

Hsinchun Chen
Christopher C. Yang (Eds.)

Intelligence and Security Informatics

Techniques and Applications



Springer

Hsinchun Chen and Christopher C. Yang (Eds.)

Intelligence and Security Informatics

Studies in Computational Intelligence, Volume 135

Editor-in-Chief

Prof. Janusz Kacprzyk

Systems Research Institute

Polish Academy of Sciences

ul. Newelska 6

01-447 Warsaw

Poland

E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage:
springer.com

Vol. 116. Ying Liu, Aixin Sun, Han Tong Loh, Wen Feng Lu
and Ee-Peng Lim (Eds.)
Advances of Computational Intelligence in Industrial Systems,
2008
ISBN 978-3-540-78296-4

Vol. 117. Da Ruan, Frank Hardeman
and Klaas van der Meer (Eds.)
Intelligent Decision and Policy Making Support Systems, 2008
ISBN 978-3-540-78306-0

Vol. 118. Tsau Young Lin, Ying Xie, Anita Wasilewska
and Churn-Jung Liau (Eds.)
Data Mining: Foundations and Practice, 2008
ISBN 978-3-540-78487-6

Vol. 119. Sławomir Wiak, Andrzej Krawczyk
and Ivo Dolezel (Eds.)
Intelligent Computer Techniques in Applied Electromagnetics,
2008
ISBN 978-3-540-78489-0

Vol. 120. George A. Tsihrintzis and Lakhmi C. Jain (Eds.)
Multimedia Interactive Services in Intelligent Environments,
2008
ISBN 978-3-540-78491-3

Vol. 121. Nadia Nedjah, Leandro dos Santos Coelho
and Luiza de Macedo Mourelle (Eds.)
Quantum Inspired Intelligent Systems, 2008
ISBN 978-3-540-78531-6

Vol. 122. Tomasz G. Smolinski, Mariofanna G. Milanova
and Aboul-Ella Hassanien (Eds.)
Applications of Computational Intelligence in Biology, 2008
ISBN 978-3-540-78533-0

Vol. 123. Shuichi Iwata, Yukio Ohsawa, Shusaku Tsumoto, Ning
Zhong, Yong Shi and Lorenzo Magnani (Eds.)
Communications and Discoveries from Multidisciplinary Data,
2008
ISBN 978-3-540-78732-7

Vol. 124. Ricardo Zavala Yoe
*Modelling and Control of Dynamical Systems: Numerical
Implementation in a Behavioral Framework*, 2008
ISBN 978-3-540-78734-1

Vol. 125. Larry Bull, Bernadó-Mansilla Ester
and John Holmes (Eds.)
Learning Classifier Systems in Data Mining, 2008
ISBN 978-3-540-78978-9

Vol. 126. Oleg Okun and Giorgio Valentini (Eds.)
*Supervised and Unsupervised Ensemble Methods
and their Applications*, 2008
ISBN 978-3-540-78980-2

Vol. 127. Régis Gras, Einoshin Suzuki, Fabrice Guillet
and Filippo Spagnolo (Eds.)
Statistical Implicative Analysis, 2008
ISBN 978-3-540-78982-6

Vol. 128. Fatou Xhafa and Ajith Abraham (Eds.)
*Metaheuristics for Scheduling in Industrial and Manufacturing
Applications*, 2008
ISBN 978-3-540-78984-0

Vol. 129. Natalio Krasnogor, Giuseppe Nicosia, Mario Pavone
and David Pelta (Eds.)
*Nature Inspired Cooperative Strategies for Optimization
(NICSO 2007)*, 2008
ISBN 978-3-540-78986-4

Vol. 130. Richi Nayak, Nikhil Ichalkaranje
and Lakhmi C. Jain (Eds.)
Evolution of the Web in Artificial Intelligence Environments,
2008
ISBN 978-3-540-79139-3

Vol. 131. Roger Lee and Haeng-Kon Kim (Eds.)
Computer and Information Science, 2008
ISBN 978-3-540-79186-7

Vol. 132. Danil Prokhorov (Ed.)
Computational Intelligence in Automotive Applications,
2008
ISBN 978-3-540-79256-7

Vol. 133. Manuel Graña and Richard J. Duro (Eds.)
Computational Intelligence for Remote Sensing, 2008
ISBN 978-3-540-79352-6

Vol. 134. Ngoc Thanh Nguyen and Radoslaw Katarzyniak (Eds.)
New Challenges in Applied Intelligence Technologies, 2008
ISBN 978-3-540-79350-0

Vol. 135. Hsinchun Chen and Christopher C. Yang (Eds.)
Intelligence and Security Informatics, 2008
ISBN 978-3-540-69207-2

Hsinchun Chen
Christopher C. Yang
(Eds.)

Intelligence and Security Informatics

Techniques and Applications



Springer

Hsinchun Chen
University of Arizona
Department of Management Information Systems
The University of Arizona
1130 E. Helen St.
Tucson AZ 85721
430Z McClelland Hall
USA
Email: hchen@eller.arizona.edu

Christopher C. Yang
Drexel University
College of Information Science and Technology
3141 Chestnut Street
Philadelphia, PA 19104
USA
Email: chris.yang@ischool.drexel.edu

ISBN 978-3-540-69207-2

e-ISBN 978-3-540-69209-6

DOI 10.1007/978-3-540-69209-6

Studies in Computational Intelligence

ISSN 1860949X

Library of Congress Control Number: 2008925308

© 2008 Springer-Verlag Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, 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.

Type set & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed in acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

The IEEE International Conference on Intelligence and Security Informatics (ISI) and Pacific Asia Workshop on Intelligence and Security Informatics (PAISI) conference series (<http://www.isiconference.org>) have drawn significant attention in the recent years. Intelligence and Security Informatics is concerned with the study of the development and use of advanced information technologies and systems for national, international, and societal security-related applications. The ISI conference series have brought together academic researchers, law enforcement and intelligence experts, information technology consultant and practitioners to discuss their research and practice related to various ISI topics including ISI data management, data and text mining for ISI applications, terrorism informatics, deception and intent detection, terrorist and criminal social network analysis, public health and bio-security, crime analysis, cyber-infrastructure protection, transportation infrastructure security, policy studies and evaluation, information assurance, among others. In this book, we collect the work of the most active researchers in the area. Topics include data and text mining in terrorism, information sharing, social network analysis, Web-based intelligence monitoring and analysis, crime data analysis, infrastructure protection, deception and intent detection and more.

Scope and Organization

The book is organized in four major areas. The first unit focuses on the terrorism informatics and data mining. The second unit discusses the intelligence and crime analysis. The third unit covers access control, infrastructure protection, and privacy. The forth unit presents surveillance and emergency response.

There are twenty-two chapters contributed by authors coming from nine different countries, including Belgium, Canada, Israel, Italy, Northern Cyprus, Singapore, Taiwan, United Kingdom and United States. The titles of the twenty-two chapters are listed below:

- | | |
|------------|---|
| Chapter 1: | Assured Information Sharing: Technologies, Challenges and Directions |
| Chapter 2: | Automatic Event Extraction for the Security Domain |
| Chapter 3: | Knowledge Discovery and Information Visualization for Terrorist Social Networks |
| Chapter 4: | Understanding the Nexus of Terrorist Web Sites |

Chapter 5:	Multi-lingual Detection of Web Terrorist Content
Chapter 6:	Modeling Anticipatory Event Transitions
Chapter 7:	Exploring Gray Web Forum: Analysis and Investigation of Forum-Based Communities in Taiwan
Chapter 8:	Identifying Interesting Networks of Criminal Activity
Chapter 9:	Name Matching in Law Enforcement Database
Chapter 10:	Discovering Investigation Clues through Mining Criminal Databases
Chapter 11:	Automated Filtering on Data Streaming for Intelligence Analysis
Chapter 12:	Personal Information Management for Intelligence Tasks
Chapter 13:	A Data Miner's Approach to Country Corruption Analysis
Chapter 14:	Protecting Private Information in Online Social Networks
Chapter 15:	Protection of Database Security via Collaborative Inference Detection
Chapter 16:	Suspect Vehicle Identification for Border Safety
Chapter 17:	Optimization Problems for Port-of-Entry Detection Systems
Chapter 18:	Modeling and Validation of Aviation Security
Chapter 19:	Anomaly Detection in Moving Object
Chapter 20:	Intelligent Face Recognition
Chapter 21:	Video Analysis of Vehicles and Persons for Surveillance
Chapter 22:	Video-based Deception Detection

Audience

1. Public and private sector practitioners in the national/international and homeland security area.
2. Consultants and contractors engaged in on-going relationships with federal, state, local, and international agencies on projects related to national security.
3. Graduate level students in Information Sciences, Public Policy, Computer Science, Information Assurance, and Terrorism.
4. Researchers engaged in security informatics, homeland security, information policy, knowledge management, public administration, and counterterrorism.

We hope that the readers will find the book valuable and useful in their study or work. We also hope that the book will be a contribution to the ISI community. The researchers and practitioners in this community will continue to grow and share our research findings to contribute to the national safety around the world.

USA
USA

Hsinchun Chen
Christopher C. Yang

Author Biographies

Hsinchun Chen, Ph.D.



Dr. Hsinchun Chen is McClelland Professor of Management Information Systems at the University of Arizona and Andersen Consulting Professor of the Year (1999). He received the B.S. degree from the National Chiao-Tung University in Taiwan, the MBA degree from SUNY Buffalo, and the Ph.D. degree in Information Systems from the New York University. Dr. Chen is a Fellow of IEEE and AAAS. He received the IEEE Computer Society 2006 Technical Achievement Award. He is author/editor of 18 books, 17 book chapters, 150 SCI journal articles, and 110 refereed conference articles covering digital library, intelligence analysis, biomedical informatics, data/text/web mining, knowledge management, and Web computing. His recent books include: *Medical Informatics: Knowledge Management and Data Mining in Biomedicine* and *Intelligence and Security Informatics for International Security: Information Sharing and Data Mining*, both published by Springer. Dr. Chen was ranked #8 in publication productivity in Information Systems (CAIS 2005) and #1 in Digital Library research (IP&M 2005) in two recent bibliometric studies. He serves on ten editorial boards including: *ACM Transactions on Information Systems*, *ACM Journal on Educational Resources in Computing*, *IEEE Transactions on Intelligent Transportation Systems*, *IEEE Transactions on Systems, Man, and Cybernetics*, *Journal of the American Society for Information Science and Technology*, *Decision Support Systems*, and *International Journal on Digital Library*.

Dr. Chen has served as a Scientific Counselor/Advisor of the National Library of Medicine (USA), Academia Sinica (Taiwan), and National Library of China (China). He has been an advisor for major NSF, DOJ, NLM, DOD, DHS, and other international research programs in digital library, digital government, medical informatics, and national security research. Dr. Chen is founding director of Artificial Intelligence Lab and Hoffman E-Commerce Lab. The UA Artificial Intelligence Lab, which houses 40+ researchers, has received more than \$25M in research funding from NSF, NIH, NLM, DOD, DOJ, CIA, DHS, and other agencies. The Hoffman E-Commerce Lab, which has been funded mostly by major IT industry partners, features one of the

VIII Author Biographies

most advanced e-commerce hardware and software environments in the College of Management. Dr. Chen is conference co-chair of ACM/IEEE Joint Conference on Digital Libraries (JCDL) 2004 and has served as the conference/program co-chair for the past eight International Conferences of Asian Digital Libraries (ICADL), the premiere digital library meeting in Asia that he helped develop. Dr. Chen is also (founding) conference co-chair of the IEEE International Conferences on Intelligence and Security Informatics (ISI) 2003-2007. The ISI conference, which has been sponsored by NSF, CIA, DHS, and NIJ, has become the premiere meeting for international and homeland security IT research. Dr. Chen's COPLINK system, which has been quoted as a national model for public safety information sharing and analysis, has been adopted in more than 200 law enforcement and intelligence agencies in 20 states. The COPLINK research had been featured in the New York Times, Newsweek, Los Angeles Times, Washington Post, Boston Globe, and ABC News, among others. The COPLINK project was selected as a finalist by the prestigious International Association of Chiefs of Police (IACP)/Motorola 2003 Weaver Seavey Award for Quality in Law Enforcement in 2003. COPLINK research has recently been expanded to border protection (BorderSafe), disease and bioagent surveillance (BioPortal), and terrorism informatics research (Dark Web), funded by NSF, CIA, and DHS. In collaboration with Customs and Border Protection (CBP), the BorderSafe project develops criminal network analysis and vehicle association mining research for border-crosser risk assessment. The BioPortal system supports interactive geospatial analysis and visualization, chief complaint classification, and phylogenetic analysis for public health and biodefense. In collaboration with selected international terrorism research centers and intelligence agencies, the Dark Web project has generated one of the largest databases in the world about extremist/terrorist-generated Internet contents (web sites, forums, and multimedia documents). Dark Web research supports link analysis, content analysis, web metrics analysis, multimedia analysis, sentiment analysis, and authorship analysis of international terrorism contents. The project was featured in the Discover magazine, Arizona Republic, and Toronto Star, among others. Dr. Chen is the founder of the Knowledge Computing Corporation, a university spin-off company and a market leader in law enforcement and intelligence information sharing and data mining. Dr. Chen has also received numerous awards in information technology and knowledge management education and research including: AT&T Foundation Award, SAP Award, the Andersen Consulting Professor of the Year Award, the University of Arizona Technology Innovation Award, and the National Chaio-Tung University Distinguished Alumnus Award.

Christopher C. Yang, Ph.D.

Dr. Christopher C. Yang is an associate professor in the College of Information Science and Technology at Drexel University. He received his B.S., M.S., and Ph.D. in Electrical and Computer Engineering from the University of Arizona. He has been an associate professor in the Department of Systems Engineering and Engineering



Management and the director of the Digital Library Laboratory at the Chinese University of Hong Kong and an assistant professor in the Department of Computer Science and Information Systems at the University of Hong Kong. He has also been a research scientist in the Department of Management Information Systems at the University of Arizona. His recent research interests include security informatics, information visualization, social network analysis, cross-lingual information retrieval and knowledge management, Web search and mining, text summarization, multimedia retrieval, digital library, and electronic commerce. He has published over 150 referred journal and conference papers in *Journal of the American Society for Information Science and Technology (JASIST)*, *Decision Support Systems (DSS)*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Robotics and Automation*, *IEEE Computer*, *Information Processing and Management*, *Journal of Information Science*, *Graphical Models and Image Processing*, *Optical Engineering*, *Pattern Recognition*, *International Journal of Electronic Commerce*, *Applied Artificial Intelligence*, *IWWWC*, *SIGIR*, *ICIS*, *CIKM*, and more. He has edited several special issues on multilingual information systems, knowledge management, and Web mining in *JASIST* and *DSS*. He chaired and served in many international conferences and workshops, including the IEEE International Conference on Intelligence and Security Informatics and Pacific Asia Workshop on Intelligence and Security Informatics. He has also frequently served as an invited panelist in the NSF Review Panels in US. He was the chairman of the Association for Computing Machinery Hong Kong Chapter.

Contents

Part I: Terrorism Informatics and Data Mining

1 Assured Information Sharing: Technologies, Challenges and Directions <i>Bhavani Thuraisingham</i>	1
2 Automating Event Extraction for the Security Domain <i>Clive Best, Jakub Piskorski, Bruno Pouliquen, Ralf Steinberger, Hristo Tanev</i>	17
3 Knowledge Discovery and Information Visualization for Terrorist Social Networks <i>Christopher C. Yang</i>	45
4 Understanding the Nexus of Terrorist Web Sites <i>Jennifer Xu, Hsinchun Chen</i>	65
5 Multi-lingual Detection of Web Terrorist Content <i>Mark Last, Alex Markov, Abraham Kandel</i>	79
6 Modeling Anticipatory Event Transitions <i>Qi He, Kuiyu Chang, Ee-Peng Lim</i>	97
7 Exploring Gray Web Forums: Analysis and Investigation of Forum-Based Communities in Taiwan <i>Jau-Hwang Wang, Tianjun Fu, Hong-Ming Lin, Hsinchun Chen</i>	121

Part II: Intelligence and Crime Analysis

8 Identifying Interesting Networks of Criminal Activity <i>Byron Marshall</i>	135
---	-----

9 Name Matching in Law Enforcement Database <i>Olcay Kursun, Michael Georgopoulos, Kenneth Reynolds</i>	151
10 Discovering Investigation Clues through Mining Criminal Databases <i>Patrick S. Chen</i>	173
11 Automated Filtering on Data Streaming for Intelligence Analysis <i>Yiming Ma, Dawit Yimam Seid, Sharad Mehrotra</i>	199
12 Personal Information Management for Intelligence Tasks <i>Antonio Badia</i>	215
13 A Data Miner's Approach to Country Corruption Analysis <i>Johan Huysmans, Bart Baesens, Jan Vanthienen</i>	227

Part III: Access Control, Infrastructure Protection and Privacy

14 Protecting Private Information in Online Social Networks <i>Jianming He, Wesley W. Chu</i>	249
15 Protection of Database Security Via Collaborative Inference Detection <i>Yu Chen, Wesley W. Chu</i>	275
16 Suspect Vehicle Identification for Border Safety <i>Siddharth Kaza, Hsinchun Chen</i>	305
17 Optimization Problems for Port-of-Entry Detection Systems <i>Endre Boros, Elsayed Elsayed, Paul Kantor, Fred Roberts, Minge Xie</i>	319
18 Modeling and Validation of Aviation Security <i>Uwe Glässer, Sarah Rastkar, Mona Vajihollahi</i>	337

Part IV: Surveillance and Emergency Response

19 Anomaly Detection in Moving Object <i>Xiaolei Li, Jiawei Han, Sangkyum Kim, Hector Gonzalez</i>	357
20 Intelligent Face Recognition <i>Adnan Khashman</i>	383
21 Video Analysis of Vehicles and Persons for Surveillance <i>Sangho Park, Mohan M. Trivedi</i>	407

22 Video-Based Deception Detection	
<i>Matthew L. Jensen, Thomas O. Meservy, Judee K. Burgoon, Jay F. Nunamaker Jr.</i>	425
Subject Index	443
Author Index	449

Assured Information Sharing: Technologies, Challenges and Directions

Bhavani Thuraisingham

Erik Jonsson School of Engineering and Computer Science,
The University of Texas at Dallas, USA
bhavani.thuraisingham@utdallas.edu

Abstract. This paper describes issues, technologies, challenges, and directions for Assured Information Sharing (AIS). AIS is about organizations sharing information but at the same time enforcing policies and procedures so that the data is integrated and mined to extract nuggets. This is the first in a series of papers we are writing on AIS. It provides an overview including architectures, functions and policies for AIS. We assume that the partners of a coalition may be trustworthy, semi-trustworthy or untrustworthy and investigate solutions for AIS to handle the different scenarios.

1.1 Introduction

Data from the various data sources at multiple security levels as well as from different services and agencies including the Air Force, Navy, Army, Local, State and Federal agencies have to be integrated so that the data can be mined, patterns and information extracted, relationships identified, and decisions made. The databases would include for example, military databases that contain information about military strategies, intelligence databases that contain information about potential terrorists and their patterns of attack, and medical databases that contain information about infectious diseases and stock piles. Data could be structured or unstructured including geospatial/ multimedia data. Data also needs to be shared between healthcare organizations such as doctors' offices, hospitals and pharmacies. Unless the data is integrated and the big picture is formed, it will be difficult to inform all the parties concerned about the incidences that have occurred. While the different agencies have to share data and information, they also need to enforce appropriate security and integrity policies so that the data does not get into the hands of unauthorized individuals. Essentially the agencies have to share information but at the same time maintain the security and integrity requirements.

This is the first in a series of papers we are writing on Assured Information Sharing describing our research. The papers that follow will include applying game theoretical techniques for AIS among semi-trustworthy partners, defending against malicious attacks while data sharing, applying RBAC (role-based access control) with UCON (Usage Control) extensions for AIS and carrying out offensive operations against untrustworthy partners. We are also investigating risk-based access control, data origin and provenance issues as well as geospatial data management for AIS.

In this paper we describe Assured Information Sharing that will ensure that the appropriate policies for confidentiality, privacy, trust, release, dissemination, data quality and provenance are enforced. We discuss technologies for AIS as well as novel approaches based on game theoretical concepts. In Sect. 1.2 we will provide an overview of an AIS architecture. The policies that are of interest to a wide variety of domain applications including defense, intelligence, medical and financial are discussed in Sect. 1.3. Data integration and analysis technologies for AIS will be discussed in Sect. 1.4. Security policy aspects including enforcement and integration will be discussed in Sect. 1.5. Integrity and dependability issues such as data provenance and quality and real-time processing will be discussed in Sect. 1.6. Balancing conflicting requirements including security vs. real-time processing will be discussed in Sect. 1.7. Some novel approaches will be discussed in Sect. 1.8. In particular applications of game theoretical techniques for handling semi-trustworthy partners will be discussed. Approaches for handling untrustworthy partners will be discussed in Sect. 1.9. The paper is concluded in Sect. 1.10.

1.2 Coalition Data Sharing

A coalition consists of a set of organizations, which may be agencies, universities and corporations that work together in a peer-to-peer environment to solve problems such

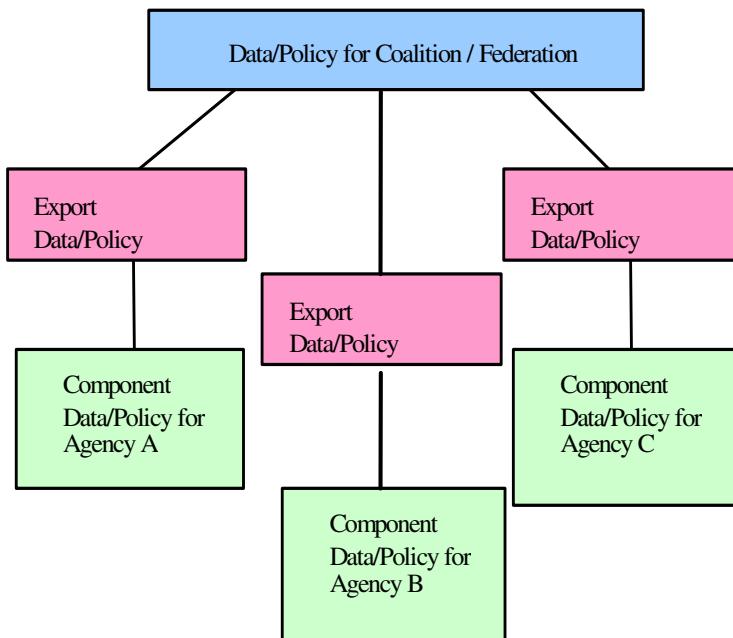


Fig. 1.1. Architecture for Organizational Data Sharing

as intelligence and military operations as well as healthcare operations. Fig. 1.1 illustrates an architecture for a coalition where three agencies have to share data and information. Coalitions are usually dynamic in nature. That is, members may join and leave the coalitions in accordance with the policies and procedures. A challenge is to ensure the secure operation of a coalition. We assume that the members of a coalition, which are also called its partners, may be trustworthy, untrustworthy or partially (semi) trustworthy.

Various aspects of coalition data sharing are discussed in the Markle report [12]. However, security including confidentiality, privacy, trust, integrity, release and dissemination has been given little consideration. Much of the prior work on security in a coalition environment has focused on secure federated data sharing. Thuraisingham was one of the first to propose multilevel security for federated database systems [22]. Discretionary security was proposed in [15]. None of the previous work has focused on determining the amount of information that is lost for conducting military operations by enforcing security. Furthermore, developing flexible policies in a coalition environment are yet to be examined. Enforcing security while meeting timing constraints remains a largely unexplored topic. A discussion of information survivability issues and the need for flexible policies for enforcing security and meeting timing constraints are given in [24] and [20]. However, to our knowledge, no research has been reported on secure (including confidentiality, privacy, trust and integrity) and timely data sharing for a coalition environment. Some of the challenges include the following:

Data Sharing: One of the main goals of coalition data sharing is for organizations to share the data but at the same time maintain autonomy. For example, one database could be used for travel data while another database could be used to manage data pertaining to airplanes. For counter-terrorism applications and military operations, the key is to make links and associations as rapidly as possible. We need policies and procedures to determine what data to share under what conditions.

Data Mining: Data mining techniques extract patterns and trends often previously unknown from large quantities of data [23]. However data mining tools could give out false positives and false negatives. This is especially critical for applications such as counter-terrorism and military operations as it could result in catastrophic consequences [25]. Therefore, we need human analysts to examine the patterns and determine which ones are useful and which ones are spurious. The challenge is to develop automated tools to sift through the data and produce only the useful links and associations.

Security: Confidentiality, privacy, integrity, trust, real-time processing, fault tolerance, authorization and administration policies enforced by the component organizations via the local agencies have to be integrated at the coalition level. As illustrated in Fig. 1.1, each organization may export security policies and data to the coalition. The component systems may have more stringent access control requirements for foreign organizations. The challenge is to ensure that there is no security violation at the coalition level.

In Sects. 1.3 through 1.7 we discuss various aspects on AIS assuming that the partners are trustworthy. Semi-trustworthy partners will be discussed in Sect. 1.8. Untrustworthy partners will be discussed in Sect. 1.9.

1.3 Policy Issues

Before we discuss technologies and ways to enforce security policies we need to determine what types of policies are going to be considered by the organizations. In this section we will discuss some policies. While we will collectively call them security policies, they will include confidentiality, privacy, and trust policies among others.

Confidentiality: These policies will determine the types of access that the users will have on the data objects. The types of policies include access control policies as well as the more sophisticated role-based access control and usage control policies [16], [PARK04].

Need to know: Even if the user has access to the data, does he really have a need to know for the data? These policies will determine who has the need to know for the data.

Need to share: While need to know policies were developed during the cold war era, since 9/11, there is emphasis on need to share. Essentially the focus is on organizational data sharing. Therefore, even if a user does not have access to the data, an organization may need to share the data under different situations.

Privacy: Note that different organizations view privacy differently. In the healthcare domain privacy mainly deals with the information a user decides to release about himself. In the case of intelligence applications, an agency determines what information it can release about a user to another agency.

Trust: These policies will determine the level of “trust” that an organization or user has about another organization or user. If John has 100% trust in Jane, then he may release all of the information to Jane. If on the other hand he has only 50% trust in Jane, he may not want to release the sensitive information to Jane.

Integrity: There are various types of integrity policies. One set of policies will determine who can modify the data. Another set of policies may determine the accuracy of the data. A third set of policies may determine whether the data is complete. Essentially we have included data quality related policies as part of the integrity policies.

Data Provenance: Data provenance is about maintaining the history of the data from the time it was created. Data provenance policies will determine what rules are enforced on maintaining the history of the data.

Release: These policies will determine how and to whom data is released.

Dissemination: These policies will determine how data is disseminated after it is released. That is, once data has been released to John, can John disseminate the data to Jane?

Sanitization: These policies will determine how data is sanitized before releasing to the user.

Downgrading: These policies will determine how highly classified data may be downgraded to a lowly cleared user.

Service policies: Many organizations are using services to carry out their operations including contracting and outsourcing. These policies will determine how services are carried out.

Other Policies: Organizations also enforce policies for governance, auditing, accountability, authentication, and data distribution among others.

1.4 Data Integration and Analysis Technologies

Data Integration: As illustrated in Fig. 1.2, data from the various data sources at multiple levels such as local, state and federal levels have to be integrated so that the data can be mined, patterns extracted and decisions made. Data integration has been attempted for about 20 years. Until recently brute force integration techniques consisting of translators and gateways were used between the multiple data management systems. Standards such as RDA (Remote Database Access) were developed initially for client-server interoperability. Later object-base wrappers were used to encapsulate the multiple systems including the legacy systems. For example, distributed object management standards were used to encapsulate systems and applications into objects. However, common representation of the data remained a challenge. It is only recently that we have a good handle on syntactic integration through standards such as XML (eXtensible Markup Language). The idea is as follows: each data system publishes its schema (also called metadata) in XML. Since all the systems now represent their schema in XML, the systems can talk to each other in a seamless fashion.

A major challenge for data integration is semantic heterogeneity. While much progress has been made on syntactic integration, not much work has been reported on semantic integration. For example, multiple systems may use different terms for the same data; the procedure EKG (Electro Cardiogram) is called ECG in the United Kingdom. Even within the same state, different hospitals may use different terms to mean the same entity. For example, one hospital may use the term influenza while another hospital may use the term flu. In some cases, the same term may be used to represent different entities. While repositories and dictionaries have been built, a satisfactory solution for semantic heterogeneity is still not available. The development of semantic web technologies including the Resource Description Framework (RDF) language standard shows promise to handle semantic heterogeneity.

Multimedia and Geospatial Data: Data will include structured data as well as unstructured data such as text, voice, video and audio. Data emanating from multiple data sources including sensor and surveillance data have to be integrated and shared. Managing, integrating and mining multimedia data remains a challenge. We need efficient indexing techniques as well as XML and RDF based representation schemes. Furthermore, the data has to be mined so that patterns and trends are extracted. Video data could be data emanating from surveillance cameras or news feeds such as CNN (Cable News Network) video data. Emergency response systems have to integrate geospatial data such as maps together with structured data, make sense out of the data and rapidly produce summaries so that the emergency response teams can read and understand the data [2].

