribute to the classification performance. A hidden layer unit connected only with small weight values to the output layer might be a candidate for pruning. For each hidden layer unit, an "absolute weight sum" was calculated. This is the sum of rectified weight values to all output units. Fig. 4A shows the distribution of this "absolute weight sum" for the best performing network ($n=8000$ hidden layer units). There are no units with a sum below 0.05, and most units have a sum around 0.125. Fig. 4B shows how the performance drops if all hidden layer units with small sums are successively removed. Above 0.1, the performance drops rapidly. Without considerable loss in performance, only 10% of hidden layer units could be removed. Hence, network pruning is ineffective in case of the MNIST database, and might be ineffective on other

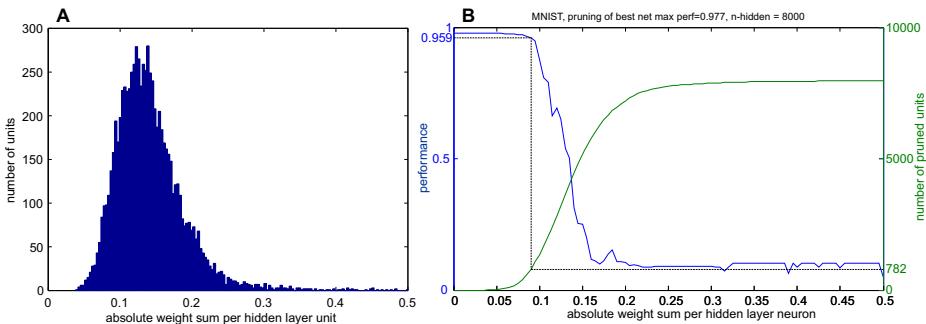


Fig. 4. Network pruning. A) Distribution of the "absolute weight sum" (sum of all rectified weight values of a hidden layer unit to all output units). Shown is the best overall network with a n=8000 units in the hidden layer. B) Performance starts to drop for absolute weight sums above 0.1, resulting in only 10% of hidden layer units prune-able.

| Neural Nets | performance | publication |
|--|-------------|--|
| 2-layer NN, 300 hidden units | 4,70 | LeCun et al. 1998 |
| 2-layer NN, 1000 hidden units | 4,50 | LeCun et al. 1998 |
| 3-layer NN, 300+100 hidden units | 3,05 | LeCun et al. 1998 |
| 3-layer NN, 500+150 hidden units | 2,95 | LeCun et al. 1998 |
| NO-PROP, 8000 hidden units | 2,30 | |
| 2-layer NN, 1000 hidden units, scaled conjugate gradients backprop | 1,91 | |
| 6-layer NN 784-2500-2000-1500-1000-500-10 (on GPU) [elastic distortions] | 0,35 | Ciresan et al. Neural Computation 10, 2010 |

Fig. 5. Comparison of different MNIST benchmark results. Benchmark results for different neural networks with no special pre-processing of the input images were selected from the website <http://yann.lecun.com/exdb/mnist>. No-PROP with a large hidden layer performs slightly better than a 3 layer neural network trained with classic back-propagation.

data-sets, too. In contrast to [12], where an evolutionary optimization of the dynamic reservoir of an echo state network greatly improved performance, optimizing the hidden layer weight matrix had little to no effect, see Fig. 3A.

3 Eye Tracking and Neural Nets

The term eye tracking denotes the process of monitoring and recording the participants' gaze positions when they look at 2D or 3D stimuli, for example presented on a computer monitor. Researchers are interested in exact *gaze positions* measured in 2D or 3D coordinates and their temporal course, i.e., *spatial-temporal scan paths*. The analysis of eye movements yields valuable insights into the cognitive processes underlying information processing ("eyes are a window to the mind") [13].

Neural nets have been applied to the field of eye tracking to reliably estimate the 3D gaze point from a participant's binocular eye-position data [14], based on the observation that specialized, individually calibrated artificial neural networks could be used to significantly reduce the error in gaze-position measurement. Further application fields are the real time tracking of human gaze direction to improve human-machine interaction by using low-cost hardware, like webcams and the modeling of human eye

movements [15]. Other lines of research apply neural nets for the classification of eye movements. In [16] a self-organizing map formed 2 areas where dyslexic patients were clustered - showing that the trained map can be used for classification. [17] recorded the eye gaze point and pupil size of five participants during an overnight driving simulation task. The gaze data was pre-processed by several signal processing routines, resulting in 80 spectral power density values serving as inputs to a LVQ net. [18] applied neural networks to validate if the way a person reads influences the way he understands information and proposed a novel method of detecting the level of engagement in reading based on gaze-patterns. [19] applied their hybrid fuzzy signatures with Levenberg-Marquardt optimization method for recognizing different eye-gaze patterns while viewing faces or text documents.

All these approaches have in common that the gaze data needs a high amount of pre-processing before it can serve as input to the neural networks. In a previous study, we showed that raw eye scan paths can be classified with a plain but large BP net[20]. Here, we will show that our No-Prop-*fast* method can also classify gaze data without sophisticated pre-processing. Fast learning techniques can enable artificial systems to react adaptively in a broad variety of application fields. For example, intelligent machines can detect age specific problems and actively adapt user interaction. Before we describe the classification results of the No-Prop-*fast* method, we first motivate and sketch our eye-tracking study on a typical daily routine task.

Cartoon Study: Motivation, Method and Experimental Results. The perception of text and images is an important day-by-day task. In newspapers, advertisements or internet pages, we always have to relate text and picture information to understand the content. This process requires a certain amount of cognitive effort, whereby, besides memory processes, empathy and other competences, particularly executive functions of the working memory are important. In the scientific literature the deficits in different capacities, particularly in the executive functions, with ongoing age are well documented [21]. There are many eye-tracking studies on age-based differences on perception and text processing (for an overview see [22]) and images (for example, [23]), but only relatively few on the combination of both, particularly on cartoons. Therefore, we recorded the eye movements of 75 German native speakers (41 females and 34 males) with an age between 19 to 60 while they read silently cartoons with and without text (see Figure 6). Group 1 were young people (15 females and 10 males) from 18 to 30 years. Group 2 (10 females and 15 males) consisted of middle aged people from 31 to 45 years. Finally, group 3 were older people (16 females and 9 males) from 46 to 60 years. For the cartoons, the screen was subdivided into 4 boxes, similar in size to the original cartoons (see Figure 6), in order to prevent influences on the reading behavior resulting from different positioning. For the cartoons without and with text we selected examples from the father and son [24] and Calvin and Hobbes [25] comics, respectively. For the later ones, we standardized the font without changing the spatial positioning of figures, text and spaces. Each category consisted of 25 different stimuli which were presented in a randomized order to prevent learning or priming effects. Participants eye movements were recorded using a remote binocular Eyegaze Analysis System with a sampling rate of 120 Hertz and a gaze position accuracy of 0.4°. The task of the participants was to read the cartoons and to press a button to signal that they had understood

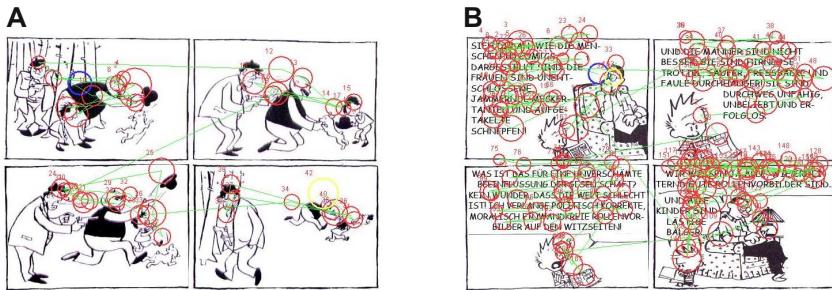


Fig. 6. Two example figures from the eye-tracking dataset. Green: scan path. Red: fixations. The diameter indicates the fixation duration. A) Cartoon without text. B) Cartoon with text. A clear difference is the higher number of fixations in the cartoon with text.

the content. For each trial the time between stimulus appearance and button press was recorded. When participants did not react within 60 seconds after stimulus presentation (a threshold calculated on the basis of a pre-study), the trial was automatically stopped and the stimulus disappeared. The analysis of the eye movements revealed significant differences between both cartoons and the three groups. The average viewing time for Father and Son and Calvin and Hobbs cartoons was 9.8 and 28.3 seconds, respectively. Old people needed more time to understand the cartoons and showed more regressions. These findings are in accordance to [26] and show that it is more difficult for them to integrate the text and picture information. These results support a gradual decrease of cognitive abilities with increasing age, because young people show the lowest number of regressions. We also found significant differences in the number of fixations and fixation durations, particularly between old people and the other groups, but not between group 2 and 1.

4 Neural Network Classification Results

First, we investigated if the No-Prop-*fast* network is able to distinguish the differences in the overall scan paths for cartoons with and without text that have been found in the empirical study. The training data fed into the net had three data channels: the x- and the y-component of each fixation point from participants' scan path with the corresponding fixation duration f . Shorter scan paths were filled with zeros to ensure that the training data was normalized to the same length. Each channel was rescaled to a standard deviation of 1 and zero mean. The data was then spatially unfolded to the input vector $\mathbf{u} = (x_0, y_0, f_0, x_1, y_1, f_1 \dots x_n, y_n, f_n)$.

A feed-forward network with 510 hidden units, trained with No-Prop-*fast*, was able to distinguish eye-tracking scan paths recorded while watching a cartoon with or without text (see Figure 6). With ten-fold cross-validation, on average 88.9% of all samples were correctly classified ($n=10$ repetitions).

To analyze the contribution of the individual components of the eye tracking data, a network with 510 hidden units was trained ($N=50$ repetitions) separately on each

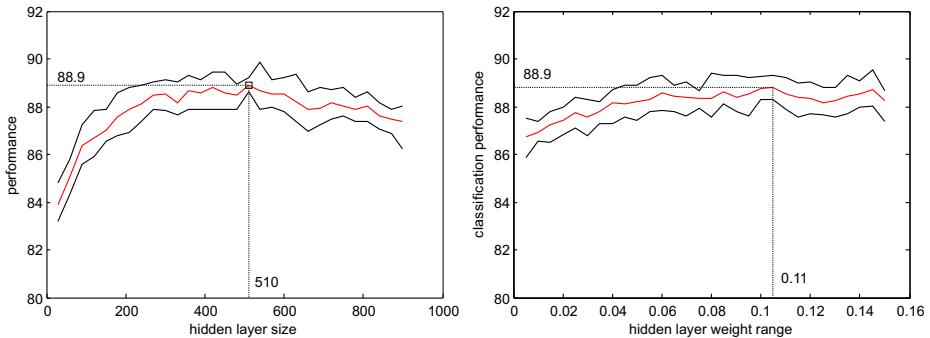


Fig. 7. No-Prop classification performance depending on the hidden layer size and weightrange. Eye-tracking data recorded from cartoon pictures with- and without text, see Fig.6. N=10.

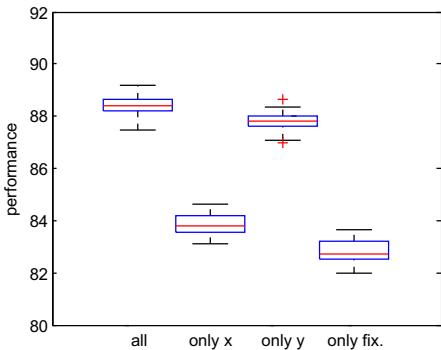


Fig. 8. Individual contribution of the three fixation parameters to the classification performance on the cartoon vs. cartoon with text task. A network with 510 hidden layer units was used, N=50 training repetitions. The input to the network was either the x-, the y-component or the fixation duration, only.

data channel. Fig. 8 shows that training the networks given each channel alone did not reach the performance of the network that was fed with all three data channels. The figure also indicates that the vertical component of the scan path contributes the most to classification success.

In a second step, we tested if the No-Prop-*fast* network with 100 hidden units is able to distinguish the age specific differences in the number of regressions between text and pictures areas in the Calvin and Hobbs cartoons (see Figure 6B). Therefore, we provided a second training data set with region specific fixation information for each text and image area appearing in the cartoon (such as overall and average number of fixations, mean and cumulative fixation duration). Using ten-fold cross-validation, an average classification performance of 83% was achieved (n=10 repetitions).

The speed advantage of No-prop-*fast* on the cartoon vs. cartoon+text dataset (3000 samples) compared to MNIST (60,000) was even larger. Learning the dataset took on average 420 ms (510 hidden units), while SCG-BP (90 hidden units) using Matlab2012a took on average 31 seconds (both methods 88% avg. perf.). Hence, in this case No-prop-*fast* is 74 times faster.

5 Conclusions

We proposed a faster implementation of the No-Prop algorithm [2], coined No-Prop-*fast* that avoids the iterative nature of the delta rule. We showed that No-Prop-*fast*, compared to the modern SCG-BP algorithm, does not only provide similar classification rates on the MNIST database, but is also **ten times faster** on large data sets. The algorithm is also able to classify differences in the overall scan paths over different cartoons and age specific attention shifts between text and picture areas. On average sized data-sets, No-Prop-*fast* provides near real-time training speeds. The presented eye-tracking dataset with 3000 samples can be learnt in 420 ms, while the substantially larger dataset of the MNIST database requires only 100 seconds on a common 1.7 GHz laptop. On the smaller eye-tracking dataset we found a 74x speedup. There is no risk of stalled learning due to local minima and the algorithm has only two parameters to tune. Further, spread of performance values and cpu-time is minimal, which is good for requirement engineering. The only apparent drawback is that the whole dataset and the collected network-states during training have to fit into the working memory.

Fast, almost real-time network training results in lower product development turn-around cycles and opens new paths, ranging from interactive scientific data analysis to industrial / commercial applications of neural networks, such as continuously learning, user adaptable terminals and intelligent mobile robots.

Acknowledgment. This research was supported by the DFG CoE 277: Cognitive Interaction Technology (CITEC). We wish to thank Gesa Sophie Lichtner for her assistance with the experiments and the data analysis.

References

1. Widrow, B., Hoff Jr., M.E.: Adaptive switching circuits. *IRE WESCON Convention Record* 4, 96–104 (1960)
2. Widrow, B., Greenblatt, A., Kim, Y., Park, D.: The no-prop algorithm: A new learning algorithm for multilayer neural networks. *Neural Networks* 37, 182–188 (2013)
3. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In: Kremer, S.C., Kolen, J.F. (eds.) *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press (2001)
4. LeCun, Y.A., Bottou, L., Orr, G.B., Müller, K.-R.: Efficient backProp. In: Orr, G.B., Müller, K.-R. (eds.) *NIPS-WS 1996. LNCS*, vol. 1524, pp. 9–50. Springer, Heidelberg (1998)
5. Jäger, H., Haas, H.: Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science* 304, 78–80 (2004)
6. Wang, Z.Q., Manry, M., Schiano, J.: Lms learning algorithms: misconceptions and new results on convergence. *IEEE Transactions on Neural Networks* 11(1), 47–56 (2000)
7. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical Recipes in C: The Art of Scientific Computing*, 2nd edn. Cambridge University Press (1992)
8. Møller, M.F.: A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* 6(4), 525–533 (1993)
9. Ranzato, M., Poultney, C., Chopra, S., LeCun, Y.: Efficient learning of sparse representations with an energy-based model. In: Platt, J., et al. (eds.) *Advances in Neural Information Processing Systems (NIPS 2006)*, vol. 19. MIT Press (2006)

10. Ciresan, D.C., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. CoRR abs/1202.2745 (2012)
11. Ciresan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Deep big simple neural nets excel on handwritten digit recognition. CoRR (2010)
12. Krause, A.F., Dürr, V., Bläsing, B., Schack, T.: Evolutionary optimization of echo state networks: multiple motor pattern learning. In: Madani, K. (ed.) 6th ANNIIP 2010, Funchal, Madeira, pp. 63–71 (June 2010)
13. Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., van de Weijer, J.: Eye tracking - A comprehensive guide to methods and measures. Oxford University Press, New York (2011)
14. Essig, K., Pomplun, M., Ritter, H.: A neural network for 3d gaze recording with binocular eye trackers. International Journal of Parallel, Emergent and Distributed Systems 21(2), 79–95 (2006)
15. Zhang, Y., Zhao, X., Fu, H., Liang, Z., Chi, Z., Zhao, X., Feng, D.: A time delay neural network model for simulating eye gaze data. Journal of Experimental & Theoretical Artificial Intelligence 23(1), 11–126 (2011)
16. Macaš, M., Lhotská, L., Novák, D.: Bio-inspired methods for analysis and classification of reading eye movements of dyslexic children. Technical report, University in Prague, Algarve, Portugal, October 3-5 (2005)
17. Sommer, D., Hink, T., Golz, M.: Application of learning vector quantization to detect drivers dozing-off. In: European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems, pp. 119–123 (2002)
18. Vo, T., Mendis, B.S.U., Gedeon, T.: Gaze pattern and reading comprehension. In: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) ICONIP 2010, Part II. LNCS, vol. 6444, pp. 124–131. Springer, Heidelberg (2010)
19. Zhu, D., Mendis, B.S.U., Gedeon, T., Asthana, A., Goecke, R.: A hybrid fuzzy approach for human eye gaze pattern recognition. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) ICONIP 2008, Part II. LNCS, vol. 5507, pp. 655–662. Springer, Heidelberg (2009)
20. Krause, A.F., Essig, K., Essig-Shih, L.-Y., Schack, T.: Classifying the differences in gaze patterns of alphabetic and logographic L1 readers – A neural network approach. In: Iliadis, L., Jayne, C. (eds.) EANN/AIAI 2011, Part I. IFIP AICT, vol. 363, pp. 78–83. Springer, Heidelberg (2011)
21. Buckner, R.: Memory and executive functioning in aging and ad: Multiple factors that cause decline and reserve factors that compensate. Neuron 44, 195–208 (2004)
22. Rayner, K.: Eye movements in reading and information processing: 20 years of research. Psychological Bulletin 124, 372–422 (1998)
23. Henderson, J., Ferreira, F.: Scene perception for psycholinguists. In: The Interface of Language, Vision and Action: Eye Movements and the Visual World, pp. 1–58. Psychology Press, New York (2004)
24. Plauen, E.: Vater und Sohn (3 Bde.). Südverlag Konstanz, Konstanz (2000)
25. Watterson, B.: Calvin und Hobbes: Der Jubelband: 10 Jahre: 10 Jahre Jubel Buch. Carlsen Verlag, Hamburg (2008)
26. Kramer, A., Hahn, S., Irwin, D., Theeuwes, J.: Age differences in the control of looking behavior: Do you know where your eyes have been? Psychological Science 11, 210–217 (2000)

CPL Criterion Functions and Learning Algorithms Linked to the Linear Separability Concept

Leon Bobrowski

Faculty of Computer Science, Białystok University of Technology
1.bobrowski@pb.edu.pl

Abstract. Linear separability of learning sets is a basic concept of neural networks theory. Exploration of the linear separability can be based on the minimization of the perceptron criterion function. Modification of the perceptron criterion function have been proposed recently aimed at feature selection problem. The modified criterion functions allows, among others, for discovering minimal feature subset that assure linear separability. Learning algorithm linked to the modified function is formulated in the paper.

Keywords: learning sets, linear separability, perceptron criterion function, modified criterion function, learning algorithms.

1 Introduction

The convergence of the *Perceptron* error correction algorithm used in the theory of neural networks depends on the linear separability of learning sets [1], [2]. The term *linear separability* refers to the possibility of two learning sets separation by a hyperplane in a given feature space [3], [4].

The perceptron criterion function linked to the error correction algorithm has been defined on feature vectors that belong to learning sets [3], [5]. It has been proved that the minimal value of the perceptron criterion function is equal to zero if and only if the learning sets are linearly separable. More generally, the level of linear separability can be evaluated through the minimization of the perceptron criterion function [5].

Modifications of the perceptron criterion function have been proposed recently that include introduction of the costs of individual features. The modified criterion functions allow, among others, for discovering minimal feature subsets that still assure linear separability. Such property is used in the *relaxed linear separability (RLS)* method of feature (gene) subset selection [6].

The perceptron criterion function and their modifications belong to the family of convex and piecewise linear (*CPL*) criterion functions [5]. The basis exchange algorithms, which are similar to the linear programming, allow to find efficiently the minimal value of the *CPL* criterion functions [7] even in the case of numerous or multidimensional data sets.

The leaning algorithms linked to the perceptron criterion function and their *CPL* modifications through the Robbins-Monro procedure of stochastic approximation have been formulated in this work [8].

2 Linear Separability of Learning Sets

We are assuming that m objects O_j ($j = 1, \dots, m$) contained in a database are represented in a standardised manner as feature vectors $\mathbf{x}_j = [x_{j1}, \dots, x_{jn}]^T$ or as points in the n -dimensional feature space F ($\mathbf{x}_j \in F$). The component x_{ji} of the vector \mathbf{x}_j is the numerical value of the i -th feature x_i ($i = 1, \dots, n$) of the j -th object O_j . The component x_{ji} can be treated, for example, as the numerical result of particular examination of the j -th object O_j . Components x_{ji} of feature vectors \mathbf{x}_j can be binary ($x_{ji} \in \{0,1\}$), or a real number ($x_{ji} \in R^1$).

Let us take into consideration two disjoined learning sets: the *positive set* G^+ and the *negative set* G^- containing m^+ and m^- feature vectors \mathbf{x}_j ($G^+ \cap G^- = \emptyset$), adequately:

$$G^+ = \{\mathbf{x}_j: j \in J^+\} \text{ and } G^- = \{\mathbf{x}_j: j \in J^-\} \quad (1)$$

where J^+ and J^- are the sets of indices j .

We are considering the possibility of the learning sets G^+ and G^- separation by the hyperplane $H(\mathbf{w}, \theta)$ in a feature space:

$$H(\mathbf{w}, \theta) = \{\mathbf{x}: \mathbf{w}^T \mathbf{x} = \theta\} \quad (2)$$

where $\mathbf{w} \in R^n$ is the *weight vector*, $\theta \in R^1$ is the *threshold*, and $\mathbf{w}^T \mathbf{x}$ is the inner product.

Definition 1: The learning sets G^+ and G^- are *linearly separable* in the feature space F , if and only if there exists such a *weight vector* $\mathbf{w} = [w_1, \dots, w_n] \in R^n$, and a threshold $\theta \in R^1$ that these sets can be separated by the hyperplane $H(\mathbf{w}, \theta)$:

$$\begin{aligned} (\exists \mathbf{w}, \theta) \quad (\forall \mathbf{x}_j \in G^+): \quad \mathbf{w}^T \mathbf{x}_j &> \theta \text{ and} \\ (\forall \mathbf{x}_j \in G^-): \quad \mathbf{w}^T \mathbf{x}_j &< \theta \end{aligned} \quad (3)$$

If the above relation is fulfilled, then all the feature vectors \mathbf{x}_j from the set G^+ are located on the positive side of the hyperplane $H(\mathbf{w}, \theta)$ (3) and all the vectors \mathbf{x}_j from the set G^- are located on the negative side of this hyperplane. Let us introduce the *augmented* feature vectors \mathbf{y}_j [4]:

$$\begin{aligned} (\forall \mathbf{x}_j \in G^+) \quad \mathbf{y}_j &= [\mathbf{x}_j^T, 1]^T \text{ and} \\ (\forall \mathbf{x}_j \in G^-) \quad \mathbf{y}_j &= -[\mathbf{x}_j^T, 1] \end{aligned} \quad (4)$$

and the *augmented* weight vector $\mathbf{v} = [\mathbf{w}^T, -\theta]^T$.

The linear separability inequalities (3) with the augmented vectors \mathbf{y}_j take the following form:

$$(\exists \mathbf{v}) \quad (\forall \mathbf{y}_j \in G^+ \cup G^-) \quad \mathbf{v}^T \mathbf{y}_j > 0 \quad (5)$$

The above inequalities can be represented also as [5]:

$$(\exists \mathbf{v}) \quad (\forall \mathbf{y}_j \in G^+ \cup G^-) \quad \mathbf{v}^T \mathbf{y}_j \geq 1 \quad (6)$$

These inequalities have been used in the definition of the perceptron penalty functions.

3 The Perceptron Penalty Functions and Criterion Function

Let us define the penalty function $\varphi_j(\mathbf{v})$ for each feature vector \mathbf{x}_j (1) (augmented vector \mathbf{y}_j (4)):

$$(\forall \mathbf{x}_j \in G^+ \cup G^-) \quad \varphi_j(\mathbf{v}) = \begin{cases} 1 - \mathbf{y}_j^T \mathbf{v} & \text{if } \mathbf{y}_j^T \mathbf{v} < 1 \\ 0 & \text{if } \mathbf{y}_j^T \mathbf{v} \geq 1 \end{cases} \quad (7)$$

The convex and piecewise-linear (*CPL*) penalty function $\varphi_j(\mathbf{v})$ are aimed at reinforcement of the desired inequalities (6) (Fig. 1).

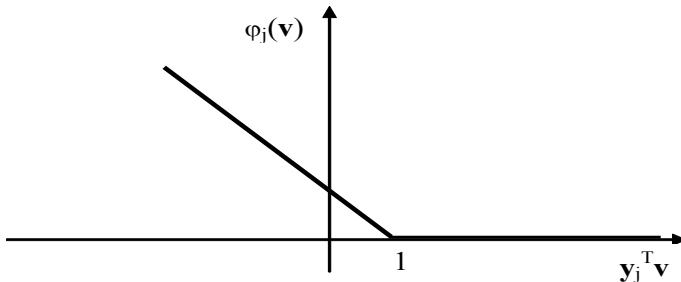


Fig. 1. The perceptron penalty functions $\varphi_j(\mathbf{v})$ (7)

The *perceptron* criterion function $\Phi(\mathbf{v})$ is defined as the weighted sum of the penalty functions $\varphi_j(\mathbf{v})$ (7) [4], [5]:

$$\Phi(\mathbf{v}) = \sum_{j \in \{1, \dots, m\}} \alpha_j \varphi_j(\mathbf{v}) \quad (8)$$

where the positive parameters α_j ($\alpha_j > 0$) determine *prices* of particular feature vectors \mathbf{x}_j . The default (standard) values of the parameters α_j are specified below:

$$\begin{aligned} &\text{if } \mathbf{x}_j \in G^+, \text{ then } \alpha_j = 1/(2m^+), \text{ and} \\ &\text{if } \mathbf{x}_j \in G^-, \text{ then } \alpha_j = 1/(2m^-) \end{aligned} \quad (9)$$

The non-negative criterion function $\Phi(\mathbf{v})$ (8) is convex and piecewise-linear (*CPL*) as the sum with the positive coefficients α_j of the *CPL* penalty functions $\varphi_j(\mathbf{v})$ (9). The optimal vector \mathbf{v}^* constitutes the minimal value Φ^* of the *CPL* criterion function $\Phi(\mathbf{v})$ (8) defined on elements \mathbf{x}_j of the data sets G^+ and G^- (1):

$$(\exists \mathbf{v}^*) \ (\forall \mathbf{v} \in R^{n+1}) \ \Phi(\mathbf{v}) \geq \Phi(\mathbf{v}^*) = \Phi^* \geq 0 \quad (10)$$

The *basis exchange algorithms*, which are similar to the linear programming, allow to find efficiently the minimal value Φ^* of the convex and piecewise linear (*CPL*) criterion function $\Phi(\mathbf{v})$ (10) and the optimal vector of parameters \mathbf{v}^* (12) [7]. The below theorems has been proved [5]:

Theorem 1: The minimal value $\Phi^* = \Phi(\mathbf{v}^*)$ (10) of the perceptron criterion function $\Phi(\mathbf{v})$ (8) is equal to zero ($\Phi^* = 0$) if and only if the learning sets G^+ and G^- (1) are linearly separable (5).

It can be proved also [5], that the minimal value $\Phi^* = \Phi(\mathbf{v}^*)$ (10) of the criterion function $\Phi(\mathbf{v})$ (8) with the parameters α_j specified by (9) is near to one ($\Phi^* \approx 1.0$) if the learning sets G^+ and G^- (1) are completely overlapped. The below standardization has been introduced on this basis:

$$0 \leq \Phi(\mathbf{v}^*) \leq 1.0 \quad (11)$$

One of important properties of the minimal value Φ^* (10) is their *invariancy* due to nonsingular affine transformations [5]:

$$\text{if } (\forall \mathbf{x}_j \in G^+ \cup G^-) \quad \mathbf{z}_j = \mathbf{A} \mathbf{x}_j + \mathbf{b} \text{ where } \mathbf{A}^{-1} \text{ exists, then } \Phi_z^* = \Phi_x^* \quad (12)$$

where \mathbf{A} is a nonsingular matrix, \mathbf{b} is a vector, and Φ_z^* is the minimal value of the perceptron criterion function $\Phi_z(\mathbf{v})$ defined (8) on the transformed vectors \mathbf{z}_j .

4 The Modified Criterion Function $\Psi_\lambda(\mathbf{v})$

The modified criterion function $\Psi_\lambda(\mathbf{v})$ contains the perceptron function $\Phi(\mathbf{v})$ (8) and extra *CPL* penalty functions in the form of absolute values of components $|w_i|$ of vector \mathbf{w} (6) multiplied by the *costs* γ_i ($\gamma_i > 0$) of particular features x_i [6]:

$$\Psi_\lambda(\mathbf{v}) = \Phi(\mathbf{v}) + \lambda \sum_{i=1, \dots, n} \gamma_i |w_i| \quad (13)$$

where $\lambda \geq 0$ is the *cost level*. The values of the parameters γ_i may be equal to one:

$$(\forall i = 1, \dots, n) \quad \gamma_i = 1.0 \quad (14)$$

The modified criterion function $\Psi_\lambda(\mathbf{v})$ (14) is used in the *relaxed linear separability (RLS)* method of feature subset selection [6]. The regularization component $\lambda \sum \gamma_i |w_i|$ used in the function $\Psi_\lambda(\mathbf{v})$ (13) is similar to that used in the *Lasso* method developed for the model selection in the framework of the regression analysis [9]. The main difference between the *Lasso* and the *RLS* methods is in the types of the basic criterion functions. The basic criterion function typically used in the *Lasso* method is the *residual sum of squares*, whereas the perceptron criterion function $\Phi(\mathbf{v})$ (8) is the basic function used in the *RLS* method. This difference effects among others the computational techniques used to minimize the criterion function. The criterion function $\Psi_\lambda(\mathbf{v})$ (13), similarly to the function $\Phi(\mathbf{v})$ (8), is convex and piecewise-linear (*CPL*). The basis exchange algorithm allows to find their minimum:

$$(\exists \mathbf{v}_\lambda^*) \quad (\forall \mathbf{v}) \quad \Psi_\lambda(\mathbf{v}) \geq \Psi_\lambda(\mathbf{v}_\lambda^*) \quad (15)$$

where (6):

$$\mathbf{v}_\lambda^* = [(\mathbf{w}_\lambda^*)^T, -\theta_\lambda^*]^T = [w_{\lambda 1}^*, \dots, w_{\lambda n}^*, -\theta_\lambda^*]^T \quad (16)$$

The optimal parameters $w_{\lambda i}^*$ are used in the below features *reduction rule* [6]:

$$(w_{\lambda i}^* = 0) \Rightarrow (\text{the feature } x_i \text{ is omitted}) \quad (17)$$

The reduction of feature x_i which is related to the weight $w_{\lambda i}^*$ equal to zero ($w_{\lambda i}^* = 0$) does not change the value of the inner product $(\mathbf{w}_\lambda^*)^T \mathbf{x}_j$. This means that the positive location $(\mathbf{w}_\lambda^*)^T \mathbf{x}_j > \theta_\lambda^*$ or the negative location $(\mathbf{w}_\lambda^*)^T \mathbf{x}_j < \theta_\lambda^*$ of all feature vectors \mathbf{x}_j in respect to the optimal hyperplane $H(\mathbf{w}_\lambda^*, \theta_\lambda^*)$ (3) remains unchanged.

5 Properties of the Optimal Parameter Vectors

Each of the augmented feature vectors \mathbf{y}_j (4) defines the below hyperplane h_j^1 in the $(n + 1)$ -dimensional parameter space:

$$(\forall \mathbf{y}_j \in G^+ \cup G^-) \quad h_j^1 = \{\mathbf{v}: \mathbf{y}_j^T \mathbf{v} = 1\} \quad (18)$$

Similarly, each of the unit vectors \mathbf{e}_i defines the hyperplane h_i^0 in the parameter space:

$$(\forall i \in \{1, \dots, n\}) \quad h_i^0 = \{\mathbf{v}: \mathbf{e}_i^T \mathbf{v} = 0\} \quad (19)$$

Let us consider a set S_k of $(n + 1)$ linearly independent feature vectors \mathbf{y}_j ($j \in J_k$) and unit vectors \mathbf{e}_i ($i \in I_k$). Vectors \mathbf{y}_j and \mathbf{e}_i belonging to the set S_k define the vertex \mathbf{v}_k in the $(n + 1)$ -dimensional parameter space through the below system of $(n + 1)$ linear equations [5]:

$$\begin{aligned} (\forall j \in J_k) \quad & \mathbf{v}_k^T \mathbf{y}_j = 1 \\ \text{and } (\forall i \in I_k) \quad & \mathbf{v}_k^T \mathbf{e}_i = 0 \end{aligned} \quad (20)$$

where J_k is the k -th set of indices j of the base feature vectors \mathbf{y}_j and I_k is the k -th set of indices i of the base unit vectors \mathbf{e}_i in the $(n + 1)$ -dimensional parameter space.

The linear equations (20) can be represented in the matrix form, as:

$$\mathbf{B}_k^T \mathbf{v}_k = \mathbf{1}_k \quad (21)$$

where \mathbf{B}_k is the matrix with columns defined by the base vectors \mathbf{y}_j or \mathbf{e}_i , and $\mathbf{1}_k$ is the vector with components equal to 1 or 0 in accordance with the sets J_k and I_k (20).

The nonsingular matrix \mathbf{B}_k is called the k -th *basis* of the parameter space and the vertex \mathbf{v}_k linked to this basis can be computed by the below matrix equation:

$$\mathbf{v}_k = (\mathbf{B}_k^T)^{-1} \mathbf{1}_k \quad (22)$$

Remark 5: The k -th *basis* (nonsingular matrix) \mathbf{B}_k linked to the vertex \mathbf{v}_k (22) contains m_k ($m_k \leq n+1$) augmented feature vectors \mathbf{y}_j (4) ($j \in J_k$) and $(n+1 - m_k)$ unit vectors \mathbf{e}_i ($i \in I_k$) (21). In this case, the vertex \mathbf{v}_k is located at the intersection of m_k base hyperplanes h_j^1 (18) and $n + 1 - m_k$ base unit hyperplanes h_j^0 (19).

If the learning sets G^+ and G^- (1) are linearly separable (3), then the minimal value $\Phi^* = \Phi(\mathbf{v}^*)$ (10) of the perceptron criterion function $\Phi(\mathbf{v})$ (8) is equal to zero. This means that all the penalty function $\varphi_j(\mathbf{v})$ (7) have to be equal to zero in the optimal point \mathbf{v}^* (10).

$$(\forall j = 1, \dots, m) \quad \varphi_j(\mathbf{v}^*) = 0 \quad (23)$$

All the penalty functions $\varphi_j(\mathbf{v})$ (7) are equal to zero in the point \mathbf{v} if this point is located on the positive side of all the hyperplanes h_j^1 (18):

$$(\forall j = 1, \dots, m) \quad \mathbf{y}_j^T \mathbf{v} \geq 1 \quad (24)$$

Let us define the set \mathbf{R} of such parameters \mathbf{v} , that fulfils the above conditions:

$$\mathbf{R} = \{\mathbf{v}: (\forall j = 1, \dots, m) \quad \mathbf{y}_j^T \mathbf{v} \geq 1\} \quad (25)$$

The set \mathbf{R} is nonempty if and only if the learning sets G^+ and G^- (1) are linearly separable (5). In this case the set \mathbf{R} is the convex polyhedron in the $(n + 1)$ -dimensional parameter space. The optimal parameter vector \mathbf{v}^* (10) can be located in any of the vertices \mathbf{v}_k (22) of the set \mathbf{R} [5]. This means that the minimal value $\Phi(\mathbf{v}^*)$ (13) of the perceptron criterion function $\Phi(\mathbf{v})$ (10) is equal to zero in any vertex of the set \mathbf{R} (25). The equations (23) are fulfilled in each vertex \mathbf{v}_k of the polyhedron \mathbf{R} (25).

In accordance with the definition of the hyperplane h_i^0 (20), the i -th component w_{ki} of the vector (vertex) $\mathbf{v}_k = [w_{k1}, \dots, w_{kn}, -\theta_k]^T$ (23) is equal to zero ($w_{ki} = 0$) if the unit vector \mathbf{e}_i ($i \in I_k$) is contained in the basis \mathbf{B}_k :

$$(i \in I_k) \Rightarrow (w_{ki} = 0) \quad (26)$$

We can remark that the same set J_k of m_k base vectors \mathbf{y}_j can be used to define different vertices $\mathbf{v}_{k'}$ (22) of the set \mathbf{R} (25). These vertices $\mathbf{v}_{k'}$ (22) are defined by different subsets of the base unit vector \mathbf{e}_i ($i \in I_{k'}$).

The minimization (16) of the modified criterion function $\Psi_\lambda(\mathbf{v})$ (13) with the condition that $\gamma_i = 1$ (15) allow to characterize the best vertex $\mathbf{v}_k^* = [(\mathbf{w}_k^*)^T, -\theta_k^*]^T = [w_1^*, \dots, w_n^*, -\theta^*]^T$ (17) by the below property:

$$\min_{i \in \{1, \dots, n\}} \{ \sum |w_{ii}| : \mathbf{v} \in \mathbf{R} \} = \sum_{i \in \{1, \dots, n\}} |w_i^*| \quad (27)$$

The relation (27) means that the optimal vertex \mathbf{v}_k^* (15) of the set \mathbf{R} (25) has the lowest L_1 length of the weight vector \mathbf{w}_k^* . There is a certain similarity to the optimal vector \mathbf{w}_{SVM}^* found in accordance with the *support vector machines (SVM)* [10]:

$$\min \{ \mathbf{w}^T \mathbf{w} : \mathbf{v} \in \mathbf{R} \} = (\mathbf{w}_{SVM}^*)^T \mathbf{w}_{SVM}^* \quad (28)$$

The optimal *SVM* vector \mathbf{w}_{SVM}^* is characterized by the lowest Euclidean L_2 norm, in contrast to the L_1 norm used in the relation (27) resulting from the *CPL* approach [5].