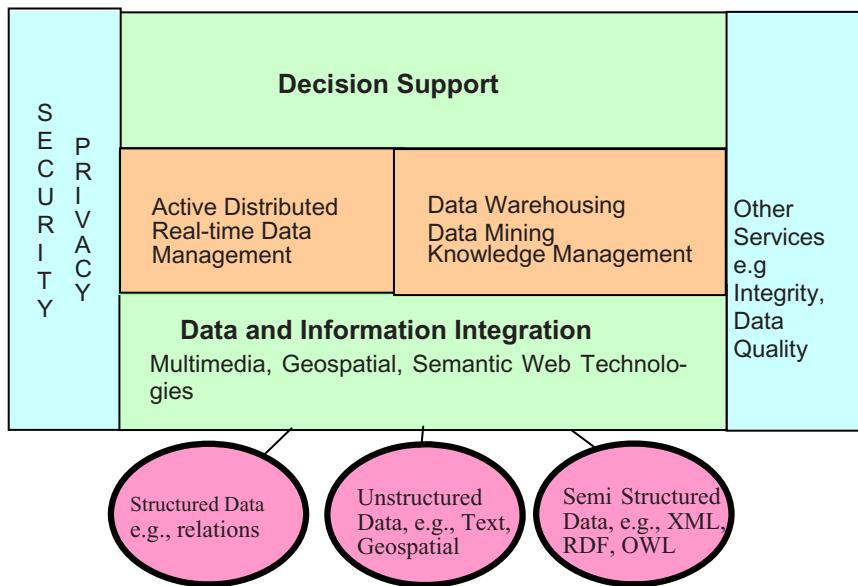


Fig. 1.2. Data Integration and Analysis

Data Mining: Integrated data may be mined to extract patterns for suspicious and unusual behavior. Much of the work in data mining has focused on mining relational and structured databases. While some work has been reported on text, image, audio and video data mining, much remains to be done. For example, how can one mine integrated geospatial and multimedia data? How can false positives and false negatives be eliminated or at least reduced? What are the training models used for multimedia data? What are the appropriate outcomes for multimedia data mining? Does it make sense to extract metadata and then mine the metadata? Much remains to be done before operational tools for multimedia and geospatial data mining are developed.

Web Services: The Department of Defense (DoD) as well as other agencies are migrating toward service oriented architectures (SOA). For example, the Network Centric Operations Architecture is based on SOA and the services are called Network Centric Enterprise Services (NCES). Furthermore, The Global Information Grid (GIG) is based on SOA. In a coalition environment, the agencies will publish their policies and schema as illustrated in Fig. 1.1, and communicate with each other using web services technology.

Semantic Web: Semantic web is the vision of Tim Berners Lee and is utilized by many applications including e-business [10]. Due to the extensive investments by the DoD (Department of Defense) and other agencies, many semantic web technologies such as XML, RDF and Ontologies have been developed for applications such as interoperability. Furthermore, semantic web technologies are being developed for different communities. These technologies are critical for AIS. For example, we need ontologies specified in languages such as OWL (web ontology language) to represent objects so that multiple systems can work with the ontologies to handle semantic

heterogeneity. A member organization of a coalition can publish its schema in languages such as XML or RDF to facilitate interoperability and information extraction.

While semantic webs are being developed for different communities, there is little work on enforcing security, privacy and trust for these semantic webs. XML, RDF and Ontologies have to be secure. Furthermore, there is a need to incorporate trust negotiation for the semantic web. We are developing secure semantic web technologies for AIS [3, 26, 29].

1.5 Security Policy Enforcement

Security policies include policies for confidentiality, privacy, trust, release, dissemination and integrity. A broader term is dependable systems or trustworthy systems that also include real-time processing and fault tolerance. We will discuss dependability in the next section. By confidentiality we mean that data is only released to individuals who are authorized to get the data. Privacy in general deals with the situation where an individual determines what information should be released about him/her. (Note that different definitions of privacy have been proposed.) Trust policies may add further restriction to privacy and confidentiality policies. For example, a user may be authorized to get the data according to the confidentiality policies, but the system may not trust the individual in which case the data is not released. Similarly a person may give permission to release certain private information about him or her but that person may not trust a particular web site in which case the private information is not released to the web site. Alternatively one could argue that one needs to establish trust first before establishing the confidentiality and privacy policies. For example, a user's (or web site's) trust is established before determining that the user (or web site) can receive confidential (or private) information. Release policies specify rules for releasing data while dissemination policies specify rules for disseminating the data. Integrity within the context of security ensures that only authorized individuals can modify the data so that the data is not maliciously corrupted [30]. We are conducting extensive investigation on privacy preserving data mining [11].

Security for relational databases has been studied extensively and standards such as secure SQL (Structured Query Language) have been developed [27]. In addition several secure data management system products have been developed. There has been research on incorporating security into next generation data management systems. There is also work on data quality as well as trust management. Security has also been investigated for secure object request brokers as well as for secure e-commerce systems. Finally W3C (World Wide Web Consortium) is specifying standards for privacy such as the P3P (Platform for Privacy Preferences). While there is research on incorporating security for semantic webs and heterogeneous data systems, this research is in the early stages. There is an urgent need to develop operational systems that enforce security. Furthermore, security has conflicting requirements with real-time processing. We need to enforce flexible policies and subsequently standards for specifying these policies. Security is critical for many of the information technologies we have discussed here. For a discussion of secure data sharing and related standards we refer to [5].

Security Policy Integration: There is a critical need for organizations to share data as well process the data in a timely manner, but at the same time enforce various security policies. Fig. 1.3 illustrates security policy integration in a coalition environment. In this example, A and B form a coalition while B and C form a second coalition. A could be California, B could be Texas and C could be Oklahoma. California and Texas could form a coalition as part of the border states in the US and Texas and Oklahoma could form a coalition as part of the neighboring states in the South of US for emergency management. California and Texas could share immigration related data while Oklahoma and Texas could share healthcare data. For both types of data confidentiality and privacy are important considerations. Furthermore, states may prefer to use standards-based data sharing. The standards efforts in this area include Role-based access control (RBAC) [16] as well as P3P (Platform for Privacy Preferences).

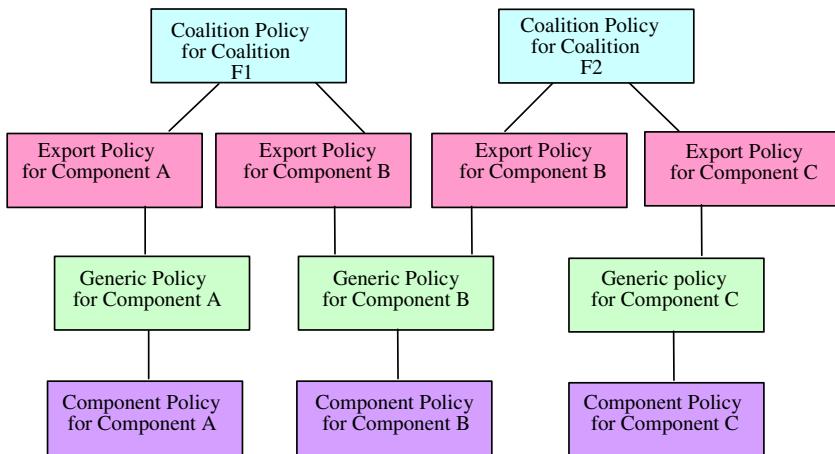


Fig. 1.3. Security Policy Integration and Transformation for Coalitions

1.6 Dependability Aspects

By dependable systems we mean systems that are fault tolerant and meet timing constraints. The time-critical, information-sensitive goals of managing a crisis include actions such as the early confirmation of cases and correct identification of exposed populations over a relevant time period. Early confirmation means that triggers have to be activated when certain situations (such as anomalies) occur. Suppose a hospital is flooded with 30 patients within 15 minutes who are all reporting a temperature of 105 degrees. There has to be a rule such as “If more than 30 patients register at a hospital within 20 minutes with temperature greater than 102 degrees then alert the emergency response system”. To effectively process a large number of rules, we need active data management. Furthermore, the various parties involved such as federal, state and local governments have to be informed within a certain time. That is, if the authorities are notified after say 2 hours then it will be difficult to contain the spread

of the disease. This means we need real-time data management capabilities. Some initial research on dependable and secure systems is discussed in [7].

While there are techniques for active real-time data management, the challenge is to develop an integrated system for end-to-end data management. For example, the data manager will ensure that the data is current and the transactions meet the timing constraints. However in an emergency situation there are numerous dependencies between different data sources. For example when rule A gets triggered, that would result in rules C, D, and E getting triggered in multiple data management systems. Such chain rule processing remains a challenge. We also need end-to-end real-time processing. That is, in addition to the data manager, the infrastructure, the network and the operating system have to meet timing constraints. This remains a challenge. Incorporating security into real-time processing techniques remains largely unexplored. For example, in an emergency situation, real-time processing and activating triggers may be more critical than enforcing access control techniques. Furthermore, the system must ensure that the deadlines are not missed due to malicious code and attacks (e.g., denial of service).

While integrity within the context of security implies that the data is not maliciously corrupted, integrity also includes policies for data quality and data provenance management. Data quality determines the accuracy of the data. This would depend on who updated the data, who owns the data and what is the accuracy of the source of the data. That is, as data moves from organization to organization, its quality may vary. Some measure to compute the quality of the data is needed. Data provenance is about maintaining the history of the data. That is, information as to who accessed the data from start to finish is needed to determine whether data is misused.

1.7 Balancing Conflicting Requirements

There are two types of conflicting requirements: one is security vs. data sharing. The goal of data sharing is for organizations to share as much data as possible so that the data is mined and nuggets obtained. However when security policies are enforced then not all of the data is shared. The other type of conflict is between real-time processing and security. The war fighter will need information at the right time. If it is even say 5 minutes late the information may not be useful. This means that if various security checks are to be performed then the information may not get to the war fighter on time.

We are conducting research in both areas. For example, we are integrating the data in the coalition databases without any access control restrictions and apply the data mining tools to obtain interesting patterns and trends. In particular, we are developing associations between different data entities such as “A and B are likely to be in a location 50 miles from Baghdad”. Next we are using the same tool on the integrated data after enforcing the policies. We can then determine the patterns that might be lost due to enforcing the policies (note that there is some relationship between this work and the research on privacy preserving data mining). Our research is described in [1].

In addition, we are conducting research on examining the extent to which security affects timing constraints. For example, we enforce timing constraints on the query

algorithms. That is, we first process the query using the enforcement algorithms without enforcing any of the policies. Then we enforce the security policies and determine whether the timing constraints can be met. This will determine the extent to which security impacts timely information processing.

Our goal is to develop flexible approaches and balance conflicting requirements. That is, if timely processing of data is critical then security has to be relaxed. Similarly say during non combat operations, security will have to be given full consideration. The same applies for data sharing vs. security. If during an emergency operation such as say the operation just before, during or soon after Hurricane Katrina, then several agencies will need the data without any restrictions. However during non emergency operations, security policies need to be enforced.

Another aspect of our research on AIS is risk analysis. For example, if the security risks are high and the cost to implement security features are low, then security should be given high consideration. If the risks are low and the cost is high, one needs to evaluate whether it is worth the effort and cost to incorporate security. Our research on risk based access control for AIS is reported in [4].

1.8 Game Theory Applications and Semi-trustworthy-partners

In the previous sections we assumed that the organizations were trustworthy and would enforce the policies while data sharing. However in many cases the organization may be semi-honest or completely dishonest. In the case of semi-trustworthy (also called semi-honest) partners, organizations may have to play games to extract data. In the case of dishonest and untrustworthy partners, one may not only have to defend against malicious code, but also have to figure out what the partner is up to by monitoring his machine. In this section we will address semi-trustworthy partners and in the next we will discuss untrustworthy partners.

Semi-Honest Partners and Game Playing

To handle secure data sharing especially with semi-trustworthy partners, modeling the query processing scenario as a non cooperative game may be more appropriate especially between two partners. The players are the partners, which could be agencies or countries of a coalition. Lets assume we have Agency A and B as two partners. The objective of agency A is to extract as much information as possible from agency B. Essentially agency A wants to compromise information managed by Agency B. B's goal is to prevent this from occurring. Cooperative games on the other hand may have applications among friendly partners of a coalition. A mixture of cooperative and non-cooperative strategies may be applied for multi-party coalition.

Two-party information sharing: Information sharing between two agencies A and B may be modeled as a non-cooperative game. A has a specific objective; for example, it may know that B has some sensitive data and it wants to extract the value of that data from B. B knows A's objective. A move made by A is a query. A move made by B is the response. The game continues until A achieves its objectives or gets tired of playing the game. As stated in [6], the game can be represented as a graph theoretic tree of vertices and edges. The tree has a distinguished vertex, which is the initial state. There is a payoff function, which assigns a pair of values say (X,Y) where X is

the payoff for A and Y is the payoff for B for each move. The payoff for A is high if it is close to obtaining the sensitive value. The payoff for B is high if the response does not reveal anything about the sensitive value. Note that if B does not give out any information or if it gives erroneous information then it cannot be regarded as a game, That is, the aim here is for B to participate in the game without giving away sensitive information.

Multi-party information sharing: The idea here is that certain parties play cooperative games while certain other parties play non-cooperative games. We illustrate with an example consisting of three parties. Let's consider the following situation. Suppose the year is 2006 and the UK has obtained some sensitive information on Operation Iraqi Freedom that the US needs. However, the UK is reluctant to share this information. The US in the meantime has formed an alliance with Argentina by giving some incentive either in the form of money or weapons. When the UK hears this, it is concerned thinking about the Falklands situation. However, in reality the US has no intention of doing anything about the Falklands but does not want the UK to know the truth. So the UK may reason about the benefits it receives by sharing the data with the US and makes a determination.

Cooperative games have also been called Coalition games. In a true coalition the players are friendly and therefore share the information and determine a collective payoff. However in our environment, organizations form coalitions only to solve a particular problem. An agency that is a trustworthy party in a particular coalition may turn against its partner at a later time and divulge the information gathered during the coalition operation.

We have conducted some initial research on game theory applications for AIS. Our objective has been to consider the interaction of participants within a loose coalition. In particular, we are interested in a scenario in which those involved have made a reluctant but necessary decision to trade information to achieve some goal. A great deal of work has already been done in the areas of secret sharing and protocol enforcement. However, even if agreements to exchange are kept, there is no guarantee what is shared is legitimate. The ultimate goal of this research is to create a behavior which works optimally against lying agencies while taking advantage of implicit trust. Our results at this point in the research suggest our algorithm is effective against basic opponents, though more refinement is needed. We report which behaviors work for the players and why, with regards to the motivating factors for each strategy. Our research is described in [9].

1.9 Handling Untrustworthy Partners

Note that in fighting the global war on terrorism we have to work with our allies as well as with countries that we may not trust. If our partners are untrustworthy, then we have to not only defend against malicious code but also figure out what the partners are doing both with their computers as well as their activities. Essentially we need to conduct information operations [19]. We will first discuss our research on defensive operations and then discuss some aspects of offensive operations.

Defensive Operations: In the case where partners are untrustworthy we have to defend ourselves against malicious code such as viruses and worms planted by our partners. In order to accomplish this, we are applying data mining techniques to detect such malicious code. Some of our research in this area can be found in [13].

Offensive Operations: There is little work in the unclassified published literature on offensive operations. However recently we are seeing articles published in Signal magazine on the importance of monitoring the adversaries' computing activities [17, 18]. Three of the techniques for handling untrustworthy partners include the following:

Trojan Image Exploitation: Modern anti-virus and anti-spy ware detection packages rely on the presence of malicious code within an executable or script to prevent attacks. This is done by detection methods that are carried out when the program first loads. In theory, it is possible to circumvent this detection by designing a program without any explicit malicious code; instead, a memory leak in this program's security is purposefully created. This weakness is exploited by downloading a tailored file from the Internet, such as a picture, after the program is loaded. As a result, this program could be used as a staging area for a malicious attack.

Web Browser Customization: Web browsers have been enhanced dramatically in the past year to prevent attacks from malicious web pages. For the benefit of the user, these features are frequently made optional, allowing a great deal of customization. By compromising a user's customization features covertly, it becomes possible to execute potential attacks without the user detecting any warning signs normally visible in the user's browser such that the attacker's methods can be hidden from the user. The attacker could use browser customization, such as enabling JavaScript, to create a shadow copy of the web and gain classified information from the victim without certain warning signs, such as URLs being correctly displayed. All user-entered information would be funneled through the attacker's spoofed world and thus the attacker could easily take advantage of the situation in order to retrieve any type of information.

Message Interception: Enron data set (publicly available) may be used to send emails to the partners of the coalition as well as to those outside of the coalition. Messaging may be simulated in such a way that they are sent at random intervals. We can then determine whether interception techniques can be used to extract some of the messages sent. This is a very challenging problem.

1.10 Summary and Directions

In this paper we have defined Assured Information Sharing (AIS) and discussed issues, technologies, challenges and directions for this area. The goal of AIS is for organizations to share data but at the same time enforce security policies. Security includes confidentiality, privacy, trust, and integrity policies. We discussed approaches for AIS when the partners of a coalition are trustworthy, semi-trustworthy and untrustworthy. In particular, we discussed security policy enforcement, game theory applications and defending against worms and viruses. We also discussed AIS technologies including data integration, data mining, and the semantic web.

There are several areas that need further investigation. We need to develop policies for accountability. This is especially important in a coalition environment. In such an environment, there are numerous pieces of hardware and software that interact with each other. Therefore, the action of all the processes has to be recorded and analyzed. Furthermore, risk analysis studies are needed to determine the risks and developing appropriate solutions. For example, in a high risk low cost security environment, there will be no questions about implementing security solutions. However in a low risk high cost environment one needs to think twice before enforcing the security policies. Essentially we need some form of risk-based AIS. We also need to develop web services for AIS. Essentially we need to integrate AIS and semantic web technologies. Finally we need to investigate several additional technologies such as collaborative services, social network analysis, surveillance data sharing, digital identity management, metadata extraction and management as well as policies for identification and authentication for AIS. We also need to investigate the use of standards as well as infrastructures such as data grids for AIS. Some of our preliminary research in some of these topics is reported in [8, 4, 28].

We are conducting extensive investigation on AIS with our partners George Mason University and Purdue University as well as with Arizona State University and the University of Maryland at Baltimore County. In addition to the technical aspects discussed in this paper, we are also investigating the connection between AIS and the Global Information Grid as well as Network centric Operations. While our primary application is counter-terrorism, we are also focusing on other applications such as Emergency preparedness and Healthcare. Future papers will focus on the design of our approaches as well as our experimental results for AIS.

Acknowledgements

The work reported in this paper is supported by AFOSR under contract FA9550-06-1-0045 and by the Texas Enterprise Funds. I thank Dr. Robert Herklotz for funding our research on Information Operations Across Infospheres from which I got the ideas to write this paper. I thank my colleagues Profs. Latifur Khan, Murat Kantacioglu, Elisa Bertino, Ravi Sandhu and Tim Finin as well as Dr. Mamoun Awad and Dr. Ebru Celikel and many others for discussions and inputs on AIS. I also thank my students Ryan Layfield, Nathalie Tsybulnik, Li Liu, Alam Ashraful, Ganesh Subbiah, Gal Lavee, Kim Jungin, I. Srinivasan, and Abin Chandrasekaran, as well as many others for discussions on AIS, and especially Ryan Layfield, Nathalie Tsybulnik and Li Liu for writing the techniques for information operations in section 9. A version of this paper was presented at a workshop held in Berkley California to honor Prof. Ramamoorthy' 80th Birthday on May 5, 2006. A version will be published as a book chapter on Security Informatics edited by Prof. H. C. Chen et al.

References

1. Awad, M., Thuraisingham, B., Khan, L., et al.: Assured Information Sharing: vol. 2, Experimental Analysis of Data Integration, Mining and Security, Technical Report, The University of Texas at Dallas, UTDCS44-06 (2006)

2. Ashraful, A., Thuraisingham, B.: Geography Resource Description Framework (GRDF) and Secure GRDF (S-GRDF), Technical Report, The University of Texas at Dallas, UTDCS-03-06 (January 2006); (also presented at W3C Working Group Meeting (October 2006); and Geospatial semantic web workshop, November 2006 (Athens, GA) (2006)
3. Bertino, E., Carminati, B., Ferrari, E., Thuraisingham, B.: Secure Third Party Publication of XML Documents. *IEEE Transactions on Knowledge and Data Engineering* (October 2004)
4. Celikel, E., Kantacioglu, M., Thuraisingham, B.: Assured Information Sharing: Risk-based Data Sharing. Technical Report, The University of Texas at Dallas (to appear, 2007)
5. Harris, D., Khan, L., Paul, R., Thuraisingham, B.: Standards for Secure Data Sharing across Organizations. *Computer Standards and Interfaces Journal* (accepted, 2006)
6. Jones, A.: Game Theory, Mathematical Models of Conflict. Halstead Press (1980)
7. Kim, J., Thuraisingham, B.: Dependable and Secure TMO Scheme. In: Proceedings of IEEE ISORC Conference (April 2006) (also UTD report UTDCS-02-06)
8. Layfield, R., et al.: Design of a Social Network Analysis System. In: Proceedings of the ACM SIGKDD Conference Workshop on Multimedia Data Mining (2005)
9. Layfield, R., Kantacioglu, M., Thuraisingham, B.: Assured Information Sharing: vol 3: Using Game Theory to Enforce Honesty Within a Competitive Coalition. Technical Report, The University of Texas at Dallas, UTDCW-46-06 (October 2006)
10. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American* (2001)
11. Liu, L., Kantacioglu, M., Thuraisingham, N., Khan, L.: An Adaptable Perturbation Model of Privacy Preserving Data Mining. In: Proceedings of the IEEE ICDM Data Mining Conference Workshop on Privacy preserving Data Mining (2005); (also published as technical report, UTDCS-03-06, January 2006)
12. Vatis, M. (ed.): Markle Report, Creating a Trusted Network for Homeland Security (2003)
13. Masud, M., Khan, L., Thuraisingham, B., Awad, M.: Detecting New malicious Executables Using Data Mining. UTDCS-27-06 Technical Report, The University of Texas at Dallas (June 2006)
14. The Implementation of Network Centric Warfare. Office of Force Transformation (2003)
15. Olivier, M.S.: Self-protecting Objects in a Secure Federated Database. In: Proceedings of the IFIP Database Security Conference, NY (August 1995)
16. Sandhu, R., Coyne, E., Hal Feinstein, H., Youman, C.: Role-Based Access Control Models. *IEEE Computer* 29(2) (February 1996)
17. Signal Magazine, AFCEA (May 2005)
18. Signal Magazine, AFCEA (February 2005)
19. Spitzner, L.: Honeypots, Tracking Hackers. Addison Wesley, Reading (2002)
20. Son, S., David, R., Thuraisingham, B.: An Adaptive Policy for Improved Timeliness in Secure Database Systems. In: Proceedings of the 9th IFIP Working Conference in Database Security, New York (August 1995)
21. Thuraisingham, B.: Novel Approaches to the Inference Problem. In: Proceedings of the 3rd RADC Database Security Workshop, New York (June 1990)
22. Thuraisingham, B.: Security Issues for Federated Database Systems, December 1994. Computers and Security. North Holland, Amsterdam (1994)
23. Thuraisingham, B.: Data Mining: Technologies, Techniques, Tools and Trends, December 1998. CRC Press, Boca Raton (1998)
24. Thuraisingham, B., Maurer, J.: Information Survivability for Real-time Command and Control Systems. *IEEE Transactions on Knowledge and Data Engineering* (January 1999)
25. Thuraisingham, B.: Web Data Mining and Applications in Business Intelligence and Counter-terrorism. CRC Press, Boca Raton (2003)

26. Thuraisingham, B.: Security Standards for the Semantic Web. Computer Standards and Interfaces Journal (2005)
27. Thuraisingham, B.: Database and Applications Security: Integrating Information Security and Data Management. CRC Press, Boca Raton (2005)
28. Thuraisingham, B.: Assured Information Sharing: vol 1: Overview. Technical Report, The University of Texas at Dallas (2006)
29. Thuraisingham, B.: Building Trustworthy Semantic Webs, September 2007. CRC Press, Boca Raton (2007)
30. Tsyblinik, N., Thuraisingham, B., Ashraful, A.: CPT: Confidentiality, Privacy and Trust for the Semantic Web. UTDCS-06-06, Technical Report, the University of Texas at Dallas (March 2006); Also to appear in the Journal of Information Security and Privacy

Automating Event Extraction for the Security Domain

Clive Best, Jakub Piskorski, Bruno Pouliquen, Ralf Steinberger,
and Hristo Tanev

Joint Research Center of the European Commission,
Web and Language Technology Group of IPSC, Italy
`firstname.lastname@jrc.it`

Abstract. This chapter presents on-going efforts at the Joint-Research Center of the European Commission for automating event extraction from news articles collected through the Internet with the Europe Media Monitor system. Event extraction builds on techniques developed over several years in the fields of information extraction, whose basic goal is to derive quantitative data from unstructured text. The motivation for automated event tracking is to provide objective incident data with broad coverage on terrorist incidents and violent conflicts from around the world. This quantitative data then forms the basis for populating incident databases and systems for trend analysis and risk assessment.

A discussion of the technical requirements for information extraction and the approach adopted by the authors is presented. In particular, we deploy lightweight methods for entity extraction and a machine-learning technique for pattern-based event extraction. A preliminary evaluation of the results shows that the accuracy is already acceptable. Future directions of improving the approach are also discussed.

2.1 Introduction

The increase in security concerns since 9/11 especially those related to terrorism have focused research interest on the problem of developing automatic monitoring systems for Open Source Intelligence. Terrorist use of the Internet for propaganda, training and planning purposes has been well documented and continues to expand [31]. The flood of information published on the Internet, especially live news reports from around the world provide an ever changing view of world events and opinions. Automatic systems should ideally monitor these reports round the clock, analyze their content, log all incidents and references to known persons and groups and alert authorities to potential threats. This paper discusses the technical challenges of providing such an idealized service. The first challenge is to identify and retrieve the most relevant data from within the sea of data on the Internet and elsewhere. The second challenge is to classify and organize this information into relevant topics. Specialized techniques are needed to identify the relevant information, to collect it and then to filter it for later processing. Having collected this textual information, the goal is to extract data concerning events and their associated details. The whole procedure can be seen as a two step process namely Information Retrieval (IR) and Information Extraction (IE). The primary goal of Information Extraction is to transform unstructured text into structured data suitable for storing in a database. This is illustrated in Fig. 2.1.

Goal of Information Extraction

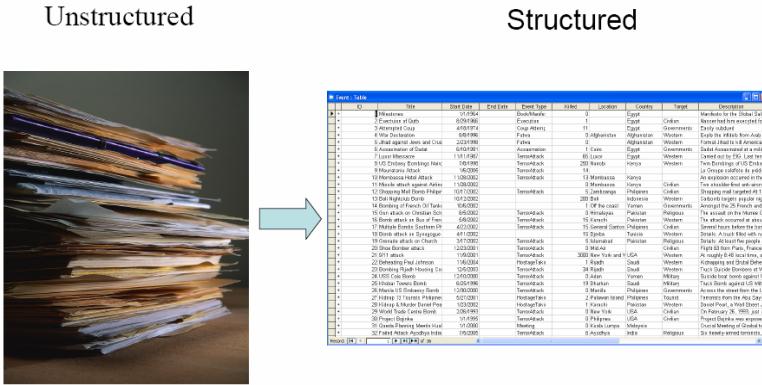


Fig. 2.1. The Goal of Information Extraction

The motivation for this work is twofold. Firstly to automate the population of incident databases in order to better understand conflict and terrorism. The second motivation is to develop early warning systems which attempt to detect precursors for threats in the fields of conflict and health, and study trends over time.

2.1.1 Incident Databases

Event data extracted from news reports can then be stored in so-called incident databases. Although there are potentially many incident databases that can be recorded, the following two examples are relevant for Security and Terrorism Informatics.

Terrorism

Terrorism Informatics applies information technology to record incidents which help understand the threats of terrorism, their root causes and countermeasures. Reference data is essential and several groups have supported terrorism knowledge bases. These systems consist of databases of terrorist attacks, profiles of each terrorist group and information on the persons involved. Traditionally all this data is entered by skilled editors, where most of the information comes from open sources. As the quantity of information has increased, so too has the manual load on the editors. Therefore systems which can automate this process at least as far as selecting and organising proposed input are needed. This information task can be seen as “event extraction”. Event extraction should ideally derive data from text to answer the question: Who did what to whom, when, where and with what consequences? This is an event and the software should attempt to identify as many “facts” as possible from the collection of news article describing each event.

Existing Terrorism databases that are in the public domain include the following; MIPT Terrorism Knowledge Base (TKB) [17], the ICT Terrorism database [13], and

the South Asian Terrorism Portal [26]. There are additional databases not available on-line held at various institutions and services. These are currently all maintained by human editors, but attention is now focusing on automating this process. One advantage of automation is that it removes the human subjective element. Thus even if the precision is not perfect, at least a consistent selection criteria is ensured.

Conflict

Political science research in the area of international conflicts has often been based on the analysis of world events as reported by news media. Event data research was developed in the 70's - 90's by defining coding schemes such as WEIS (World Event Interaction Survey) and more recently IDEAS (Integrated Data for Event Analysis) [4]. Since the 90's automatic coding software has been developed which can handle large quantities of reports. Another advantage of automatic event coding systems as opposed to human event coding, is that subjective influences can be avoided. The main software packages available for event coding are KEDS or TABARI [24] and VRA [30]. Traditionally, event coding software is based on analysis of the leading sentences in Reuters News Reports. This has the advantage that each report is already country related and concerns a single "event" that has happened somewhere in the world. Event coding software relies on natural language parsing. The event codes themselves are classified according to the severity of the action on a standard such as the Goldstein scale [9]. Likewise dictionaries of actors are used to identify classes of actors, be they civil or governmental persons. This event coding software is only fully available in English, and relies on exclusive rather than open sources.

A new approach to recording incidents is based on Semantic Web technology. The domain of violent events is described by an Ontology. The ontology defines the attributes and relationships between people, organizations and events. The incident data is recorded in RDF identifying the who as persons and organizations and the what as events and the relations between them. Repositories of such data are usually called a knowledge base which can then be queried and further information inferred through the ontology.

2.1.2 Trends and Early Warning

The objective of studying trends is to understand better the evolution of conflicts and general themes like natural disasters and the like. Initial results in this area have been achieved using simple statistics of the numbers of articles referring to countries and themes. This works in general for large events where multiple reports on the same subject measure in some sense the gravity, through simple media attention. An example is shown in Fig. 2.2 which plots two normalized indicators for terrorist attacks based purely on news report statistics. Although this represents the overall trends for the major events and allows to compare relative indicators between countries, it cannot be used for detailed studies, intelligence gathering or for fact extraction. Event tracking allows a much deeper study of underlying trends and provides the possibility of early warning based on small signals or patterns. It also allows a fine tuning of what is monitored namely keyword or terms, individuals and organizations. This is important both for Terrorism and Conflict early warning.

Another application area of particular relevance is that of disease outbreak monitoring based on Internet reports. There are currently two systems which perform this task for government agencies and the World Health Organisation (WHO). These are the Canadian GPHIN [11] and the MediSys system [16]. Incoming news reports, plus medical reports from worldwide public health web sites are monitored and subscribers are alerted of potential new outbreaks and threats. A specialised fact extraction system for disease incident detection has also been developed [32]. Each incoming report is scanned to identify one of a large list of diseases and to extract information on the

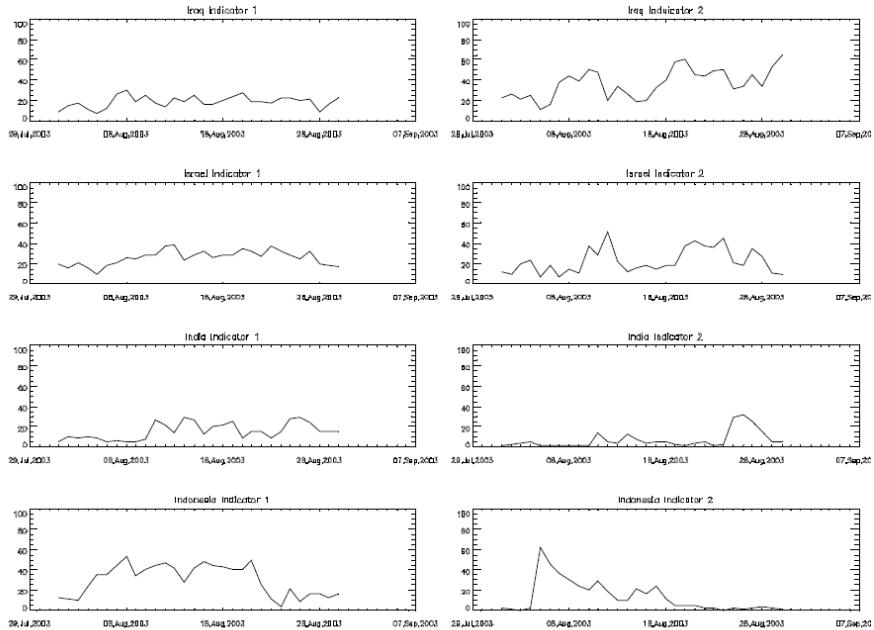


Fig. 2.2. TerroristAttack indicators for Iraq, India, Israel and Indonesia for August 2003. During this period each country suffered terrorist bomb attacks. Indicator 1 (Ijc) gives the percentage of media coverage on terrorism for each country whereas Indicator 2 (Icj) gives the percentage of media coverage for each country concerning terrorism. Indicator 2 shows the main incidents clearly.

number of human cases, their location and time. This is fed to a disease incident database and can also detect outbreaks defined as similar incidents at the same location. Eventually an automatic early warning of disease outbreaks should become possible. Such systems are also important for identifying potential threats from acts of bioterrorism to alert an analyst. This subject is particularly important with an increasing public concern on health risks such as Avian Flu.

In summary, event tracking aims to automatically extract detailed data about events reported in the text. These details try to answer the question - *who did what to whom where when and perhaps why?*. They are recorded in incident databases and knowledge

bases. The rest of this chapter reviews the on-going research, and results in this area, focusing on the application EMM. Sect. 2.2 reviews the field of Information Extraction and focuses on methods and challenges for event extraction. Sect. 2.3 describes how news reports from the Internet are collected and processed for the final step of event extraction presented in Sect. 2.4.

2.2 Information Extraction

This section defines relevant concepts, gives an overview of the state-of-the-art (Sect. 2.2.1), and explains the main methods in information extraction in general and event extraction in particular (Sect. 2.2.2). The discipline covering this type of work is called *Computational Linguistics*, which is itself multi-disciplinary because it exploits techniques from the fields of linguistics, computer science, machine learning, statistics, artificial intelligence and more.

While the term *Information Retrieval* (IR) refers to the more generic task of finding and gathering information (or rather: text that contains information), the term *Information Extraction* (IE) refers more specifically to extracting specific, structured information out of unstructured free text documents. A well-known example of IE is Named Entity Recognition (NER), i.e. the recognition of the names of persons, organisations and geographical locations and other entities. The category of the extracted information is known, it can be stored and thus searched and retrieved more efficiently. Further information that can be extracted from text includes dates, measurements, currency expressions, contact details (e.g., addresses, email addresses), national insurance numbers and bank details, references to weapons, vehicles, drugs, etc. Techniques to carry out these IE tasks are well-known and software for a number of different languages is widely available, although many other languages are not yet covered.

All of the above-mentioned information types are single entities that are not related to each other. An additional challenge is to extract relational information (such as person-organisation memberships, involving two entities and a type of relation), or even event information, involving a number of different entities and their relationships. Sample event extraction application is, for instance, succession scenario (X replaces Y on date A in position B in company C). The aim of the current work carried out at the JRC is to answer the questions: Who (Actor) did What (Event Type) To Whom (Victim or Affected), Where (Location), When (Time), Why (Motivation) and with What Damage (persons hurt or dead; infrastructure destroyed, financial or ecological damage, etc.). Obviously, any further attribute information for each entity, such as nationality or organisation membership of the actors, is of interest, as well. For each event to be extracted, it is necessary to model a scenario with different pieces of information that need to be found. Event extraction has accordingly been described as a scenario template filling task, where each piece of information is a slot of the template that needs to be filled. It goes without saying that some texts may not contain all pieces of information, or that an automatic system may not be able to fill all slots successfully.

2.2.1 History and State-of-the-Art of Event Extraction

The interest in event extraction was raised considerably with the beginning of a series of DARPA-initiated Message Understanding Conferences (MUC-1 to MUC-7, 1987-1998) (see [18]), which were kind of invitations to the scientific community to compete in pre-defined IE tasks. These extraction tasks covered a wide range of topics, e.g., fleet operation information, terrorist activities, joint ventures and leadership changes in business, plane crashes and missile launches. The later MUC conferences distinguished several different subtasks in order to facilitate the evaluation. In addition to Event Extraction (referred to in MUC as Scenario Template Task), these include Named Entity Recognition, Co-reference Resolution (identify all mentions of the same person or entity), Template Element filling (identify descriptive information about entities such as nationality or title) and Template Relation filling (filling two-slot templates such as for person-organisation membership).

The results achieved in the last MUC conference, MUC-7, give a good idea of what performance can be expected from IE technology. The best English language system managed to capture 42% of the event information present in the text (recall), and 65% of the automatically extracted results were correct (precision). In comparison, NER systems did much better, with recall and precision values of 92% and 95%, respectively. The results differ very much depending on the text type and complexity. Little work has been done to extract event information for languages other than English.

The ACE (Automatic Content Extraction) Program [1] is a follow-up to MUC endeavor. It defines new extraction tasks focusing on entity, relation and event extraction which are significantly harder to tackle than MUC-defined tasks (e.g., varying source and quality of input data, wider spectrum of template structures and entity (relation) types).

Several commercial event extraction software applications have become available over the recent years, focusing on the application domains business, medicine and security. These include the INFOTRACT system by Cymfony [12], their intelligence spin-off SEMANTEX by Janya [25] and Teragrams ENTITIES AND EVENTS EXTRACTOR [29]. The products DISCOVERER EXTRACTOR by TEMIS [6] and THINGFINDER PROFESSIONAL by InXight [14] not only offer ready-made extraction software for selected application domains, but also provide software that allows users to write their own extraction rules. A very interesting application, already mentioned in Sect. 1.1, was developed by Virtual Research Associates [30]. VRAs event coding software only analyses a small set of sources.

2.2.2 IE Methods and Challenges

The basic method to extract information from free text is to match patterns against the text. If the pattern matches, the information can be extracted. For instance, if a pattern such as [Uppercase Words] “killed” [Uppercase Words] is used and a text segment such as *Mary killed John* is found, we can say (with a certain degree of confidence) that *Mary* is the actor of a killing event in which *John* is the victim. The challenge is thus to identify possibly all patterns that describe the information we are looking for. However, as natural language has an almost infinite number of different

ways to express the same fact, it is not trivial to identify these patterns. The simple killing event involving *Mary* (A) and *John* (B), for instance, could also be described in the following ways:

- [a] A killed B
- [b] A shot/strangled/poisoned B
- [c] A has/had killed B
- [d] B was killed by A
- [e] A, a linguist from Munich, killed 56-year-old B
- [f] A has beaten B to death
- [g] Having killed B, A fled
- [h] B(*) is dead. A later admitted to having hacked and frozen her husband(*)

These variations can be classified into different types and they are of varying complexity. While some of the mentioned paraphrases are basically lexical (a vs. b) or morphological variants (a vs. c), others such as the active/passive variation (a vs. d) and the apposition in (e) are relatively simple syntactic variants. (f) shows that recognition patterns may be discontinuous (e.g., *beaten ... to death*). In (g), a deep syntactic analysis has to identify that A is the subject of the indefinite verb group *having killed*. Example (h) not only is complicated syntactically (the information is spread over two sentences), but also pragmatically (*hack* and *freeze* implies *killing*). Additionally, we have to infer that *her husband* refers to the same person as B.

Identifying the large number of possible patterns and converting them into a computer-readable grammar is a rather time-consuming task requiring the skills of trained linguists. It is thus desirable to reduce this effort to a minimum.

Reducing the number of different patterns

There are two main ways to reduce the task of finding patterns: (A) Simplify the text representation by pre-processing the text with linguistic parsing software – this will reduce the number of variants; and (B) automatically learn linguistic variants from large amounts of documents (e.g., via exploiting the Internet), using Machine Learning methods. Both methods can be combined.

Input simplification can be done to various levels. At the simplest level, it is possible to lemmatise all words (*killing*, *kills* and *killed* all get matched to *kill* or to use synonym dictionaries that contain the information which words have approximately the same meaning (*shoot*, *strangle* and *poison* are equivalent to *kill*). At the next level, parsers can simplify the sentence structure. Parsers can identify, for instance, that *linguist from Munich* and *56-year-old* are different types of modifiers of the persons A and B in example sentence (e), so that these modifiers can be omitted to reveal the bare sentence A killed B, which is then easier to analyse. Dependency parsers, which produce an even more abstract sentence representation, would furthermore produce the same sentence structure for the active and passive sentences (a) and (d), even though B is the grammatical object in the active sentence (a) while being the grammatical subject in the passive sentence (d). The more abstract and sophisticated the linguistic abstraction of each sentence, the less patterns are needed. However, the likelihood that the linguistic pre-processing software makes a mistake grows. At the JRC, we are experimenting with various levels of linguistic pre-processing in order to

determine the most efficient way to identify and express the patterns while trying to keep the system simple. Simplicity is crucial for being able to port the system to new domains and languages other than English, which is a priority in the context of the JRCs multi-disciplinary and multilingual work. Therefore, we also deploy machine learning techniques in order to automatically acquire patterns.

Factuality of reported events

Another challenge for event extraction is that we cannot be sure about the degree of factuality of the information automatically extracted from texts. There are three reasons for this: (A) some recognition patterns are more ambiguous and thus less reliable than others; (B) Some information sources are less reliable than others, and (C) the text may explicitly or implicitly mention how sure we can be that the reported fact is true. For instance, the following formulations could be ranked from ‘rather certain’ to ‘less certain’ or even ‘unlikely’: *A killed B*, *A was reported to have killed B* *A claimed responsibility for the killing of B*, *A denied responsibility for having killed B*, *A convinced the court that she did not kill B*, etc. A possible solution to these issues is to use a scoring system, where ideally each event slot is scored separately.

Aggregating IE results into one event template

Event templates consist of various pieces of information (slots) which need to be filled one by one. When trying to fill the templates, we identified three different types of problem. These challenges can be labeled as (A) summary reporting, (B) multiple reporting and (C) compactness of reporting.

The first challenge has to do with the fact that the number of events in active conflict zones such as post-war Iraq is so high that summary reports are frequently found. An example would be the sentence *This weekend, two car bombs and a suicide attack in Baghdad and Basra killed 13 and wounded 20*. While the total damage to persons is clearly stated, it will be difficult to identify time, location and victims for each separate event, and an extra effort has to be made to keep track of the number of events in the country.

The second problem is multiple reporting about the same event, both from the same news source written over time (e.g. updates on the number of victims of a recent event) and from different news sources. The Europe Media Monitor gathers English news articles around the clock from hundreds of sources. The speed of gathering the news and the redundancy in reporting is not only an advantage (higher coverage including multinational and multicultural viewpoints), but also a challenge for event extraction. The reason is that multiple counting of victims for the same event would make summary information inaccurate. It is not always easy to recognise that two articles report on the same event because we may find conflicting information for the same slot, such as *seven victims*, *12 wounded persons*, *at least 10*, *15*, etc., all referring to the same event. Such conflicting information can be solved via averaging, via voting, by weighting some news sources higher than others, or by weighting more recent articles more than earlier ones, etc.

The third challenge for event information aggregation is due to the fact that a single template is unlikely to capture the information for all event slots. Instead, various independent patterns (for the event, for the actor, for the victims, for time and place)

are needed, which means that there is a risk that the extracted information is not related to the same event. For instance, the time and location information found in a text may not belong to the identified violent event, but it may describe when and where the event was reported. A possible solution to this is to score information found close by in the text higher than information found in more distant parts of the text.

This section summarised current efforts and methods of tackling the task of extracting information from unstructured text. Comprehensive surveys can be found in [2] and [19].

2.3 News Gathering and Preprocessing

Before the event extraction process can proceed, news articles are gathered by dedicated software for electronic media monitoring and further linguistically preprocessed in order to produce a more abstract representation of the texts. This section presents this process in detail. Firstly, in Sect. 2.3.1 news gathering is addressed. Secondly, in Sects. 2.3.2 and 2.3.3 the effort of grouping news into clusters ideally including documents on one topic is described. Next, in Sect. 2.3.4 the in-house tools for performing lightweight linguistic preprocessing are briefly presented. Finally, Sects. 2.3.5 and 2.3.6 explain the scope of entity and event-keywords recognition performed and introduces the tagset for marking up the news respectively.

2.3.1 News Gathering

Round the clock news monitoring on the Internet is made possible by a dedicated software system called Europe Media Monitor (EMM) [3], which regularly checks for updates of headline across multiple sites. Another example of a news aggregator is Google News. EMM uses a technique called scraping which converts each news section of the web site to a single RSS (Really Simple Syndication). This process parses the HTML into a simplified XHTML form before applying style sheets to complete the process. The topic selection is performed by the Alert System, which applies a specialized technique to process text from web pages.

EMM is implemented as a suite of WEB applications. Fig. 2.3 gives an overview of all the systems and how they interact.

EMM processes incoming news articles in a modular process. Some 1000 web sites are scanned up to every 10 minutes by Web-scraper. This system is driven by two configuration files. One defines the sites to monitor and then for each site a recipe file tells scraper how to analyse the site. The current headline content of each site is cached in RSS files one per site.

The Alert System is called after each scrape and checks whether any new headlines have been detected since the last scrape. If so it follows the URL and extracts the pure text from the web page and filters the content through its 650 Alert definitions containing 10,000 keywords. Each alert is defined as lists of multilingual keyword patterns which can each be weighted or combined in Boolean combinations. The algorithm is based on a finite state machine and keeps track of the total score per alert of the article. Articles which trigger a topic alert are appended to the relevant RSS topic feeds.

The Breaking News System (BNS) processes separately the content of all articles to identify a sudden appearance of new topics, and the indexer maintains a free text search index of recent articles. The results from the Alert system and BNS are updated continually through XML files, which update the user Web interface - NEWSBRIEF every 10 minutes. This can be accessed publicly at <http://press.jrc.it>

EMM also provides an e-Mail alerting facility both for individual topics and for breaking news. Some 35,000 articles are processed per day by EMM and presented live on the NEWSBRIEF. In a second step after midnight all the articles for that day are then clustered in each language to identify the main stories of the day and further information extraction performed as described in the next sections. The NEWSEXPLORER interface on EMM provides an interactive overview of this process.

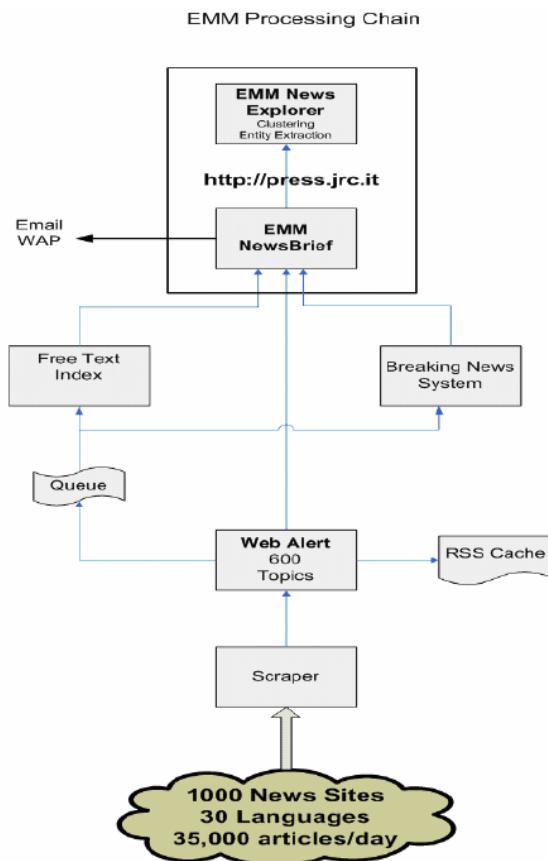


Fig. 2.3. Overview of EMM Processing Chain

2.3.2 Clustering: Grouping Similar Articles

In order to avoid duplicates and make use of the redundancy of the information, a clustering program reduces the news of the day (about 7000 English articles) to the

most important ones (about 200 clusters). It allows the events to be detected beyond the document level to the story level.

We use a *bottom up agglomerative clustering* algorithm, based on a similarity measure between documents. Each document is represented by a vector consisting of a list of keywords and their keyness. Keywords are the words of the document. Their keyness is computed by comparing the frequency of the word in the document to the frequency of the word in a reference corpus. This vector is enhanced by a normalised country score, depending on the number of references to each country (including country name, capital city, cities, inhabitant name, etc.). Adding the country score to the list of keywords helps to distinguish clusters with contents that are similar (such as two reports on terrorist attacks), but that happen in two different countries.

The similarity between two documents is a simple *cosine* of their vector. The algorithm compares all pairs of documents, then groups together the pair having the highest similarity. This new group is treated as a single document and the process is re-launched iteratively until every document belongs to a group or forms a group by itself.

On the NEWSEXPLORER (see next section) clusters are linked over time and across languages. Two clusters are linked over time in the same language if they share their most important keywords. We automatically identify the similar clusters of the past seven days. Linking clusters across languages is another challenge because, to do this, we first need to find a language-independent representation for the texts in each language. For this purpose we use a combination of four representations: A selection of classes from the multilingual thesaurus EUROVOC, lists of person and organisation names, lists of the geographical references, and the words themselves. The whole process is described thoroughly in [27].

2.3.3 NewsExplorer: Exploring News through Clusters and Entities

NEWSEXPLORER is an online portal (<http://press.jrc.it/NewsExplorer/>) allowing anyone to explore the main international News of the last three years. The main page displays the clusters containing most of the articles (importance of an event being computed according to the number of newspapers reporting this event; for example, the 2005 London bombing event was reported in 200 articles in a single day). Each cluster can then be displayed showing the main article (the article closest to the average representation), the other articles belonging to this cluster, the entities extracted (countries, persons, organisations) and information about related articles in the past seven days or in other languages (see Fig. 2.4).

Each entity has its own page showing the various spellings of the name, the trigger words associated, the associated persons/organisations, and the clusters where it appeared. A frequency-based algorithm for counting co-occurrence of entity pairs in the same clusters automatically discovers relations between people and organisations. A weighting method furthermore links personalities that are mainly mentioned together (see ‘related people’ and ‘associated people’ in Fig. 2.5).

The NEWSEXPLORER currently covers 13 languages (soon 17). Every day, it compiles the main news of the day and updates information about each person in the news.



Fig. 2.4. Example of NewsExplorer cluster

2.3.4 Corleone

A crucial prerequisite to tackle the problem of IE from huge document collections is the capability of robustly and efficiently performing basic linguistic operations, including splitting a text into word-like units, associating them with their corresponding syntactic category/base form, and recognising of domain-relevant terms and entities. Therefore, we have developed CORLEONE (**C**ore **L**inguistic **E**ntity **O**nline **E**xtraction) [21] - a pool of general-purpose Unicode-aware basic lightweight linguistic processing resources for carrying out the aforementioned tasks. Currently, it consists of a tokenizer, morphological analyzer, and a domain-specific dictionary look-up component whose functionality and peculiarities are described briefly in this subsection. We heavily exploited state-of-the-art finite-state techniques for implementing the tools since finite-state machines guarantee efficient processing and have an expressive power to model relevant language phenomena in the context of processing online news texts. Any linguistic resources used may be manipulated and parameterized in a straightforward manner, which eases adapting the tools to processing texts in a new language.

Our tokenizer performs two tasks: segmentation of the text into word-like units called tokens, and their fine-grained classification. In order to facilitate the definition of IE patterns, circa 40 default IE-oriented token classes are provided, e.g. hyphenated-word-with-apostrophe-first-capital (*Shiite-dominated*), all-capital-word (*NATO*), word-number-first-capital (*F10*), etc. These token classes are in general language-independent and cover well all Euro-zone languages.

The morphological analyzer maps each token into its linguistic properties, including its base form, syntactic category and other morphosyntactic features, e.g., morphological analysis for the word *attacked* would yield as one of the interpretations: *base-form:attack cat:verb type:main tense:past form:indicative*, i.e., a indicative form of the verb *attack* in the past tense (crucial information for defining higher-level IE patterns). In order to cope with the multilinguality, for encoding the

morphosyntactic data (full forms) we have chosen the MULTEXT format [8] which uses the same uniform tagset for all languages. In addition, there are freely available resources for several languages which we benefit of. Finally, the morphological analyzer allows for triggering language-specific components for handling complex structures. In particular, we have developed a language-specific component for English which among other tries to morphologically analyze hyphenated words, which are frequently used in the security domain and they can not be simply lexicalized since the process of constructing such structures is very productive. Assigning hyphenated words their part-of-speech is realized via application of semi-automatically created patterns (ca. 500) which map combinations of part-of-speech information of single elements of hyphenated words into a part-of-speech information for the whole construction, e.g., U-V → A means that hyphenated word consisting of an unknown word and a verb is an adjective (e.g., *Shiite-dominated* matches this pattern).

The domain-specific dictionary look-up component matches an input stream of characters against a list of named entities and named expressions, e.g., people names, organizations, currency units, violence keywords, etc. and associates identified text fragments with additional information present in the domain knowledge base, represented as an arbitrary list of attribute-value pairs. For instance the phrase *New York's* would be associated with the following annotation:

```
concept:new_york
type:city
country:usa
case:genitive
latitude:40'47
longitude:73'58
```

Since exhaustive dictionaries play an enormous role in building IE systems, a novel space and time-efficient storage models for handling huge IE dictionaries have been developed [20].

2.3.5 Entity Recognition

Entities are temporal expressions (*Wednesday*), numbers (*two*), geographical references (*in Medan*), persons (*officers*), organisation names (*police*) and specific terms (*explosion*). After having identified these entities, the templates for event extraction can make use of this meta information rather than to the raw texts. The final aim is to mark up all articles in a convenient way so the event extraction can work on a more abstract representation of a text.

Recognising names in text is done mainly using two techniques: guessing names (looking for patterns that trigger an unknown name) or dictionary lookup (looking in a repository of known names). Looking for geographical places belongs to the second category. We cannot rely solely on the context to “guess” place names (the context alone would not help us to identify that *Another Iraqi bomb* refers to the country *Iraq*). However, lists of geographical place names (gazetteers) exist and can be used. Our repository contains more than 500 000 geographical places around the world (we have used a compilation of three gazetteers as described in [22]). For looking up place names, we use a memory and time-efficient automaton described in detail in

Sect. 2.3.4). Looking up place names is not sufficient because of high ambiguity. Place names can have other meanings, e.g., *Split* (city or verb) and *Bush* (city or person). An additional difficulty is to choose between various homonyms (e.g. *Paris* is the name of 45 different cities and villages in the world). For this purpose we deploy disambiguation rules as described in [22], relying on features such as the importance of the place, the context of the document, kilometric distance, etc.

Person name recognition, on the other hand, cannot rely only on a dictionary. We guess names using an in-house tool currently relying on an extensive list of first names (including foreign names, i.e. *Peter*, *Pierre*, *Pietro*, *Petro*, *Petr*, *Peetrus*, etc. and a list of “trigger words” (or character-level patterns). The current patterns are quite simple but easily extendable to new languages. The “trigger words” are words which precede or succeed person name (like *Sir*, *spokesman*, *Secretary General*). They can also consist of more sophisticated patterns like [Number]+-year-old, Mayor of [Upper-case word], leader of [Organisation], which cover text fragments such as *12-year-old*, *Mayor of London*, *leader of Al Qaeda*, respectively. We envisage to develop and apply a more complex grammar based on regular expressions over feature structures via the utilisation of the pattern engine described in Sect. 2.4.3 in order to improve the coverage and to associate identified entities with more fine-grained output structures.

The daily processing of the news guesses 1000 new person names per day. These names are stored in a database which currently consists of about 600,000 persons. These names are then compiled into an optimised finite-state representation. A description of the name recognition process is available in [23]. The original aspect of our repository is that it automatically recognises various spellings of the same name (*Bin Laden*, *bin-Laden*, *Ben Laden*, *Ben Ladin*, etc.). The most used trigger words near a person name are also stored in the repository (which allows us to show them on the NEWSEXPLORER, see Fig. 2.5).

Additionally, further information can be derived from our repository, like the nationality, the occupation type (governmental, civil, religious, etc.).

2.3.6 Markup of News

Using the techniques described earlier, we are able to mark up each incoming article with a new format (encoded in XML), including the recognised entities.

The example in Fig. 2.6 shows the result of the pre-processing, where entities and some phrases are enclosed in the following tags:

- ACTION: Encloses verbs or phrases used to design a man-made-violent action (*bomb*, *exploded*, *injured*, *killed*, *suicide-attack*, etc.);
- DATE: a temporal expression (*yesterday*, *last Sunday*, *13/01/2003*, *13th of January 2004*);
- ORG: An organisation name (*Al Qaeda*);
- GEO: A named place, could be a country name (*Iraq*) inhabitant names (*Iraqis*) or a city (*Baghdad*). It contains the attributes Latitude/Longitude so that we can display the event on a geographical map;

- LOCATION: A more general geographical position relying on some patterns (*In the South Province of ..., about 10 kms North of the capital, near the Pakistani border, in the Bagdad area ...*);
- NUM: A number (possibly in letters);
- TRP: Trigger word for Person, any word or complex pattern that can be used to refer to persons (*men, others*);
- PERSON: Can be a named person (i.e. *Al Zarqawi*, found in our repository) or a combination of number and TRP (i.e. *ten Iraqi soldiers*).

Replacing the recognised entities with their more generic place holders reveals the more generic pattern of the sentence:

"on" [DATE] "an" [EXPLOSION] "in" [LOCATION] [KILLED] [PERSON]
 [PERSON]
 "and" [INJURED] [PERSON(s)], "a" [PERSON] "said".



Fig. 2.5. Example of NEWSEXPLORER entry: Bin Laden

On **<DATE d="05/04/2006">Wednesday</DATE>** an **<ACTION>explosion</ACTION>** in **<LOCATION>the headquarters of the <ORG>paramilitary police command</ORG></LOCATION>** **<LOCATION>in the western Indonesian city of <GEO lat="3.5852" lon="98.6751">Medan</GEO></LOCATION>** **<ACTION>killed</ACTION>** **<PERSON><NUM n="2">two</NUM>** **<TRP>officers</TRP>** **<PERSON>** and **<ACTION>injured</ACTION>** **<PERSON>several</TRP>** **<TRP>others</TRP>** **<PERSON>**, **<PERSON>a <TRP>police spokesman</TRP>, <TRP>Brig. Gen.</TRP>** Anton Bahru Alam**</PERSON>** said.

Fig. 2.6. Markup example

2.4 Event Extraction

This section describes our core endeavor towards fully automatic event extraction. In order to tackle this task we use a blend of machine learning and knowledge-based techniques. Firstly, we acquire patterns for recognition of partial information concerning events in a semi-automatic manner. Contrary to other approaches, the learning phase is done via exploiting clustered news (see Sect. 2.3.2), which intuitively guarantees better precision of the learned patterns. The details thereof are discussed in Sect. 2.4.1. Secondly, we enhanced the set of automatically learned patterns by adding manually created multi-slot extraction patterns. The on-line application of the patterns and currently applied methods for merging partial information into fully-fledged event descriptions are presented in Sect. 2.4.2. For encoding the patterns we have developed our own finite-state based pattern specification language which is somewhat of an amalgam between two known IE-pattern engine frameworks. Its short description is provided in Sect. 2.4.3. Finally, we report on some preliminary evaluation of our approach and initial deployment in Sects. 2.4.4 and 2.4.5 respectively.

2.4.1 Machine Learning for Pattern Acquisition

State-of-the art IE systems make use of patterns to recognize named entities, semantic relations and roles. In the context of event extraction patterns are used to find information relevant to events detected in a text: participants, places, dates, consequences, damages, etc. Consider the following sample patterns.

```
[1] [NUMBER] "people were killed"
[2] [ORGANIZATION] "launched an attack"
```

The first pattern ([1]) may be used to extract the number of the victims in a violent event, whereas the second extracts actors which initiate violent events. In general, IE patterns can be linear sequences or syntactic trees, they may contain NE classes or semantic elements. Usually, each pattern is a language construction which contains one or more slots and some context. In event extraction, patterns are intended to extract text entities which have different roles. In the case of violent events we have to learn patterns for affected dead, affected wounded, material damages, actors, places, dates, etc.