6 Learning Algorithms Linked to the CPL Criterion Functions

The criterion functions $\Phi(\mathbf{v})$ (8) have originated from the error correction algorithm used in the framework of the *Perceptron* model [3]. The learning process is based on the learning sequence $\{(\mathbf{x}[k], s[k])\}$, where $k = 1, 2, 3, \dots$:

$$(\mathbf{x}[1], s[1]), (\mathbf{x}[2], s[2]), (\mathbf{x}[3], s[3]), \dots \quad (29)$$

where $\mathbf{x}[k]$ is the *input vector* of the formal neuron selected during the k -th learning step from the learning sets (1) ($\forall k = 1, 2, 3, \dots$) $\mathbf{x}[k] = \mathbf{x}_j$, and $s[k]$ ($s[k] = 1$ iff $\mathbf{x}_j \in G^+$, and $s[k] = 0$ iff $\mathbf{x}_j \in G^-$) is the *teacher's decision*. The decision rule $r(\mathbf{w}[k], \theta[k]; \mathbf{x}[k])$ of the formal neuron $NF(\mathbf{w}[k], \theta[k])$ during the k -th learning step is specified below:

$$\begin{aligned} r[k] &= r(\mathbf{w}[k], \theta[k]; \mathbf{x}[k]) = \\ &\quad \begin{cases} 1 & \text{if } \mathbf{w}[k]^T \mathbf{x}[k] \geq \theta[k] \\ 0 & \text{if } \mathbf{w}[k]^T \mathbf{x}[k] < \theta[k] \end{cases} \end{aligned} \quad (30)$$

In accordance with the classical *error correction* algorithm, the formal neuron parameters $\mathbf{v}[k] = [\mathbf{w}[k]^T, -\theta[k]]^T$ (4) are changed during the k -th step if and only if the *neuron's decision* $r[k]$ differs from *teacher's decision* $s[k]$ ($r[k] \neq s[k]$):

$$\text{if } r[k] \neq s[k], \text{ then } \mathbf{v}[k+1] = \mathbf{v}[k] + \mathbf{y}[k], \text{ else } \mathbf{v}[k+1] = \mathbf{v}[k] \quad (31)$$

where $\mathbf{y}[k] = (2s[k] - 1) [\mathbf{x}[k]^T, 1]^T$ ($\mathbf{y}[k] = [\mathbf{x}[k]^T, 1]^T$ or $\mathbf{y}[k] = -[\mathbf{x}[k]^T, 1]^T$ (4)).

The proof of convergence of the error correction algorithm (31) in a finite number of steps is based on the assumption of linear separability (3) [2]. The algorithm (31) never stops the learning sets G^+ and G^- (1) are not linearly separable (3).

The error correction algorithm (31) has been modified in order to assure convergence in a finite number of steps also when the learning sets G^+ and G^- (1) are not linearly separable [5]. Three types of modifications were included:

i. *random generation* of the learning sequence $\{(\mathbf{x}[k], s[k])\}$ in accordance with a stationary probability distribution $p(\mathbf{x}_j, s_i)$, where $\mathbf{x}_j \in G^+ \cup G^-$ (1) and $s_i = 1$ or $s_i = 0$:

$$(\forall k = 1, 2, 3, \dots) P\{\mathbf{x}[k] = \mathbf{x}_j, s[k] = s_i\} = p(\mathbf{x}_j, s_i) \quad (32)$$

ii. *positive margin* δ ($\delta = 0$)

iii. *decreasing gain parameters* β_k ($\beta_k > 0$) that fulfill the stochastic approximations conditions [8]:

$$\sum_{k=1, \dots, \infty} \beta_k = \infty \quad \text{and} \quad \sum_{k=1, \dots, \infty} (\beta_k)^2 < \infty \quad (33)$$

The modified learning algorithm (31) can be represented in the below manner [5]:

$$\text{If } \mathbf{v}[k]^T \mathbf{y}[k] < 1, \text{ then } \mathbf{v}[k+1] = \mathbf{v}[k] + \beta_k \mathbf{y}[k], \text{ else } \mathbf{v}[k+1] = \mathbf{v}[k] \quad (34)$$

If the learning sets G^+ and G^- (1) are not linearly separable (4), then the algorithm (34) with the assumptions (32), (33) generates such random sequence $\{\mathbf{v}[k]\}$ that converges with probability 1 to the optimal (stationary) vector \mathbf{v}^* [5]:

$$\underset{k \rightarrow \infty}{\mathbf{v}[k]} \rightarrow \mathbf{v}^* \quad (35)$$

The algorithm (34) can be considered as the Robbins-Monro procedure of stochastic approximation aimed at minimization of the below regression type function $Q(\mathbf{v})$ [8]:

$$Q(\mathbf{v}) = \sum_{j=1,\dots,m} \sum_{i=0,1} p(\mathbf{x}_j, s_i) q(\mathbf{x}_j, s_i; \mathbf{v}) \quad (36)$$

where $p(\mathbf{x}_j, s_i)$ determine the probability distribution (32) and $q(\mathbf{x}_j, s_i; \mathbf{v})$ is a penalty function linked to the learning pair (\mathbf{x}_j, s_i) (29).

The Robbins-Monro procedure is based on the criterion function $Q(\mathbf{v})$ (32) through the gradient $\nabla_{\mathbf{v}} q(\mathbf{x}_j, s_i; \mathbf{v})$ of the penalty function $q(\mathbf{x}_j, s_i; \mathbf{v})$ defined by the learning pair (\mathbf{x}_j, s_i) (29) [8]:

$$\mathbf{v}[k+1] = \mathbf{v}[k] - \beta_k \nabla_{\mathbf{v}} q(\mathbf{x}_j, s_i; \mathbf{v}) \quad (37)$$

where the parameters β_k ($\beta_k > 0$) fulfill the conditions (34).

Theorem 2: If the criterion function $Q(\mathbf{v})$ (36) is convex, then the stationary point \mathbf{v}^* (35) of the Robbins-Monro procedure (37) constitutes the minimum $Q(\mathbf{v}^*)$ of this function [8]:

$$(\forall \mathbf{v}) \quad Q(\mathbf{v}) \geq Q(\mathbf{v}^*) \quad (38)$$

Remark 8: The modified error correction algorithm (34) with the assumptions (32), (33) can be treated as the Robbins-Monro procedure (37) aimed at minimization of the perceptron criterion function $\Phi(\mathbf{v})$ (8) with the prices $\alpha_j = 1$ [5].

It can be also shown that Robbins-Monro procedure (37) aimed at the minimization of the modified criterion function $\Psi_{\lambda}(\mathbf{v})$ (13) can have the below formula:

$$\begin{aligned} \text{if } \mathbf{v}[k]^T \mathbf{y}[k] < 1, \text{ then } \mathbf{v}[k+1] &= \mathbf{v}[k] + \beta_k \{\mathbf{y}[k] - \lambda [\mathbf{\Gamma} \mathbf{w}[k]]^T, 0\}^T, \\ \text{else } \mathbf{v}[k+1] &= \mathbf{v}[k] - \beta_k \lambda [\mathbf{\Gamma} \mathbf{w}[k]]^T, 0 \end{aligned} \quad (39)$$

where $\mathbf{\Gamma}$ is the diagonal matrix of the dimension $n \times n$, with the *features costs* γ_i ($\gamma_i > 0$) (13) on the diagonal.

7 Concluding Remarks

The perceptron criterion function $\Phi(\mathbf{v})$ (8) has been defined on feature vectors that belong to learning sets [4], [5]. This function is linked to the error correction algorithm considered in a framework of neural network theory [1], [2]. The perceptron criterion function $\Phi(\mathbf{v})$ (8) belongs to the family of the convex and piecewise-linear (CPL) functions.

The criterion function $\Phi(\mathbf{v})$ (8) has the property that its minimum value $\Phi(\mathbf{v}^*)$ (12) is equal to zero if and only if the learning sets are linearly separable. As a consequence, the linear separability of a given pair of learning sets G^+ and G^- (1) can be detected through minimization of the criterion function $\Phi(\mathbf{v})$ (8) defined on elements \mathbf{x}_j of these learning sets.

Evaluation of the linear separability degree of two learning sets can be carried out through minimization of the criterion functions $\Phi(\mathbf{v})$ (8). The modified *CPL* criterion function $\Psi_\lambda(\mathbf{v})$ (13) is used in the relaxed linear separability (*RLS*) method of optimal feature subsets selection [6]. The minimal value and the optimal vector of particular *CPL* criterion functions $\Phi(\mathbf{v})$ (8) or $\Psi_\lambda(\mathbf{v})$ (13) can be computed efficiently even in the case of large high-dimensional data sets by using the basis exchange algorithms, which are similar to the linear programming [7].

The minimization the perceptron criterion function $\Phi(\mathbf{v})$ (8) can also be carried out during the learning process by using the modified error correction algorithm (34). The learning algorithm (34) has resulted from the Robbins-Monro procedure (37) applied to the perceptron criterion function $\Phi(\mathbf{v})$ (8).

The modified learning algorithm (39) has been formulated in this work. This algorithm has resulted from the Robbins-Monro procedure (37) applied to the modified criterion function $\Psi_\lambda(\mathbf{v})$ (13). The algorithm (39) allows, among other things, for a broader interpretation of the *RLS* method of feature selection [6]. This algorithm can give new insight into the formation of important feature subsets during a learning process.

Acknowledgments. This work was supported by the project S/WI/2/2013 from the Białystok University of Technology.

References

1. Rosenblatt, F.: Principles of neurodynamics. Spartan Books, Washington (1962)
2. Minsky, M.L., Papert, S.A.: Perceptrons - Expanded Edition. An Introduction to Computational Geometry. The MIT Press, Cambridge (1987)
3. Duda, O.R., Hart, P.E., Stork, D.G.: Pattern classification. J. Wiley, New York (2001)
4. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, New York (1995); Rosenblatt F.: Principles of neurodynamics. Spartan Books, Washington (1962)
5. Bobrowski, L.: Eksploracja danych oparta na wypukłych i odcinkowo-liniowych funkcjach kryterialnych (Data mining based on convex and piecewise linear (*CPL*) criterion functions). Technical University Białystok (2005) (in Polish)
6. Bobrowski, L., Łukaszuk, T.: Relaxed Linear Separability (*RLS*) Approach to Feature (Gene) Subset Selection. In: Xia, X. (ed.) Selected Works in Bioinformatics, pp. 103–118. Intech (2011)
7. Bobrowski, L.: Design of piecewise linear classifiers from formal neurons by some basis exchange technique. Pattern Recognition 24(9), 863–870 (1991)

8. Kushner, H.J., Clark, D.S.: Stochastic Approximation for Constrained and Unconstrained Systems. Springer, Berlin (1978)
9. Wu, T.T., Chen, Y.F., Hastie, T., Sobel, E., Lange, K.: Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics* 25(6), 714–721 (2009)
10. Vapnik, V.N.: Statistical Learning Theory. J. Wiley, New York (1998)

Learning Errors of Environmental Mathematical Models

Dimitri Solomatine^{1,2}, Vadim Kuzmin³, and Durga Lal Shrestha⁴

¹ UNESCO-IHE Institute for Water Education,
Westvest 7, P.O. Box 3015, Delft, The Netherlands

² Water Resources Section, Delft University of Technology

³ Russian State Hydrometeorological University (RSHU),
St. Petersburg, Russia

⁴ CSIRO, Australia

d.solomatine@unesco-ihe.org

Abstract. In solving civil engineering problems the use of various models for forecasting environmental variables (for example, water levels in a river during flooding) is a must. Mathematical models of environmental processes inevitably contain errors (even if models are calibrated on accurate data) which can be represented as realizations of a stochastic process. Parameters of this process vary in time and cannot be reliably estimated without making (unrealistic) assumptions. However the model errors depend on various factors characterizing environmental conditions (for example, for extreme events errors are typically higher), and such dependencies can be reconstructed based on data. We present a unifying approach allowing for building machine learning models (in particular ANN and Local weighted regression) able to predict such errors as well as the properties of their distributions. Examples in modelling hydrological processes are considered.

Keywords: neural network, local weighted regression, learning, errors of models.

1 Introduction

In planning civil engineering and infrastructural projects it is important to know the future states of natural environment, and the use of various models for forecasting environmental variables (for example, water levels in a river during flooding) is a must. Mathematical models of environmental processes inevitably contain errors which can be represented as realizations of a stochastic process. Parameters of this process vary in time and cannot be reliably estimated without making (unrealistic) assumptions. Assumptions of normality of errors that are sometimes made are far from reality since the model errors depend on various factors characterizing environmental conditions (for example, for extreme events errors are typically higher), and such dependencies can be reconstructed based on data.

There was earlier work aimed at estimating quantiles - the method of Quantile Regression (QR) introduced by Koenker and Bassett [1]. We tested it for hydrological

models in Solomatine and Shrestha 2009 and found that it cannot reliably predict quantiles. The reasons are: it is based on a linear model, uses only the time series itself without the use of other relevant variables, and builds the model using the whole data set, as opposed to the approach taken in our method which builds local specialized non-linear (and hence more accurate) models.

We present an approach allowing for building machine learning models able to predict such errors as well as the properties of their distributions, using information "hidden" in the variables that have influence on the forecasted variable. Examples in modelling hydrological processes are considered.

2 Methodology of Learning Errors

A deterministic model M of a real-world system predicting a system output variable y^* given input vector \mathbf{x} ($\mathbf{x} \in \mathbf{X}$) is considered (Figure 1). Mathematical model of a hydrological process predicts a system output variable y^* given input vector \mathbf{x} . Data-driven (machine learning) model here could be of two different types: a) predicting at each time step the model errors, treating them either as a deterministic variable (model V), or b) treating errors either as random variables and thus predicting at each time step their statistical properties or pdf (model U).

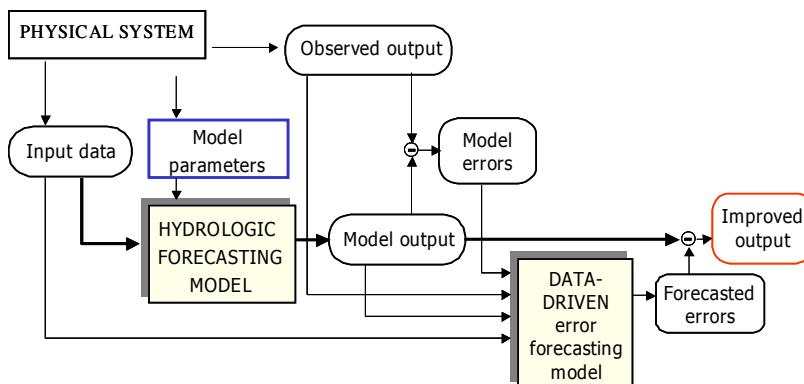


Fig. 1. A deterministic (hydrological) model, and a data-driven model predicting its errors or error's statistical properties

Let y be the measurement of an unknown true value y^* , made with error ε_y . Various types of errors propagate through the model M while predicting the observed output y and have the following form:

$$y = y^* + \varepsilon_y = M(\mathbf{x}, \theta) + \varepsilon_s + \varepsilon_\theta + \varepsilon_x + \varepsilon_y \quad (1)$$

where θ is a vector of the model parameters values, ε_s , ε_θ , and ε_x , are the errors associated with the model structure, parameter θ and input vector \mathbf{x} , respectively.

In most cases it is very difficult to estimate the individual error components in (1), so these components are generally treated as a single variable and equation (1) can be written as:

$$y = \hat{y} + \varepsilon \quad (2)$$

where \hat{y} is the model output and ε is the remaining (or residual) error.

We aim here at building two models (Figure 1):

- 1)a machine learning model V to forecast error ε (treating the error as a deterministic variable)
- 2)a machine learning model U able at every time step to estimate the statistical properties of the error ε (ideally, its pdf or at least its quantiles) conditioned to the input and/or state variables of the process model M.

A challenge is to try to combine these two models into one, however at the moment of this paper submission we are not yet ready to report the results.

We make here an important assumption. It is assumed that the uncertainty associated with the wrong model structure, inaccurate parameter values, and observational errors (if any) are manifested implicitly in the model residuals. This type of uncertainty analysis based on the model residuals is principally different from the classical uncertainty analysis methods [2] where uncertainty of parameters, input data (presented by pdf) or plausible model structures are propagated to the pdf of the output, typically using Monte Carlo simulations.

2.1 Model V to Predict Deterministic Error

First, the input variables (predictors) for model V have to be identified. We defined these as descriptive, Relevant Variables (RV). For example, for hydrological conceptual models these can be differently lagged and preprocessed rainfall data, soil moisture and flow rate. Average Mutual Information (AMI) and correlation analysis can be used to select these predictors.

We used ANN as machine learning methods due to their widely reported reliability and accuracy. Experiments show that after correction the model error is reduced considerably (for various experiments the error reduction was between 50 and 70%).

2.2 Model U to Predict Statistical Properties of the Error

The easiest idea is to find the quantiles from the empirical distribution of error across the available data and to use for all future estimates of y . However this approach does not take into account the fact that quantiles depend on many factors and have to be different at each time step.

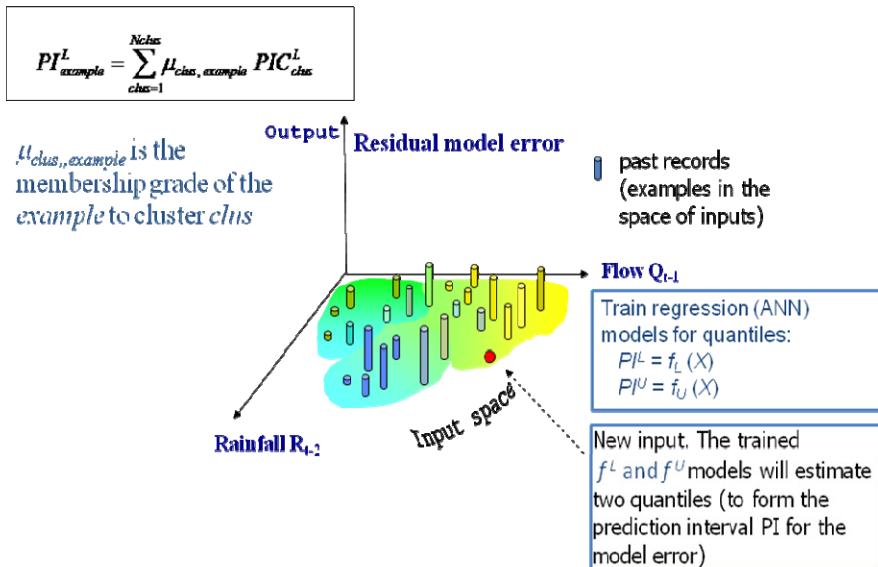


Fig. 2. Essence of UNEEC method

To solve this problem machine learning methods were also used for building the U model. Initially we aimed at estimation of two quantiles (5 and 95%) of the residual error (Figure 2), and are extending it to predict more quantiles. All data is presented in the input space of “relevant variables” X (in this case two), with the associated error value as output. PI^L is the lower quantile (5%), and PI^U – the upper one (95%). Model U is constituted of a pair of models f_L and f_U treating errors either as random variables and thus predicting at each time step their statistical properties.

The basics of this approach were worked out in earlier publications ([3], [4]) and this approach was termed UNEEC (UNcertainty Estimation based on local Errors and Clustering). It includes the following steps:

- In the data set representing the past model performance (errors), find fuzzy clusters of data (fuzzy c-means clustering method is used). Clustering is done in the space of specially selected descriptive variables (for hydrological conceptual models these are lagged rainfall data, soil moisture and flow). Again, average Mutual Information (AMI) and correlation analysis can be used to select these predictors. Optimality of cluster numbers must be ensured. Various indicators have been proposed for offline identification of the best number of clusters in the literature (e.g. the method proposed by Xie and Benies [5]).
- For each cluster c calculate M percentiles Q_c^m ($m=1,\dots,M$) of the empirical distribution of the model error for each cluster c , taking however into account the membership degree of each data point (weighted counting is conducted).

- Using the calculated percentiles Q_c^m for each cluster c , calculate the “global” estimate of the percentile Q^m for each data vector, by weighting the cluster percentile by the corresponding degree of membership of the given data vector to this cluster. Form the input-output data tables for each percentile m where the “global” percentile values Q^m are used as outputs.
- For each percentile m , train a machine learning (statistical) model (e.g. an ANN or use Instance-based learning) able to predict the percentile value Q^m given the input vector of relevant variables RV. If $M=2$, the percentiles will form the confidence interval with the confidence level determined by the percentiles used.

The most important advantage of UNEEC is that it makes no assumptions about the error distribution, and that the use of clustering allows for taking into account the local character of dependency of the error percentiles on the descriptive (relevant) variables, and the use of fuzzy logic reflects the smooth nature of variability in environmental variables and makes the transition between these “local” models gradual.

3 Results

We tested the applicability of models V and U on an example of a hydrological model representing the relationship between rainfall and the resulting runoff (flow) in a catchment (watershed). For this purpose we used the HBV model, a conceptual, deterministic rainfall-runoff model of the Brue catchment in the United Kingdom.