While in the past, IE systems used patterns created manually by human experts, the modern approach is to use automatic or semi-automatic techniques for their acquisition [15]. The strong side of ML is its ability to generate in a relatively short time a large set of pattern constructions. While human experts will not be able always to predict all the possible linguistic variants of a statement, the ML approach using vast amount of textual data can efficiently capture the language variations.

However, since ML approaches are error-prone, building a high quality pattern library still requires some manual intervention. A realistic and efficient method for acquisition is to use first ML for automatic learning of an initial set of patterns and next to manually filter out the bad ones from this list. Due to the data sparsity sometimes it is necessary to manually add entries to this library. Going one step further, the human experts can create more abstract rules on the basis of the automatically learned ones.

Pattern Learning Approach

We adopted a combined iterative approach for pattern learning which involves both ML and manual validation. Here are the basic stages of our method:

1. We annotate manually a small corpus with event-specific information, where for each event we specify date, place, actors, affected dead, injured, etc. As an example consider the annotation in Fig. 2.7. In this text fragment different entities (e.g. *five people*) are labeled with their roles (e.g. *affected_dead*).
2. Learn automatically single-slot patterns as described in Sect. 2.4.1. Each pattern covers a different type of information (wounded, dead, actors). One of the patterns which could be extracted from the above text fragment is [ORGANIZATION] “claimed the responsibility”, where the entity filling the slot [ORGANIZATION] will be assigned the role *actor*.
3. Manually check and filter out low quality patterns. If necessary, individual ones can be changed or new patterns can be added to the list. If the size of the list is over a certain threshold - stop the process.
4. The patterns are matched against the articles from EMM. Then entities which fill the slots of the patterns and comply to the semantic constraints of the slot are taken as *anchor entities*. If an anchor entity *A* (e.g. *five people*) is assigned a role *R* (e.g. *affected_dead*) in the news cluster *C* (see Sect. 2.3.2), we may assume with high confidence that in the cluster *C* entity *A* appears mostly in the same role *R*. Following this assumption, we annotate automatically all the occurrences of *A* in *C* with the semantic label *R*.
5. Go to step 2.

```
<AFFECTED_DEAD>five people</AFFECTED_DEAD>were killed and
<AFFECTED_WOUNDED>one</AFFECTED_WOUNDED> was injured during a
<ACTION>suicide bombing</ACTION> <PLACE>North of Baghdad</PLACE>.
<ACTOR>The terrorist group XXX</ACTOR> claimed the responsibility
```

Fig. 2.7. Sample annotation of input data

In this schema the iterative learning process may run until the desired coverage of the acquired library is reached. Consider the following simplified example for learning patterns for the semantic slot *affected_dead*:

- Suppose after performing steps 1, 2, and 3 two patterns have been learned: [PERSON] “was killed” and [PERSON] “was murdered”
- In step 4 the system introduces additional annotation in the news corpus with these two patterns. That is, each time it encounters a text like *John Brown was killed* or *John Brown was murdered*, *John Brown* will be annotated as *affected_dead* in the whole news cluster in which the text appears.
- Since the annotation of our corpus was enriched, we run again the ML algorithm in step 2. Let us suppose we acquire at this step two new constructions [PERSON] “was arrested” and “stabbed to death” [PERSON].
- In step 3, the pattern [PERSON] “was arrested” will be filtered out by the human expert as irrelevant to the semantic role *affected_dead*.

- Annotate the corpus with the newly acquired pattern in step 4. For example, in the text fragment *An unknown man stabbed to death a 23-years old woman*. the system will annotate *23-years old woman* as *affected_dead* both in this text and in the whole cluster.
- Learn new patterns in step 2.
- This process may be repeated a predefinite number of times or it may run until no more constructions can be learned or the human expert says that the learning should stop in step 3.

A similar kind of learning using the Web as a corpus is described in [28]. Disadvantage of Web-based approaches is that usually they are dependent on the speed of a public search engine; moreover, the linguistic quality of the Web documents in general is lower than the quality of a news corpus. Another disadvantage is that the Web documents are not clustered and therefore it is difficult to capture the set of documents where an anchor appears in the same semantic role.

Learning Linear Patterns from Anchor Contexts

In step 2 the pattern acquisition schema has to learn patterns from a corpus with labeled entities (anchors). We developed a ML approach which learns sequences of tokens which have a slot on the left or on the right side. That is why we consider separately the left and right contexts of all the equally labeled anchors.

Suppose our system has to learn a pattern from the following right contexts of anchors labeled as *affected_wounded*, where the position of the anchor entity (i.e. the slot) is denoted by P:

```
P "was heavily wounded in a mortar attack"
P "was heavily wounded when a bomb exploded"
P "was heavily injured in an accident"
P "was heavily injured by a bomb"
P "was heavily injured when a bomb exploded"
```

There are different pattern candidates. For example P "was heavily" is the most general one, but it is not very informative. On the other hand, the pattern P "was heavily wounded in a mortar attack" is too specific and contains redundant words. Our criterion for pattern selection is based on the so called *local context entropy maximum*. For each candidate pattern we take as its context all the adjacent words to all its occurrences. If the slot of the pattern (e.g. P in the above example) is on the left side, we take as context the words which occur on the right and vice versa. Considering again the above list of phrases, the pattern P "was heavily" has two context words: *wounded* and *injured*, each of which co-occurs twice with the pattern. Taking this into consideration, a *context entropy* for each pattern t can be calculated using the following formula:

$$\text{context_entropy}(t) = \sum_{w \in \text{context}(t)} p(w|t) - \ln(p(w|t)^{-1}) \quad (2.1)$$

where $\text{context}(t)$ is the set of the words in the immediate context of the pattern t and $p(w|t)$ is the probability that a word appears in this context.

Intuitively, the more words we add to a phrase, the lower its context entropy becomes. However, when a pattern is semantically consistent and complete, it may have higher context entropy than some of its sub-patterns. This is because a complete phrase is less dependent on the context and may appear in different linguistic constructions, while the incomplete phrases appear in a limited number of immediate contexts which complete it. For example, the phrase P "was heavily injured" has higher right-context entropy than P "was heavily" which can be completed by only two words in the above example.

In order to introduce formally our selection criterion, we have to consider a partial order of the pattern constructions. For simplicity we will explain this partial order for constructions in which the slot is on the left: We say that a phrase τ_1 precedes τ_2 , when τ_2 can be obtained from τ_1 by adding one word on the right. For example P "was heavily" precedes P "was heavily wounded". Considering this ordering, we may introduce our LOCAL CONTEXT ENTROPY MAXIMUM CRITERION: A pattern τ satisfies this criterion only when all the patterns which precede it have the same or lower context entropy and all the patterns it precedes has lower context entropy. Finally, we select only the patterns which satisfy this context entropy maximum criterion and do not contain other patterns which satisfy it. In the example above we will select only P "was heavily wounded" and P "was heavily injured".

2.4.2 On-Line Extraction and Information Aggregation

NEXUS (**N**ews cluster **E**vent **e**xtraction **U**sing language **S**tructures) is our prototype event extraction system which firstly uses linear patterns acquired via the process described in Sect. 2.4.1 in order to capture partial information concerning an event and secondly merges the single pieces into an event descriptions. The system processes the news clusters on a daily basis and for each detected violent event it produces a frame, whose main slots are: date and location, number of killed and injured, kidnapped people, actors (who initiated the violence), and type of event, i.e. shooting, bombing, missile attack, etc.

NEXUS uses heuristics to preselect documents describing security-related events. The pre-selection algorithm relies on keywords, patterns matched, and statistical clues and finds the news clusters which are security-related. Next, the system tries to detect and extract the main event of each cluster via analyzing all documents in the cluster. Since the news articles refer to this main event in the title and in the first sentence, the system applies the slot-filling patterns only based on the title and the first sentence of each news article.

After the pattern matching phase, the system has a set of text entities with semantic roles assigned for each cluster in the collection. If one and the same entity has two roles assigned, a preference is given to the role assigned by the most reliable group of patterns. The double-slot patterns like X shot down Y which extract two entities at the same time are considered the most reliable. Regarding the one-slot constructions, the system considers the ones for detection of affected_dead, affected_wounded, and affected_kidnapped as more reliable than the ones for extraction of the actor (the latter one being more generic). All these preference rules are based on empirical observations which have been confirmed by the evaluation presented in Sect. 2.4.4.

Another ambiguity arises from the contradictory information which news sources give about the number of killed and wounded. NEXUS uses an ad-hoc algorithm for computing the most probable estimation for these numbers. After this estimation is computed, the system discards from each news cluster all the articles whose reported numbers of killed and wounded are significantly different from the estimated numbers. The entities which are extracted from the discarded articles are also filtered out. In this way, only the most important named entities of the news clusters are taken into consideration when merging the pieces of information and filling the slots of the final event description frame.

The sketched approach works in most cases satisfactorily on our news data. However, it has some limitations, the most important of which is that it considers only one main event per news cluster, ignoring events with smaller importance or incidents subsumed by the main event. In the security related domain it is often necessary to detect links between events. For example, a kidnapping typically includes capturing a hostage, a statement by the kidnappers in which they declare what they want to liberate the abducted person, police action to liberate the hostage, and finally his or her liberation. The current implementation of *nexus* is able to detect these events separately, but it cannot aggregate them into one complex event since *nexus* does not have a temporal reasoning component, i.e. references to past events are captured as current events.

Although there is clearly space for improvement, *nexus* already produces compact and structured event descriptions with satisfactory quality - potentially useful for political analysts. Some evaluation figures are presented later in Sect. 2.4.4

2.4.3 Pattern Matching Engine

Recently, several high-level specification languages for creating IE patterns have been developed. The widely-known GATE platform which has been heavily exploited in the last years for construction of IE systems comes with JAPE (Java Annotation Pattern Engine) [5] for the development of IE grammars. A JAPE grammar consists of pattern-action rules. The left-hand side (LHS) of a rule is a regular expression over arbitrary atomic feature-value constraints (the recognition part), while the right-hand side (RHS) constitutes a so-called *annotation manipulation statement* which specifies the output structures to be produced once the pattern matches. The RHS may call native code (e.g., C or Java), which on the one side provides a gateway to the outer world, but on the other side makes pattern writing difficult for non-programmers.

A similar, but more declarative and linguistically elegant pattern specification formalism called XTDL is used in SPROUT [7]. In XTDL the LHS of a pattern (recognition part) is a regular expression over typed feature structures¹ with functional operators and coreferences, and the RHS is a typed feature structure, specifying the output production. Coreferences express structural identity, create dynamic value assignments, and serve as means of data transfer from LHS to RHS of a pattern. Functional operators are primarily utilized for forming the slot values in the output structures

¹ Typed feature structures (TFS) are related to record structures in many programming languages. They are widely used as a data structure for natural language processing and their formalizations include multiple inheritance and subtyping, which allow for terser descriptions.

(e.g., concatenation of strings, converting complex number expressions into their corresponding numeric values) and, secondly, they can act as Boolean-valued predicates, which can as well be utilized for introducing complex constraints in the rules. The aforementioned features make XTDL more amenable pattern formalism than JAPE since they eliminate writing ‘native code’, and, secondly, TFSs with coreferencing feature allow for precise description of linguistic phenomena in a very compact manner. Nevertheless, processing such patterns involves performing unification, which is known to be a rather expensive operation.

In order to find a trade-off between ‘compact linguistic descriptions’ and efficient processing we have developed our own grammar formalism which is similar in spirit to JAPE, but also encompasses some features and syntax borrowed from XTDL. In our formalism the LHS of a pattern is a regular expression over flat TFSs, i.e., non-recursive typed features structures without coreferencing, where types are not ordered in a hierarchy. Unlike JAPE, we allow for specifying variables² tailored to string-valued attributes on the LHS of a pattern in order to facilitate information transport into the output descriptions. Further, like in XTDL, functional operators can be used on the RHSs for performing some operations in order to produce proper slot values. Finally, we adapted the JAPEs feature of associating patterns with multiple actions, i.e., producing more than one annotation (eventually nested) for a given text fragment. The following pattern for matching partial information concerning events, where one person kills another by shooting, gives an idea of the syntax.

```

killing-event :> ((person & [FULL-NAME: #name1]):killed
                     key-phrase & [SURFACE: "shot down by"]
                     (person & [FULL-NAME: #name2]):killer):event
-> killed: victim & [NAME: #name1],
    killer: actor & [NAME: #name2],
    event: violence & [TYPE: "killing",
                       METHOD: "shooting",
                       ACTOR: #name2,
                       VICTIM: #name1,
                       ACTOR_IN_EVENTS: inHowManyEvents(#name2)]
```

The pattern matches a sequence consisting of: a structure of type `person` representing a person or group of persons who is (are) the victim of the event, followed by the phrase *shot down by* and another structure of type `person` representing the actor. The symbol `&` links a name of the structure type with a list of constraints (in form of attribute-value pairs) which have to be fulfilled. The variables `#name1` and `#name2` establish variable bindings to the names of both humans involved in the event. Further, there are three labels on the LHS (`killed`, `killer`, and `event`) which specify the start/end position of the annotation actions specified on the RHS of the pattern. The first two actions (triggered by the labels `killed` and `killer`) on the RHS simply produce structures of type `victim` and `actor` respectively, where the value of the `NAME` slot is created via accessing the variables `#name1` and `#name2`. Finally,

² Although coreferences are not necessarily indispensable in our patterns, we have designed the formalism in such a way so that variables can be upgraded to have ‘coreference’ functionality at a later stage.

the third action produces an output structure of type violence. The value of the ACTOR_IN_EVENTS attribute is computed via a call to a functional operator `inHowManyEvents()` which contacts some knowledge base to find out the number of all events the current actor was involved in the past (such information might be useful on higher strata). It is worthwhile to note that grammars can be cascaded, i.e., output produced by one grammar can be used as input for the grammar on higher level.

2.4.4 Preliminary Evaluation

A preliminary evaluation of the NEXUS performance has been carried out on 26333 English-language news articles grouped into 826 clusters from the period 24-28 October 2006. NEXUS classified as security-related 47 events; for 39 of them the classification was correct. Since only the texts returned by the system were taken into consideration, evaluation measured the precision of NEXUS and not its coverage. Additionally, for the slot actor the recall of the system was measured, but still considering only the texts which the system returns.

For each news cluster estimated as security related, NEXUS considers the title and the first sentence of each article to find an estimate for the number of killed and wounded. Finally, the system returns only the title and the first sentence from these articles where the numbers of killed and wounded are close to the cluster estimate. The event extraction is performed on these texts and the rest are ignored. Using these article fragments, the `nexus` accuracy was evaluated with respect to security related events classification, date and place detection, and the pattern-based slot filling for dead, wounded, kidnapped, and actors. Since NEXUS assumes that the date of the news cluster is the date of the extracted event, the date is considered correctly detected, only if the cluster reports a current event. On the other hand, each reference to a past event is considered incorrect date detection. The evaluation of the slot filling was carried out only on the 39 events which were correctly classified as security-related. In Table 2.1 precision figures are presented.

Our evaluation shows that there is still space for improvement of date detection and security-related events classification. Regarding the identification of places, our geolocation algorithm is precise at the level of country (95%). However, some temporary technical problems in the current implementation did not allow us to achieve high accuracy (28%) at the level of city, town and village.

Table 2.1. NEXUS performance

Detection task (slot)	Precision (%)
security-related events classification	83
Date	76
Country identification	95
City, town, village	28
Affected dead	91
Affected wounded	91
Affected kidnapped	100
Actors	69

Regarding the pattern-based slot filling (the last four rows of Table 2.1), the precision varies between 69% (actors) and 100% (kidnapped). The recall of the actor extraction was measured and was found to be 63%. Taking into account that NEXUS relies on superficial patterns and no syntactic information is exploited, these figures can be considered quite satisfactory at this stage.

As stated previously, the scope of this evaluation was to measure the system precision. The coverage is still to be addressed in future evaluations.

2.4.5 Initial Deployment

Events from 2005 and 2006 have been processed and filtered according to their geolocation. Two initial results are presented here for aggregated casualty figures. The first in Fig. 2.8 concerns known conflicts in the Middle East where the monthly death tolls are summed and presented as trends. This period covers the Lebanon conflict and the Iraq and Afghan conflicts. A comparison of the casualty figures for Iraq with manually recorded figures at the JRC from March to August 2006 was possible. The figures agree for five out of six months, and are overestimated for March. One suspected systematic effect is double reporting of the same incident across different days, since the analysis assumes a single event per day. This in effect weights the casualty figures by media attention, and is something that can be avoided by linking the same events across time, or through manual moderation. The second example in Fig. 2.9 shows casualty counts for three other known conflicts. Here the political trends in each conflict are clearly visible. Nepali insurgents activities has seen fases of

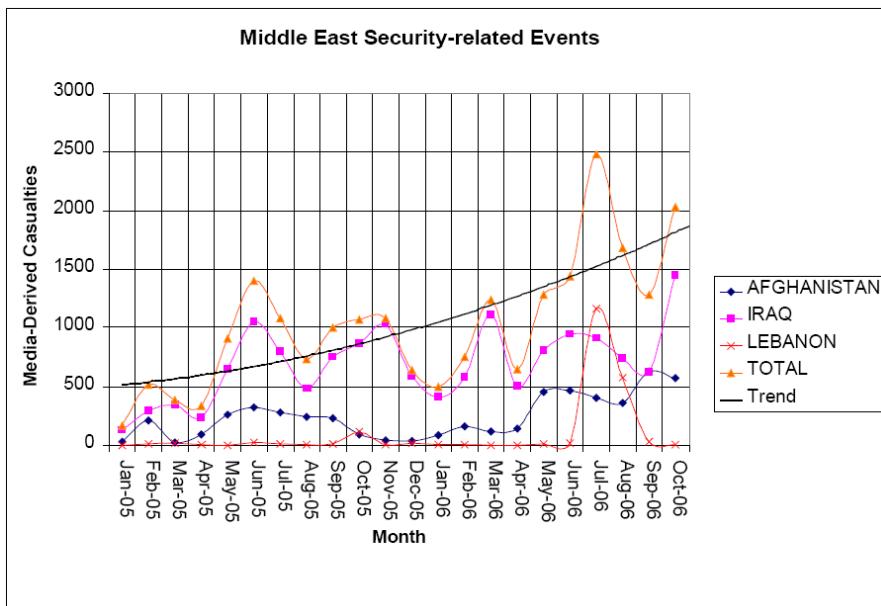


Fig. 2.8. Monthly Casualty Figures for 3 countries in the Middle East derived from automatic event extraction

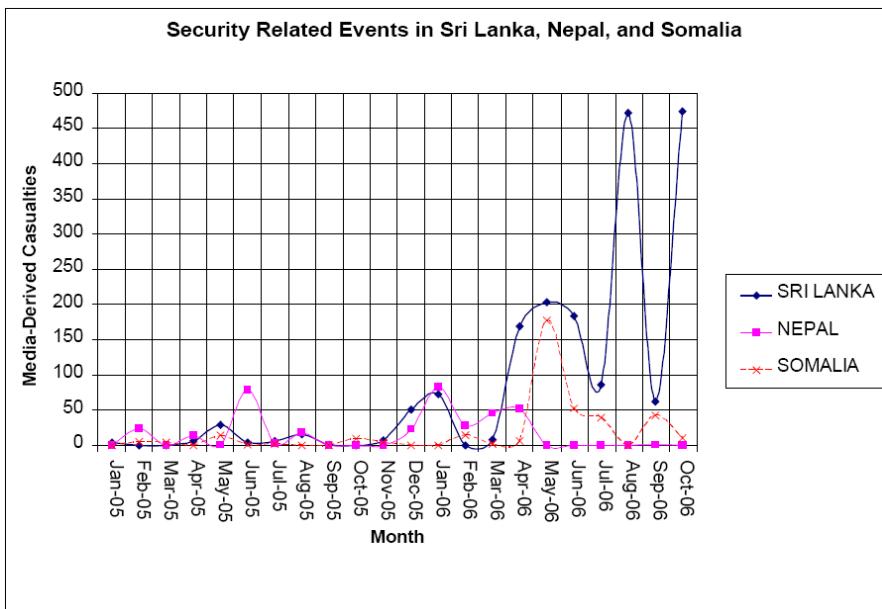


Fig. 2.9. Monthly Casualty Figures for 3 countries with internal conflict derived from automated event extraction

attacks followed by a peaceful period since May 2006. The up-shoot in violence in Somalia corresponds to the takeover of Mogadishu by Islamist militants. The breaking of the cease fire by the Tamil Tigers and the Sri Lankan government has seen a flareup of attacks with heavy casualties in Sri Lanka in 2006, after a quiet 2005. Again there is some evidence of over counting. Future work will aim to improve coverage and to detect multiple event recording across different days.

2.5 Conclusions and Future Work

This chapter has presented on-going research and initial results in automatic event extraction for the EMM project. Automatic event tracking and event extraction is important for security applications and terrorism informatics. The main advantage is processing large quantities of news report in order to track entities, to extract “facts” about events, and eventually derive causal information. The first results have shown that fact extraction - namely casualties, actors and locations has reached an acceptable degree of accuracy, and that entity tracking is operational. Clearly automated procedures will have some systematic effects due to the feedback effect of media coverage, but this is countered by the advantage of objectivity and consistency. Systematic effects can be avoided by human moderation, for detailed intelligence applications. The data is already suitable to feed incident databases both with or without human moderation.

In order to improve further on the semantics derived from event reports several enhancements to the presented approach are envisaged. Firstly, manually created

extraction patterns will be produced since pure machine-learning methods pose problems when dealing with data sparseness and are not good at tackling more complex linguistic phenomena e.g., inter-sentence dependencies. Secondly, a long-term goal is to automatically discover hierarchical structure of events, i.e., discovering sub-events of the main event and incidents which are part of it. Clearly such structure would provide a more meaningful information for discovering complex patterns by an inference apparatus on higher strata. Finally, some work on improving the coverage of pre-processing modules is planned.

One future application that uses the automatically extracted information about violent events can be a so-called “Knowledge Base for Violent Events”. Such a knowledge base includes a “Violent Events Ontology” to describe the formal model (the concept) of violent events, whereas the automatically extracted information provides the real world data. As the knowledge base makes use of the ontology, therefore also logic and metadata, it can be a useful technology for (i) outlining connections between seemingly unrelated violent events, (ii) gaining additional information about violent events via inference rules (iii) facilitating intelligent searches such as identifying aliases for names or different transliterations of a given name.

Acknowledgements

We would like to acknowledge the EMM team without whom the presented work could not be possible. In particular, we thank Erik van der Goot, Teofilo Garcia, David Horby, Pinar Oezden and Andrea Heyl. Finally we thank the JRC and in particular Delilah Al Khudhairy and Freddy Dezeure for their support.

References

1. ACE, <http://projects.ldc.upenn.edu/ace>
2. Appelt, D.: Introduction to Information Extraction Technology. In: IJCAI 1999, Tutorial, Stockholm, Sweden (1999)
3. Best, C., van der Goot, E., Blackler, K., Garcia, T., Horby, D.: Europe Media Monitor - System Description. Technical Report EUR 22173 EN, European Commission (2005)
4. Bond, D.: Integrated Data for Event Analysis (IDEA) (1998-2002), <http://vranet.com/idea>
5. Cunningham, H., Maynard, D., Tablan, V.: JAPE: a Java Annotation Patterns Engine (2rd edn). Technical Report, CS-00-10, University of Sheffield, Department of Computer Science (2000)
6. Discoverer Extractor, <http://www.temis-group.com>
7. Drożdżyński, W., Krieger, H.-U., Piskorski, J., Schäfer, U., Xu, F.: Shallow Processing with Unification and Typed Feature Structures — Foundations and Applications. Künstliche Intelligenz 2004(1), 17–23 (2004)
8. Erjavec, T.: MULTEXT - East Morphosyntactic Specifications (2004), Web document, <http://nl.ijz.si/ME/V3/msd/html>
9. Global Public Health Information Network
10. Goldstein, J.: A Conflict-Cooperation scale for WEIS Events data. Journal of Conflict Resolution 36(2), 369–385 (1992)
11. <http://www.phacaspc.gc.ca/media/nr-rp/2004/2004gphin-rmispbke.html>

12. Infoxtract, <http://www.cymfony.com>
13. Institute for Counter Terrorism, <http://www.itc.org.il>
14. Inxight ThingFinder Professional, <http://www.inxight.com>
15. Jones, R., McCallum, A., Nigam, K., Riloff, E.: Bootstrapping for Text Learning Tasks. In: Proceedings of IJCAI 1999 Workshop on Text Mining: Foundations, Techniques, and Applications, Stockholm, Sweden (1999)
16. Medical Intelligence System, <http://medisys.jrc.it>
17. MIPT Terrorism Knowledge Base (TKB), <http://www.tkb.org>
18. MUC, <http://www.itl.nist.gov/iaui/894.02/related/projects/muc>
19. Piskorski, J.: Advances in Information Extraction. In: Abramowicz, W. (ed.) Knowledge Based Information Retrieval and Filtering from Internet. Kluwer Academic Publishers, Dordrecht (2003)
20. Piskorski, J.: On Compact Storage Models for Gazetteers. In: Yli-Jyrä, A., Karttunen, L., Karhumäki, J. (eds.) FSMNLP 2005. LNCS (LNAI), vol. 4002. Springer, Heidelberg (2006)
21. Piskorski, J.: CORLEONE - Core Linguistic Entity Online Extraction. Technical Report, European Commission (to appear, 2007)
22. Pouliquen, B., Kimler, M., Steinberger, R., Ignat, C., Oellinger, T., Blackler, K., Fuart, F., Zaghouani, W., Widiger, A., Forslund, A.C., Best, C.: Geocoding multilingual texts: Recognition, Disambiguation and Visualisation. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy, pp. 24–26 (2006)
23. Pouliquen, B., Steinberger, R., Ignat, C., Temnikova, I., Widiger, A., Zaghouani, W., Zizka, J.: Multilingual person name recognition and transliteration. Journal CORELA - Cognition, Representation, Langage. Special issue: Le traitement lexicographique des noms propres (2005)
24. Schrodt, P.: Kansas Event Data Project (KEDS). Dept. of Political Science, University of Kansas, <http://www.ku.edu/~keds/project.html>
25. Semantex, <http://www.janyainc.com>
26. South Asian Terrorism Portal, <http://www.satp.org>
27. Steinberger, R., Pouliquen, B., Ignat, C.: Navigating multilingual news collections using automatically extracted information. Journal of Computing and Information Technology - CIT 13, 257–264 (2005)
28. Szpektor, I., Tanev, H., Dagan, I., Coppola, B.: Scaling Web-based acquisition of Entailment Relation. In: Proceedings of EMNLP 2004, Barcelona, Spain (2004)
29. Teragram, <http://www.teragram.com>
30. Virtual Research Associates, <http://www.vranet.com>
31. Weimann, G.: Terror on the Internet. USIP Press (2006) ISBN 1929223714
32. Yangarber, R., Jokipii, L., Rauramo, A., Huttunen, S.: Information Extraction from Epidemiological Reports. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005), Vancouver, Canada (2005)

Online Resources

1. Automatic Content Extraction Program (NIST) ACE
<http://projects.ldc.upenn.edu/ace/>.
2. Message Understanding Conference MUC:
<http://www.itl.nist.gov/iaui/894.02/related/projects/muc/>

3. Semantic Web resources: <http://www.w3.org/2001/sw/>
4. Commercial Event Coding provider VRA: <http://www.vranet.com>
5. Kansas Event Data Project (KEDS):
<http://www.ku.edu/keds/project.html> (AND Open source software TABARI).
6. KIM platform from OntoText: <http://www.ontotext.com/kim/>
7. CYC Ontology and Knowledge Base: <http://www.cyc.com>
8. IDEA event classification codes (IDEA = Integrated Data for Event Analysis) at:
<http://vranet.com/idea>
9. The Goldstein conflict-cooperation scale for events:
<http://jcr.sagepub.com/cgi/content/abstract/36/2/369>
<http://gking.harvard.edu/events/data/VRA1990-2004.doc>
10. MIPT Terrorism Knowledge Base: <http://www.tkb.org>

Questions for Discussions

1. Why can it be difficult to recognise references to geo-locations in text? Explore solutions to overcome these difficulties.
2. To extract information from text (such as references to persons and locations, for instance), it is possible to write recognition patterns manually or to learn patterns using Machine Learning techniques. Discuss the relative advantages of each approach.
3. What can be gained from using an ontology to model security-related events compared to a traditional RDBMS?
4. Try to plan the architecture of an automatic system to extract event information from free text.
5. What are the challenges when extracting event information from multiple information sources (e.g. various newspapers talking about the same event)? Discuss possible solutions to these challenges.
6. What are the challenges when extracting event information from texts written in different languages? How can the multilingual information be aggregated and presented to monolingual users?
7. Propose a method to evaluate an automatic event extraction system. Is there an efficient method to find out whether the system will perform well for different text types?
8. Assume that you have a working system extracting security-related event information from free text. What is the effort to adapt this system to a new domain (e.g. the recognition of disease outbreaks in the medical domain).
9. Can you gage the limitations of automatic event extraction for a given domain and a given language? Can machines ever get better than human beings doing the same job?
10. What are the advantages to using computers rather than manual effort for the extraction of events?

Knowledge Discovery and Information Visualization for Terrorist Social Networks

Christopher C. Yang

College of Information Science and Technology
Drexel University
Philadelphia, PA 19104
USA

Abstract. Terrorist social networks present information about the relationships between terrorists which is important for investigators in combating the war of terrorism. A good analysis of terrorist social networks is essential for discovering knowledge about the structure of terrorist organizations. Typical social network analysis techniques discover patterns of interactions between social actors, for example, detecting subgroup and identifying central individuals. Such knowledge is important for developing effective combating strategies against terrorism. A typical terrorist social network has several hundreds to thousands of nodes. Given the large number of nodes and links, it is difficult to integrate the extracted pattern with the global structure of the terrorist social network to further analyze the related actors. With the support of interactive visualization tools, investigators can further explore the network by selecting the focus of interests and extract the useful information and patterns. In this chapter, we present a social network analysis and interactive visualization technique for complex terrorist social networks. Case studies are presented to illustrate how such techniques are capable to extract the hidden relationships among terrorists in the network through user interactions.

3.1 Terrorist Social Networks

A social network is a social structure of people who are related to each other through a common relation or interest directly or indirectly. A terrorist social network (TSN) is a network representation of terrorists who have different types of relationships with each other. The relationships include acquaintance, friends, relatives, nuclear family member, teachers and religious leaders. In this chapter, we use the Global Salafi Jihad social network for illustration. There are totally 366 terrorists in the Global Salafi Jihad social network, who are related to other terrorists by one or more type of relations. There are totally 1275 links in the social network. The Global Salafi Jihad social network is generated by the data provided by Sageman, who has authored an authoritative terrorism monograph [5].

A terrorist social network is represented by a weighted graph $G = (V, E; w)$. V is a set of nodes. Each node represents a terrorist. E is a set of links ($V \times V$). w is a function mapping each link, $(u, v) \in E$, to a weight w_{uv} in the range $[0, 1]$. The weight indicates the strength of association between the corresponding nodes. The weight is computed by normalizing the link score, where the link score is the summation of the importance score of each type of relationship that the link possesses.

3.1.1 Social Network Analysis – Centrality Measurement

The degree centrality is defined as the degree of a node normalized with the maximum degree of a network. Given a social network with n nodes, the degree centrality formulation is

$$\text{degree centrality}(u) = \frac{\text{degree of } u}{n-1} \quad (3.3)$$

The closeness centrality is measured by the distance from a node to all other nodes. Let the shortest distance of a path from u to v be $d(u,v)$. The closeness centrality formulation is

$$\text{closeness centrality}(u) = \frac{n-1}{\sum_{v=1}^n d(u,v)} \quad (3.4)$$

The betweenness centrality measures the control of a node over other pairs of nodes in a social network. Let p_{uv} be the number of shortest paths between u and v . The betweenness of w is defined as the number of shortest paths that pass w ($p_{uv}(w)$) normalized by the number total number of shortest paths of all pairs of nodes not including w . The betweenness centrality formulation is

$$\text{betweenness centrality}(w) = \frac{2 \sum_{u < v} p_{uv}(w)}{(n-1)(n-2)} \quad (3.5)$$

The size of nodes in Fig. 3.1 is adopting the degree centrality.

3.1.2 Example

Given two networks, Network A and Network B, as shown in Fig. 3.2, we illustrate the computation and the impact strength of different centrality measurements.

Table 3.1 presents the degree and degree centrality of the nodes in Network A and Network B. The computation is straight forward. Table 3.2 presents the distance between each pair of nodes and the total distance for each node to all other nodes. Table 3.3 presents the result of the closeness centrality. The computation in closeness centrality is more expensive.

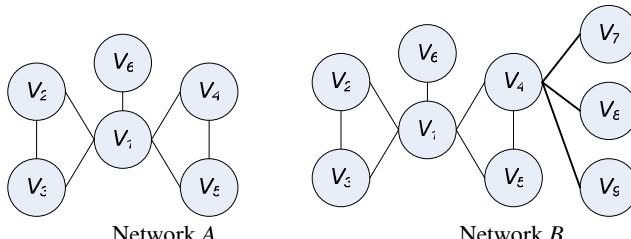


Fig. 3.2. A sample network with six nodes

Table 3.1. Degree and degree centrality of the nodes in Network A and Network B

Network A:

	V_1	V_2	V_3	V_4	V_5	V_6
Degree	5	2	2	2	2	1
Degree Centrality	1.0	0.4	0.4	0.4	0.4	0.2

Network B:

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9
Degree	5	2	2	5	2	1	1	1	1
Degree Centrality	0.625	0.250	0.250	0.625	0.250	0.125	0.125	0.125	0.125

Table 3.2. Distance between each pair of nodes in Network A and Network B

Network A:

Distance	V_1	V_2	V_3	V_4	V_5	V_6	$\sum d(u,v)$
V_1	0	1	1	1	1	1	5
V_2	1	0	1	2	2	2	8
V_3	1	1	0	2	2	2	8
V_4	1	2	2	0	1	2	8
V_5	1	2	2	1	0	2	8
V_6	1	2	2	2	2	0	9

Network B:

Distance	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	$\sum d(u,v)$
V_1	0	1	1	1	1	1	2	2	2	11
V_2	1	0	1	2	2	2	3	3	3	17
V_3	1	1	0	2	2	2	3	3	3	17
V_4	1	2	2	0	1	2	1	1	1	11
V_5	1	2	2	1	0	2	2	2	2	14
V_6	1	2	2	2	2	0	3	3	3	18
V_7	2	3	3	1	2	3	0	2	2	18
V_8	2	3	3	1	2	3	2	0	2	18
V_9	2	3	3	1	2	3	2	2	0	18

Table 3.3. Closeness centrality of the nodes in Network A and Network B

Network A:

	V_1	V_2	V_3	V_4	V_5	V_6
Closeness Centrality	1.0	0.625	0.625	0.625	0.625	0.556

Network B:

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9
Closeness Centrality	0.727	0.471	0.471	0.727	0.571	0.444	0.444	0.444	0.444

Table 3.4 presents the shortest paths, the number of shortest path between each pair of nodes and the number of times that the shortest paths pass through individual node. Table 3.5 presents the result of the betweenness centrality. The computational cost of betweenness centrality is approximately the same as the computational cost of closeness centrality because the shortest path of each pair of nodes has to be first computed.

Table 3.6 presents the comparison of the degree centrality, closeness centrality and betweenness centrality for Network A and Network B.

Table 3.4. The shortest path between each pair of nodes

Network A:

	Shortest path	Number of Shortest path, $P_{u,v}$	$P_{u,v}(v_1)$	$P_{u,v}(v_2)$	$P_{u,v}(v_3)$	$P_{u,v}(v_4)$	$P_{u,v}(v_5)$	$P_{u,v}(v_6)$
V_1, V_2	(V_1, V_2)	1		0	0	0	0	0
V_1, V_3	(V_1, V_3)	1		0	0	0	0	0
V_1, V_4	(V_1, V_4)	1		0	0	0	0	0
V_1, V_5	(V_1, V_5)	1		0	0	0	0	0
V_1, V_6	(V_1, V_6)	1		0	0	0	0	0
V_2, V_3	(V_2, V_3)	1	0		0	0	0	0
V_2, V_4	(V_2, V_1, V_4)	1	1		0	0	0	0
V_2, V_5	(V_2, V_1, V_5)	1	1		0	0	0	0
V_2, V_6	(V_2, V_1, V_6)	1	1		0	0	0	0
V_3, V_4	(V_3, V_1, V_4)	1	1	0		0	0	0
V_3, V_5	(V_3, V_1, V_5)	1	1	0		0	0	0
V_3, V_6	(V_3, V_1, V_6)	1	1	0		0	0	0
V_4, V_5	(V_4, V_5)	1	0	0	0		0	0
V_4, V_6	(V_4, V_1, V_6)	1	1	0	0		0	0
V_5, V_6	(V_5, V_1, V_6)	1	1	0	0	0		0
$\sum p_{uv}(w)$		8	0	0	0	0	0	0

Network B:

	Shortest path	Number of Shortest path, $P_{u,v}$	$P_{u,v}(v_1)P_{u,v}(v_2)P_{u,v}(v_3)P_{u,v}(v_4)$	$P_{u,v}(v_5)P_{u,v}(v_6)P_{u,v}(v_7)P_{u,v}(v_8)P_{u,v}(v_9)$	
V_1, V_2	(V_1, V_2)	1	0	0	0
V_1, V_3	(V_1, V_3)	1	0	0	0
V_1, V_4	(V_1, V_4)	1	0	0	0
V_1, V_5	(V_1, V_5)	1	0	0	0
V_1, V_6	(V_1, V_6)	1	0	0	0
V_1, V_7	(V_1, V_4, V_7)	1	0	1	0
V_1, V_8	(V_1, V_4, V_8)	1	0	1	0
V_1, V_9	(V_1, V_4, V_9)	1	0	1	0
V_2, V_3	(V_2, V_3)	1	0	0	0
V_2, V_4	(V_2, V_1, V_4)	1	1	0	0
V_2, V_5	(V_2, V_1, V_5)	1	1	0	0
V_2, V_6	(V_2, V_1, V_6)	1	1	0	0
V_2, V_7	(V_2, V_1, V_4, V_7)	1	0	1	0
V_2, V_8	(V_2, V_1, V_4, V_8)	1	0	1	0
V_2, V_9	(V_2, V_1, V_4, V_9)	1	0	1	0
V_3, V_4	(V_3, V_1, V_4)	1	1	0	0

Table 3.4. (continued)

V ₃ ,V ₅ (V ₃ ,V ₁ ,V ₅)	1	1	0	0	0	0	0	0	0
V ₃ ,V ₆ (V ₃ ,V ₁ ,V ₆)	1	1	0	0	0	0	0	0	0
V ₃ ,V ₇ (V ₃ ,V ₁ ,V ₄ ,V ₇)	1	0	1	0	0	0	0	0	0
V ₃ ,V ₈ (V ₃ ,V ₁ ,V ₄ ,V ₈)	1	0	1	0	0	0	0	0	0
V ₃ ,V ₉ (V ₃ ,V ₁ ,V ₄ ,V ₉)	1	0	1	0	0	0	0	0	0
V ₄ ,V ₅ (V ₄ ,V ₅)	1	0	0	0	0	0	0	0	0
V ₄ ,V ₆ (V ₄ ,V ₁ ,V ₆)	1	1	0	0	0	0	0	0	0
V ₄ ,V ₇ (V ₄ ,V ₇)	1	0	0	0	0	0	0	0	0
V ₄ ,V ₈ (V ₄ ,V ₈)	1	0	0	0	0	0	0	0	0
V ₄ ,V ₉ (V ₄ ,V ₉)	1	0	0	0	0	0	0	0	0
V ₅ ,V ₆ (V ₅ ,V ₁ ,V ₆)	1	1	0	0	0	0	0	0	0
V ₅ ,V ₇ (V ₅ ,V ₄ ,V ₇)	1	0	0	0	1	0	0	0	0
V ₅ ,V ₈ (V ₅ ,V ₄ ,V ₈)	1	0	0	0	1	0	0	0	0
V ₅ ,V ₉ (V ₅ ,V ₄ ,V ₉)	1	0	0	0	1	0	0	0	0
V ₆ ,V ₇ (V ₆ ,V ₁ ,V ₄ ,V ₇)	1	0	0	1	0	0	0	0	0
V ₆ ,V ₈ (V ₆ ,V ₁ ,V ₄ ,V ₈)	1	0	0	1	0	0	0	0	0
V ₆ ,V ₉ (V ₆ ,V ₁ ,V ₄ ,V ₉)	1	0	0	0	1	0	0	0	0
V ₇ ,V ₈ (V ₇ ,V ₄ ,V ₈)	1	0	0	0	1	0	0	0	0
V ₇ ,V ₉ (V ₇ ,V ₄ ,V ₉)	1	0	0	0	1	0	0	0	0
V ₈ ,V ₉ (V ₈ ,V ₄ ,V ₉)	1	0	0	0	1	0	0	0	0
$\sum p_m(w)$	17	0	0	18	0	0	0	0	0

Table 3.5. Betweenness centrality of Network A and Network B

Network A:

	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆
Betweenness Centrality	0.8	0	0	0	0	0

Network B:

	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇	V ₈	V ₉
Betweenness Centrality	0.607	0	0	0.643	0	0	0	0	0

Table 3.6. Ranking of nodes by different centrality measurements

Network A			Network B				
Degree Centrality	Closeness Centrality	Betweenness Centrality	Degree Centrality	Closeness Centrality	Betweenness Centrality		
Rank	Node	Rank	Node	Rank	Node		
1	V ₁ (1.0)	1	V ₁ (1.0)	1	V ₁ (0.727)	1	V ₄ (0.643)
2	V ₂ (0.4)	2	V ₂ (0.625)	2	V ₂ (0.0)	2	V ₄ (0.727)
	V ₃ (0.4)		V ₃ (0.625)		V ₃ (0.0)	3	V ₁ (0.607)
	V ₄ (0.4)		V ₄ (0.625)		V ₄ (0.0)	2	V ₂ (0.0)
	V ₅ (0.4)		V ₅ (0.625)		V ₅ (0.0)	3	V ₂ (0.471)
3	V ₆ (0.2)	3	V ₆ (0.556)	3	V ₆ (0.0)	4	V ₃ (0.471)
					V ₆ (0.125)	4	V ₅ (0.0)
					V ₇ (0.125)	5	V ₆ (0.0)
					V ₈ (0.125)	6	V ₇ (0.0)
					V ₉ (0.125)	7	V ₈ (0.0)
					V ₉ (0.125)	8	V ₉ (0.0)

Network A is simple and symmetric. The rankings computed by the degree centrality and closeness centrality on Network A are exactly the same although the values are not the same. The rankings computed by the betweenness centrality is very close to those computed by the degree centrality and closeness centrality except that V_6 is ranked second rather than third. The degree centrality's computation is straight forward, which only consider the degree of a node but do not consider the distance to other nodes. In a simple network as Network A, the result is very similar to other centrality measurements because the distances from one node to other nodes are not large. However, the betweenness centrality considers whether a shortest path passes through individual node. In Network A, almost half of the shortest paths have a distance of 1; that means, the shortest path is a path directly goes from the starting node to the target node. As a result, many shortest paths do not pass through any other nodes. V_1 is the only node in Network A that has shortest paths with length longer than 1 pass through it while all other nodes have no shortest paths passing through them.

Network B is a bit more complex than Network A and it is not symmetric. The rankings computed by the three centrality measurements are different. The rankings computed by degree centrality and closeness centrality are the same except V_5 is ranked higher than V_2 and V_3 by closeness centrality instead of the same ranking as V_2 and V_3 by degree centrality. In terms of degree, V_2 , V_3 , and V_5 are the same. However, V_5 is closer to other nodes in terms of distance. Therefore, V_5 obtains a higher closeness centrality. Simply using the degree of nodes, such difference will not be obtained. Similar to Network A, most of the nodes in Network B have a value of zero on betweenness centrality because most of the shortest paths do not pass through these nodes. However, the betweenness centrality ranks V_4 higher than V_1 but other centrality measurements do not have any difference on the rankings of V_1 and V_4 . That means, V_1 and V_4 have the same degree and the same closeness to other nodes but there are more shortest paths passing through V_4 .

As the size of the network increases, the difference between the rankings obtained by the degree centrality, closeness centrality and betweenness centrality increases. Depending on the analysis of the terrorist social networks and the computational efficiency, different centrality measurements may have advantages. The degree centrality is computation inexpensive and obtains a fair approximation on the importance of a node. The closeness centrality and betweenness centrality are more computational expensive but they have better differentiation on the importance of nodes. The closeness centrality is measuring how close a node is to all other nodes. The betweenness centrality is measuring how important a node is in terms of the relationship between other nodes in the network. In terms of analyzing terrorist social networks, the closeness centrality is helpful in identifying the leader in a subgroup. The betweenness centrality is useful in determining the gateway between two subgroups. After extracting the leaders and gateways from a terrorist social network, we have a better understanding of the related terrorists in a particular investigation.

3.2 Information Visualization

A main objective of visualization is mapping information onto graphical representation to gain insight for users. Information visualization assists users to view and locate

the information of interest in a limited space by interacting with the visualization systems.

In general, there are two stages of information visualization. The first stage is information interpretation and mapping. The second stage is information display and control. In the stage of information interpretation and mapping, data are transformed from multi-dimensional quantitative and qualitative features into a two or three dimensional graphical representation for subsequent operations. The transformed representations are then rendered and displayed [8].

Given a terrorist social network, we have a set of nodes, V , representing the terrorists, a set of links, E , representing the relationships possessed by the corresponding terrorists, and a set of weights, w , one for each link in V , computed by the types and number of relationships possessed by the corresponding terrorists. In order to obtain an effective visualization, there are a number of objectives when the nodes and links of a terrorist social network are mapped to a two-dimensional space. (1) Nodes should have an optimal distance from each other so that the two-dimensional space can be fully utilized. (2) Nodes and edges should not be cluttered. (3) Distance between two nodes should correspond to the strength of their associations. The spring embedder algorithm, which model nodes as charged particles with mutual repulsion and links as springs attached to their end nodes can achieve the above objectives.

In the spring embedder algorithm, there are two opposite forces, (1) repulsive force and (2) spring force. The repulsive force attempts to avoid nodes being cluttered together. It tries to keep nodes with the longest distance within the limited two-dimensional display window. The spring force attempts to maintain a desirable distance between nodes. It tries to keep two nodes as close as possible if the weight of their corresponding link is high. After applying the spring embedder algorithm, the two-dimensional coordinates of all nodes in V are computed and the terrorist social network is mapped onto a two-dimensional display window as illustrated in Fig.3.1. The first stage of information visualization, information interpretation and mapping, is completed.

The second stage, information display and control, is very important in visualizing a complex terrorist social network. Given the presentation of a terrorist social network with 366 nodes and 1275 links as shown in Fig. 3.1, users are not able to visualize the details and conduct any analysis on the network since there is too much information to be displayed on a limited two-dimensional space.

In the following two sections, we introduce two interactive visualization techniques, fisheye views and fractal views. Fisheye views are using distortion approach while fractal views are using information reduction approach. Fisheye views maintain the global structure of a social network but explode the area of interest. Fractal views filter the information that is less relevant to the focus of interest.

3.2.1 Fisheye Views

Fisheye views perform as a lens placing on top of an area of interest. Fisheye views [1, 2, 5, 8] enlarge the area of interest while diminish other regions. As a result, the local details in the area of interest and the global structure can be presented within one display window. By moving the focus point, users may explore different areas of the social network. For example, users may start from the node with the highest closeness

centrality, which represent the leader of the subgroup, as the focus point and then explore his neighborhoods to identify any suspicious terrorists in a particular event. However, the distortion requires mental integration when the focus is shifting. If the focus is shifted to an adjacent node, the mental integration is minimal. However, if the focus is shifted to another node that is further away from the current node, the mental integration is large and may cause some confusion to users.

Before the fisheye views transformation, each node has a pair of normal coordinates (x_{norm}, y_{norm}) . Given a focus point, (x_{focus}, y_{focus}) , selected by an user, the fisheye coordinates of a node, (x_{feye}, y_{feye}) , is computed by the following formulation using the polar coordinate system.

θ is the angle of a node using the polar coordinate system. That means, $x_{norm} = r_{norm} \cos \theta$ and $y_{norm} = r_{norm} \sin \theta$. r_{max} is the maximum possible radius limited by the window size for a node if the node is moving along the line formed by the focus point and the node. d is a distortion factor of the fisheye views that controls the degree of explosion.

$$\langle x_{feye}, y_{feye} \rangle = \langle x_{focus} + r_{feye} \cos \theta, y_{focus} + r_{feye} \sin \theta \rangle \quad (3.6)$$

$$\text{where } r_{feye} = r_{norm} \cdot \frac{d+1}{d \cdot \frac{r_{norm}}{r_{max}} + 1}$$

$$r_{norm} = \sqrt{(x_{norm} - x_{focus})^2 + (y_{norm} - y_{focus})^2}$$

$$\theta = \tan^{-1} \left(\frac{y_{norm} - y_{focus}}{x_{norm} - x_{focus}} \right)$$

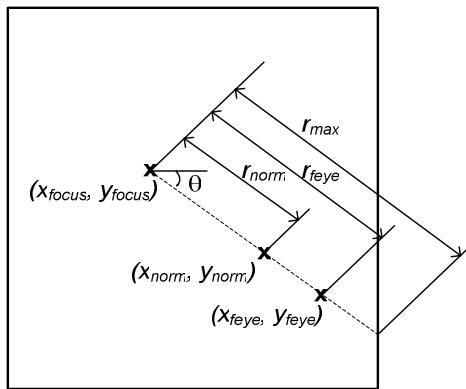


Fig. 3.3. Illustration of fisheye views computation

If $d = 0$, $r_{feye} = r_{norm}$. That means the coordinates of nodes are not changed. If $d > 0$, the node is moving away from the focus. On the contrary, if $d < 0$, the node is moving towards the focus. Since the purpose of the fisheye views is exploding the area of interest but not shrinking the area of interest, d is always greater zero.

Given two nodes, (x_1, y_1) and (x_2, y_2) , and r_1 is shorter than r_2 , the distance between (x_1, y_1) and the transformed coordinates of (x_1, y_1) is always larger than the distance between (x_2, y_2) and the transformed coordinates of (x_2, y_2) . In the fisheye effect, the closer the node to the focus point is, the larger the degree of explosion is. The further a node to the focus point is, the smaller the degree of explosion is. As a result, the area that is closer to the focus has a larger explosion effect. Fig. 3.4. illustrates the visual effect of fisheye views on the Global Salafi Jahid terrorist social network. The node in red is the focus point. The region that is close to the focus is enlarged. However, the regions that are further away from the focus is diminished.

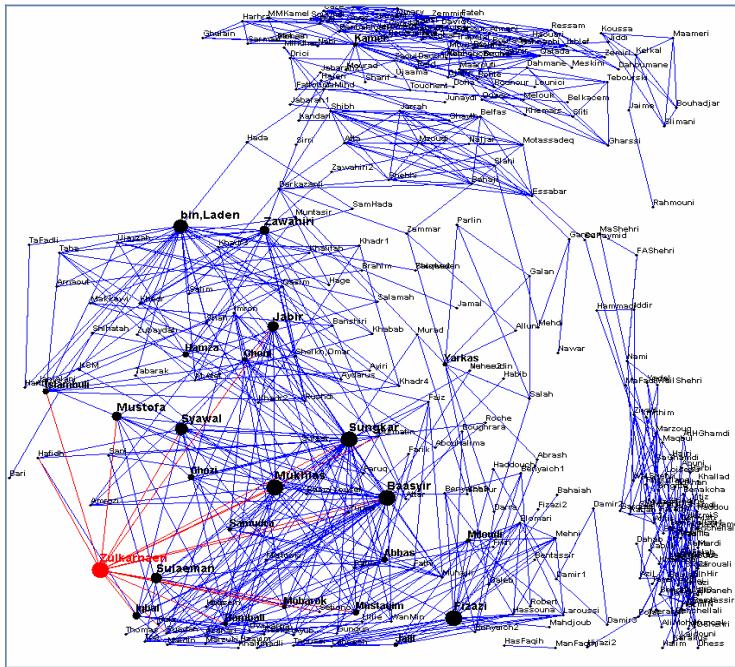


Fig. 3.4. The Fisheye views on the Global Salafi Jihad terrorist social network

As illustrated in Fig. 3.4, the global structure of the terrorist social network is maintained. All nodes and links are maintained in the display except that the coordinates of the nodes are changed. With the distortion, the details near the focus point are visible. It was not visible in the original display of the network as shown in Fig. 3.1. However, the details of the other two subgroups, which are located at the top and the right hand side of the display, are now invisible because these regions are diminished.

The advantage of maintaining the global structure is easy to determine the relative location of the focus point and the exploded region in the whole network. The disadvantage of fisheye views is the difficulty in mental integration when the focus point is changed from one node to another node that is far from the current one. It is because the new exploded area is likely to be a diminished area before shifting of the focus point. Our previous study shows that such difficulty in mental integration will

decrease the efficiency when users are applying the fisheye views in exploring the terrorist social networks.

3.2.2 Fractal Views

Fractal views [4, 9] are adopting the information reduction approach in information visualization. The global structure of the social network is not maintained it is in fisheye views. Fractal views is developed by utilizing the concept of fractals [2] to abstract complex objects and control the amount of displayed information specified by users using a threshold.

Fractal views were developed to apply on tree structures to control the number of displayed nodes without relation to the shape of trees [5]. The root node of a tree is assigned a fractal value of 1. The fractal values of other nodes are propagated from the root node using a fractal value propagation formulation. The nodes with fractal values lower than a threshold are trimmed off to reduce the less relevant information from the visually overloaded structure.

The fractal values propagation formulations for a tree structure is as follow:

$$FV_{child\ of\ x} = r_x \times FV_x \quad (3.7)$$

where $FV_{child\ of\ x}$ and FV_x are the fractal values of the child of node x

and node x ,

$$r_x = C N_x^{-1/D}$$

C is a constant, D is the fractal dimension, and N_x is the number of children of node x

Given a simple tree structure in Fig. 3.5, the computation of fractal values is given in Table 3.7. If the threshold is 0.15, V_7, V_8, V_9 are trimmed. If the threshold is 0.20, V_5, V_6, V_7, V_8, V_9 are trimmed.

Table 3.7. Computation of fractal values for the tree in Fig. 3.5.

V_{root}	1
V_1, V_2, V_3	$1 \times 3^{-1} = 0.33$
V_4	$0.33 \times 1^{-1} = 0.33$
V_5, V_6	$0.33 \times 1^{-2} = 0.17$
V_7, V_8, V_9	$0.33 \times 1^{-3} = 0.11$

A terrorist social network is a network or cyclic graph rather than a hierarchical tree. Therefore, the fractal view algorithm cannot be simply applied on abstracting a terrorist social network. Besides, the fractal views propagation formulation does not consider any association strength between nodes. It only considers the number of children a node has. In order to develop the fractal views for terrorist social network, we need to (1) transform a network topology into a hierarchical tree structure and (2)

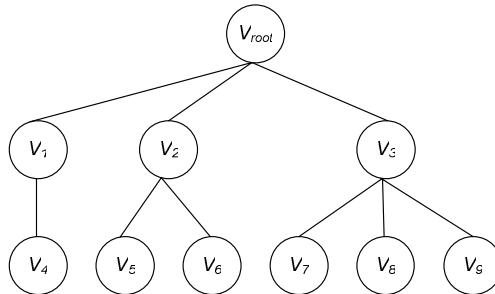


Fig. 3.5. A simple tree for illustration of fractal views

modify the fractal values propagation formula to take different association strengths between a node and its adjacent nodes into consideration so that the propagated fractal values reflect the association between the parent node and the child node.

To transform the social network structure to a hierarchical tree structure, we take the focus node as the root node to construct a tree such that the paths from the root node to any nodes in the transformed tree represent the path with the strongest association strength (i.e. a path with the shortest distance). The fractal views propagation formulation is modified as:

$$F_c = \left(\frac{w_{cp}}{\sum_{c' \in \text{children_of}(p)} w_{c'p}} \right)^{1/D} F_p \quad (3.8)$$

where c is a child of p

w_{cp} is a weight between c and p corresponding to the inverse of the association strength as computed in Eq. 3.1

D is the fractal dimension

The fractal values are normalized so that the sum of the fractal values of all children equals to the fractal values of the parent.

Example

Fig. 3.6 is an example of a social network with 9 nodes, V_1 to V_9 . The weights labeled on the links correspond to the inverse of the association strengths of their links. The lower the value of the weight is, the stronger the associate strength between the corresponding nodes is. To apply the fractal views on this network, we need to select a focus point (i.e. a node in the network) first. Given the selected focus node, we transform the network structure into a hierarchical tree structure in which the path from the root node of the hierarchical tree to any particular node represents the shortest path between the root node and the node in the network. The shortest path between any two nodes is the path that the total weight of the links that are included in the path is the smallest among all the possible paths between the corresponding nodes. For example, between V_1 and V_5 , there are two paths, (V_1, V_5) and (V_1, V_4, V_5) . The total weight on the path (V_1, V_5) is 5 and the total weight on the path (V_1, V_4, V_5) is 2.

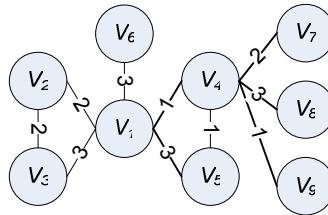


Fig. 3.6. A social network with 9 nodes and weights corresponding to the inverse of the association strengths of their links

Therefore, the shortest path between V_1 and V_5 is (V_1, V_4, V_5) although this path goes through more number of nodes.

Figs. 3.7(a) and (b) are the transformed hierarchical tree with V_1 and V_4 as the root nodes, respectively. Given different focus nodes, the structures of the transformed hierarchical trees are totally different. The width and depth of the trees and the leave nodes of the trees vary as the root nodes are changed. It exhibits why and how different nodes are filtered by fractal views as we change the focus of the fractal views. The filtering of nodes by fractal views is not solely depend on the transformed tree structure but also on the propagation of the fractal values from the root nodes to other nodes in the tree. The propagation depends on the weights between the parent node and the child nodes, the number of child nodes that the parent node has, and the fractal value of the parent node. As the fractal value is propagated from the root node to the nodes in the lower level of the hierarchical tree, the fractal values decrease.