The model was calibrated using the Nash-Sutcliffe coefficient as a performance measure often used in hydrological studies ([6], [7]) (calibration set: hourly data from 1994/06/24 to 1995/06/24, verification set: from 1995/06/24 to 1996/05/31).

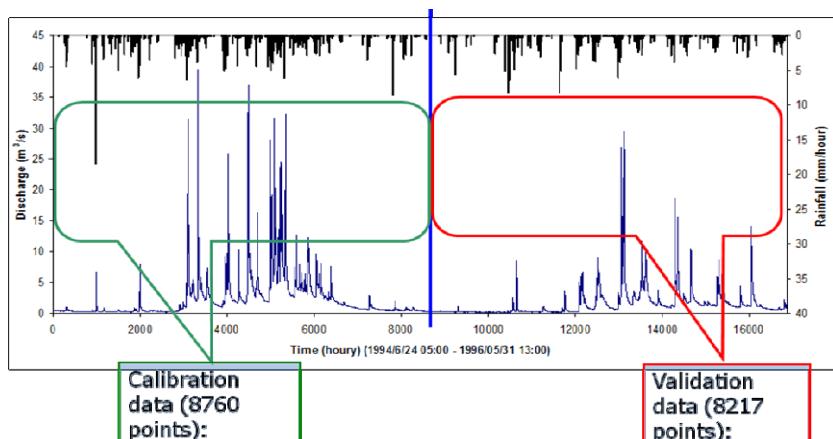


Fig. 3. Rainfall (*top*) and Discharge (*bottom*) data used in the modeling exercise

Analysis of error dependency on environmental variables revealed that the maximum dependency is observed for the following parameters: rainfall equivalent (RE) and discharge (Q) at previous time steps:

$$\text{RE}(t-8), \text{RE}(t-9), \text{RE}(t-10), Q(t-1), Q(t-2), Q(t-3).$$

We present here only the U model. Two U models were built (both were MLP ANNs):

$$q5 = U^L(\text{RE}(t-8), \text{RE}(t-9), \text{RE}(t-10), Q(t-1), Q(t-2), Q(t-3))$$

$$q95 = U^U(\text{RE}(t-8), \text{RE}(t-9), \text{RE}(t-10), Q(t-1), Q(t-2), Q(t-3))$$

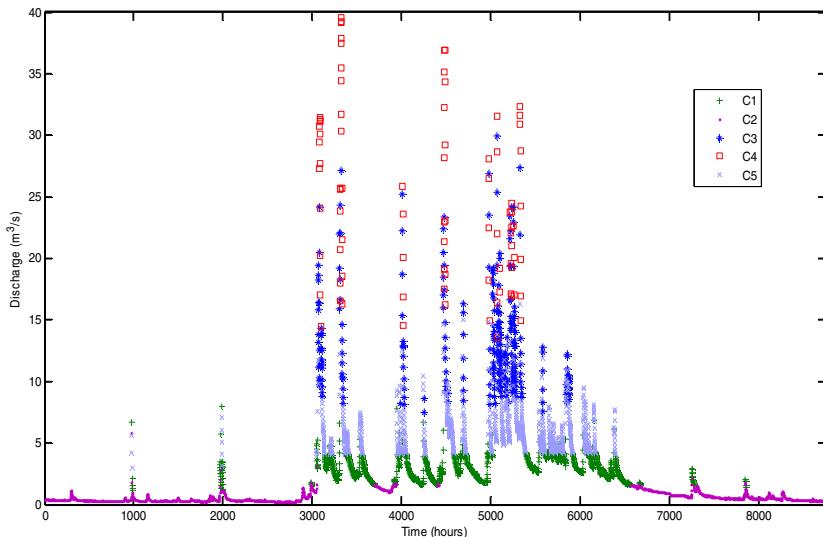


Fig. 4. Figure 4. Results of c-means fuzzy clustering. The quantiles are calculated separately for each cluster.

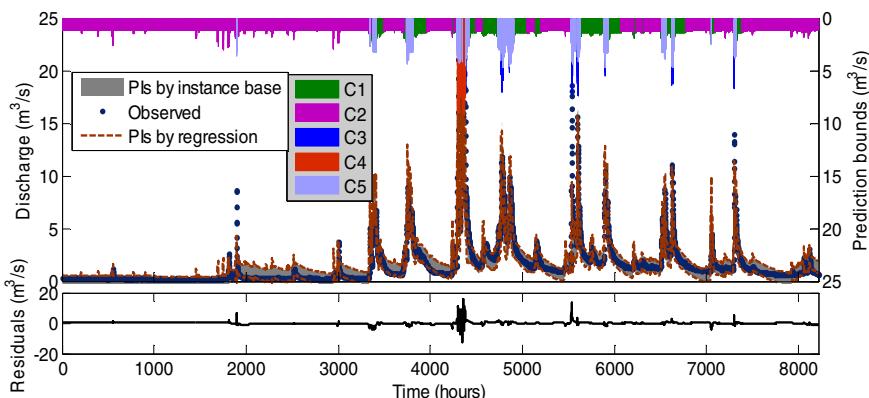


Fig. 5. Modelling results, together with the prediction intervals calculated by the U models. We used two techniques: MLP ANN and Instance-based learning (local weighted regression).

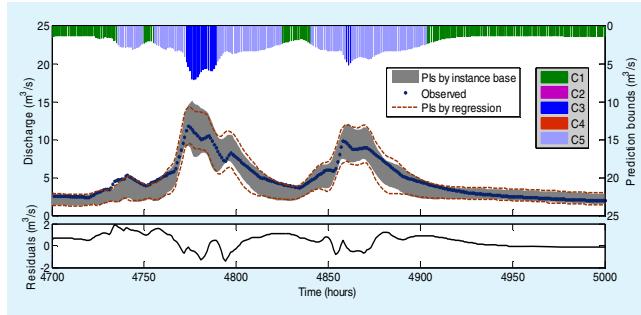


Fig. 6. Fragment of the plot given on Figure 5

For training MLP ANN we used Levenberg-Marquardt optimization algorithm, and the number of hidden nodes was selected by exhaustive search (training ANN for 2,3,..., 10 hidden nodes and then selecting the architecture with the minimum cross-validation error); we found that 4 hidden nodes give the best results.

Results are presented on Figures 4-6. On Figure 4 It can be seen that low, medium and high flows belong to different clusters, so that automatic clustering quite reliably determines physical properties of flow. Figure 5 shows the prediction intervals which can be used by practitioners as indicators of certain levels of uncertainties associated with the deterministic forecasts. Consultation with experts in hydrology revealed that the results represent the residual uncertainty quite well.

One of the indicators to assess the quality of uncertainty forecasting methods could be the Prediction interval coverage probability (PICP) that shows how much data is inside the prediction interval:

$$\text{PICP} = \frac{1}{n} \sum_{t=1}^n C \quad (3)$$

with $C = \begin{cases} 1, & PI_t^L \leq y_t \leq PI_t^U \\ 0, & \text{otherwise} \end{cases}$

In our experiments we were consistently obtaining PICP around 80% (for the theoretical level to be at 90%).

4 Conclusions and Future Work

The use of Machine learning methods allows for explicit representation and forecasting of mathematical models errors. These methods are computationally efficient and can be used in real time application.

Future work will be focused at testing other ML methods and exploring the possibilities of merging the V and U models; indeed, both V and U models are aiming at modeling the error so it would be natural to try to combine them into one unified model. One of the possibilities to be explored is using a committee of specialized models (V, U or their combination) instead of training a global model as we did in this paper.

Acknowledgments. The authors acknowledge partial financial support of the WeSenseIt EU FP7 project.

References

1. Koenker, R., Bassett, G.: Regression quantiles. *Econometrica* 46(1), 33–50 (1978)
2. Kuczera, G., Parent, E.: Monte Carlo assessment of parameter uncertainty in conceptual catchment models: The Metropolis algorithm. *J. Hydrology* 211, 69–85 (1998), doi:10.1016/S0022-1694(98)00198-X
3. Shrestha, D.L., Solomatine, D.P.: Machine learning approaches for estimation of prediction interval for model output. *Neural Networks* 19, 225–235 (2006)
4. Solomatine, D.P., Shrestha, D.L.: A novel method to estimate model uncertainty using machine learning techniques. *Water Resources Research* 45, W00B11 (2009), doi:10.1029/2008WR006839
5. Xie, X.L., Beni, G.A.: Validity measure for fuzzy clustering. *IEEE Trans. PAMI* 3(8), 841–846 (1991)
6. Madsen, H.: Automatic calibration of a conceptual rainfall-runoff model using multiple objectives. *J. Hydrol.* 235(3-4), 276–288 (2000)
7. Kuzmin, V.A.: Algorithms of Automatic Calibration of Multi-parameter Models Used in Operational Systems of Flash Flood Forecasting. *Russian Meteorology and Hydrology* 34(7), 1068–3739 (2009) ISSN 1068-3739

On Mining Opinions from Social Media

Vicky Politopoulou and Manolis Maragoudakis

Department of Information and Communication Systems Engineering,
University of the Aegean,
83200, Karlovassi, Greece
`{v.politopoulou,mmarag}@aegean.gr`

Abstract. The broad use of social media nowadays has led many users to express their opinions on various subjects through them. The need for these opinions to be automatically labeled and categorized according to their sentiment, has also arisen. In this paper, a novel sentiment analysis approach is introduced, which takes into account the total number of idiomatic expressions and emoticons that are used in the text, and simultaneously processes the original text in Greek along with its automatic translation in English as well. Moreover, the novelty of the proposed solution lies in the difficulty of Modern Greek language and the fact that the text in social media is mainly unstructured. The proposed methodology is tested on two distinct data sets of opinions regarding a certain matter, which have been collected from Facebook and Twitter respectively. Finally, we discuss the performance of various classification algorithms and we compare the extracted experimental results.

Keywords: opinion mining, social media analysis, sentiment analysis, sentiment classification.

1 Introduction

In the last years, social media have become an integral part of people's lives as well as of modern day business' operations, and cannot be ignored. According to Alexa Internet Inc. [4] two out of ten most popular web pages are Facebook and Twitter. Users spend more time on Facebook than any other site [9]. Facebook has almost 1 billion users while Twitter has approximately 500 millions.

These millions of users with different educational background share their thoughts through social media. In Twitter, everyone can be an influencer [5] and in Facebook after 2009, using a specific application (like) a user can inform his/her friends for his/her likings. Due to this richness of the available content, the analysis of opinions (or *opinion mining*) in social media has turned into a very important issue not only for the academia but also for the industry [24].

Generally speaking, opinion mining is the computational study of opinions, sentiments and emotions expressed in texts. In terms of well-structured texts, various approaches have been developed in order to track the opinion of Internet users, which received an extra boost due to recent advances in machine learning and natural language processing.

However, this is not the case with unstructured text; with the appearance of social media, a variety of issues in opinion mining arose due to the unstructured way the information is being represented and the difficulty in getting relevant information out of large volumes of data in a useful way. Moreover, the constraints in the maximum length of messages must be considered when we deal with opinion mining in social media, as Facebook posts can have a maximum length of 5,000 characters (Facebook comments have a maximum length of 8,000 characters) while Twitter messages, or else tweets, can have a maximum length of 140 characters. Therefore, we have two platforms that are commonly used by Internet users with the above mentioned special characteristics and much information to be analyzed.

Opinion mining approaches on social media have involved the use of classifiers in varying languages and various features. To the best of our knowledge though, there isn't a work that combines a classifier with features as the count of total idiomatic expressions, i.e. the total number of capital words in the text, and the total number of emoticons, i.e. the total number of punctuation marks that are used for emphasis in a corpus with data in those two languages, Greek and English.

In this paper a novel approach regarding opinion mining from social media platforms is presented, that is using - among others - a categorization based on idiomatic expressions and emoticons. The proposed methodology is evaluated with the use of 700 tweets and 700 Facebook posts about wind energy, which are employed as the corpuses of the experiments. The original posts and tweets are in Greek language, but are also translated in English in order to have more features to work with.

The goal is to find out how much the performance of three classifiers (SVM, Naive Bayes and Neural Net) is affected, when taking into account additionally the number of idiomatic expressions and the number of emoticons and the corpus consists of text in two different languages simultaneously, while the second one is the translated version of the original text through a web translation service. Finally, the performance of these three classifiers is measured, when they are tested in a social networking/microblogging sites like Facebook and Twitter.

The rest of the paper is organized as follows. In section 2 previous work on opinion mining and sentiment analysis in social media is presented. Section 3 describes the proposed method. The experimental results are presented in Section 4 and in section 5 future implementations are discussed.

2 Related Work

The approaches that have been used so far in sentiment classification include:

- *unsupervised* techniques in order to find patterns according to a lexicon,
- *supervised* techniques using classifiers, such as SVM, Naive Bayes, etc., with various syntactic and/or linguistic features [10,14], and
- *semi-supervised* techniques that use a combination of a lexicon and a classifier.

Part-of-speech (POS) information and punctuation marks have been exploited in sentiment analysis techniques in order to indicate the existence of an opinion [15], POS n-gram patterns [13] have been used in order to represent the expression of a sentiment and modifiers [25] were used in order to indicate the presence of appraisal.

Authors in [1] proposed sentiment analysis in multiple languages, while in [7] a method for domain adaptation for sentiment classifiers is introduced. Aisopos et. al [3] proposed an N-gram graph to categorize the sentiment of social medias content. Tromp [23] proposed an approach which consists of four steps, language identification, Part-of Speech tagging, subjectivity detection and polarity detection and Agarwal et al [2] introduced POS-specific prior polarity features.

Rule based techniques use sentiment dictionaries along with rules. Maynard et al. [11] used a rule based technique to analyze political tweets. Taboada et al [22] used a rule based technique in order to detect positive and negative sentiment of ePinions reviews on the web. In 2012 the authors of [12] analyzed opinions expressed in social media about two very different domains using a rule based technique which performs shallow linguistic analysis and builds on a number of linguistic subcomponents to find the sentiment for each comment.

Boiy and Moens [8] tested machine learning techniques in English, Dutch and French texts based on unigram features augmented with linguistic features while Ritter et al. [18] proposed a new approach using POS tagging through Chunking, to Named Entity Recognition for Twitter messages. Bal et al [6] proposed a multilingual pipeline for evaluating how an author's sentiment is conveyed in different languages, English and Deutch. Stajner et al [21] explored the influence of using background knowledge in the form of different sentiment lexicons, as well as the influence of various lexical surface features. Said et al [19] proposed a language-independent, semi-supervised Twitter sentiment analysis method and used emoticons as noisy labels.

3 Proposed Methodology

This paper introduces an approach, considered as a supervised technique, which uses one of SVM, Naive Bayes and Neural Net each time as classifier with the following features: the total number of idiomatic expressions in each comment, and the total number of emoticons along with bigram features in a corpus which contains text in two languages, Greek and English.

The proposed methodology comprises of the following steps:

1st Step: A corpus of 700 tweets is collected using Twitter API, which are labeled as positive, negative or neutral. Moreover, the same number of Facebook posts concerning wind energy is collected. The available data were labeled manually by the authors. Those texts are translated in English and the corpus that will be used to train the classifiers contains the same text in two different languages, Greek and English. The corpus has approximately the same number of comments in all three categories.

2nd Step: The next step is to pre-process the available data in order to be presented in an appropriate way for the classifiers. Firstly, an additional procedure is carried out for the available tweets in order to remove the url links, Twitter user names and hashtags. Afterwards, the discrete steps that are described below are followed, both for the two datasets:

- a. *Tokenization*, in order to extract the words from the corpus. From the available tokens, only those with minimum length of five are retained.
- b. *Lemmatization*, using Stem Snowball [17], in order to reduce the variant forms of a word to a common form.
- c. *Feature selection*. In order to identify the most important words in the text, the TF-IDF score is used as the term presence metric, based on word frequency in each comment and in the entire corpus. Finally, text is represented using bigram model which generates all possible two-word sequence pairs found on the text. Bigrams have the potential of retaining more information regarding opinion polarity than unigrams. In order to further reduce the vector's size, each word's weights (normalized from 0 to 1) are calculated using SVM, which uses the coefficients of the normal vector of a linear SVM as feature weights, and afterwards the words having a weight greater than 0,55 are selected. Therefore, the selected bigrams are the input features for the classifiers that are used in our experiments. Consequently, data are composed of bag-of-words features as well as number of idiomatic expressions and emoticons.

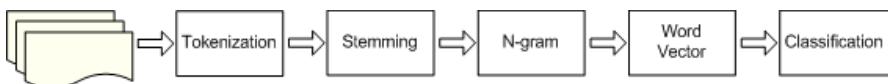


Fig. 1. The classification procedure

- d. *Classification*. Three sentiment classifiers (Naive Bayes, SVM and Neural Net) are built, out of which the first one (Naive Bayes) proves to be the most effective.

3.1 Naive Bayes

The Naive Bayes Classifier is probably the most popular classifier due to its good performance and computational efficiency despite its simplicity. Moreover, this classifier is very fast during the training procedure. The model assumes that all features are fully independent even though this never happens in real situations. Despite the unreal assumptions being made and the simplistic design of the classifier, it shows good results if the classification problems are not so complex, as is the case in this work.

3.2 SVM

The SVM classifier, even though it is very powerful, it is complex and difficult in training, as well as sensitive to noisy data. Also, this classifier is non-linear.

The general idea behind SVM is that data can better be classified by increasing the dimensionality. More particularly, SVM uses an n -dimensional space, where n is the number of samples in the training set.

3.3 Neural Networks

Neural networks are inspired by the way biological nervous systems, such as the brain, process information. The system is composed of a large number of highly interconnected processing elements called neurons. Neural networks learn by example. We used the Multilayer Perceptron Network which learns a model by means of a feed-forward neural network. This learning is done via back-propagation.

A detailed analysis of the three above mentioned classifiers can be found in [20].

4 Experimental Results

In order to determine whether the proposed approach is suitable for sentiment analysis on social media text, when taking into account additionally the total number of idiomatic expressions and emoticons used for emphasis in the text, a set of experiments using the three classifiers mentioned above is conducted.

More specifically, firstly Greek texts regarding a specific subject (wind energy) are collected from the two most famous social network platforms, i.e. Facebook and Twitter. Then those texts are translated in English and thus two different corpuses are created, one for each platform, that will be the base of our experiments.

Consequently, for each platform, a 10-fold cross validation approach is used to evaluate the performance of the three classifiers (Bayes, SVM, Neural Net), and for the most efficient of them, the effect of the combined use of Greek and English language is studied, as compared to only having one of them. The evaluation is conducted with the use of recall, precision and accuracy defined by equations (1,2,3) respectively:

$$\text{Recall} = \frac{t_p}{t_p + f_n} \quad (1) \qquad \text{Precision} = \frac{t_p}{t_p + f_p} \quad (2)$$

$$\text{Accuracy} = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}, \quad (3)$$

where t_p are the correct positive predictions (true positives), f_p are the unexpected results (false positives), f_n are the missing results (false negatives) and t_n are the correct negative predictions (true negatives).

In other words, recall is the relative number of the correctly classified instances that were actually classified, precision is the relative number of correctly

classified instances among all those classified and accuracy is the proportion of the instances of the testing set that were classified correctly against all instances.

Finally, the results of those two platforms are compared in order to examine if and in what way is the proposed methodology *platform-independent*. The results of the conducted experiments, together with our comments, can be found in the next subsections.

4.1 Experiments with Facebook Corpus

Facebook corpus consists of 700 posts, in Greek and in English, regarding wind energy. Out of these 700 posts, 300 are negative comments, 245 are positive and 155 are neutral.

Naive Bayes Classifier. For the Facebook case, a Naive Bayes classifier is trained using unigrams, as well as bigrams, and the results prove that the latter is more efficient. Thus, for the next experiments, all classifiers will always be used with bigrams. Tables 1,2 show the results of the bayesian classifier for unigrams and bigrams respectively.