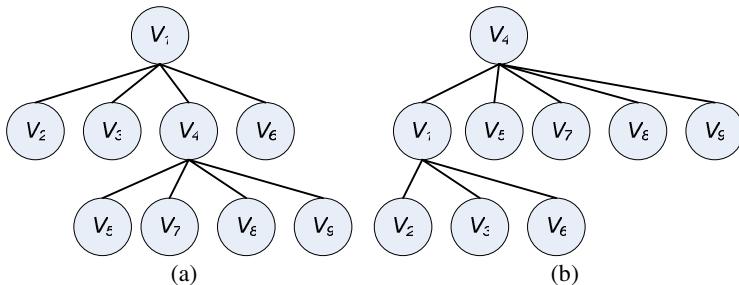
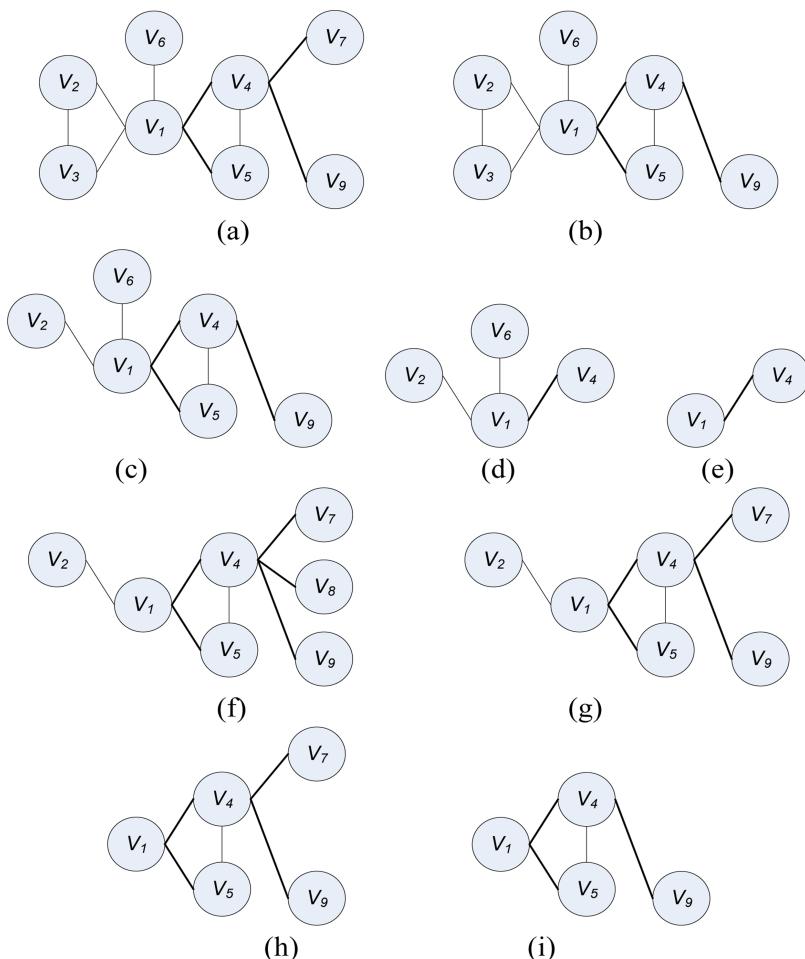


Fig. 3.7. Transformation of the network in Fig. 3.6 to a hierarchical with (a) V_1 as a focus node (b) V_4 as a focus node

Table 3.8 presents the propagated fractal values of the nodes in Figs. 3.7(a) and (b). The fractal values of the root nodes (V_1 in Fig. 3.7(a) and V_4 in Fig. 3.7(b)) are always 1. When the fractal value of V_1 in Fig. 3.7(a) propagates to the nodes on the second level, V_2, V_3, V_4, V_6 , the fractal value of V_1 is distributed to these nodes. That means the total of the fractal values of V_2, V_3, V_4, V_6 equal to the fractal value of V_1 (i.e. $0.231 + 0.154 + 0.462 + 0.231 = 1.000$). V_4 is propagated with a higher fractal value than other three nodes because the weight between V_1 and V_4 is the lowest. After

Table 3.8. Fractal values of nodes in the network shown on Fig. 3.6

	V ₁ as focus node	V ₄ as focus node
V ₁	1.000	0.261
V ₂	0.231	0.112
V ₃	0.154	0.075
V ₄	0.462	1.000
V ₅	0.162	0.261
V ₆	0.231	0.075
V ₇	0.083	0.130
V ₈	0.055	0.087
V ₉	0.162	0.261

**Fig. 3.8.** Fractal views of the network as shown in Fig. 3.6. with threshold gradually increasing
(a) – (e) V₁ as a focus node, (f) – (i) V₄ as a focus node

computing the fractal values of the nodes on the second level, the fractal values of V_4 is then propagated to V_5 , V_7 , V_8 , V_9 . Similar propagation for the hierarchical tree in Fig. 3.7(b) is also computed from V_4 to V_1 , V_5 , V_7 , V_8 , V_9 and then from V_1 to V_2 , V_3 , V_6 .

Fig. 3.8(a) to (e) is the fractal views of the social network in Fig. 3.6 with V_1 as the focus node. As we decrease the threshold of the fractal views, the nodes in the fractal views are filtered. In Fig. 3.8(a), V_8 is first filtered. V_7 is then filtered (Fig. 3.8(b)) and then V_3 (Fig. 3.8(c)), V_5 and V_9 (Fig. 3.8(d)). At the end, V_2 and V_6 are filtered. V_1 and V_4 are left as shown in Fig. 3.8(e).

Similar effect can be observed from Fig. 3.8 (f) to 3.(i) where V_4 is selected as focus node. V_3 and V_6 are first filtered (Fig. 3.8(f)). V_8 is then filtered (Fig. 3.8(g)) and V_2 is filtered next (Fig. 3.8(h)). At the end, V_7 is filtered and V_1 , V_4 , V_5 , V_9 are remained because V_1 , V_5 , V_9 have the same fractal values.

3.3 Case Studies

In this section, we present a few case studies of the information visualization on the Global Salafi Jihad terrorist social netowk.

3.3.1 Case 1 – Extracting the Hamburg Cell of the 911 Attack

Extracting a subgroup who involves in a terror attack from a terrorist social network is an important task of the investigators who combat the terrorism. The 911 attack in 2001 was a shocking event to the whole world and caused tremendous damages and heavy casualties. Many are still suffering in the lost of their love ones.

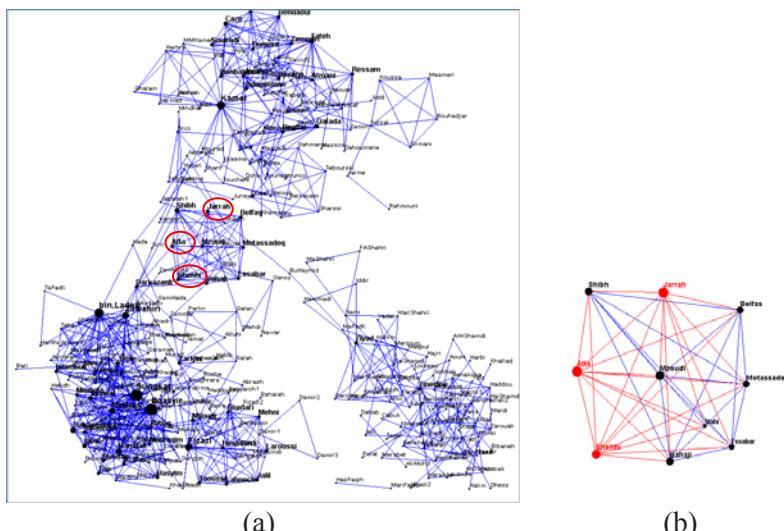


Fig. 3.9. (a) The Global Salafi Jihad terrorist social network with Atta, Jarrah, and Shehhi identified, (b) Extracting Hamburg Cell with Atta, Jarrah, and Shehhi as focuses

The 911 attack is known to be carried out by the Hamburg Cell after years of investigation. The Hamburg Cell is composed of nine people in an upper-middle class expatriate student community. Most of them have studied in the Technical University of Hamburg-Harburg (TUHH) or the University of Applied Sciences (UAS). In includes Belfas, Atta, Mzoudi, Mzoudi, Motassadeq, Shihb, Bahaji, Essabar, and Shehhi. These group of people meet each other through a study group at al-Quds Mosque in Hamburg, which was conducted by Belfas.

It was found that Atta, Jarrah and Shehhi received aircraft training in US and participated in the 911 operation. Given these three suspects, we can identify the other members of the Hamburg Cell in the Global Salafi Jihad terrorist social network. Using these three nodes as focuses (Fig. 3.9(a)) in fisheye views and fractal views and adjusting the threshold value in fractal views, we extract the Hamburg Cell as shown in Fig. 3.9(b).

Slahi is also extracted in the fractal views in addition to the nine members of the Hamburg Cell. Slahi has not participated in the 911 attack but he is extracted together with the Hamburg Cell because he has direct relationships with all three suspects, Atta, Jarrah, and Shehhi. Only one incorrect member is extracted but all six remaining members of the Hamburg Cell other than the three focuses (Shibh, Belfas, Mzoudi, Motassadeq, Bahaji, and Essabar) are extracted. Using the fractal views to extract the Hamburg Cell achieves 100% recall and 85.7% precision. It shows how such tools greatly reduce the manual efforts of the investigators in identifying the members of the Hamburg Cell.

3.3.2 Case 2 – Determine the Gatekeepers between Two Terrorist Leaders

Determining the gatekeepers between two subgroups in a terrorist social network is a typical task in understanding the relationships between different groups of terrorists. In this case study, we present how we determine the gatekeepers by selecting the leaders of subgroups as the focuses in the fisheye views and fractal views.

Bin Laden is the leader of the al Qaeda terrorist group. Fateh was the leading person in the US Millennial Plot. However, Bin Laden and Fateh do not have direct relationship according to the terrorist social network in Fig. 3.10(a). Selecting Bin Laden and Fateh as the focuses of the fisheye views and fractal views and adjusting the threshold to filter the less important nodes and less relevant paths, we extract a shortest path between bin Laden and Fateh and other relevant terrorists along the path as shown on Fig. 3.10(b). The shortest path includes six terrorists {bin Laden, Hada, ZaMihd, Kamel, Caze, Fateh}. The fisheye views and fractal views also extract two other paths (including seven terrorists) which are only a bit longer than the shortest path: {bin Laden, Zawahiri, Birri, ZaMihd, Kamel, Caze, Fateh} and {bin Laden, Hada, ZaMihd, Kamel, Caze, Omary, Fateh}. That means Zawahiri, Birri, and Omary are probably important persons between the relationship of Bin Laden and Fateh.

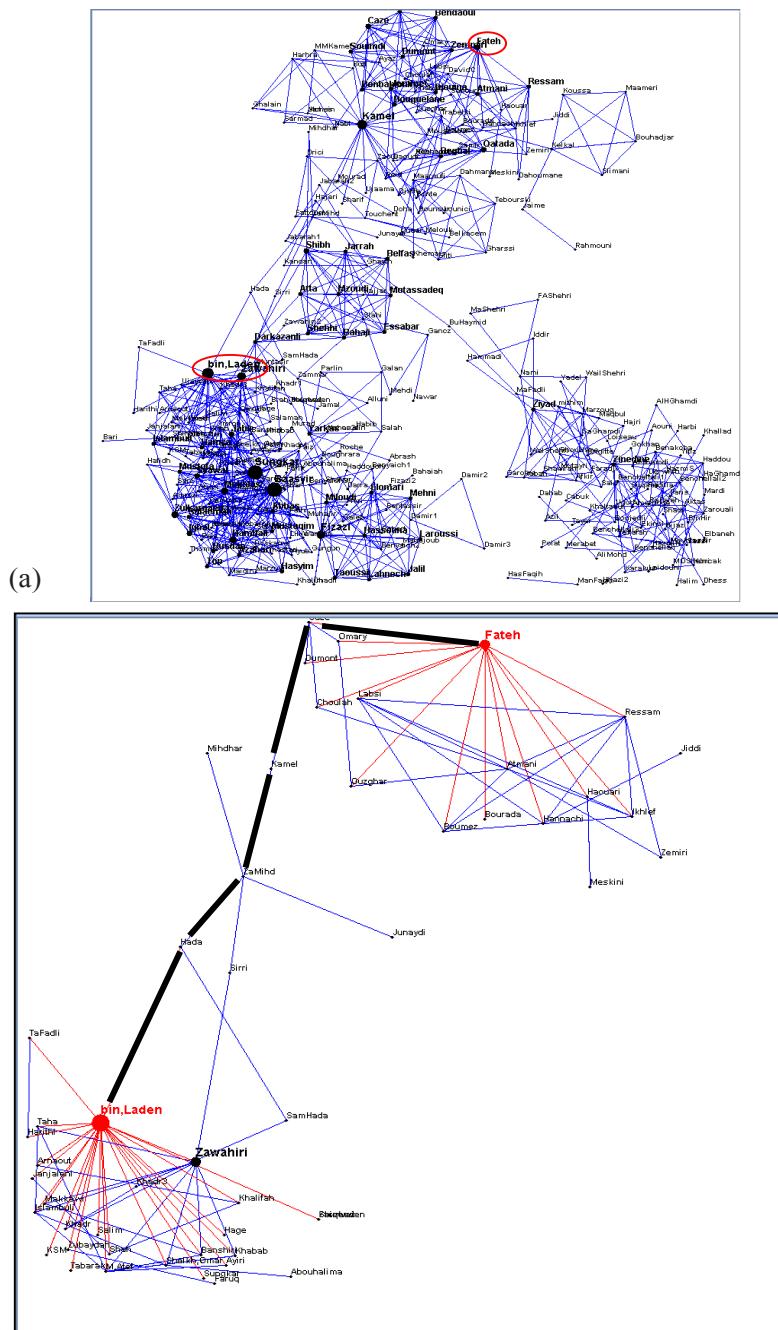


Fig. 3.10. (a) The Global Salafi Jihad terrorist social network with bin Laden and Fateh as the focus nodes, (b) Extracting the shortest path between bin Laden and Fateh (bin Laden, Hada, ZaMihd, Kamel, Caze, Fateh).

3.3.3 Case 3 – Determining the Relationships of the Terrorists in the US Millennial Plot Operation

It was reported that Ressam carried out the bomb mission in December 1999, known as the US Millennial Plot, but he failed. Fateh Kamel is the hub of the terrorist group who organized the US Millennial Plot. There were several other terrorists who had involved in the operation: Omary, Ikhlef, Haouari, Meskini, Labsi, Atmani, and Ouzghar [7].

According to the report, Ressam was supposed to carry out the bomb mission with Meskini's assistance in December 1999. Ressam attempted to infiltrate from Canada to US through US border at Port Angeles, Washington on December 14, 1999. He put the chemical material in a rented car. A custom inspector found him suspicious and asked him to pull over for inspection. The US custom discovered the chemical material and arrested Ressam. Meskini, who lived in US, was supposed to assist Ressam after he crossed the border but he had not meet Ressam at the US border because Ressam was already arrested. Meskini was later arrested by US FBI in New York.

Before the mission, seven terrorists were involved in supporting the mission. Omary set up the network of supporters with Fateh for the Bosnia Jihad. Ikhlef helped Ressam with the planning of the Los Angeles airport bombing. Ikhlef had been implicated in a 1992 bombing at Algiers airport that killed eleven and injured more than one hundred. Haouari provided some money, false credit cards and logistical support. Meskini, who is Haouari's childhood friend, helped Ressam to deliver the bomb to the airport but had never meet Ressam at the US border. Labsi was supporting Ressam by forwarding Meskini's visa to Ressam. Atmani and Ouzghar were invited to Canada by Fateh to support the mission. (More detail information about the US Millennial Plot can be found at pages 99-103 in [7].)

As shown in Fig. 3.11(a), if we select Fateh as the focus node, we can extract the most relevant nodes in the terrorist social network that are relevant to the US Millennial Plot since Fateh is the hub who organized the plot. Applying fisheye views and fractal views as shown in Fig. 3.11(b), the most relevant nodes are captured and exploded in the view. Sixteen terrorists that are most relevant are extracted in the view. Eight of them (circled in Fig. 3.11(b)) are the active members in the US Millennial Plot. The distance between the nodes reflects the strength of association between the corresponding nodes.

In the extracted terrorist social network, we also find that Ressam does not have direct relationship with Meskini. If we select Ressam and Meskini as focus nodes, we extract the path between Ressam and Meskini, which is (Ressam, Fateh, Haouari, Meskini). Such path reflects that Ressam did not know Meskini but through Fateh's arrangement to obtain Meskini's assistance in the Millennial Plot. Meskini got involved in the plot through the introduction of his childhood friend Haouari. The extracted path by fractal view as shown in Fig. 3.11(c) clearly illustrated such relationship.

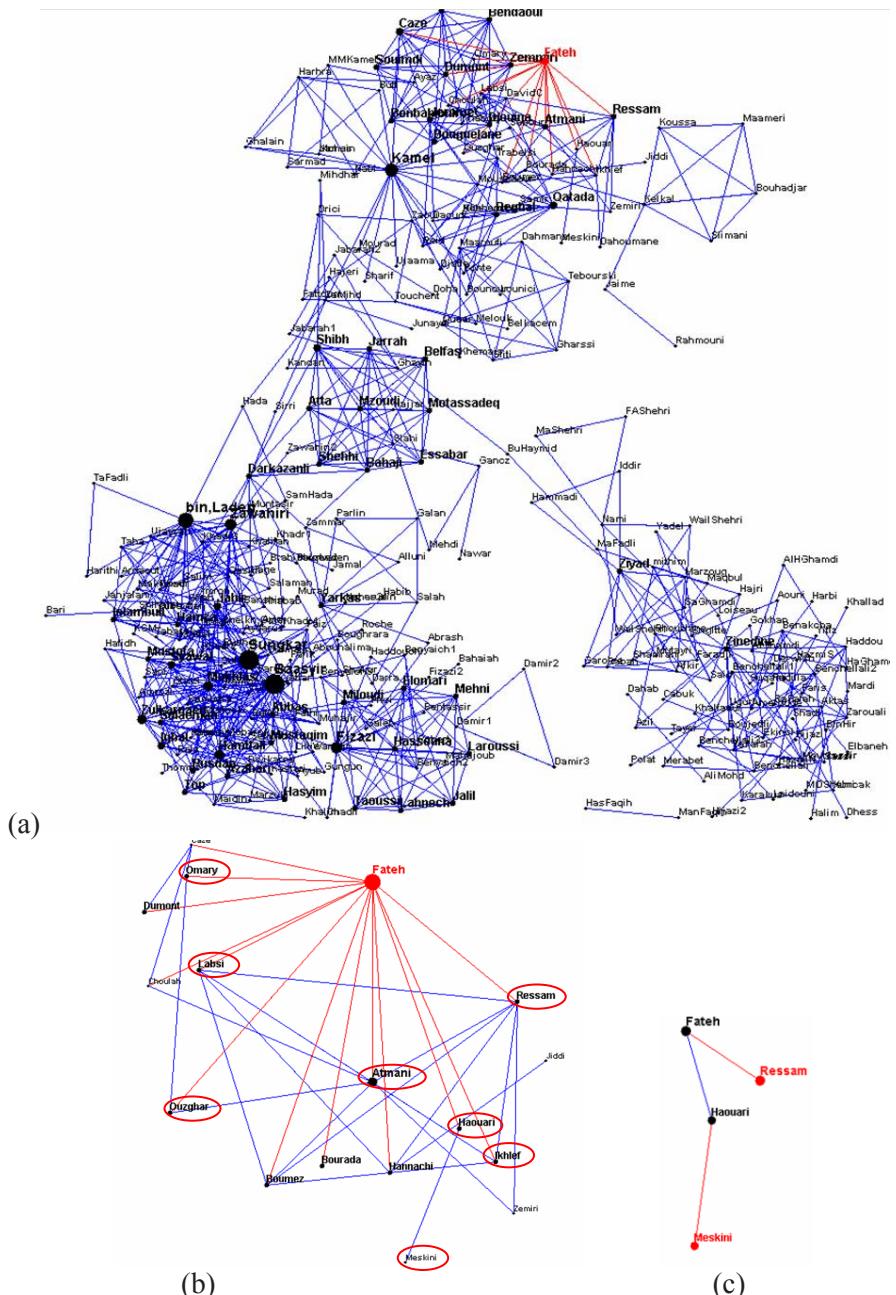


Fig. 3.11. (a) The Global Salafi Jihad terrorist social network with Fateh as the focus node, (b) Extracting the subgroup with Fateh as the center, (c) Extracting the path between Ressam and Meskini.

3.4 Conclusion

In this chapter, we have introduced the terrorist social network and its potential to support investigation and combat against terrorism. Centrality measurements are important techniques to extract the important nodes in the social networks, which are likely to be the leaders and gateways. We have also introduced the visualization techniques such as fisheye views and fractal views to provide interactive visualization to discover knowledge from the terrorist social networks.

References

1. Brown, M.H., Meehan, J.R., Sarkar, M.: Browsing Graphs using a Fisheye View. In: Proceedings of ACM on Human Factors in Computing Systems, Amsterdam, Netherlands (1993)
2. Eades, P.: A Heuristic for Graph Drawing. Congressus Numerantium 42 (1984)
3. Deter, J.: Fractals. Plenum, New York (1988)
4. Furnas, G.W.: Generalized Fisheye Views. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (1986)
5. Herman, I., Melancon, G., Marshall, M.S.: Graph Visualization and Navigation in Information Visualization: A Survey. IEEE Transactions on Visualization and Computer Graphics 6(1), 24–43 (2000)
6. Koike, H., Yoshihara, H.: Fractal Approaches for Visualizing Huge Hierarchies. In: Proceedings of IEEE Symposium on Visual Languages, Bergen, Norway, August 24–27 (1993)
7. Manojit, S., Brown, M.H.: Graphical Fisheye Views. Communications of the ACM 37(12), 73–83 (1994)
8. Sageman, M.: Understanding Terror Networks. University of Pennsylvania Press (2004)
9. Yang, C.C., Chen, H., Hong, K.: Visualization of Large Category Map for Internet Browsing. Decision Support Systems 35(1), 89–102 (2003)
10. Yang, C.C., Liu, N., Sageman, M.: Analyzing the Terrorist Social Networks with Visualization Tools. In: Proceedings of the IEEE International Conference on Intelligence and Security Informatics, San Diego, CA, US, May 23–24 (2006)
11. Yang, C.C., Ng, T.D., Wang, J.H., Wei, C., Chen, H.: Analyzing and Visualizing Gray Web Forum Structure. In: Proceedings of the Pacific Asia Workshop on Intelligence and Security Informatics, Chengdu, China, April 11–12 (2007)Online Resources

Online Resources

NetDraw, <http://www.analytictech.com/Netdraw/netdraw.htm>

Questions for Discussions

1. Describe the three types of centrality measurements and how they can be of use in terrorist social network analysis.
2. What are the six types of relationships between terrorists that are adopted in the terrorist social network?
3. Plot a three-dimensional graph to illustrate how the distortion factor and the radius of a node affect the fisheye view.
4. Explain how the fractal views extract the important paths between leaders in a terrorist social network.

Understanding the Nexus of Terrorist Web Sites

Jennifer Xu¹ and Hsinchun Chen²

¹ Computer Information Systems Department,
Bentley College, USA
jxu@bentley.edu

² Department of Management Information Systems, Eller College of Management,
The University of Arizona, USA
hchen@eller.arizona.edu

Abstract. In recent years terrorist groups have been using the World-Wide Web to spread their ideologies, disseminate propaganda, and recruit members. Studying the terrorist Web sites may help us understand the characteristics of these Web sites and predict terrorist activities. In this chapter, we propose to apply network topological analysis methods on systematically collected the terrorist Web site data and to study the structural characteristics at the Web page level. We conducted a case study using the methods on three collections of terrorist Web sites: Middle-Eastern, US domestic, and Latin-American. We found that the Web page networks from these three collections have the small-world and scale-free characteristics. We also found that smaller size Web sites which share similar interests tend to make stronger inter-site linkages, which help them form the giant component in the networks.

4.1 Introduction

Terrorism and terrorist activities substantially threaten national security and have far reaching economic, political, psychological and social impacts. Since the tragic events of September 11, authorities have taken extensive counter-terrorism measures that have to some extent reduced the occurrences of terrorist attacks. However, instead of feeling safer the public believes that the risk is still high because terrorist groups remain active. In order to design highly successful, effective counter-terrorism strategies a thorough understanding of the terrorism phenomena from different perspectives is warranted. Because terrorist groups often operate in network forms in which individual perpetrators cooperate with each other and exploit information technology to plan and implement their attacks [20], we could gain valuable knowledge about the terrorist groups by studying various structural properties of terrorist network operations in the real world and the cyberspace. This chapter focuses on the “virtual” social movements of terrorist groups in the cyberspace and seeks to understand the structure of the network formed by the Web pages of terrorist groups and their supporters. This network is considered to be the alternate side of the Web and referred to as the Dark Web.

Nowadays, terrorist groups are taking advantage of a variety of media to disseminate propaganda, seek support, and recruit new members. The World-Wide Web, an effective information presentation and dissemination tool, has been widely used by

terrorist groups as a communication medium [28]. The Web presence of these terrorist groups reflects their characteristics and may provide information about planned terrorist activities. Thus, monitoring and studying the content and structural characteristics of terrorist Web sites may help us analyze and even predict the activities of terrorist groups. In addition, the knowledge gained may help researchers understand the root causes, agenda setting, information operations of groups, and the emergence of the new types of “virtual” social movements.

Although the Dark Web has recently received government and media attention, our systematic understanding of how terrorists use the Web is limited. Recently, researchers have employed content analysis and Web structure mining to reveal the characteristics of terrorist Web sites at the site level. This chapter presents our study of the structural characteristics of terrorist Web sites at a lower granularity—page level. Based on a systematically collected Dark Web data set, we conducted topological analysis to compare the hyperlink structures of terrorist Web sites from three geographical regions: Middle-East, the United States, and Latin-America. The study is intended to address three research questions:

1. What is the structure of the Dark Web?
2. Are there structural differences in the Dark Web in different regions?
3. What are the implications of these structural properties?

The remainder of this chapter is organized as follows. Sect. 4.2 reviews previous work on Web structure mining, the structure of terrorist Web sites, and the topological analysis methods. In Sect. 4.3, we present our data collection methods and the Dark Web dataset. In Sect. 4.4, we report and discuss our findings from the analysis. Sect. 4.5 concludes the chapter with implications and future research directions.

4.2 Literature Review

In this section, we review prior research on Web structure mining, the network topology analysis methodology, and the structure of terrorist Web site networks.

4.2.1 Web Structure Mining

Web mining is about the automatic discovery of information, service, and valuable patterns from the content of Web documents (Web content mining), the structure of hyperlinks (Web structure mining), and the usage of Web pages and services (Web usage mining) [7]. An important application of Web mining is to improve the design of online search engines and crawlers to help users find what they look for more effectively and efficiently [5].

With the extensive hyperlink structure of the Web, Web structure mining is highly promising. Previous studies have shown that the link structure of the Web represents a considerable amount of latent human annotation [10]. For example, when there is a direct link from page A to page B, it often means that the author of page A recommends page B because of the relevant contents in page B. Moreover, similar to citation analysis in which frequently cited articles are considered more important, Web pages with more incoming links are often considered to be better than those with

fewer incoming links. As a result, the network of hyperlinks to a large extent resembles social network and thus can be mined for previously hidden patterns of information such as important, high quality Web pages and Web communities that consist of similar topics.

In Web structure mining research, the *HITS* [12] and *PageRank* [3] algorithms are the two most widely used methods for locating high-quality documents on the Web. The HITS algorithm can find authoritative pages that receive many in-links from other pages, and hub pages that contain many out-links pointing to authoritative pages. The HITS algorithm has also been used widely in methods for detecting Web communities [10, 13]. Kumar, et al. [13], for example, propose a trawling approach to find a set of core pages containing both authoritative and hub pages for a specific topic. The core is a directed bipartite subgraph whose node set is divided into two sets with all hub pages in one set and authoritative pages in the other. The core and the other related pages constitute a Web community.

4.2.2 Network Topological Analysis

Statistical analysis of network topology [1] is a recent development in network structure analysis research. In network topological analysis, entities, regardless of their contents, are treated as nodes and their relations are treated as links. The result is a graph representation of the network. Three topologies have been widely studied recently, namely, random network [6], small-world network [24], and scale-free network [2]. Different topologies have different structural characteristics and implications. In a random network the probability that two randomly selected nodes are connected is a constant p . As a result, each node has roughly the same number of links and nodes are rather homogenous. In addition, communities are not likely to exist in random networks. Small-world networks, in contrast, have a significantly high tendency to form groups and communities.

Three statistics are used to categorize the topology of a network: *average shortest path length*, *clustering coefficient*, and *degree distribution*. Random networks are characterized by small shortest path length, low clustering coefficient, and Poisson degree distribution with a single characterizing average degree. The small shortest-path length together with the high clustering coefficient of small-world networks reflects the *six degrees of separation* phenomenon [16]. The distinctive characteristic of the scale-free network is its power-law degree distribution, which is skewed toward small degrees and has a long, flat tail for large degrees.

It has been found that most empirical networks such as social networks, biological networks, and the Web are nonrandom networks [1]. Actually, many of these networks are found to be small-world networks. Moreover, many of these networks are also scale-free, meaning that a large percentage of nodes have just a few links, while a small percentage of nodes have a large number of links. Studies have shown that the WWW, in general, is both a small-world and a scale-free network [1].

A number of measures and approaches, many of which are borrowed from Social Network Analysis (SNA) research [23], can be employed to reveal other structural patterns of a network. For example, centrality measures in SNA are often used to locate key nodes. Freeman [8] provides definitions of the three most popular centrality measures: *degree*, *betweenness*, and *closeness*.

Degree measures how active a particular node is. It is defined as the number of direct links a node has. “Popular” nodes with high degree scores are the leaders, experts, or hubs in a network. In the counter-terrorism and crime fighting context, the removal of key offenders is often an effective disruptive strategy [21, 14]. *Betweenness* measures the extent to which a particular node lies between other nodes in a network. The betweenness of a node is defined as the number of geodesics (shortest paths between two nodes) passing through it. Nodes with high betweenness scores often serve as gatekeepers and brokers between different parts of a network. They are important communication channels through which information, goods, and other resources are transmitted or exchanged [23]. *Closeness* is the sum of the length of geodesics between a particular node and all the other nodes in a network. It actually measures how far away one node is from other nodes and is sometimes called “farness” [8]. A node with low closeness may find it very difficult to communicate with other nodes in the network. Such nodes are thus more “peripheral” and can become outliers in the network [21, 27].

The topological analysis has been used in previous studies on terrorist networks [21, 14, 15] and terrorist Web site structural studies at the site level [28].

4.2.3 Web Mining Studies on Terrorist Web sites

The World Wide Web has been increasingly used by terrorists to spread their ideologies. According to the Southern Poverty Law Center (SPLC) [22], there were 708 active extremist and hate groups in the US in 2002. These groups had 443 Web sites in 2002 and this number increased to 497 in 2003.

Researchers and watchdog organizations such as SPLC have started monitoring, tracing and studying terrorist Web sites. The traditional approach is to study the contents and structure of these Web sites [26]. This approach is limited in the size of dataset and cannot be used to keep track of the dynamic characteristics of terrorist Web sites. Like other Web sites, terrorist Web sites quickly emerge; the content and hyperlinks are frequently modified, and they may also swiftly disappear [25].

Web content mining and structure mining techniques have been used to study terrorist Web site. Studies on the content of terrorist Web sites have shown that terrorist Web sites present different characteristics from other ordinary Web sites. For example, Gerstenfeld, et al. found that many terrorist Web sites contain multimedia contents and racist symbols [9]. Gustavson and Sherkat found that terrorist groups used the Internet mainly for ideological resource sharing [11]. This finding was also supported by a few other studies such as Zhou, et al. [28], which analyzed the contents of terrorist Web sites in the United States and found that sharing ideology was one of the most important purposes for building these Web sites.

Structure analysis based on hyperlinks has also been seen in several previous studies. It is reported that Web site hyperlink relations provide reasonably accurate representation of terrorist groups’ inter-organizational structure [28, 4]. However, most of these studies focus on the hyperlink structures at the site level. There are few studies that analyze the hyperlink structure of Web sites among different terrorist groups at the page level, which may provide insight into the structure of terrorist groups. The topological characteristics of these Web sites at a lower granularity (page level)

remain unknown. In this chapter, we analyze the topological characteristics of Dark Web to reveal the structural properties of terrorist Web sites at the page level.

4.3 The Dark Web Dataset

We call the special section of the Web that is used by terrorists, extremist groups, and their supporters the “Dark Web.” As a long-term research project, we have kept collecting and tracing the content and hyperlinks of several terrorist groups’ Web sites and created a Dark Web test bed [19].

In our research, we focused on terrorist groups from three geographical areas: the United States, Latin-America, and Middle-Eastern countries. By November 2004, we had collected three batches of Dark Web data by spidering these Web sites. To identify the correct terrorist groups to spider, we used the reports from authoritative sources suggested by a domain expert with 13 years of experience. The sources include: Anti-Defamation League, FBI, Southern Poverty Law Center, Militia Watchdog, United States Committee for a Free Lebanon, Counter-Terrorism Committee of the UN Security Council, and etc. From these resources, a total of 224 US domestic terrorist groups and 440 international terrorist groups were identified.

We then identified an initial set of terrorist group URLs and expanded them. In the initial set of URLs, all US domestic group URLs and some international group URLs were directly obtained from US State Department reports and FBI reports. Additional international group URLs were identified through online searches. We constructed three terrorism keyword lists in terrorist groups’ native languages, which contain terrorist organization name(s), leader(s)’ and key members’ names, slogans, special words used by terrorists, etc. From the search results, those Web sites that were explicitly purported to be official sites of terrorist organizations and that contained praise of or adopt ideologies espoused by a terrorist group were added to the initial URL set. The initial URL set is expanded by adding the URLs’ in-link and out-link Web sites. Manual filtering was performed on the expanded links to ensure their quality.

After the URL of a terrorist group Web site was identified, we used a digital library building toolkit to collect the contents and hyperlinks from the sites. We identified 108 US domestic terrorist Web sites, 68 Latin-American terrorist Web sites, and 135 Middle-Eastern terrorist Web sites.

After we collected the Web pages, the static HTML files and dynamic files were parsed and the hyperlinks in the files were extracted. We created a Dark Web page network, whose nodes were Web pages in the Dark Web collection and links were hyperlinks between these pages. For the US domestic collection, the network contained 97,391 nodes. The Latin-American collections contained 152,414 nodes. The Middle-Eastern collection contained 298,761 nodes.

4.4 Results and Discussion

4.4.1 Network Topological Analysis

Table 4.1 shows the basic statistics of the networks resulted from the three collections. In Table 4.1, the number of nodes is the total number of Web pages in each collection.

The number of directed links is the number of hyperlinks between pages. For example, if there is a hyperlink in page A pointing to page B and a hyperlink from B to A, we consider them two links. The number of undirected links, in contrast, ignores the direction of a hyperlink. The two hyperlinks between A and B thus would be considered as one link.

The remaining statistics in Table 4.1 were all based on the undirected networks, in which the directions of the hyperlinks were ignored. The average degree, $\langle k \rangle$, is the average number of (undirected) links a node has. We define clustering coefficient as [17]:

$$C = \frac{3 \times \text{number of triangles on the graph}}{\text{number of connected triples of nodes}} \quad (4.1)$$

The average path length, l , is defined as the mean of the lengths of all-pair shortest paths in a graph, and the diameter, D , as the maximum of the shortest path length. Because the page-level networks were rather large, we employed an approximation algorithm called ANF [18] to calculate the average path lengths. The ANF approach uses a *neighborhood function* for categorizing the importance of a node. The neighborhood function for a node u at distance h is the total number of nodes that can be reached from u within h or fewer hops. An important router in a computer network, for example, will be the one that can reach most of the routers within a few hops. Unlike the traditional Dijkstra's method that calculates the exact shortest paths in a graph, the ANF approach computes the approximate shortest paths and is much faster than the Dijkstra's method. It is thus more applicable and feasible for very large graphs such as the Web.

The statistics, C_{rand} and l_{rand} , are the average clustering coefficient and the average path length for the random network counterpart. A counterpart for a network was generated by creating a random graph containing the same number of nodes and links as in the network. For each network, we generated 30 random network counterparts and computed the C_{rand} and l_{rand} .

To estimate the diameters of the three networks, we constructed three samples by randomly selecting 400 nodes in each network and calculated the shortest path for each sample. Because these diameters were estimates from the samples, we do not present the exact values but their ranges in Table 4.1.

We found that these networks were not connected. Each network consisted of many isolated components, between which no links existed. Nodes could reach any other node within the same component through a path. We report the number of components for each network in Table 4.1, and the number of nodes and number of links in the largest connected component for each network. The largest connected component is usually called giant component in graph theory.

Comparing the topological characteristics of the three networks we found that the Middle-Eastern network was much larger than the US domestic network and the Latin American Network. The number of nodes in the Middle-Eastern network (298,761) is almost three times as many as that in the US domestic network (97,391) and two times as many as that in the Latin-American network (152,414). Among the three networks, the Middle-Eastern network has the highest average degree (12.66), indicat-

ing that their Web pages tend to refer to each other more often than those in the US domestic network and in the Latin-American network. The size of the Middle-Eastern network and the high average degree may indicate their relatively active status and strong intention to cooperate with each other.

All the three networks have rather high clustering coefficients and small average path length comparing with their random network counterparts. The high clustering coefficient indicates that the networks contain dense and strongly connected local clusters. In this case, it is obvious that the Web pages are more likely to point to pages within the same Web site, resulting in site-based local clusters. Note that the clustering coefficient of the Middle-Eastern network is smaller than those of the other two networks. This may be caused by two reasons. First, the US domestic and Latin-American networks have a much larger number of components than the Middle-Eastern network. These components are site-based local clusters, causing the overall average clustering coefficients of the whole networks to be higher. Second, it may be because that the pages in the giant component of the Middle-Eastern network are more decentralized.

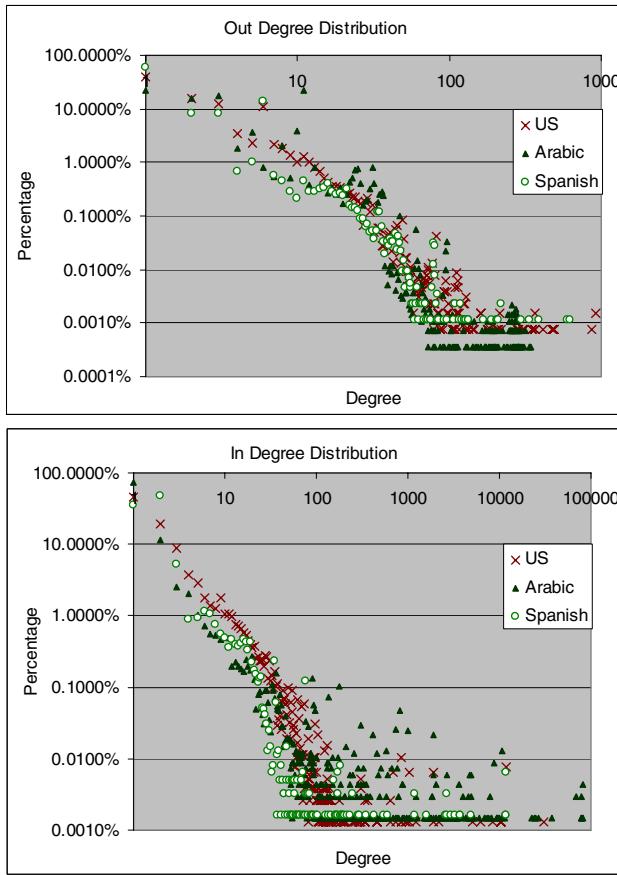
The degree distribution, $p(k)$, is the probability that a node has exactly k links. For degree distributions, we studied the original directed networks in which hyperlinks had their directions. Fig. 4.1 shows the in-degree distributions and out-degree distributions of the three networks in log-log plots. All the six degree distributions have long, flat tails which are often observed in large empirical networks [1]. The in-degree distributions of the Middle-Eastern network and the US domestic network follow a power law degree distribution. The out-degree distribution of the three networks and the in-degree distribution of the Latin American network present somewhat two-regime power-law characteristics.

Table 4.1. Basic statistics of the three collections of Dark Web page level networks

Collections	US domestic	Latin-American	Middle-Eastern
No. of Nodes	97,391	152,414	298,761
No. of Directed Links	296,330	586,115	1,914,099
No. of Undirected Links	239,572	475,748	1,890,728
$\langle k \rangle$	4.92	6.24	12.66
C	0.32	0.31	0.06
C_{rand}	5.05E-05	4.1E-05	4.24E-05
l	3.33	4.70	3.52
l_{rand}	7.21	6.52	4.97
D	≥ 39	≥ 46	≥ 38
No. of Components	4,134	1,110	674
No. of Nodes in the Largest Component	81,803	22,175	255,699
No. of Links in the Largest Component	239,982	95,346	1,718,626

Table 4.2. Exponents of the three networks' degree distributions

Collections	<i>US domestic</i>	<i>Latin-American</i>	<i>Middle-Eastern</i>
In degree exponent	1.94	a. 2.16; b. 2.53	1.60
Out degree exponent	a. 1.95; b. 2.30	a. 2.24; b. 2.26	a. 1.88; b. 2.44

**Fig. 4.1.** Degree distributions of the three networks

The power-law distribution takes the form of $p(k) \sim k^{-\gamma}$. The values of the exponents of the six distributions are shown in Table 4.2. The special two-regime shape of the out-degree distributions of the three networks may be because a Web page normally does not contain so many hyperlinks to the other pages. Thus, the likelihood of such high out-degree pages will quickly drop as degree increases. For the in-degree distribution, as the Latin-American network contains several small components and very few large components, it is difficult for the high in-degree nodes to emerge. As the in-degree increases, the number of nodes with high in-degrees decreases quickly.

Table 4.3. The node percentages of the top 5 components in the three networks

<i>Component Size Rank</i>	<i>US domestic</i>	<i>Latin-American</i>	<i>Middle-Eastern</i>
1	53.67% pages 54 Web sites	22.77% pages 9 Web sites	85.62% pages 68 Web sites
	2.31% pages 1 Web site	6.58% pages 1 Web site	2.73% pages 1 Web site
3	0.68% pages 1 Web site	5.84% pages 10 Web sites	1.66% pages 1 Web site
	0.56% pages 1 Web site	4.61% pages 11 Web sites	1.35% pages 2 Web sites
5	0.43% pages 1 Web site	2.79% pages 1 Web site	1.13% pages 10 Web sites
	42.35% pages	57.41% pages	7.51% pages
Other Components			

4.4.2 Giant Component Analysis

Although the Middle-Eastern network is the largest network, it has fewer components than the other two networks. Table 4.3 shows the three networks' top five components' node percentage in their networks. The three networks all have a giant component (The Latin-American network's largest component has 22.77% of the nodes. But it is still very big compared with the other components in the network.). We also observed that these giant components usually are composed of several terrorist Web sites. The giant component of the Latin-American network contains fewer Web sites than the giant components in the other two networks. This may be because that these Latin-American terrorist groups have diverse ideologies and beliefs. As a result, it is less likely for them to refer to each other on their Web sites or to seek cooperation.

The three giant components compose the bulk of the three networks. We thus focused only on the giant components of the three networks.

In general, there is a positive correlation between a Web site's size (number of pages) and number of internal links. This is also observed in the Web sites and pages included in the giant components of the three networks (Fig. 4.2). However, these terrorist networks show special characteristics on inter-site links. Fig. 4.3 presents the relationship between the Web site sizes and the number of inter-site links of the three networks. In this figure, the vertical axis represents the number of hyperlinks between a pair of Web sites in the giant component and the other two axes represent the number of pages in the two Web sites. For all three networks, we observe that most of the inter-site links are not present between large Web sites. For example, in the Middle-Eastern network, most of the inter-site links appear between Web sites that have less than 10,000 pages. It is normal for large Web sites to share a large number of inter-links. However, if two Web sites with relatively small number of pages are connected by many inter-links, it means that the two Web sites must have a close relationship.

To further study the relationships between the small- and middle- sized Web sites, which usually are connected by many inter-site links, we selected and examined some of these Web sites. For example, in the US domestic network, there are 4,875

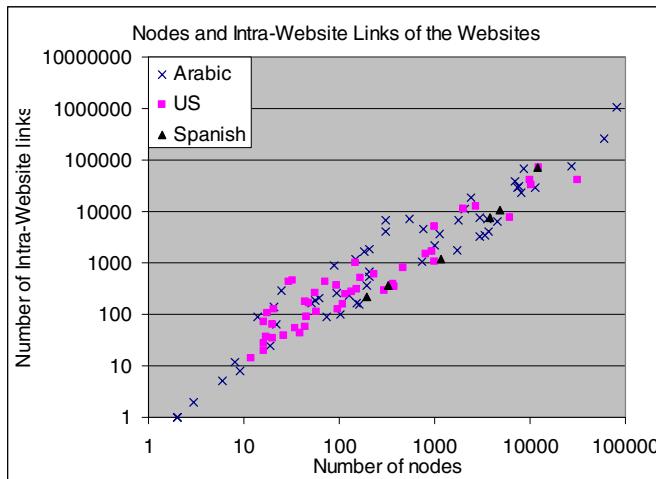


Fig. 4.2. Web site size and the number of internal links

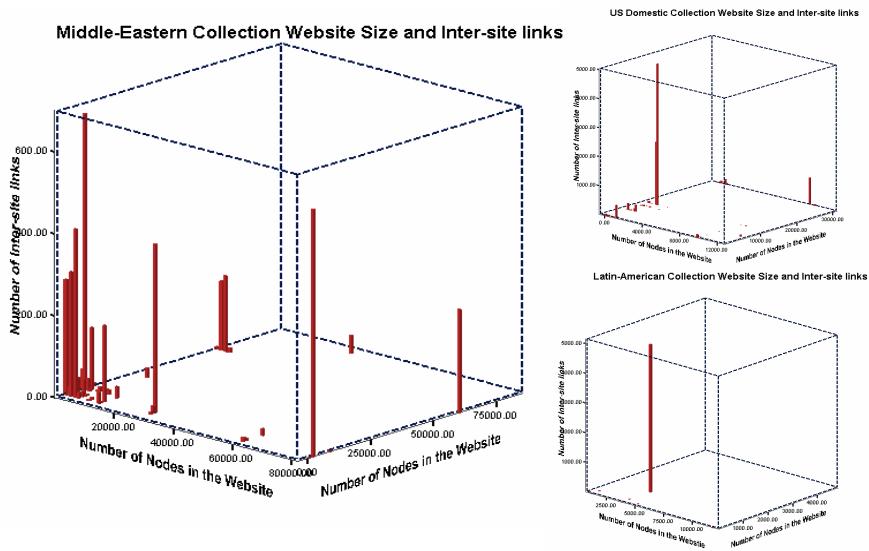


Fig. 4.3. Web site size and the number of inter-site links

inter-site links between www.resistance.com (12,188 pages) and www.natall.com (943 pages), 2,173 links between www.resistance.com and www.natvan.com (814 pages), and 414 links between www.natvan.com and www.natall.com. After we examined their Web sites, we found that the three Web sites have very close relationship. www.natall.com is the official Web site of the National Alliance, a white supremacist group. The www.natvan.com is another domain name of www.natall.com. The www.resistance.com is an e-commerce Web site owned by

Resistance Records, which is a music production company affiliated with National Alliance. Therefore, the dense inter-site hyperlinks reflect the close relationship between the organizations.

In the Latin-American Eastern network, clajadep.lahaine.org (3,796 pages) and www.carteleralibertaria.org (1,177 pages) are connected by 4,979 links. The Clajadep group (clajadep.lahaine.org) is focused more on broadcasting affairs in Mexico, while the Cartelera Libertaria group (www.carteleralibertaria.org) more on Spain. These two groups both belong to a terrorist alliance called “La Haine”, which has people from different Spanish-speaking and Latin America countries. The dense inter-site links may result from the fact that members in La Haine share similar ideologies, beliefs and interests.

Similarly, in the Middle-Eastern network, there are some small Web sites of close relationship. For example, www.daymohk.info (737 pages) and www.chechen.org (7,042 pages) have 676 links, which are both Web sites for extremists in Chechnya. www.palestine-info.info (3,698 nodes) and www.infopalestina.com (550 nodes) shared 410 interlinks, which are both news Web sites for Palestinians.

These cases show that the similarities in terrorist groups’ ideologies, beliefs, interests, and geographical closeness may cause their Web sites to frequently point to each other. From Fig. 4.3, we can see that the Middle-Eastern giant component has more and denser inter-site links than the other two giant components. It implies that the terrorist groups in Middle-Eastern have relatively closer relationships and more interconnections than those in the United States and Latin-American. Such dense inter-site links also enable the emergence of the giant component in the network.

4.5 Conclusions and Future Directions

In this research, we analyzed the structural properties of the Dark Web at the page level based on systematically connected terrorist Web sites data. Our goal was to reveal the characteristics of these Web sites. We conducted a case study based on a Dark Web test bed of US domestic, Latin-American, and Middle-Eastern terrorist Web sites. The findings from the case study help us answer our research questions.

1. What is the structure of the Dark Web?

We found that the three networks are small worlds with relatively short path lengths and high clustering coefficients. The small-world characteristics are often related to the efficiency of communication. Since terrorist Web pages are “close” to each other and form a small world, it is very easy for a visitor of a page to traverse to other parts of the network and explore many other pages related to terrorism information. We also found that the three networks’ in-degree and out-degree distributions roughly follow a power-law degree distribution, indicating that they have the scale-free characteristics. The scale-free characteristics imply that some pages are very popular and can attract much attention (in-links) or very “informative” by directing attention (out-links) to many other pages containing terrorism related resources. Our giant component analysis also reveals that the giant components tend to consist of Web sites that share similar interests. The dense inter-site hyperlinks help them form clusters and establish close relationships.

2. Are there structural differences in the Dark Web in different regions?

We found that the Middle-Eastern network was much larger than the US domestic network and the Latin American Network. The large size and high average degree may indicate that these Middle-Eastern terrorist groups are relatively active and constantly seek cooperation among them. We also found some differences in the degree distributions of these networks. For example, the in-degree distribution of the Latin American network presents a two-regime power-law, which is different from the single-regime power-law distributions observed in the Middle-Eastern and the US domestic networks.

3. What are the implications of these structural properties?

The structural properties found in the three networks imply that terrorism remains active in all regions. The Middle-Eastern areas are more active and also tend to be more cooperative among terrorist groups. More thorough and deeper analysis of the content, the usage, and the activities of these terrorism related Web sites should be done to obtain a more detailed, complete picture of these terrorist groups. We suppose that the difference in the structures of these networks should lead to the different counter-terrorism strategies in the different geographical areas.

The limitation of our study is that we focused only on the structural properties of the Dark Web without performing content analysis that might reveal important insights into the ideology, mission, and other information about these terrorists groups. Cautions must be made when any interpretation is drawn based solely on the structure of the Dark Web. In the future, we plan to perform in-depth content analysis on these terrorist web sites and combine it with other structural analysis methods such as cluster analysis from the network structural perspective to advance our knowledge of the Dark Web.

References

- Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. *Reviews of Modern Physics* 74, 47–97 (2002)
- Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* 286, 509–512 (1999)
- Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. In: The 7th WWW Conference, Brisbane, Australia (1998)
- Burris, V., Smith, E., Strahm, A.: White Supremacist Networks on the Internet. *Sociological Focus* 33, 215–235 (2000)
- Chau, M., Zeng, D., Chen, H., Huang, M., Hendriawan, D.: Design and evaluation of a multi-agent collaborative Web mining system. *Decision Support Systems* 35, 167–183 (2003)
- Erdos, P., Renyi, A.: On random graphs. *Publ. Math-Debrecen* 6, 290–297 (1959)
- Etzioni, O.: The World Wide Web: Quagmire or gold mine. *Communications of the ACM* 39, 65–68 (1996)
- Freeman, L.C.: Centrality in social networks: Conceptual clarification. *Social Networks* 1, 215–240 (1979)

9. Gerstenfeld, P.B., Grant, D.R., Chiang, C.P.: Hate Online: A Content Analysis of Extremist Internet Sites. *Analyses of Social Issues and Public Policy* 3, 29 (2003)
10. Gibson, D., Kleinberg, J., Raghavan, P.: Inferring web communities from link topology. In: The 9th ACM Conference on Hypertext and Hypermedia, Pittsburgh, PA (1998)
11. Gustavson, A.T., Sherkat, D.E.: Elucidating the Web of Hate: The Ideological Structuring of Network Ties among White Supremacist Groups on the Internet. In: Ann. Meeting Am. Sociological Assoc. (2004)
12. Kleinberg, J.: Authoritative sources in a hyperlinked environment. In: The 9th ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA (1998)
13. Kumar, S.R., Raghavan, P., Rajagopalan, S., Tomkins, A.: Trawling the web for emerging cyber-communities. *Computer Networks* 31, 1481–1493 (1999)
14. McAndrew, D.: The structural analysis of criminal networks. In: Canter, D., Alison, L. (eds.) *The Social Psychology of Crime: Groups, Teams, and Networks*, I, Aldershot, Dartmouth. Offender Profiling Series, vol. III, pp. 53–94 (1999)
15. McIlwain, J.S.: Organized crime: A social network approach. *Crime, Law & Social Change* 32, 301–323 (1999)
16. Milgram, S.: The small world problem. *Psychology Today* 2, 60–67 (1967)
17. Newman, M.E., Watts, D.J., Strogatz, S.H.: Random graph models of social networks. *Proc. Natl Acad Sci.* 99(1), 2566–2572 (2002)
18. Palmer, C.R., Gibbons, P.B., Faloutsos, C.: ANF: A fast and scalable tool for data mining in massive graphs. In: The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada (2002)
19. Qin, J., Zhou, Y., Lai, G., Reid, E., Sageman, M., Chen, H.: The Dark Web portal project: Collecting and analyzing the presence of terrorist groups on the web. In: Kantor, P., Muresan, G., Roberts, F., Zeng, D.D., Wang, F.-Y., Chen, H., Merkle, R.C. (eds.) *ISI 2005. LNCS*, vol. 3495, pp. 623–624. Springer, Heidelberg (2005)
20. Ronfeldt, D., Arquilla, J.: What next for networks and net-wars? In: Arquilla, J., Ronfeldt, D. (eds.) *Networks and Netwars: The Future of Terror, Crime, and Militancy*, Rand Press, Santa Monica (2001)
21. Sparrow, M.K.: The application of network analysis to criminal intelligence: An assessment of the prospects. *Social Networks* 13, 251–274 (1991)
22. S.P.L. Center, Hate Groups, Militias on Rise as Extremists Stage Comeback (2004), <http://www.splcenter.org/center/splcreport/article.jsp?aid=71>
23. Wasserman, S., Faust, K.: *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge (1994)
24. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. *Nature* 393, 440–442 (1998)
25. Weimann, G.: How Modern Terrorism Uses the Internet. United States Institute of Peace (2004), <http://www.terror.net> (Special Report 116)
26. Whine, M.: Far Right on the Internet. In: Loader, B. (ed.) *Governance of Cyber-space*, Routledge, pp. 209–227 (1997)
27. Xu, J., Chen, H.: CrimeNet Explorer: A framework for criminal network knowledge discovery. *ACM Transactions on Information Systems* 23, 201–226 (2005)
28. Zhou, Y., Qin, J., Lai, G., Reid, E., Chen, H.: Building Knowledge Management System for Researching Terrorist Groups on the Web. In: Proceedings of the Eleventh Americas Conference on Information Systems, Omaha, NE, USA (2005)

Online Resources

1. Terrorism Knowledge Portal at the AI Lab of University of Arizona:
<http://aidemo.eller.arizona.edu/tkp/servlet/tkp?event=search>
2. Anti-Defamation League: <http://www.adl.org/>
3. Southern Poverty Law Center: <http://www.splcenter.org/>
4. Militia Watchdog: <http://www.adl.org/mwd/m1.asp>
5. The Hate Directory: <http://www.bcpl.net/~rfrankli/hatedir.htm>
6. Counter-Terrorism Committee of the UN Security Council:
<http://www.un.org/sc/ctc/>

Questions for Discussions

1. What do the small-world properties imply about the function of the network of terrorist Web sites?
2. What do the scale-free properties imply about the function of the network of terrorist Web sites?
3. In addition to the three regions mentioned in the case study in this chapter, what are other regions where terrorism related activities are also observed and reported?
4. How do other Web media such as chat rooms, discussion forums, newsgroups, and recently, blogs, help terrorist groups disseminate their ideologies, spread propaganda, and recruit new members?

Multi-lingual Detection of Web Terrorist Content

Mark Last¹, Alex Markov¹, and Abraham Kandel²

¹ Department of Information Systems Engineering,
Ben-Gurion University of the Negev, Israel

{mlast, markov}@bgu.ac.il

² Department of Computer Science and Engineering,
University of South Florida, USA
kandel@csee.usf.edu

Abstract. The role of the Internet in the infrastructure of the global terrorist organizations is increasing dramatically. Beyond propaganda, the WWW is being heavily used for practical training, fundraising, communication, and other purposes. Terrorism experts are interested in identifying who is behind the material posted on terrorist web sites and online forums and what links they have to active terror groups. The current number of known terrorist sites is so large and their URL addresses are so volatile that a continuous manual monitoring of their multi-lingual content is definitely out of question. Moreover, terrorist web sites and forums often try to conceal their real identity. This is why *automated multi-lingual detection* methods are so important in the cyber war against the international terror. In this chapter, we describe a classification-based approach to multi-lingual detection and categorization of terrorist documents. The proposed approach builds upon the recently developed graph-based web document representation model combined with the popular C4.5 decision-tree classification algorithm. Two case studies are performed on collections of web documents in Arabic and English languages respectively. The first case study demonstrates that documents downloaded from several known terrorist sites in Arabic can be reliably discriminated from the content of Arabic news reports using a compact set of filtering rules. In the second study, we induce an accurate classification model that can distinguish between the English content posted by two different Middle-Eastern terrorist organizations (Hamas in the Palestinian Authority and Hezbollah in Lebanon).

5.1 Introduction

While the World Wide Web has evolved as a global technological and social phenomenon, terrorists have quickly learned to use it as a convenient and cost-effective information infrastructure. Web sites, online forums, and file-sharing services are routinely used by terrorist organizations for propaganda, recruitment, communications, and even basic training in kidnapping, explosive preparation, and other “core” terrorist activities. As indicated by the former US Deputy Defense Secretary Paul D. Wolfowitz, the Internet became “a tool that the terrorists use to conceal their identities, to move money, to encrypt messages, even to plan and conduct operations remotely” [12].

The military pressure put on the al-Qaeda leadership in Afghanistan after 9/11 has dramatically increased the role of the so-called “Dark Web” in the global activities of

the Jihadi terrorist organizations. In terrorism expert Peter Bergen's words: "They lost their base in Afghanistan, they lost their training camps, they lost a government that allowed them do what they want within a country. Now they're surviving on the internet to a large degree. It is really their new base" [4]. Bergen's statement is strengthened by Michael Doran, a Near East scholar and terrorism expert at Princeton University who says: "When we say al-Qaeda is a global ideology, this is where it exists—on the Internet" [26]. According to a recent estimate, the total number of Jihadi websites has increased from only 12 on September 10, 2001 to close to 5,000 in 2006 [6].

There is increasing evidence that terrorists are using the Web to distribute tactical orders to their sympathizers. For instance, in December 2003, a web site in Arabic published a detailed plan by Bin-Laden associates to force the US and its allies to withdraw from Iraq. The document specifically mentioned the fact that the Spanish government may lose the upcoming elections if Madrid were to be hit by terrorist attacks¹. In March 2004, just shortly before the Election Day in Spain, Madrid was hit by a fierce terrorist attack, which had apparently affected the election results and the subsequent policy of the new Spanish government with respect to Iraq. The direct link between the Islamist cell, which carried out the attack, and this Internet posting was later emphasized in the Spanish court ruling [8]. The July 7, 2005 bombings in London are also believed to have much stronger connections to the Internet than to any specific terrorist organization [13].

The Iraqi insurgency has raised terrorist abuse of the Internet to a new and more gruesome level. In 2004, the Web was awash with raw video footage of hostage beheadings perpetrated by followers of Abu Musab al-Zarqawi, starting with the murder of the American civilian contractor Nicholas Berg—which law enforcement agents believe was carried out by Zarqawi himself. From the terrorists' perspective, their online campaign was a tremendous success, since the full, raw videos of their murders spread rapidly around the Web and were downloaded by millions of users [26]. Another example of using the Internet for psychological warfare was a false announcement about the execution of two Italian hostages posted by an Islamic web site in September 2004 [5]. The two women were safely released a few days later.

As a continuous worldwide threat to the lives of innocent people, to the Democratic values of Western societies, and to the world stability in general, the international terrorism phenomenon has become an important research subject for experts in social and engineering sciences. The extensive terrorist activities on the Internet are also attracting an increasing attention of computer science and information systems researchers who are interested in identifying who is behind the material posted on terrorist web sites and online forums, what links they have to other terrorist web sites and to active terror groups behind them, and what threat, if any, they may pose. They would also like to identify temporal trends and abrupt changes in terrorist-related content as well as track down the "target audience" of individual and public online messages.

Reid, et al. [19] have developed an integrated approach to the study of the Jihad terrorism Web Infrastructure. Their study was based on more than 300,000 Web pages downloaded from 39 terrorist Websites, which were identified and filtered

¹ Walla, March 13, 2004 [www.walla.co.il] (in Hebrew).

using a series of semi-automatic steps supervised by domain experts. The analysis of the downloaded Web pages included automated identification of hyperlinked online communities and attribute-based content analysis of each web site, which was again performed with the assistance of domain experts. In [29], a similar approach was applied to the Web sites of US domestic extremist groups, where eight high-level attributes (communications, fundraising, propaganda, etc.) were used for characterizing the content of each web site.