Table 1. Bayes with unigrams

| | | | |
|---------------------|----------|----------|---------|
| accuracy: 84.17% | negative | positive | neutral |
| class precision | 84.21% | 87.50% | 76.92% |
| class recall | 88.89% | 96.55% | 55.56% |

Table 2. Bayes with bigrams

| | | | |
|---------------------|----------|----------|---------|
| accuracy: 86.67% | negative | positive | neutral |
| class precision | 84.62% | 87.88% | 90.91% |
| class recall | 91.67% | 100.00% | 55.56% |

It is clear that Naive Bayes performs very well when bigrams are used, as they hold much more useful information than unigrams. The fact that its performance regarding neutral class is not so good is not alarming, since the classification of the examples in Neutral class is difficult even for humans and many times varies among different evaluators.

SVM. Next, SVM classifier is tested with bigram features, and results are shown on Table 3.

Table 3. SVM

| | | | |
|------------------|----------|----------|---------|
| accuracy: 68.19% | negative | positive | neutral |
| class precision | 61.02% | 84.21% | 100.00% |
| class recall | 100.00% | 55.17% | 27.78% |

SVM doesn't show such good performance as Naive bayes. This is mainly due to the fact that SVM shows better results when we have a bigger dimensionality and therefore more number of samples in the training set. Furthermore, SVM is sensitive to noisy data which is usually the case in social media text.

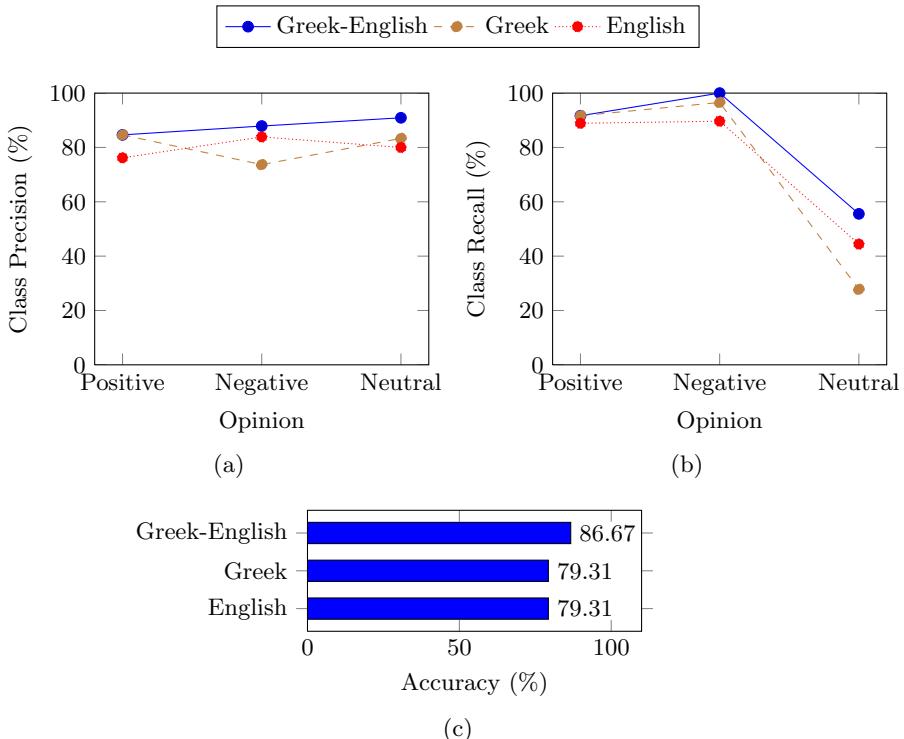
Table 4. Neural Net

| | | | |
|------------------|----------|----------|---------|
| accuracy: 59.71% | negative | positive | neutral |
| class precision | 77.14% | 74.07% | 38.10% |
| class recall | 75.00% | 68.97% | 44.44% |

Neural Net. Finally, for the Neural Net classifier results are shown on Table 4.

Best Classifier. It is derived from the results that the most efficient classifier is the Bayesian.

Subsequently, the effect of the two languages selection is examined, as compared to having just one of them. To do that, the experiment is conducted again, but it is done so for just the English and just for the Greek language. The results are depicted on Figure 2.

**Fig. 2.** Bayes performance for different corpuses - the Facebook case

Observing the results we can make the conclusion that our method performs well, regarding all classes - positive, negative, neutral- even in the case that we have more than one language, simultaneously.

4.2 Experiments with Twitter Corpus

Naive Bayes Classifier. For the Twitter case, a Naive Bayes classifier, as well as all the other classifiers, is trained using bigrams. The corpus used consists of 700 Twitter messages, 300 are positive, 240 are negative and 160 are neutral. Table 5 shows the results of the bayesian classifier for bigrams.

Table 5. Bayes with bigrams

| | | | |
|------------------|----------|----------|---------|
| accuracy: 63.33% | negative | positive | neutral |
| class precision | 59.09% | 80.00% | 66.67% |
| class recall | 100.00% | 50.00% | 22.22% |

From the above mentioned results it is concluded that the proposed method doesn't perform very well when Twitter messages are used, because tweets have only 140 characters while Facebook posts can have 5,000 and therefore tweets don't have so many idiomatic expressions and emoticons as Facebook posts.

SVM. For the SVM classifier, results are shown on Table 6.

Table 6. SVM

| | | | |
|------------------|----------|----------|---------|
| accuracy: 56.67% | negative | positive | neutral |
| class precision | 64.29% | 50.00% | 50.00% |
| class recall | 69.23% | 87.50% | 11.11% |

SVM shows worse performance comparing to that of Naive Bayes, as we not only have few characters in each comment but also the data are very noisy.

Neural Net. For the Neural Net classifier, results are shown on Table 7.

Table 7. Neural Net

| | | | |
|------------------|----------|----------|---------|
| accuracy: 53.33% | negative | positive | neutral |
| class precision | 52.63% | 100.00% | 37.50% |
| class recall | 76.92% | 37.50% | 33.33% |

Best Classifier. It can be seen from the results that the most efficient classifier is once again the Bayesian.

In analogy to the Facebook case, the effect of the two languages selection is once again examined, as compared to having just one of them. To do that, the experiment is repeated, separately for the English and the Greek language. The results are depicted on Figure 3.

It is clear that in Twitter corpus the proposed method presents better accuracy when using the corpus containing text in the two languages but not so good as in Facebook case. Consequently, the proposed method is more appropriate for the Facebook case.

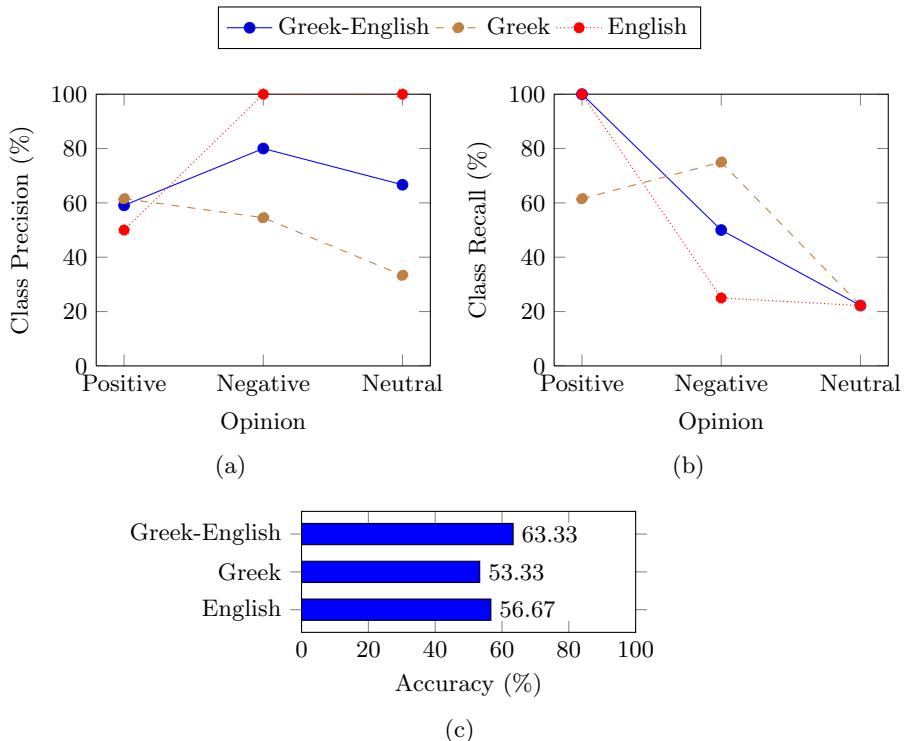


Fig. 3. Bayes performance for different corpuses - the Twitter case

5 Conclusions

In this paper a novel solution was presented for the detection of sentiment in social media, and especially in Facebook and Twitter platform. The proposed methodology uses a classifier combined with content-based features such as the total number of idiomatic expressions and emoticons that are used in the text we want to process, when the used corpus consists of text in Greek and English.

Experiments were conducted by training three classifiers, namely Naive Bayes, SVM and Neural Net, on source sets of hundreds of tweets and Facebook posts in Greek and English. The results of the three classifiers were compared with each other, as well as between using corpus containing text in one only language and when using two languages, Greek and English. Finally, the results between Facebook and Twitter corpus were compared.

The above mentioned experiments showed that the classifier that achieves the best performance is the Naive Bayes combined with bigram features regardless of the corpus used (Facebook or Twitter). Moreover, it is shown that best results are achieved when two languages are simultaneously present. Finally, it is derived from the results that the proposed methodology performs better with Facebook comments than with tweets. Future work might include tracking the sentiment as time passes and measuring the performance when having text in real time.

Acknowledgement. Research supported by EC FP7 project NOMAD - “Policy Formulation and Validation through non-Moderated Crowdsourcing” (www.nomad-project.eu).

References

1. Abbasi, A., Chen, H., Salem, A.: Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. *ACM Trans. Inf. Syst.* 26(3) (2008)
2. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of Twitter data. In: *LSM 2011* (2011)
3. Aisopos, F., Papadakis, G., Varvarigou, T.: Sentiment analysis of social media content using N-Gram graphs. In: *3rd ACM SIGMM*, pp. 9–14
4. Alexa - The Web Information Company, www.alexa.com (last accessed: March 2013)
5. Bakshy, E., Hofman, J.M., Mason, W.A., Watts, D.J.: Everyone’s an influencer: quantifying influence on Twitter. In: *WSDM 2011*, pp. 65–74 (2011)
6. Bal, D., Bal, M., van Bunningen, A., Hogenboom, A., Hogenboom, F., Frasincar, F.: Sentiment Analysis with a Multilingual Pipeline. In: Bouguettaya, A., Hauswirth, M., Liu, L. (eds.) *WISE 2011*. LNCS, vol. 6997, pp. 129–142. Springer, Heidelberg (2011)
7. Blitzer, J., Dredze, M., Pereira, F.: Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In: *ACL 2007* (2007)
8. Boiy, E., Moens, M.F.: A machine learning approach to sentiment analysis in multilingual Web texts. *Information Retrieval* 12(5), 526–558 (2009)
9. Comscore Inc, The comScore, Europe Digital Year in Review. s.n., s.l. (2011)
10. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford (2009)
11. Maynard, D., Funk, A.: Automatic Detection of Political Opinions in Tweets. In: *ESWC Workshops 2011*, pp. 88–99 (2011)
12. Maynard, D., Bontcheva, K., Rout, D.: Challenges in developing opinion mining tools for social media. In: *Workshop at LREC 2012* (2012)
13. Nasukawa, T., Yi, J.: Sentiment analysis: capturing favorability using natural language processing. In: *K-CAP 2003*, pp. 70–77 (2003)
14. Pak, A., Paroubek, P.: Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In: *LREC 2010* (2010)
15. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques, CoRR cs.CL/0205070
16. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval* 2, 1–135 (2008)
17. Porter, M.F.: Snowball: A Language for Stemming Algorithms (2001), <http://www.snowball.tartarus.org/texts/introduction.html>
18. Ritter, A., Clark, S., Mausam, Etzioni, O.: Named Entity Recognition in Tweets: An Experimental Study. In: *EMNLP 2011* (2011)
19. Said, A., Jain, B.J., Narr, S., Plumbaum, T.: Users and noise: The magic barrier of recommender systems. In: Masthoff, J., Mobasher, B., Desmarais, M.C., Nkambou, R. (eds.) *UMAP 2012*. LNCS, vol. 7379, pp. 237–248. Springer, Heidelberg (2012)
20. Smola, A., Vishwanathan, S.V.N.: *Introduction to Machine Learning*. Cambridge University Press (2010)

21. Mladenic, D., Stajner, T., Novalija, I.: Informal sentiment analysis in multiple domains for English and Spanish. In: IS 2012 (2012)
22. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-Based Methods for Sentiment Analysis. Computational Linguistics 37(2), 267–307 (2011)
23. Tromp, E.: Multilingual Sentiment Analysis on Social Media. Master's thesis, Eindhoven University of Technology (2011)
24. Trusov, M., Bucklin, R.E., Pauwels, K.H.: Effects of Word of Mouth Versus Traditional Marketing: Findings from an Internet Social Networking Site. Journal of Marketing 73(5), 90–102 (2009)
25. Whitelaw, C., Garg, N., Argamon, S.: Using appraisal groups for sentiment analysis. In: CIKM 2005, pp. 625–631 (2005)

Automata on Directed Graphs for the Recognition of Assembly Lines

Antonios Kalampakas^{1,2}, Stefanos Spartalis¹, and Lazaros Iliadis³

¹ Department of Production Engineering and Management,
Laboratory of Computational Mathematics, School of Engineering,
Democritus University of Thrace, V.Sofias 12, Prokat,
Building A1, 67100 Xanthi, Greece
akalampakas@gmail.com, sspart@pme.duth.gr

² American University of the Middle East, Block 3,
Building 1, Egaila, Kuwait

³ Department of Forestry & Management of the Environment & Natural Resources,
Democritus University of Thrace, 193 Pandazidou st.,
68200, Nea Orestiada, Greece
liliadis@fmenr.duth.gr

Abstract. Automata on general directed hypergraphs have been recently introduced based on the algebraic representation of hypergraphs inside graphoids. In this paper, we employ this mechanism for the construction of automata recognizing mixed model assembly lines.

Keywords: Recognizability, Graph Automata, Mixed Model Assembly Lines.

1 Introduction

Automata on general hypergraphs were constructed in [6] by utilizing the algebraic properties of graphoids, i.e., magmoids satisfying the 15 fundamental equations which are specified in [5]. A magmoid is a doubly ranked set equipped with two operations which are associative, unitary and mutually coherent in a canonical way [1]. This algebraic structure can be engaged for the finite generation of hypergraphs, in a role similar to that of monoids for the generation of strings.

Hypergraphs are a generalization of the usual directed graphs. They consist of a set of nodes and a set of hyperedges, just as ordinary directed graphs except that a hyperedge may have an arbitrary sequence of sources and an arbitrary sequence of targets. Each hyperedge is labeled with a symbol from a doubly ranked alphabet Σ in such a way that the first (second) rank of its label equals the number of its sources (targets respectively). Also, every hypergraph is equipped with a sequence of “begin” and “end” nodes.

In [10] Engelfriet and Vereijken proved that every hypergraph with edges labeled over a finite doubly ranked set Σ can be built from a specific finite set of elementary hypergraphs D , together with the elements of Σ , by using the

operations of graph product and graph sum. From this result it is derived that hypergraphs can be organized into a magmoid with operations product and sum. Although, as it was observed in [10], every hypergraph can be constructed in an infinite number of ways, this ambiguity was settled in [5] by determining a *finite* set of equations \mathcal{E} with the property that two expressions represent the same hypergraph, if and only if, one can be transformed into the other through them.

Commencing from this result, a *graphoid* \mathbf{M} is defined as a magmoid with a designated set of elements that satisfy the equations \mathcal{E} . Hence $GR(\Sigma)$ can be structured into a graphoid by virtue of the set D of elementary hypergraphs. The *relational magmoid* over a set Q is constructed by defining the operations of composition and sum on the set of relations from Q^m to Q^n . This set is structured into a *relational graphoid* over Q , by specifying a set D of relations that satisfy the equations \mathcal{E} . A relational graphoid is called abelian when a particular relation of D consists of all the transpositions in Q . In [6] graph automata, with state set Q , were introduced by virtue of a specific abelian relational graphoid and by exploiting the fact that $GR(\Sigma)$ is the free graphoid generated by Σ . In the same paper it is postulated that different kinds of graphoids will produce graph automata with diverse operation and recognition capacity. In [8] it is shown that all abelian relational graphoids are characterized in the following way: a set Q generates an abelian relational graphoid if and only if Q is partitioned into disjoint abelian groups with operations univocally instructed by D .

In the present paper we employ the theory of graph automata for the recognition of mixed-model assembly lines. Mixed-model assembly lines are found in many industrial environments and are gradually replacing the traditional single model production systems due to the growing trend for product variability and shorter life cycle [7]. In mixed-model assembly lines different products are jointly manufactured in intermixed product sequences. As a consequence, typically, all models are variations of the same base product and only differ in specific customizable product attributes, also referred to as options [4]. The design of such a system requires the assignment of specific tasks to workstations with respect to a given set of precedence constraints. The so obtained assembly sequence is visualized by using directed graphs, traditionally called *precedence graphs*, first developed by Salveson [13].

An important challenge, in connection to this, is the construction of mixed-model assembly lines by partitioning the set of tasks and associating the classes with workstations in a way that the given cycle time and the precedence relations are respected. For this, we present the construction of a graph automaton recognizing mixed-model assembly lines which consist of a specific set of tasks and a corresponding set of task times, a fixed cycle time, a set of *final* tasks, required for the completion of the model, and the associated precedence graph. Such a machine shall allow us to construct assembly lines satisfying the required characteristics and can additionally be employed for production optimization with regard to the cycle time and the number of stations.

In Section 2 we present the basic definitions for the algebraic structure we employ and we construct the magmoid of hypergraphs. In the following section we present graph automata by virtue of relational graphoids. In Section 4 we apply this mechanism for the recognition of mixed-model assembly lines. The operation of the presented graph automaton is delineated by a relevant example.

2 Magmoids and Hypergraphs

A doubly ranked set $(A_{m,n})_{m,n \in \mathbb{N}}$ is a set A together with a function $\text{rank} : A \rightarrow \mathbb{N} \times \mathbb{N}$ we set $A_{m,n} = \{a \in A \mid \text{rank}(a) = (m, n)\}$. In what follows we will drop the subscript and denote a doubly ranked set simply by $A = (A_{m,n})$. A *magmoid* is a doubly ranked set $M = (M_{m,n})$ equipped with two operations

$$\circ : M_{m,n} \times M_{n,k} \rightarrow M_{m,k}, \quad \square : M_{m,n} \times M_{m',n'} \rightarrow M_{m+m',n+n'},$$

which are associative in the obvious way, satisfy the distributivity law

$$(f \circ g) \square (f' \circ g') = (f \square f') \circ (g \square g')$$

whenever all the above operations are defined and are equipped with a sequence of constants $e_n \in M_{n,n}$, called units, such that

$$e_m \circ f = f = f \circ e_n, \quad e_0 \square f = f = f \square e_0, \quad e_m \square e_n = e_{m+n}$$

hold for all $f \in M_{m,n}$ and all $m, n \in \mathbb{N}$. Notice that, due to the last equation, the elements e_n are uniquely determined by e_1 . From now on e_1 will be simply denoted by e . The free magmoid $\text{mag}(\Sigma)$ generated by a doubly ranked set Σ is constructed in [5]. The sets $\text{Rel}_{m,n}(Q)$ of all relations from Q^m to Q^n

$$\text{Rel}_{m,n}(Q) = \{R \mid R \subseteq Q^m \times Q^n\}$$

can be structured into a magmoid with \circ being the usual relation composition and \square defined as follows: for $R \in \text{Rel}_{m,n}(Q)$ and $S \in \text{Rel}_{m',n'}(Q)$

$$R \square S = \{(u_1 u_2, v_1 v_2) \mid (u_1, v_1) \in R \text{ and } (u_2, v_2) \in S\},$$

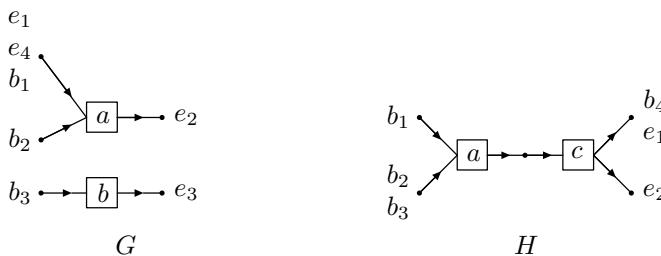
where $u_1 \in Q^m$, $u_2 \in Q^{m'}$, $v_1 \in Q^n$, $v_2 \in Q^{n'}$. Notice that $Q^0 = \{\varepsilon\}$, where ε is the empty word of Q^* . The units are given by $e_0 = \{(\varepsilon, \varepsilon)\}$ and $e = \{(g, g) \mid g \in Q\}$. We denote by $\text{Rel}(Q) = (\text{Rel}_{m,n}(Q))$ the magmoid constructed in this way and call it the *relational magmoid* of Q .

An (m, n) -hypergraph $G = (V, E, s, t, l, \text{begin}, \text{end})$ with edge labels from a doubly ranked set $\Sigma = (\Sigma_{m,n})$ is a tuple consisting of the set of nodes or vertices V , the set of edges E , the source and target functions $s : E \rightarrow V^+$ and $t : E \rightarrow V^+$ respectively, the labeling function $l : E \rightarrow \Sigma$ such that $\text{rank}(l(e)) = (|s(e)|, |t(e)|)$, for all $e \in E$, and the sequences of begin and end nodes $\text{begin} \in V^*$ and $\text{end} \in V^*$ with $|\text{begin}| = m$ and $|\text{end}| = n$. Notice that vertices can be

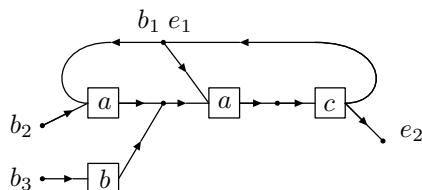
duplicated in the begin and end sequences of the graph and also at the sources and targets of the edges. Isomorphism between two graphs is defined in the obvious way and we shall not distinguish between two isomorphic graphs. The set of all (m, n) -graphs over Σ is denoted by $GR_{m,n}(\Sigma)$ and we set $GR(\Sigma) = (GR_{m,n}(\Sigma))_{m,n \in \mathbb{N}}$. Ordinary unlabeled directed graphs are obtained as a special case of hypergraphs i.e., in the case that each hyperedge is binary (has one source and one target), every edge has the same label and the sequences *begin* and *end* are the empty word.

If G is the (m, n) -graph $(V, E, s, t, l, \text{begin}, \text{end})$ and H is the (n, k) -graph $(V', E', s', t', l', \text{begin}', \text{end}')$ then their *product* $G \circ H$ is the (m, k) -graph that is obtained by taking the disjoint union of G and H and then identifying the i^{th} end node of G with the i^{th} begin node of H , for every $i \in \{1, \dots, n\}$; also, $\text{begin}(G \circ H) = \text{begin}(G)$ and $\text{end}(G \circ H) = \text{end}(H)$. The *sum* $G \square H$ of arbitrary graphs G and H is their disjoint union with their sequences of begin nodes concatenated and similarly for their end nodes.

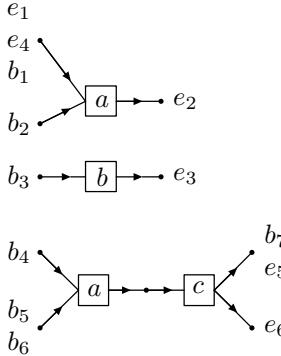
For instance let $\Sigma = \{a, b, c\}$, with $\text{rank}(a) = (2, 1)$, $\text{rank}(b) = (1, 1)$ and $\text{rank}(c) = (1, 2)$. In the following pictures, edges are represented by boxes, nodes by dots, and the sources and targets of an edge by directed lines that enter and leave the corresponding box, respectively. The order of the sources and targets of an edge is the vertical order of the directed lines as drawn in the pictures. We display two graphs $G \in GR_{3,4}(\Sigma)$ and $H \in GR_{4,2}(\Sigma)$, where the i^{th} begin node is indicated by b_i , and the i^{th} end node by e_i .



Then their product $G \circ H$ is the $(3, 2)$ -graph



and, their sum $G \square H$ is the $(7, 6)$ -graph



For every $n \in \mathbb{N}$ we denote by E_n the discrete graph of rank (n, n) with nodes x_1, \dots, x_n and $\text{begin} = \text{end} = x_1 \dots x_n$; we write E for E_1 . It is straightforward to verify that $GR(\Sigma) = (GR_{m,n}(\Sigma))$ with the operations defined above is a magmoid, whose units are the graphs E_n .

3 Graphoids and Graph Automata

Now we present graph automata by employing the algebraic structure of graphoids as introduced in [6]. We denote by $I_{p,q}$ the discrete (p, q) -graph that has a single node x and whose begin and end sequences are $x \dots x$ (p times) and $x \dots x$ (q times) respectively, Π is the discrete $(2, 2)$ -graph that has two nodes x and y and whose begin and end sequences are xy and yx , respectively, also for every $\sigma \in \Sigma_{m,n}$, we denote again by σ the (m, n) -graph having only one edge and $m + n$ nodes $x_1, \dots, x_m, y_1, \dots, y_n$. The edge is labeled by σ , and the begin (resp. end sequence) of the graph is the sequence of sources (resp. targets) of the edge, viz. $x_1 \dots x_m$ (resp. $y_1 \dots y_n$).

| | | | |
|--|---|--|----------|
| $I_{p,q}$ | Π | E_n | σ |
| $\begin{array}{c} b_1 \cdot e_1 \\ \vdots \cdot \vdots \\ b_p \cdot e_q \end{array}$ | $\begin{array}{c} b_1 \cdot e_2 \\ b_2 \cdot e_1 \end{array}$ | $\begin{array}{c} b_1 \cdot e_1 \\ \vdots \cdot \vdots \\ b_n \cdot e_n \end{array}$ | |

Engelfriet and Vereijken, in [10], presented an algorithm that inductively constructs every graph $G \in GR(\Sigma)$ from the set $\Sigma \cup \{\Pi, I_{01}, I_{21}, I_{10}, I_{12}, \}\}$ by using graph product and graph sum. However, there are infinitely many ways to construct a given graph. This was overridden by identifying a finite set \mathcal{E} of equations with the property that two expressions represent the same graph if and only if one can be transformed into the other through these equations [5]. It is evident from this discussion that the equations \mathcal{E} are satisfied in $GR(\Sigma)$. Magmoids with such a property are called *graphoids*. Formally, a graphoid $\mathbf{M} = (M, D)$ consists of a magmoid M and a set $D = \{s, d_{01}, d_{21}, d_{10}, d_{12}\}$, with $s \in M_{2,2}$, $d_{\kappa\lambda} \in M_{\kappa,\lambda}$, such that the following equations hold:

$$s \circ s = e_2 \quad (1) \quad (s \square e) \circ (e \square s) \circ (s \square e) = (e \square s) \circ (s \square e) \circ (e \square s) \quad (2)$$

$$(e \square d_{21}) \circ d_{21} = (d_{21} \square e) \circ d_{21} \quad (3) \quad (e \square d_{01}) \circ d_{21} = e \quad (4)$$

$$s \circ d_{21} = d_{21} \quad (5) \quad (e \square d_{01}) \circ s = (d_{01} \square e), \quad (6)$$

$$(s \square e) \circ (e \square s) \circ (d_{21} \square e) = (e \square d_{21}) \circ s, \quad (7)$$

$$d_{12} \circ (e \square d_{12}) = d_{12} \circ (d_{12} \square e) \quad (8) \quad d_{12} \circ (e \square d_{10}) = e, \quad (9)$$

$$d_{12} \circ s = d_{12} \quad (10) \quad s \circ (e \square d_{10}) = (d_{10} \square e), \quad (11)$$

$$(d_{12} \square e) \circ (e \square s) \circ (s \square e) = s \circ (e \square d_{12}), \quad (12)$$

$$d_{12} \circ d_{21} = e \quad (13) \quad (d_{12} \square e) \circ (e \square d_{21}) = d_{21} \circ d_{12} \quad (14)$$

$$s_{m,1} \circ (f \square e) = (e \square f) \circ s_{n,1}, \quad \text{for all } f \in mag_{m,n}(\Sigma \cup D), \quad (15)$$

where $s_{m,1}$ is defined inductively by s and represents the graph associated with the permutation that interchanges the last n numbers with the first one [5]. We point out that although (15) is a set of equations it only has to be valid for the elements of Σ in order to hold for every element of a magmoid generated by Σ [5]. Thus the pair $\mathbf{GR}(\Sigma) = (GR(\Sigma), D)$, with $D = \{\Pi, I_{01}, I_{21}, I_{10}, I_{12}\}$ is a graphoid and in fact it is the free graphoid generated by Σ as it is illustrated in [6]. Given graphoids (M, D) and (M', D') , a magmoid morphism $H : M \rightarrow M'$ preserving D -sets, i.e., $H(s) = s'$ and $H(d_{\kappa\lambda}) = d'_{\kappa\lambda}$, is called a morphism of graphoids.

Graphoids constructed from the magmoid of relations $Rel(Q)$ over a given set Q are called *relational graphoids* and a relational graphoid is called abelian when $s = \{(g_1 g_2, g_2 g_1) \mid g_1, g_2 \in Q\}$. The abelian relational graphoid $\mathbf{TSRel}(Q) = (Rel(Q), D)$ that was used for the introduction of graph automata is constructed by setting s as above and $d_{01} = \{(\varepsilon, g) \mid g \in Q\}$, $d_{21} = \{(gg, g) \mid g \in Q\}$, $d_{10} = \{(g, \varepsilon) \mid g \in Q\}$, $d_{12} = \{(g, gg) \mid g \in Q\}$.

A *relational graph automaton* is a structure

$$\mathcal{A} = (\Sigma, Q, \mathbf{Rel}(Q), \delta, I, T),$$

where Σ is the doubly ranked set of hyperedge labels, Q is the finite set of states, $\mathbf{Rel}(Q)$ is a relational graphoid over Q , $\delta : \Sigma \rightarrow \mathbf{Rel}(Q)$ is the doubly ranked transition function and I, T are initial and final rational subsets of Q^* . The function δ is uniquely extended into a morphism of graphoids $\bar{\delta} : GR(\Sigma) \rightarrow \mathbf{Rel}(Q)$, where $\bar{\delta}(I_{\kappa\lambda}) = d_{\kappa\lambda}$ and $\bar{\delta}(\Pi) = s$, and the behavior of \mathcal{A} is given by

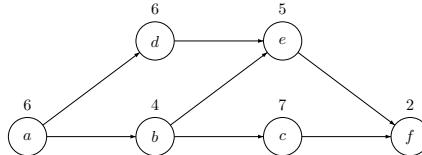
$$|\mathcal{A}| = \{F \mid F \in GR_{m,n}(\Sigma), \bar{\delta}_{\mathcal{A}}(F) \cap (I_{\mathcal{A}}^{(m)} \times T_{\mathcal{A}}^{(n)}) \neq \emptyset, m, n \in \mathbb{N}\}$$

where $I_{\mathcal{A}}^{(m)} = I_{\mathcal{A}} \cap Q^m$ and $T_{\mathcal{A}}^{(n)} = T_{\mathcal{A}} \cap Q^n$. From their construction, graph automata are finite machines due to the fact that the set of equations (1)-(15) is finite. A graph language is called recognizable whenever it is obtained as the behavior of a graph automaton. The class of all such languages over the doubly ranked set Σ is denoted by $Rec(\Sigma)$.

4 Application on Mixed-Model Assembly Lines

In this section we construct a relational graph automaton recognizing Mixed-Model Assembly Lines. An assembly line consists of workstations usually arranged along a conveyor belt or a similar material handling equipment. The jobs are consecutively launched down the line and moved from station to station while at each station certain operations (tasks) are repeatedly performed. Hence, the total amount of work necessary to assemble a work piece (job) is split up into a set of tasks.

The fundamental problem in designing an assembly line is the assignment of tasks to workstations while respecting a given set of precedence relations, see [9] for a review on the *assembly line balancing problem*. This consist of grouping the tasks in a particular way that balances the workload over a number of workstations. Traditionally an assembly sequence is visualized by virtue of *precedence graphs*, first developed by Salveson [13]. The role of the precedence graph is to instruct the particular way the product has to be assembled, e.g. a bottle can't be filled when the cap is already on.

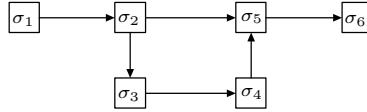


A precedence graph with 6 tasks and corresponding task times between 2 and 7 time units

Performing a task r_i takes a time $t(r_i)$ and requires certain equipment of machines and/or skills of workers. The set of tasks σ_k assigned to a station (denoted also σ_k) constitutes its station load or work content, the cumulated task time $t(\sigma_k) = \sum_{r_i \in \sigma_k} t(r_i)$ is called station time. When a fixed cycle time c is given (paced line), a line balance is feasible only if the station time of neither station exceeds c . In the case that $t(\sigma_k) < c$, the station has an idle time $c - t(\sigma_k)$ in each cycle. For the above precedence graph, a feasible line balance with cycle time $c = 11$ and 3 stations is: $\sigma_1 = \{a, b\}$, $\sigma_2 = \{d, e\}$, $\sigma_3 = \{c, f\}$.

In order to meet the diversification of consumer's preferences, mixed-model assembly lines are increasingly installed in manufacturing plants. For example, car manufacturers offer their cars in a huge number of models, which can be configured by combining the options offered and the (theoretical) number of the corresponding models grows exponentially as a function of the number of options

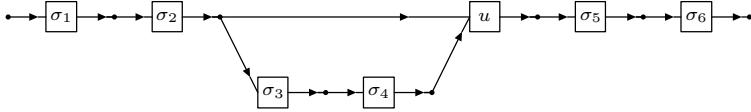
offered [11]. The large variation reduces production efficiency and may cause a line stoppage but it can be reduced by installing bypass sublines which process portions of assembly operations of products with different assembly times [14].



An assembly line \mathcal{L} with a subline

For representing this kind of mixed model assembly lines we shall use $(1, 1)$ -graphs over a specified doubly ranked alphabet. In order to create sublines we will use the elementary graph $I_{1,2}$, and in order to reunite them we will use a fixed hyperedge with label u and rank $(2, 1)$. All the rest edges have rank $(1, 1)$, i.e. they are not hyperedges, and will represent the stations of the assembly line.

For example, the above assembly line is represented by a graph as follows.



The hypergraph representation of the assembly line \mathcal{L}

Now assume that we want to construct an automaton that recognizes mixed-model assembly lines which produce at most k different models with set of tasks $R = \{r_1, \dots, r_n\}$, corresponding task times $t(r_i)$, $1 \leq i \leq n$ cycle time $c \in \mathbb{N}^*$ in time units, and a given precedence graph \mathcal{G} . We denote by $pr(r_i) \in \mathcal{P}(R)$, $1 \leq i \leq n$, the set of all tasks that precede r_i in \mathcal{G} and we set

$$pr(\sigma) = \bigcup_{r_i \in \sigma} pr(r_i).$$

Moreover, a model of the assembly line is finalized only if at least one task of a specified set of tasks $L \in \mathcal{P}(R)$ has been completed. The automaton which recognizes the mixed-model assembly line with the above characteristics is

$$\mathcal{A} = (\Sigma, Q, Rel(Q), \delta, \emptyset, T_L)$$

and consists of

- the doubly ranked alphabet $\Sigma = \Sigma_{1,1} \cup \Sigma_{2,1}$, where $\Sigma_{2,1} = \{u\}$ and

$$\Sigma_{1,1} = \{\sigma \mid \sigma \in \mathcal{P}(R) \text{ with } \sum_{r_i \in \sigma} t_i \leq c\},$$

- the state set $Q = \mathcal{P}(R) \cup \mathcal{P}(R)^2 \cup \dots \cup \mathcal{P}(R)^k$,
- the relational graphoid $Rel(Q)$,

- the transition function δ given by

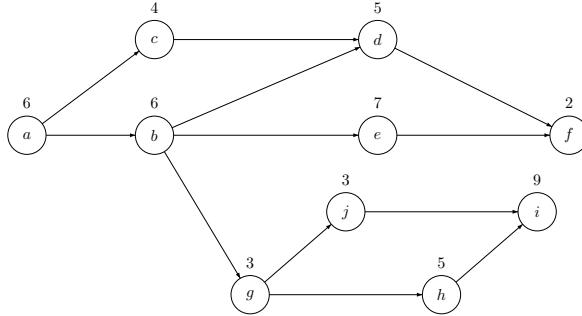
$$\begin{aligned}\delta(\sigma) &= \{(p_1 \cdots p_s, q_1 \cdots q_s) \mid p_i, q_i \in \mathcal{P}(R), \\ pr(\sigma) &\subseteq p_i, q_i = p_i \cup \sigma, 1 \leq s \leq k, 1 \leq i \leq s\}\end{aligned}$$

for all $\sigma \in \Sigma_{1,1}$ and

$$\begin{aligned}\delta(u) &= \{((p_1 \cdots p_s, q_1 \cdots q_r), p_1 \cdots p_s q_1 \cdots q_r) \mid p_i, q_i \in \mathcal{P}(R), \\ 1 \leq i \leq s, s+r &\leq k\}\end{aligned}$$

- the initial state $I = \emptyset$,
- the final state set $T_L = \{p_1 \cdots p_s \mid p_i \in \mathcal{P}(R), p_i \cap L \neq \emptyset, 1 \leq i \leq s, s \leq k\}$.

We illustrate the operation of the assembly line automaton with the construction of an automaton which recognizes all mixed model assembly lines with at most 2 different models, set of tasks $R = \{a, b, c, d, e, f, g, h, i, j\}$, cycle time 11 time units, final task $L = \{f\}$, and precedence relations as delineated in the following precedence graph \mathcal{G}_a where the related task times are also designated.

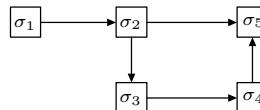


The precedence graph \mathcal{G}_a

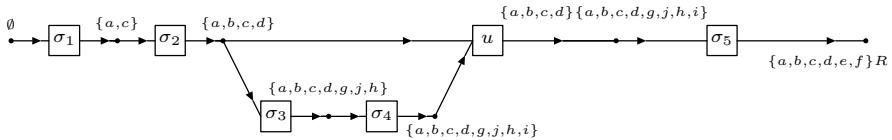
From the above discussion it turns out that the corresponding automaton is $\mathcal{A}_a = (\Sigma, Q, Rel(Q), \delta, \emptyset, T_L)$, where

- $\Sigma_{1,1} = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g\}, \{h\}, \{i\}, \{j\}, \{a, c\}, \{b, d\}, \dots\}$,
- $Q = \mathcal{P}(R) \cup \mathcal{P}(R)^2$,
- $\delta(\{a, c\}) = \{(\emptyset, \{a, c\}), (\emptyset\emptyset, \{a, c\}\{a, c\})\}$,
- $\delta(\{b, d\}) = \{(\{a, c\}, \{a, b, c, d\}), (\{a, c\}\{a, c\}, \{a, b, c, d\}\{a, b, c, d\})\}$, etc.
- $T_L = \{p, pq \mid p, q \in \mathcal{P}(R), f \in p \cap q\}$.

It is easy to verify that, for example, the assembly line



where $\sigma_1 = \{a, c\}$, $\sigma_2 = \{b, d\}$, $\sigma_3 = \{g, j, h\}$, $\sigma_4 = \{i\}$, $\sigma_5 = \{e, f\}$ belongs to the behavior of the automaton \mathcal{A}_a . Indeed, the corresponding states at every node of the hypergraph representation of this assembly line are the following.



5 Discussion

We have constructed a particular instance of a graph automaton able to recognize mixed-model assembly lines represented as hypergraphs. This mechanism can be applied for the design and verification of assembly lines as well as for optimization purposes. Indeed, we note that the set of task times, the cycle time, the set of final tasks and the precedence graph are *a priori* specified. As a result, by alternating one or more of these characteristics we obtain automata with diverse behavior and hence different sets of assembly lines. It is also noteworthy that for the representation of a mixed-model assembly line in this setup, an ordinary graph does not suffice since we have to use at least one hyperedge (labeled u , see the construction of Section 4). More generally, we emphasize that this result uncovers the effectiveness of graph automata for providing solutions in real world applications where the basic object of interest can be represented as a graph, including natural language processing [12] and formal verification [2,3].

References

1. Arnold, A., Dauchet, M.: Théorie des magmoïdes. I and II. *RAIRO Theoret. Inform. Appl.* 12, 235–257 (1978); 13, 135–154
2. Blume, C.: Recognizable Graph Languages for the Verification of Dynamic Systems. In: Ehrig, H., Rensink, A., Rozenberg, G., Schürr, A. (eds.) *ICGT 2010. LNCS*, vol. 6372, pp. 384–387. Springer, Heidelberg (2010)
3. Blume, C., Bruggink, H.J.S., Friedrich, M., König, B.: Treewidth, pathwidth and cospan decompositions with applications to graph-accepting tree automata. *Journal of Visual Languages & Computing* 24, 192–206 (2013)
4. Boysen, N., Fliedner, M., Scholl, A.: Assembly line balancing: Joint precedence graphs under high product variety. *IIE Transactions* 41, 183–193 (2009)
5. Bozapalidis, S., Kalampakas, A.: An Axiomatization of Graphs. *Acta Inform.* 41, 19–61 (2004)
6. Bozapalidis, S., Kalampakas, A.: Graph Automata. *Theoret. Comput. Sci.* 393, 147–165 (2008)
7. Bukchin, J., Dar-El, E.M., Rubinovitz, J.: Mixed model assembly line design in a make-to-order environment. *Computers and Industrial Engineering* 41, 405–421 (2001)
8. Kalampakas, A.: Graph Automata: The Algebraic Properties of Abelian Relational Graphoids. In: Kuich, W., Rahonis, G. (eds.) *Algebraic Foundations in Computer Science. LNCS*, vol. 7020, pp. 168–182. Springer, Heidelberg (2011)
9. Kriengkorakot, N., Pianthong, N.: The Assembly Line Balancing Problem: Review articles. *KKU Engineering Journal* 34, 133–140 (2007)

10. Engelfriet, J., Vereijken, J.J.: Context-free graph grammars and concatenation of graphs. *Acta Informatica* 34, 773–803 (1997)
11. Pil, F.K., Holweg, M.: Linking product variety to order-fulfillment strategies. *Interfaces* 34, 394–403 (2004)
12. Quernheim, D., Knight, K.: Towards Probabilistic Acceptors and Transducers for Feature Structures. In: Proc. 6th Workshop Syntax, Semantics and Structure in Statistical Translation, pp. 76–85. Association for Computational Linguistics (2012)
13. Salveson, M.E.: The Assembly Line Balancing Problem. *Journal of Industrial Engineering* 6, 18–25 (1955)
14. Tamura, T., Long, H., Ohno, K.: A sequencing problem to level part usage rates and work loads for a mixed-model assembly line with a bypass sub-line. *International Journal of Production Research* 60-61, 557–564 (1999)

On the Quantification of Missing Value Impact on Voting Advice Applications

Marilena Agathokleous, Nicolas Tsapatsoulis, and Ioannis Katakis

30, Arch. Kyprianos str., CY-3036, Limassol, Cyprus
mi.agathokleous@edu.cut.ac.cy, nicolas.tsapatsoulis@cut.ac.cy,
ioannis.katakis@gmail.com
<http://nicolast.tiddlyspot.com>

Abstract. Voting Advice Application (VAA) is a web application that recommends a candidate or a party to a voter. From an online questionnaire, which voters and candidates are called to answer, the VAA proposes to each individual voter the candidate who replied like him/her. It is very important the voters to reply in all questions of the questionnaire, because every question has its meaning and is responding to the political position of a each party. Missing values might mislead the VAA and impede it to have complete knowledge about the voter, as a result to offer him/her the wrong candidate. In this paper we quantitatively investigate the effect of missing values in VAAs by examining the impact of the number of missing values to different methods of voting prediction. For our experiment we have used the data obtained from the May parliamentary elections in Greece in 2012. The corresponding dataset is made freely available to other researchers working in the areas of VAA and recommender systems through the Web.

Keywords: Missing values, classifiers, recommender systems, voting advice applications.

1 Introduction

The Voting Advice Application (VAA), is an online survey tool that has recently been successfully used in many European countries and lately in countries outside the European continent [8]. It has been characterized as a new phenomenon in modern election campaigning and its utilization has increased in recent years [2]. VAAs' target is to help voters identify parties that have similar political positions with themselves. Also VAAs encourage, indirectly, citizens to exercise their right to vote and to be informed about the political stances of parties.

The operation of a VAA is briefly summarized as follows: First, parties (through representatives) are called to answer a set of questions in an online questionnaire. This questionnaire typically consists of a set of policy statements on which the parties' positions have been coded. Each question corresponds to the political positions of the parties and their reaction to the developments in the current affairs. Preferred questions are those encoding a high variance in party

positions. Once the party answers on the questionnaire were encoded citizens are able to fill in the same questionnaire by navigating to the VAA website. Both voters and candidates (party representatives) evaluate each issue by giving lower extent to those with which they do not agree at all and higher to those that perfectly expressed their position [2][6]. Usually the answering options are ‘strongly disagree’, ‘disagree’, ‘neither agree nor disagree’, ‘agree’, ‘strongly agree’ and ‘I have no opinion’. In the end, the similarity between voters and candidates is calculated, and, with the aid of a properly designed algorithm every voter is recommended the candidate with whom he/she has the higher similarity. In the following sections of this paper this, traditional, method of voting recommendation will be referred as ‘Party Coding’.

Recently, Katakis *et al.* [5] proposed an alternative method of voting recommendation in VAAs in an effort to extend their social dimension and transform them into the so-called Social VAAs (SVAAs). In SVAAs in addition to the traditional recommendation which is based on party-voter similarity, voters are recommended candidates using a collaborative filtering perspective [10] by considering SVAAs a special type of recommender systems [9]. The overall idea behind this is that voters are recommended candidates based on their similarity with other voters who expressed their voting intention. In most VAAs the voting intention of the users is recorded by asking them a corresponding complimentary question (see www.choose4greece.com and www.choose4cyprus.com). On average, 70% of the users answer this question while about 50% of them express voting intention for a specific party or candidate. Katakis gets advantage of these data and proposes several voting prediction schemes by modeling party voters using machine learning tools. He showed that party voter modeling methods are superior to party coding as far as the voting prediction is concerned and the inclusion of this type of prediction in VAAs, as an additional component, increases the enjoyment of VAA users.

Despite the recommendation method (party coding or party voter modeling) the recommendation given to the voters must be accurate. This happens whenever citizens reply in all questions of the online questionnaire, without leaving missing values. Similarly the information about candidates’ or party positions is required to be more wide, which will be fact if the candidates do not leave missing values too. Unfortunately, in many times the candidates refrain from exposing themselves to controversial issues, choosing answers in the middle of the Likert scale (i.e., ‘neither agree nor disagree’) while the voters leave unanswered several questions or give answers like ‘I have no opinion’ or ‘neither agree nor disagree’ for several reasons, including time constraints, limited information on the corresponding issues, or even due to unclear questions. Furthermore, sometimes voters forget or avoid answering a question. Also there are questions that are characterized of ambivalence or indecisiveness and those with which the respondent does not agree against the main assumptions of them [1]. These values can be characterized as missing values and, in theory, it is considered that they seriously affect VAAs voting recommendation.

In this paper we try to quantify the effect of missing values on VAAs voting recommendation effectiveness both for the traditional party coding method as well as for the party voter modeling methods. By gradually increasing, randomly, the number of missing values we measure the average precision and recall values for the various VAA recommendation methods. In this way we identify in practice the robustness of each recommendation method while we prove empirically to which extent missing values affect VAAs performance. To the best of our knowledge it is the first time that both missing values' impact quantification and recommendation method robustness are dealt with. Conclusions on these two aspects are important because they seriously affect VAAs design and VAA data utilization. The experiments were conducted on the VAA dataset obtained from Greek parliamentary elections of May, 2012, which is online available at www.choose4greece.com/datasets.

2 Problem Formulation

The basic aim of a VAA is to recommend parties of candidates to users. In such a case there is a set of N users $U = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N\}$, a set of M questions (or issues) $Q = \{q_1, q_2, \dots, q_M\}$, and a set of T political parties (or candidates) $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_T\}$. Each user $\mathbf{u}_i \in U$ and each political party $\mathbf{p}_j \in P$, has answered each question $q_k \in Q$. The answers of users are recorded through online questionnaires like the one in Choose4Greece (www.choose4greece.com). The answers of political parties are either coded by experts or answered by representatives of political parties.

Based on their answers, every political party or user can be represented in a vector space model:

$$\mathbf{u}_i = \{u_{(i,1)}, u_{(i,2)}, \dots, u_{(i,k)}, \dots, u_{(i,M)}\} \quad (1)$$

$$\mathbf{p}_j = \{p_{(j,1)}, p_{(j,2)}, \dots, p_{(j,k)}, \dots, p_{(j,M)}\} \quad (2)$$

where $u_{(i,k)}, p_{(j,k)} \in L$ are the answers of the i -th user and j -th party, respectively, to the k -th question. Usually, vectors \mathbf{u}_i and \mathbf{p}_j are named *profiles*.

A typical set of answers is a 6-point Likert scale: $L = \{-2 \text{ (Strongly disagree)}, -1 \text{ (Disagree)}, 0 \text{ (Neither agree nor disagree)}, 1 \text{ (Agree)}, 2 \text{ (Strongly agree)}, 3 \text{ (No opinion)}\}$ but in practice the sixth point it is not taken into consideration since does not correspond to a particular stance. As a result the set L , in the context of this work, becomes: $L = \{-2, -1, 0, 1, 2\}$.

The VAA recommendation *task* can be then defined as follows: Given the answers of a specific user \mathbf{u}_a suggest a ranking of political parties based on user-party relevance. In machine learning terms, the task is to approximate the hidden function $h : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}$, where $h(\mathbf{u}, \mathbf{p})$ is the estimation of the relevance of user \mathbf{u} with political party \mathbf{p} . Typically $h(\mathbf{u}, \mathbf{p}) \in [0, 1]$. In each case, the top suggestion p_a for user u_a should be:

$$p_a = \underbrace{\operatorname{argmax}_j}_{j} (h(\mathbf{u}_a, \mathbf{p}_j)) \quad (3)$$

Similarly, we could consider a function $r(\mathbf{u}, \mathbf{p}) \in [1, T]$ that returns the *rank* of the political party p for the user u , if all political parties are ranked according to relevance (similarity) with this specific user. Having learned function $h(\mathbf{u}, \mathbf{p})$ it is straightforward to calculate $r(\mathbf{u}, \mathbf{p})$.

In order to produce vote recommendations, the most simple approach is to estimate $h(\mathbf{u}, \mathbf{p})$ with the aid of a distance measure $d(\mathbf{u}, \mathbf{p})$. A number of such distance measures are discussed in [7]. In this case the top suggestion p_a for user u_a is given by:

$$p_a = \underbrace{\operatorname{argmin}_j}_{j} (d(\mathbf{u}_a, \mathbf{p}_j)) \quad (4)$$

The above mentioned recommendation approach is the one, traditionally, used in VAAs and in the context of this paper is referred to as ‘Party Coding’ method.

In many voting assistance systems, the information of *vote intention* v_i of many users \mathbf{u}_i is available as it is included as a supplementary question in the online surveys. This kind of information can be utilized to model party voters using statistical or machine learning approaches and provide an additional kind of recommendation that is based on collaborative filtering (see [5]). These type of approaches are referred to, in the context of this work, as ‘Party voter modeling’ methods.

In the framework described in this section a missing value is considered every $u_{(i,k)}, p_{(j,k)} \notin L$, that is, any question k left unanswered or given the value ‘I have no opinion’ by either a particular user i or a party representative (or candidate) j . In order to quantify the impact of the number of missing values on the VAA recommendation we need a few more definitions.

Let us define the utility matrix U as an $N \times M$ matrix whose rows are the transposed user vectors \mathbf{u}_i^T , $i = 1 \dots N$. That is:

$$U = \begin{pmatrix} u_{(1,1)} & u_{(1,2)} & \dots & u_{(1,M)} \\ u_{(2,1)} & u_{(2,2)} & \dots & u_{(2,M)} \\ \dots & \dots & \ddots & \dots \\ u_{(N,1)} & u_{(N,2)} & \dots & u_{(N,M)} \end{pmatrix} \quad (5)$$

The sparsity S_U of matrix U is defined as the percentage of missing values in U . That is:

$$S_U = \frac{|O|}{N \cdot M} \quad (6)$$

where $O = \{u_{(i,k)} | u_{(i,k)} \notin L, i = 1 \dots N, k = 1 \dots M\}$ is the set of missing values and $|O|$ is its cardinality.

The impact of missing values in U to VAA recommendations is quantified with the aid of well-known measures defined in information retrieval. In particular Precision, Recall and F-measure are computed for all voters of a particular party and a weighted average is calculated. Let us defined D_q the set of voters that expressed a voting intention for party p_q , that is:

$$D_q = \{u_i | u_i : v_i = p_q\} \quad (7)$$

Let us also define as D the set of users who expressed a voting intention for a particular party (i.e., the set of users that answer the corresponding complementary question in VAA):

$$D = \bigcup_{q=1:T} D_q \quad (8)$$

We also define the set T_q as the set of users who expressed a voting intention for party p_q and the VAA recommendation coincides with their voting intention, and F_q the set of users who expressed a voting intention for a party different than p_q but the VAA recommended them p_q , i.e.,

$$T_q = \{u_i | u_i : v_i = p_q, p_q = \underbrace{\operatorname{argmax}_j}_{j} (h(\mathbf{u}_i, \mathbf{p}_j))\} \quad (9)$$

$$F_q = \{u_i | u_i : v_i \neq p_q, p_q = \underbrace{\operatorname{argmax}_j}_{j} (h(\mathbf{u}_i, \mathbf{p}_j))\} \quad (10)$$

With the aid of the definitions above we can formally define the per party Precision (Pr^q) and Recall (Re^q) measures as follows:

$$Pr^q = \frac{|T_q|}{|T_q| + |F_q|} \quad (11)$$

$$Re^q = \frac{|T_q|}{|D_q|} \quad (12)$$

where $|A|$ denotes the cardinality of set A .

The F-measure for a particular part p_q is defined as usual with the aid of Pr^q and Re^q :

$$Fm^q = \frac{2 \cdot Pr^q \cdot Re^q}{Pr^q + Re^q} \quad (13)$$

The overall Precision (Pr), Recall (Re) and F-measure (Fm) are computed as weighted sums of the per party corresponding quantities:

$$Pr = \frac{1}{|D|} \sum_{q=1:T} (|D_q| \cdot Pr^q) \quad (14)$$

$$Re = \frac{1}{|D|} \sum_{q=1:T} (|D_q| \cdot Re^q) \quad (15)$$

$$Fm = \frac{1}{|D|} \sum_{q=1:T} (|D_q| \cdot Fm^q) \quad (16)$$

It can be easily proved that the overall Recall (Re), computed as indicated above, coincides with the well-known accuracy measure (for a definition of the accuracy measure in VAAs please see [5]).

3 Methodology

Our methodology for the quantification of missing value impact on VAAs recommendation is quite simple: The number of missing values in the utility matrix U (see eq. (5)) are progressively increased by randomly removing matrix entries and the performance of each one of the recommendation methods, mentioned next, is evaluated with the aid of Precision, Recall and F-measure defined in equations (14)-(16). The random selection of matrix entries that are removed guarantees that no specific patterns will be created within matrix U .

The following voting recommendation methods were compared using the dataset collected from www.choose4greece.com. In its initial form it includes 75,294 voters. However, only 26,355 voters expressed voting intention for a particular party. Thus, in order to be able to apply the proposed methodology for the quantification of missing value impact on VAA recommendation we have used this particular subset of the original dataset. The original sparsity (see eq. (6)) of this dataset is negligible (0.0184).

Party Coding: Party coding is the traditional VAA recommendation method. The user u_a is recommended party p_a according to eq. (4). Although alternative distance metrics were proposed for VAAs [7] in this article we stick in, for the sake of simplicity, to the classic Euclidean distance for our experiments.

$$d(\mathbf{u}_a, \mathbf{p}_j) = \sqrt{\sum_{k=1}^M (u_{a,k} - p_{j,k})^2} \quad (17)$$

Party coding is by far the most simple and computationally undemanding method. Furthermore, is the only method that serves the second basic purpose of VAAs: provide information about the party positions.

Average Voter: Average Voter is a simple community-based approach for voting recommendation. It also makes use of eq. (4) with the aid of Euclidean distance. However, the difference is that instead of using the actual profile \mathbf{p}_j of party p_j it utilizes the averaged profile $a(\mathbf{p}_j)$ of all users that belong to set D_j :

$$a(\mathbf{p}_j) = \frac{1}{|D_j|} \sum_{u_i \in D_j} \mathbf{u}_i \quad (18)$$

The advantages of this approach are its simplicity and the fact that it does not need the profile of each political party, since it depends on the voters' answers. On the other hand this can be also its main disadvantage, as it is necessary to have a sufficient number of users that were expressed voting intention for party p_j in order to estimate a representative profile for the average voter.

Mahalanobis Classifier: Mahalanobis Classifier extends the idea of the average voter by taking into account two more pieces of information contained in

the covariance matrix C_j of the voters of party p_j . It gives different weights to the various questions with the aid of the diagonal elements of matrix C_j while it records the correlations between questions through the non-diagonal elements.

The Mahalanobis Classifier makes use of eq. (3) where the relevance $h(\mathbf{u}_a, \mathbf{p}_j)$ of user u_a with political party p_j is computed through the following equation:

$$h(\mathbf{u}_a, \mathbf{p}_j) = e^{-\frac{1}{2}((\mathbf{u}_a - a(\mathbf{p}_j))^T \cdot C_j^{-1} \cdot (\mathbf{u}_a - a(\mathbf{p}_j)))} \quad (19)$$

where $a(\mathbf{p}_j)$ is the average profile of users belonging to set D_j and C_j is the corresponding covariance matrix given by:

$$C_j = \frac{1}{|D_j|} \sum_{u_i \in D_j} (\mathbf{u}_i - a(\mathbf{p}_j)) \cdot (\mathbf{u}_i - a(\mathbf{p}_j))^T \quad (20)$$

Machine Learning Approaches: If we ignore the information of political party profiles, then the problem can be defined as a single-class data classification problem, with the class obviously being the vote intention of the user. The data matrix ($D_{N \times M}$) consists of all the users profiles and the class (label) vector

$$V_{N \times 1} = \{\nu_1, \dots, \nu_N\}^T \quad (21)$$

consists of the vote intentions of all N users. Hence, $D_{N \times M}$ and $V_{N \times 1}$ constitute the training *examples* of the learning problem. In this case many classifiers can be represented as score functions $f(\mathbf{x}, y)$ that output the probability that instance \mathbf{x} belongs to class y (i.e. $P(y|\mathbf{x})$), and $\sum_{y=1}^T f(\mathbf{x}, y) = 1$. Hence, in order to solve the VAA problem, it is straightforward to set $h(\mathbf{u}, p) = f(\mathbf{x}, y)$ and use as f one of the variety of classifiers (Decision Trees Classifiers, Bayesian Classifiers, Support Vector Machines, Neural Networks, etc). In essence what is achieved in this case is the modeling of the voters' behavior based on the questions.

We used the Weka software package [11] to train a variety of indicative classifiers using the ‘one against the rest’ training approach. Datasets were split into training and test sets, with a percentage of 66 percent and 34 percent respectively. We have implemented party voter models using the Naive Bayes classifier (bayes.NaiveBayes), the Logistic Regression classifier (functions.Logistic) and the Instance-Based classifier with fixed neighborhood (lazy.IBk) in order to cover statistical learning, neural networks and decision trees respectively. All these classifiers were set with Weka default parameter settings [3].

4 Experimental Results and Discussion

Experiments were designed to investigate the performance on voting recommendation of the classifiers, mentioned in the previous section, as a function of the number of missing values in the utility matrix U . Tables 1-3 summarize these results on four distinct sparsity values: 0%, 30%, and 50% respectively. Figure 1 on the other hand illustrates graphically the overall F-measure (see eq.(16)) as a function of the number of missing values.

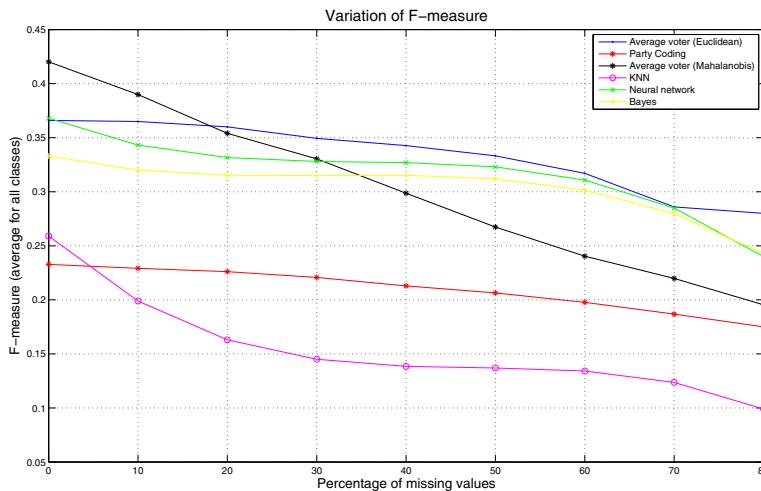


Fig. 1. Variation of F-measure as a function of the number of missing values

Table 1. Voting prediction evaluation on the original Choose4Greece dataset

| Classifier | Precision (average) | Recall (average) | F-measure (average) |
|-------------------------------------|------------------------|---------------------|------------------------|
| Party Coding | 0.268 | 0.206 | 0.233 |
| Average Voter | 0.427 | 0.328 | 0.371 |
| Mahalanobis Classifier | 0.424 | 0.417 | 0.420 |
| KNN (lazy.IBk) | 0.258 | 0.260 | 0.259 |
| Neural Network (functions.Logistic) | 0.357 | 0.399 | 0.368 |
| Bayes (bayes.NaiveBayes) | 0.328 | 0.343 | 0.333 |

Table 2. Voting prediction evaluation: Sparsity 30%

| Classifier | Sparsity 30% | | |
|-------------------------------------|--------------|--------|-----------|
| | Precision | Recall | F-measure |
| Party Coding | 0.265 | 0.189 | 0.221 |
| Average Voter | 0.423 | 0.297 | 0.349 |
| Mahalanobis Classifier | 0.364 | 0.302 | 0.330 |
| KNN (lazy.IBk) | 0.168 | 0.135 | 0.145 |
| Neural Network (functions.Logistic) | 0.319 | 0.366 | 0.328 |
| Bayes (bayes.NaiveBayes) | 0.310 | 0.334 | 0.315 |

Table 3. Voting prediction evaluation: Sparsity 50%

| Classifier | Sparsity 50% | | |
|-------------------------------------|--------------|--------|-----------|
| | Precision | Recall | F-measure |
| Party Coding | 0.262 | 0.170 | 0.207 |
| Average Voter | 0.421 | 0.276 | 0.343 |
| Mahalanobis Classifier | 0.321 | 0.229 | 0.267 |
| KNN (lazy.IBk) | 0.154 | 0.126 | 0.137 |
| Neural Network (functions.Logistic) | 0.299 | 0.350 | 0.323 |
| Bayes (bayes.NaiveBayes) | 0.308 | 0.331 | 0.312 |

It can be easily seen in the results that as the number of missing values increases the performance of all classifiers decreases. However, the rate of decrease varies. While the performance of Mahalanobis classifier and decision trees drops rapidly the other classifiers deteriorate gracefully. Overall the Bayesian classifier appears to be the more robust while the Neural Network classifier shows the highest performance in low to medium number of missing values. It is also interesting to note that even with a sparsity level as high as 50% all classifiers operate far above chance level (which in our case is 100/15 because the number of parties T is 15). Furthermore, sparsity levels higher than 10% is quite unusual in properly designed VAAs. Thus, it seems that the impact of missing values on VAA recommendation is somehow overestimated. Nevertheless, proper methods for missing value estimation are always useful and welcome.

5 Conclusion

In this article we investigated the impact of missing values on VAAs recommendation. We compared the traditional party coding recommendation method with several party voter modeling methods as far as the robustness to the number of missing values is concerned. The results show that in all cases the higher the number of missing values the lowest the recommendation accuracy (measured as per class recall in this study). However, the deterioration of performance is not as high as one might assume. Even if 80% of the expected data are missing the recommendation accuracy drops less than 50% of its highest value, showing a remarkable robustness. Among the various compared recommendation methods the traditional party coding and the ones that are based on Neural Networks and Bayesian inferencing are the more robust. The most affected methods by missing values is the one that is based on the covariance matrix of party voters (Mahalanobis Classifier) and the Decision Trees. Both are strongly related on the correlation between the questions of the questionnaire and as the number of missing values increases modeling this correlation becomes more difficult.

An implicit assumption we have made in this study is that missing values are replaced by neutral answers ('neither agree nor disagree'). However, this is not

actually the case in practice because gives an advantage to parties or candidates that avoid to express specific positions in highly controversial issues. As a result in many VAAs the similarity between candidates and voters is measured using variations of the Euclidean distance and excluding the values in the middle scale (see [7]). We are currently investigating the impact of replacing missing values with alternate values outside the scale used in VAAs. That is, if we assume the value ‘-2’ for ‘strongly disagree’ and the value ‘2’ for ‘strongly agree’ then half of the missing values will be given the value ‘-3’ and the other half the value ‘3’. Actually, some machine learning tools such as the Weka (which we have used in this study) accept missing values as inputs. However, it seems that internally they handle those values as neutral ones.

References

1. Baka, A., Figgou, L., Triga, V.: ‘Neither agree, nor disagree’: a critical analysis of the middle answer category in Voting Advice Applications. *Int. J. Electronic Governance* 5(3/4), 244–263 (2012)
2. Cedroni, L., Diego, G. (eds.): *Voting Advice Applications in Europe: The State of the Art*. ScriptaWeb, Napoli (2010)
3. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. *SIGKDD Explor. Newsl.* 11(1), 10–18 (2009)
4. Han, J., Kamber, M., Pei, J.: *Data Mining: Concepts and Techniques*, 3rd edn. Morgan Kaufmann (2011)
5. Katakis, I., Tsapatsoulis, N., Triga, V., Tziouvas, C., Mendez, F.: Clustering Online Poll Data: Towards a Voting Assistance System. In: *Semantic and Social Media Adaptation and Personalization (SMAP 2012)*, pp. 54–59. IEEE Press (2012)
6. Ladner, A., Pianzola, J.: Do Voting Advice Applications Have an Effect on Electoral Participation and Voter Turnout? Evidence from the 2007 Swiss Federal Elections. In: Tambouris, E., Macintosh, A., Glassey, O. (eds.) *ePart 2010*. LNCS, vol. 6229, pp. 211–224. Springer, Heidelberg (2010)
7. Mendez, F.: Matching voters with political parties and candidates: An empirical test of four algorithms. *International Journal of ElectronicGovernance* (2012)
8. Pianzola, J., Trechsel, A.H., Schwerdt, G., Vassil, K., Alvarez, R.M.: The Effect of Voting Advice Applications (VAAs) on Political Preferences. Evidence from a Randomized Field Experiment. In: Annual Meeting of the American Political Science Association (2012), Available at SSRN: <http://ssrn.com/abstract=2108095>
9. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.): *Recommender Systems Handbook*, pp. 39–71. Springer, Heidelberg (2011)
10. Tsapatsoulis, N., Georgiou, O.: Investigating the Scalability of Algorithms, the Role of Similarity Metric and the List of Suggested Items Construction Scheme in Recommender Systems. *International Journal on Artificial Intelligence Tools* 21(4), 12–40 (2012)
11. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn., pp. 3–9. Morgan Kaufmann (2005)

Author Index

- Agathokleous, Marilena I-496
Alexakos, Christos II-70
Alexandrides, Theodore II-174
Alexandiris, Antonios K. I-12
Alonso, Serafín I-370
Amali, Ramin I-203
Anagnostopoulos, Ioannis II-20
Anagnostopoulos, Christos-Nikolaos II-20, II-193, II-203
Anastassopoulos, George I-253, II-203
Antonelli, Fabio II-282
Antonopoulou, Hera II-70
Apolloni, B. II-302
Arbab, Masood Ahmad I-282
Asvestas, Pantelis II-40
Avlonitis, Markos II-60
Bacauskiene, Marija I-396
Bakic, Predrag R. II-146
Ballon, Pieter II-282
Barrientos, F. II-302
Barrientos, Pablo I-370
Barton, Carl II-11
Battipede, Manuela I-313
Beligiannis, Grigorios II-222
Belo, João I-182
Bentivoglio Colturato, Adimara I-406
Bentivoglio Colturato, Danielle I-406
Benyettou, Abdelkader I-273
Benyettou, Mohamed I-424
Boroş, Tiberiu I-42
Burget, Radim I-380
Calvo-Rolle, Jose Luis I-350
Caridakis, George II-257, II-269, II-302
Cassaro, Mario I-313
Castelo Branco, Luiz Henrique I-406
Chatzioannou, Aristotelis A. II-249
Chaudhari, B.N. I-61
Chochliouros, Ioannis P. II-257, II-292, II-312
Cortés Ramírez, Jorge Armando I-263
Crespo-Ramos, Mario J. I-350
Crnojević, Vladimir I-388
Ćulibrk, Dubravko I-388
Daoudi, Rima I-273
Debakla, Mohammed I-424
Delaere, Simon II-282
del Canto, Carlos J. I-370
De Moor, Bart II-222, II-241
Dimitrakopoulos, Christos II-70, II-231
Djemal, Khalifa I-273, I-424
Dolenko, Sergey I-81
Domínguez, Manuel I-370
Dumitrescu, Stefan Daniel I-42
Economou, George-Peter K. II-156
Essig, Kai I-446
Fiannaca, Antonino II-212
Fiasché, M. II-302
Fitkov-Norris, Elena I-213
Folorunso, Sakinat Oluwabukonla I-213
Fountas, Nikos I-144
Frecassetti, Mario Giovanni II-292
Fuentes, Juan J. I-370
Furtado, Edson Luiz I-406
Galliani, G. II-302
Garcia, Ana II-282
Garofalakis, John II-50
Gartsov, Alexander II-185
Gelzinis, Adas I-396
Georgakopoulos, Spiros V. I-292
Georgiadis, Panagiotis II-249
Georgiadou, Evangelia M. II-292, II-312
Giannoukos, Ioannis II-193
Gili, Piero I-313
Giotopoulos, Konstantinos II-70
Gkantouna, Vassiliki II-119
Gomes Benjamin, André I-406
Gopych, Petro I-71
Graña, M. II-302
González Guerra, José Luis I-263
Habermann, Danilo I-112
Hájek, Petr I-302, II-1
Hata, Alberto I-112
Hatzilygeroudis, Ioannis II-30
Hsieh, Bernard I-102

- Hughes, Bradley J. I-203
 Hunt, Doug I-233
- Iakovakis, Vassilis I-144
 Iliadis, Lazaros I-132, I-485
 Iliopoulos, Costas S. II-11
 Iliou, Theodoros II-203
 Ioannou, Zafeiria-Marina II-174
 Iosifidis, Alexandros I-1
 Isaev, Igor I-81
- Jadhav, Sadhana V. I-61
 Jesus, Adelaide P. I-182
- Kalampakas, Antonios I-485
 Kalita, Oksana II-185
 Kampouridis, Michael I-12
 Kanavos, Andreas II-100
 Kapsouras, Ioannis I-172
 Karagiannis, Stefanos I-144
 Karakasidis, Alexandros II-164
 Karanasiou, Irene II-40
 Karanikolos, Stylianos I-172
 Karapilafis, Georgios I-132
 Kardara, Mania II-90
 Karwowski, Jan I-122
 Karydis, Ioannis II-60
 Kasabov, Nikola I-233
 Kasampalis, Dimitrios I-360
 Katakis, Ioannis I-496
 Kateris, Dimitrios I-360
 Katsavounis, S. I-132
 Kechagias, John I-144
 Kłoszewska, Iwona II-193
 Kmet, Tibor I-52
 Kmetova, Maria I-52
 Knoll, Alois I-330
 Kokkinos, Yiannis I-340
 Kollias, Stefanos II-257, II-269, II-302
 Kompatsiaris, Ioannis I-223
 Korvesis, Panagiotis II-138
 Kossida, Sophia I-165
 Kouneli, Marianna II-50
 Krause, André Frank I-446
 Kuzmin, Vadim I-466
 Kyrtopoulos, Soterios II-249
- La Rosa, Massimo II-212
 Legierski, Jarosław I-122
 Leon, Bobrowski I-456
- Lerro, Angelo I-313
 Li, Xirong I-32
 Likothanassis, Spiros II-70, II-110, II-231, II-174
 Lopez, Merce II-282
 López-García, Hilario I-350
 Lovestone, Simon II-193
 Lucas Jaquie Castelo Branco, Kalinka Regina I-406
 Lucena, Rui I-182
- Machón-González, Iván I-350
 Maglogiannis, Ilias I-292
 Mahmud, Sahibzada Ali I-22, I-91, I-282
 Makris, Christos II-100
 Malcangi, Mario I-323
 Maltha, Sven II-282
 Maragoudakis, Manolis I-474
 Margaritis, Konstantinos I-340
 Martins, Leonardo I-182
 Masek, Jan I-380
 Matsopoulos, George K. II-40
 Mavroudi, Seferina II-231
 Mechant, Peter II-282
 Mecocci, Patrizia II-193
 Megaloikonomou, Vasilis II-138, II-146
 Merekoulias, Georgios II-80
 Mitroulias Athanasios II-110
 Mora, Ben I-192
 Morán, Antonio I-370
 Moreau, Yves II-222, II-241
 Moschopoulos, Charalampos II-222, II-241
- Moshou, Dimitrios I-360
 Moumtzidou, Anastasia I-223
 Moutafi, Konstantina II-70
 Mporas, Iosif II-138
 Muhammad Khan, Gul I-22, I-91, I-282
 Mulder, Nicola II-11
 Mylonas, Phivos II-20, II-269
- Nanopoulos, Photis II-185
 Natekin, Alexey I-330
 Nayab, Durre I-91
 Nikolaidis, Nikolaos I-172
 Nikolakopoulos, Athanasios N. II-50
 Nogueira, Pedro A. I-243
 Nuzhnaya, Tatyana II-146
- Obornev, Eugeny I-81
 Oikonomou, Maria I-414

- Okulewicz, Michał I-122
 Olej, Vladimír I-302, II-1
 Oliveira, Eugénio I-243
 Osório, Fernando I-112
 Osstyn, Dirk II-282
- Panić, Marko I-388
 Pantazi, Xanthoula Eirini I-360
 Papadopoulos, Harris I-253
 Papaioannou, Vaios II-156
 Papaoikonomou, Thanos II-90
 Parladori, Giorgio II-282
 Parry, Dave I-233
 Paschali, Kallirroi II-174
 Pavlidis, Georgios II-185
 Pegkas, Andreas II-231
 Perikos, Isidoros II-30
 Persiantsev, Igor I-81
 Piefke, Martina I-446
 Pigatto, Daniel Fernando I-406
 Pimenidis, Elias I-132
 Pitas, Ioannis I-1
 Plagianakos, Vassilis P. I-292
 Plegas, Yannis II-100
 Plerou P., Antonia I-433
 Politopoulou, Vicky I-474
 Popovic, Dusan II-222, II-241
 Prada, Miguel A. I-370
- Quaresma, Cláudia I-182
- Ratcliff, Jay I-102
 Razis, Gerasimos II-20
 Rethimiotaki, Eleni II-312
 Roschildt Pinto, Alex Sandro I-406
 Richardson, Mark I-192
 Rizzo, Riccardo II-212
 Rodrigues, Rui I-243
- San Jose, S. II-302
 Santos, Marcelo I-182
 Salazar Mendiola, Juan Luis I-263
 Schack, Thomas I-446
 Schliebs, Stefan I-233
 Serra, Artur II-282
 Sfakianakis, Evangelos II-292, II-312
 Shimelevich, Mikhail I-81
 Shrestha, Durga Lal I-466
 Sifakis, Emmanouil G. II-249
 Sifrim, Alejandro II-222, II-241
- Simmons, Andrew II-193
 Siolas, Georgios II-269, II-302
 Sioutas, Spyros II-119
 Skoura, Angeliki II-146
 Sladojević, Srdjan I-388
 Soininen, Hikka II-193
 Solomatine, Dimitri I-466
 Sourla, Efrosini II-80
 Sourla, Georgia II-119
 Spartalis, Stefanos I-132, I-485
 Spenger, Christian II-193
 Spiliopoulou, Anastasia S. II-312
 Spiridonidou, Antonia II-60
 Spiros, Likothanassis II-110
 Stafylopatis, Andreas II-269
 Stamatelatos, Makis II-282
 Stamatopoulou, Konstantina-Maria II-80
 Stephanakis, Ioannis M. II-203, II-257, II-292
 Stoykova, Velislava II-129
 Syrimpeis, Vasileios II-80
 Szupiluk, Ryszard I-154
- Tasoulis, Sotiris K. I-292
 Tefas, Anastasios I-1, I-172, I-414
 Theodoridis, Evangelos II-100
 Theofilatos, Konstantinos II-231
 Tsakalidis, Athanasios II-80, II-119, II-174
 Tsakona, Anna II-174
 Tsamandas, Athanasios II-174
 Tsapatsoulis, Nicolas I-496
 Tserpes, Konstantinos II-90
 Tsiliki, Georgia I-165
 Tsitiridis, Aristeidis I-192
 Tsolaki, Magda II-193
 Tsolis, Dimitrios II-174
 Tsouvaltzis, Pavlos I-360
 Tzimas, Giannis II-80, II-119
- Uher, Vaclav I-380
 Ullah, Fahad I-22
 Urso, Alfonso II-212
- Vaiciuškėnas, Evaldas I-396
 Valavanis, Ioannis II-249
 Vargas Luna, José Luis I-263
 Varvarigou, Theodora II-90

- Vaxevanidis, Nikolaos I-144
Vellas, Bruno II-193
Ventouras, Errikos M. II-40
Vergeti, Paraskevi II-70
Verikas, Antanas I-396
Verykios, Vassilios S. II-164
Vieira, Pedro I-182
Vlachakis, Dimitrios I-165
Vlamos, Panayiotis M. I-433
Vrochidis, Stefanos I-223
Watson, Bruce II-11
Wolf, Denis I-112
Wu, Shaohui I-32
Xu, Jieping I-32
Yang, Gang I-32
Ząbkowski, Tomasz I-154
Zacharaki, Evangelia I. II-138
Zizzo, C. II-302