The current number of known terrorist sites and active extremist forums is so large and their URL addresses are so volatile that a continuous manual monitoring of their multi-lingual content is definitely out of question. Moreover, terrorist web sites often try to conceal their real identity, e.g., by masquerading themselves as news portals or religious forums. A good example of such elusive strategy is the so-called Palestinian Information Center² currently operating in seven languages, which is nothing else but the chief portal of the Hamas movement, a militant Jihadi organization³. On the other hand, a simple keyword search (e.g., using words like “Jihad” or “Al-Qaeda”) can lead to sites about terrorism or even to fake terrorist websites [19]. This is why *automated detection methods* are so important in the research of the Internet misuse by terrorists. Particularly, there is a need of effective *filtering rules* for accurate identification of real terrorist content associated with specific terrorist groups.

Once a Web site is detected as being “terrorist”, automated categorization methods may need to perform a deeper analysis of its content to associate it with one of the known terrorist groups or organizations. This identification task requires more sophisticated techniques than the filtering task, since many groups (e.g., in the Jihadi movement) may share close views and common ideology with other groups. Automated *categorization* of web pages according to a pre-specified coding scheme (like the one defined in [19]) can also be highly beneficial for terrorism researchers.

In this chapter, we present a novel classification technique for automated identification of terror web sites in multiple languages. The chapter is organized as follows. In Sect. 5.2, we present a review of related works in the area of document classification and categorization. Two case studies based on collections of authentic web documents in English and Arabic languages are described in Sect. 5.3 and some conclusions are drawn in Sect. 5.4.

5.2 Literature Review

5.2.1 Document Categorization and Classification

Document categorization is formally defined in [25] as the task of assigning a Boolean value to each pair $\langle d_i, c_i \rangle \in D \times C$, where D is a collection of documents and $C = \{c_1, c_2, \dots, c_{|C|}\}$ is a set of pre-defined categories (topics). The value T (true) assigned

² <http://palestine-info.info/>

³ The terrorist organizations mentioned in this chapter are taken from the list of U.S.-Designated Foreign Terrorist Organizations, which is updated periodically by the U.S. Department of State, Office of Counterterrorism. The latest list can be downloaded from <http://www.infoplease.com/ipa/A0908746.html>

to $\langle d_j, c_i \rangle$ indicates the decision to associate a document d_j with the category c_i , and the value F (false) indicates the opposite. Examples of document topics in the Dark Web domain include: “terrorist”, “Hamas”, “bomb-making”, etc. Available document representations are described in the next sub-section of this chapter.

In some applications, several category labels may be assigned to each document. For example, a given *terrorist* document may belong to *Hamas* and present instructions for *bomb-making*. Such cases are called *multi-label categorization* whereas the case where exactly one category may be assigned to each document – *single-label categorization*. A special case of a single-label categorization is the *binary categorization*, where each document should be assigned to one of the two non-overlapping categories (such as *terrorist* vs. *non-terrorist*). *Multi-class categorization* involves more than two non-overlapping categories such as “PIJ”, “Hamas”, and “Al-Aqsa Brigades” (names of Middle-Eastern terrorist organizations). A multi-class problem can be reduced into multiple binary tasks (one-against-the-rest strategy)

Terrorist content detection can be seen as a binary categorization problem, if the goal is to classify a document or a Web site into the *terrorist* category or its complement – *non-terrorist*. This categorization task has the following specific requirements:

- *High accuracy.* The correct category/ categories of each document should be identified as accurately as possible.
- *Interpretability.* An automatically induced model should be subject to scrutiny by a human expert who may be able to enhance / correct the filtering rules based on her/his own knowledge in the terrorist domain.
- *Speed.* Due to the huge amount of Web sites and documents that are being posted and updated on a continuous basis, the model should be capable to process massive streams of web documents in minimal time.
- *Multilinguality.* The model induction methods should maintain a high performance level over web content in multiple languages.

Multi-lingual document classification relates to the case where documents, written in different languages should be labeled under the same category setting simultaneously as opposed to *cross-lingual classification systems* that can work with different languages but not simultaneously. Cross-lingual systems are usually integrated with machine translation tools.

With the rapid growth of the Internet, the World Wide Web has become one of the most popular sources of multilingual information. The ability to distribute multilingual information has increased the need to automatically navigate across multiple languages, and in the case of the Dark Web, finding terrorist pages in foreign languages. This is a *cross-lingual query matching* problem. Authors of [3] try to solve this problem for English and Arabic languages. The goal is to retrieve documents from an English document collection using queries in the Arabic language. To achieve this, a query should be translated as accurately as possible. Two machine translation techniques are compared in the article in terms of retrieval accuracy. Other multi-lingual solutions are presented in [1, 20].

5.2.2 Representation of Text and Web Documents

Traditional Representation of Text Documents

Standard categorization algorithms, such as decision trees [18], need a feature table as input for model induction. Each row of such table should refer to a labeled document and each column should represent a textual feature. Since different documents use different textual features (e.g., words, keyphrases, etc.), documents in their original format cannot be used as input for a classifier and there is a need to convert them into feature vectors of fixed dimensionality. In order to perform these conversions, a set of features common for all documents in the training corpus should be identified. For a text document, this is a problem of choosing a meaningful textual unit – a *term*.

The vector-space model [22] is typically used for document representation in information retrieval techniques. A set of terms $T(t_1, \dots, t_{|T|})$ that occur at least once in at least one document of the training corpus serves as a feature set and each document d_j is represented as a vector $\bar{d}_j = (w_1, \dots, w_{|T|})$, where each w_i is an importance weight of a term t_i in a document d_j . The set T is usually called *vocabulary* or *dictionary*. The popular ‘bag-of-words’ approach uses single words as terms. This approach is easily applicable to English, Arabic, and other languages, where distinct words are separated by white spaces. Alternative term definitions include n -grams, noun phrases, key-phrases, etc.

Irrelevant terms (“stop words”) can be removed the term set using a language-specific list of stop words. Normalization of English texts can be based on the “classical” Porter stemming algorithm [17]. In other languages, like Arabic, roots can be identified using morphological analysis techniques described in [1]. The dimensionality of the resulting feature set can be further reduced by removing most infrequent stems/roots in the corpus. The term weights in each document vector are typically calculated by the $TF \times IDF$ (term frequency \times inverse document frequency) measure [21], which is defined as

$$w_{ij} = TF \times IDF = TF_{ij} \times \log \frac{N}{n} \quad (5.1)$$

where:

w_{ij} = weight of Term t_j in Document d_i

TF_{ij} = frequency of Term t_j in Document d_i

N = number of documents in the collection

n = number of documents where term t_j occurs at least once

As indicated by the equation above, such a measure assigns the highest importance weight to terms that occur frequently in a specific document but do not occur at all in most other documents. Alternatively, documents may be represented by binary vectors, with each element indicating the presence or absence of the corresponding term in a given document.

Web Document Models

Most applications of web document classifiers still make use of the standard text representation techniques that were originally designed for plain-text documents. There are several reasons why such approach is not optimal. First, plain text classification

models make no use of metadata information, such as title, publication source, etc., which is available in most web documents. Second and more important, web documents contain HTML tags which are not found in plain-text documents. These tags determine document layout and can be a source of additional information about the documents. Traditional text document representations, such the vector-space model, ignore this important information. One may expect that a representation that contains more information about a document should increase the accuracy of classification methods.

In [28], five different classification approaches were presented and compared. The popular vector-space model was used for document representation, but HTML tags were treated differently in each technique. First, *no hypertext* approach made use of web document text only for classification. No additional information was extracted from HTML tags. In *encyclopedia* approach, the authors assume that all documents linked to a labeled document refer to the same category and use its words as predictive features. The same assumption was taken in *co-referencing* regularity but words from the original document obtain higher weights than words from linked documents. Available information about previously labeled documents was used in the *pre-classified* approach. For instance, if we know that a document d belongs to a category c then it makes sense to classify a document d_1 that is linked with d under c too. Under the last, *metadata* method only title and words under metadata tags are used for categorization. Experimental results of [28] show that all methods that make use of HTML tags information outperform standard text representation techniques. The limitation of these particular methods is that each of them is limited to only one type of metadata, while their combination can provide even better results.

Graph Based Representations of Web Documents

Though the ‘bag-of-words’ representation provides relatively good classification results in terms of accuracy, its limitations are obvious. This popular method of document representation does not capture important structural information, such as the ordering and the proximity of term occurrence or the location of a term within a document. Moreover, as indicated above, all vector-space models, which were developed for representation of plain text documents, do not make any use of the meta-tags presenting in any HTML document. The Graph-Theoretic Web Document Representation Technique introduced in [23] has the ability to capture important structural information hidden in the document and its HTML tags. It has been reported to outperform the vector-space model using several classification algorithms [24]. This novel representation model is briefly described below.

All graph representations proposed in [23] are based on the adjacency of terms in an HTML document. Thus, under the *standard method*, each unique term (keyword) appearing in the document becomes a node in the graph representing that document. Distinct terms (stems, roots, lemmas, etc.) can be identified by a stemming algorithm and other language-specific normalization techniques that are also used with the vector-space models. Each node is labeled with the term it represents. The node labels in a document graph are unique, since a single node is created for each distinct term even if a term appears more than once in the text. Second, if a word a immediately precedes a word b somewhere in a “section” s of the document, then there is a directed edge from the node corresponding to term a to the node corresponding to term

b with an edge label s . The ordering information is particularly important for representing texts in languages like English and Arabic, where phrase meaning strongly depends on the word order. An edge is not created between two words if they are separated by certain punctuation marks (such as periods). Sections defined for the standard representation are: *title*, which contains the text related to the document's title and any provided keywords (meta-data); *link*, which is the anchor text that appears in hyper-links on the document; and *text*, which comprises any of the visible text in the document (this includes hyperlinked text, but not the text in the document's title and keywords). Graph representations are language-independent: they can be applied to a normalized text in any language. An example of a standard graph representation of a short English web document is shown in Fig. 5.1, where *TL* denotes the title section, *L* indicates a hyperlink, and *TX* stands for the visible text.

The second type of graph representation is a “simple” representation. It is basically the same as the standard representation, except that we look at only the visible text on the page (no title or meta-data is examined) and we do not label the edges between

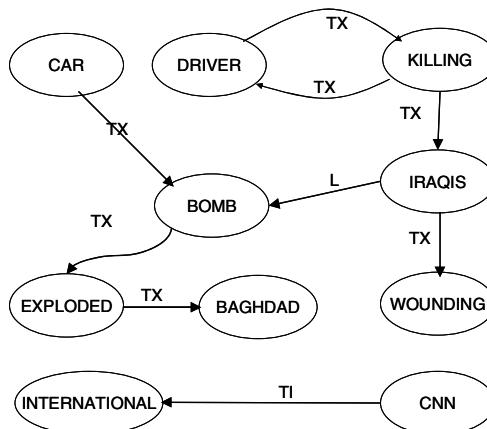


Fig. 5.1. Standard Graph Document Representation

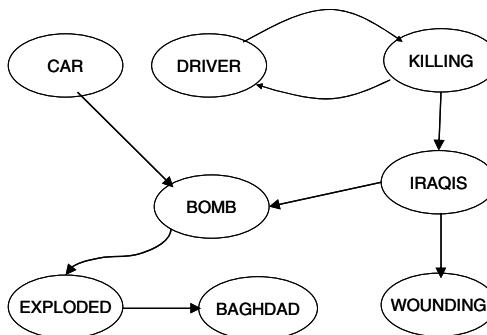


Fig. 5.2. Simple Graph Document Representation

nodes. Thus we ignore the information about the “section” of the HTML document where the two respective words appear together. An example of a simple graph representation of the same web document is shown in Fig. 5.2.

Several ways to modify the Standard and the Simple graph representations are shown in [23]. Under the *n-distance* representation, there is a user-provided parameter, n . Instead of considering only terms immediately following a given term in a web document, we look up to n terms ahead and connect the succeeding terms with an edge that is labeled with the distance between them. The *n-simple distance* is identical to n -distance, but the edges are not labeled, which means we only know that the distance between two connected terms is not more than n . The *absolute frequency* representation is similar to the simple representation (adjacent words, no section-related information) but each node and edge is labeled with an additional frequency measure. Finally, the *relative frequency* representation is the same as the absolute frequency representation but with normalized frequency values associated with the nodes and edges.

Available distance measures between two graphs allow us to classify graphs with some distance-based *lazy algorithms*⁴ like k -Nearest Neighbors. The computational complexity of such algorithms is relatively high, which makes them a poor choice for real-time categorization of massive web document streams. On the other hand, we cannot induce a classification model, such as a decision tree, from a graph structure using available data mining algorithms, which need a feature table as input for the induction process. Consequently, graphs need to be converted into a feature table for classification model induction with some model-based classifiers. For this purpose, terms should be defined and extracted from the web document graph structure. In the next section, we present a novel, hybrid method of term extraction designed specifically for the model-based classification task with documents represented by graphs.

5.2.3 Web Document Classification with the Hybrid Approach

Term Definition

In order to represent a web document, a term first has to be defined. The hybrid methodology is based on graph document representation [23]. In the hybrid representation methods, terms (discriminative features) are defined as subgraphs selected to represent a document already converted into a graph form. It is obvious that all possible subgraphs in a document graph cannot be taken as attributes because of their quantity, so some subgraph selection criteria and techniques need to be applied. In [11], two optional subgraph selection procedures are proposed, called Hybrid Naïve and Hybrid Smart.

Categorization Model Induction Based on a Hybrid Document Representation

The process for inducing a classification model from labeled web documents represented by graphs is shown in Fig. 5.3.

First we obtain a training set of labeled *web documents* $D = (d_1, \dots, d_{|D|})$ and a set of categories as $C = (c_1, \dots, c_{|C|})$, where each document $d_i \in D; 1 \leq i \leq |D|$ belongs to one and only one category $c_v \in C; 1 \leq v \leq |C|$. Then graph representation of documents

⁴ Algorithms that do not create a model in order to classify a data item.

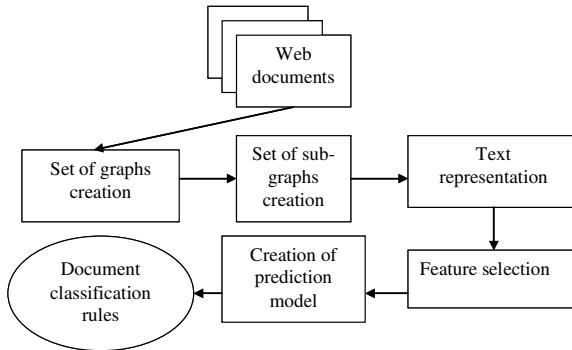


Fig. 5.3. Classification Model Induction

is generated (see Sect. 5.2.2.3 above) and a set of labeled graphs $G = (g_1, \dots, g_{|D|})$ is obtained. Now we are able to extract predictive features by identifying the subgraphs, which are most relevant for classification in a set of training graphs. The Naïve or the Smart methods can be used. A set of terms (subgraphs), or vocabulary $T = (t_1, \dots, t_{|T|})$ is the output of this stage.

Using T we can now represent all document graphs as vectors of Boolean features for every subgraph term in the set T (“1” – a subgraph from the set, created in the previous stage, appears in the graph of a particular document; “0” - otherwise). *Feature selection* may be performed to identify best attributes (Boolean features) for classification. Then prediction model creation and extraction of classification rules can be performed by one of the “eager” classification algorithms (e.g., Naïve Bayes Classifier, C4.5 decision-tree algorithm, etc.).

The Hybrid Naïve Approach

The Naïve approach to term extraction was initially introduced in [14]. All graphs representing the web documents are divided into groups by class attribute value (for instance: *terrorist* and *non-terrorist*). A frequent sub-graph extraction algorithm is then applied to each group with a user-specified threshold value t_{min} . We used the FSG algorithm [9] for frequent subgraphs extraction with all selection methods. Every subgraph more frequent than t_{min} is selected by the algorithm to be a term (discriminative feature), and stored in the vocabulary. All obtained groups of subgraphs (discriminative features) are combined into one set.

The Naïve method is based on a simple postulation that a feature explains the category best if it appears frequently in that category; in real-world cases, however, this is not necessarily true. For example if a sub-graph g is frequent in more than one category, it will be chosen as a feature by the Naïve method though it cannot make an effective distinction between documents belonging to those categories. The *Smart* extraction method presented in the next sub-section has been developed to overcome this problem.

The Hybrid Smart Approach

As in the Naïve representation, all graphs representing the web documents are divided into groups by class attribute value. In order to extract subgraphs, which are relevant for classification, some measures are defined, as follows:

SCF – Sub-graph Class Frequency:

$$SCF(g'_k(c_i)) = \frac{g'_k f(c_i)}{N(c_i)} \quad (5.2)$$

Where

$SCF(g'_k(c_i))$ - Frequency of sub-graph g'_k in category c_i .

$g'_k f(c_i)$ - Number of graphs that contains a sub-graph g'_k .

$N(c_i)$ - Number of graphs in category c_i .

ISF - Inverse Sub-graph Frequency:

$$ISF(g'_k(c_i)) = \begin{cases} \log\left(\frac{\sum N(c_j)}{\sum g'_k f(c_j)}\right) & \text{if } \sum g'_k f(c_j) > 0 \\ \log(2 \times \sum N(c_j)) & \text{if } \sum g'_k f(c_j) = 0 \end{cases} \quad \{\forall c_j \in C; j \neq i\} \quad (5.3)$$

$ISF(g'_k(c_i))$ - Measure for inverse frequency of sub-graph g'_k in category C_i .

$N(c_j)$ - Number of graphs belonging to a category c_j .

$g'_k f(c_j)$ - Number of graphs that contains g'_k belonging to a category c_j .

And finally we calculate the *CR – Classification Rate*:

$$CR(g'_k(c_i)) = SCF(g'_k(c_i)) \times ISF(g'_k(c_i)) \quad (5.4)$$

$CR(g'_k(c_i))$ - Classification Rate of sub-graph g'_k in category c_i . The interpretation of this measure is how well g'_k explains category c_i . $CR(g'_k(c_i))$ reaches its maximum value when every graph in category c_i contains g'_k and graphs in other categories do not contain it at all.

According to the Smart method, CR_{min} (minimum classification rate) is defined by the user and only sub-graphs with CR value higher than CR_{min} are selected as terms and entered into the vocabulary. The calculation of the Classification Rate for each candidate subgraph is a slightly more complicated and time-consuming procedure in the *Smart* approach than finding only the subgraph frequency because of the *ISF* (Inverse Sub-graph Frequency) calculation, where graphs from other categories are taken into account. Notwithstanding, as shown in [15], in some cases using the Smart representation produces better results in terms of accuracy.

Frequent sub-graph extraction problem

The input of the sub-graph discovery problem, in our case is a set of labeled, directed graphs and parameter t_{min} such that $0 < t_{min} < 1$ in case of the Naïve approach or CR_{min} in case of the Smart approach. The goal of the frequent sub-graph discovery is to find all connected sub-graphs that meet the threshold (e.g., occur in at least $(t_{min} * 100)$ % of the input graphs). Additional property of our graphs is that a labeled vertex is unique in each graph. This fact makes our problem much easier than the standard sub-graph

discovery case [9, 27] where such restriction does not exist. The most complex task in frequent sub-graph discovery problem is the *sub-graph isomorphism identification*⁵. It is known as NP-complete problem when nodes in the graph are not uniquely labeled but in our case it has a polynomial $O(n^2)$ complexity. We use *breadth first search* (BFS) approach and simplify the Frequent Subgraph Generation (FSG) algorithm given in [9].

5.3 Case Studies

5.3.1 Case Study 1: Identification of Terrorist Web Sites in Arabic (Based on [11])

About Document Collection

In this case study we try to classify real-world web documents into two categories (Boolean classification approach): *terrorist* and *non-terrorist*. Our collection consists of 648 Arabic documents where 200 belong to terrorist web sites and 448 to non-terrorist categories. The collection vocabulary contains 47,826 distinct Arabic words (after normalization and stopword removal).

Non terrorist documents were taken from four popular Arabic news sites: www.aljazeera.net/News, <http://arabic.cnn.com>, <http://news.bbc.co.uk/hi/arabic/news> and <http://www.un.org/arabic/news>. We automatically downloaded about 200 documents from each web site and then manually chose 448 documents while verifying that they are not belonging to the terror category. We also made sure that at least 100 documents from each web site are included into this group to ensure content and style diversity.

Terror content documents were downloaded from <http://www.qudsway.com> and <http://www.palestine-info.com/>, which are associated with Palestinian Islamic Jihad and Hamas respectively according to the SITE Institute web site (<http://www.siteinstitute.org/>). A human expert, fluent in Literary Arabic, has manually chosen 100 pages from each web site and labeled them as terror based on the entire *content* of each document rather than just occurrence of any specific keywords.

Preprocessing of Documents in Arabic

Text analysis of the Arabic language is a major challenge, as Arabic is based on unique grammar rules and structure, very different from the English language [10]. For example, orthographic variations are prevalent in Arabic; characters may be combined in different ways. Thus, sometimes in glyphs combining HAMZA or MADDA with ALIF, the HAMZA or MADDA is excluded. In addition, broken plurals are common, so the plural form might be very dissimilar to the single form.

Another problem is that many Arabic words have ambiguous meaning due to the three or four-lateral root system. In Arabic, a word is usually derived from a root containing three to four letters that might be dropped in some derivations. Also, short vowels are omitted in written Arabic and synonyms are very common. Each word can assume a very large number of morphological forms, due to an array of complex and

⁵ Means that graph is isomorphic to a part of another graph.

often irregular inflections. Furthermore, prepositions and pronouns are attached as an integral part of the word.

The first stage in text analysis is term extraction. We have defined a subset of Arabic characters in the Standard Unicode Table to be considered by the text analysis tool. The extracted terms are later stored in a data structure (array, hash table) which is called “term vocabulary”. We tend to make the vocabulary as small as possible to improve run-time efficiency and data-mining algorithms accuracy. This is achieved by normalization and stop word elimination, which are standard dimensionality reduction operations in information retrieval.

Our normalization process for Arabic is based on the following rules:

1. Normalize the initial Alif Hamza in the word to plain Alif.
2. Normalize Waw with Hamza to plain Waw.
3. Normalize the Alif Maksura to plain Ya.
4. Normalize the feminine ending, the Ta-Marbuta, to Ha.
5. Removal of Kashida (a calligraphic embellishment that has no associated meaning).
6. Removal of vowel marks (the short vowels: Fatha, Damma and Kasra).
7. Normalize original Arabic (“Hindi”) numerals to their Western (“Arabic”) counterparts.
8. Remove Shaddah, which is a consonant doubler.
9. Removal of certain letters (such as: Waw, Kaf, Ba, and Fa) appearing before the Arabic article THE (Alif + Lam).

Next, each term was compared to a list of pre-defined stop words containing several hundred terms. The list was compiled by an Arabic language expert. If the term was not found in that list, it was added to the vocabulary of terms, provided that this term was not already in the list.

Experimentation and Evaluation of Results

In order to evaluate our classification approach we used the C4.5 decision-tree classifier [18]. Decision tree models are widely used in machine learning and data mining, since they can be easily converted into a set of humanly readable if-then rules [7, 16]. The goal was to estimate classification accuracy and understand how it is affected by user-defined parameters such as document graph size N , t_{min} in case of the Naïve and CR_{min} in case of the Smart approach. We used graphs limited to 30, 40, 50 and 100 nodes in our experiments.

We used *ten fold cross validation* method to estimate classification accuracy. According to this method, the training set is randomly divided into ten parts with approximately equal number of items. Then a classification algorithm is executed ten times where each time one different part is used as a validation set and the other nine parts as the training set. The percentage of correctly classified documents is reported as the classification accuracy rate. Our experimental results for the Naïve and the Smart approach are presented in Figs. 5.4 and 5.5 respectively.

Using the Smart method, 100 nodes graph was found optimal bringing us almost 98.5% classification accuracy with the minimum classification rate CR_{min} value equal to 1.25. The resulting decision tree is shown in Fig. 5.6. The tree contains five binary attributes: four attributes representing single-node subgraphs (the words “The Zionist”

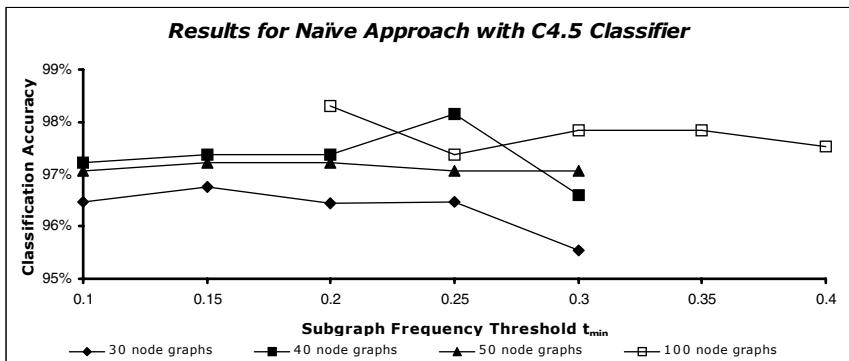


Fig. 5.4. C4.5 Classification accuracy for the Naïve Approach

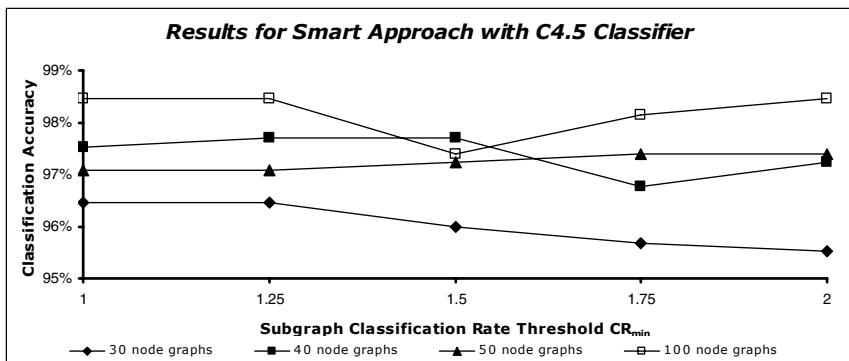


Fig. 5.5. C4.5 Classification accuracy for the Smart Approach

in two forms, “The martyr”, and “The enemy”) and one two-node subgraph (“Call [of] Al-Quds” in the document text, standing for the alias name of the Hamas web site). This simple decision tree can be easily interpreted as follows: if *at least one* of these five terms appears in an Arabic web document, it can be safely classified as “terrorist”. On the other hand, a document that contains *none* of these terms should be classified as “non-terrorist”. The decision tree induced by the Naïve approach from this corpus was nearly the same and, thus, we are not showing it here.

5.3.2 Case Study 2: Categorization of Terrorist Web Sites in English

About Document Collection

In this case study we try to identify the *source* of terrorist web documents. Our collection consists of 1,004 English documents obtained from the following two sources:

- 913 documents downloaded from a Hezbollah web site (<http://www.moqawama.org/english/>). These documents contain 19,864 distinct English words (after stemming and stopword removal).

- 91 documents downloaded from a Hamas web site (www.palestine-info.co.uk/am/publish/). These documents contain 10,431 distinct English words.

Both these organizations are located in the Middle East with Hezbollah based in Lebanon and Hamas operating from the Palestinian Authority territory. Making a distinction between the content provided by Hezbollah and Hamas is a non-trivial task, since these two Jihadi organizations are known to have close ties with each other resulting from their common cause and ideology.

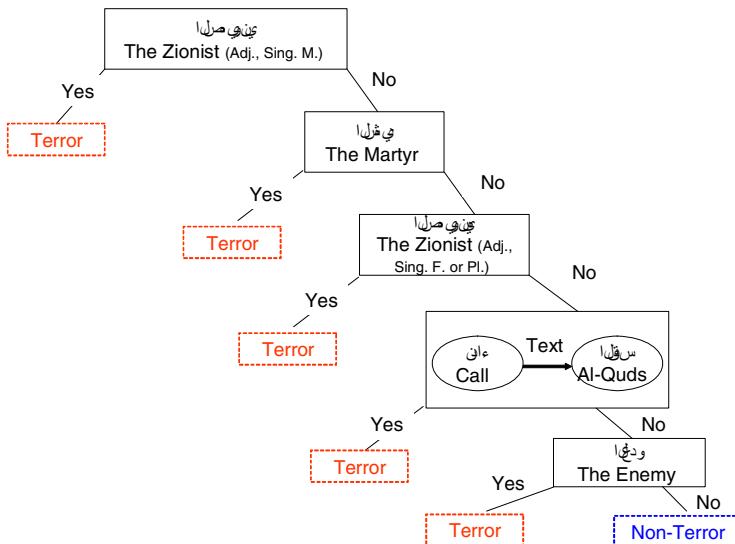


Fig. 5.6. C4.5 Decision Tree for Classification of Web Pages in Arabic

Experimentation and Evaluation of Results

In this case study we used the C4.5 decision-tree classifier [18] and the Hybrid Smart approach. The goal was to estimate the classification accuracy and the tree size as well as to understand how they are affected by the CR_{min} parameter. We used graphs limited to 100 nodes in our experiments. The *ten-fold cross validation* method was used to estimate classification accuracy. Our experimental results are presented in Fig. 5.7. It appears that $CR_{min} = 0.55$ provides the optimal trade-off between the classification accuracy (99.10) and the tree size (9 nodes only).

The resulting decision tree is shown in Fig. 5.8. The tree contains four binary attributes: two attributes representing single-node subgraphs (the words “Arab” and “PA” – Palestinian Authority) and two two-node subgraphs (a hyperlink to “Zionist Terrorism” and the expression “Holy Land” in the document text). This simple decision tree can be interpreted as follows: if *at least one* of these four terms appears in an English document coming from one of these two Web sites, it can be safely labeled as “Hamas”. On the other hand, a document that contains *none* of these terms should be labeled as “Hezbollah”.

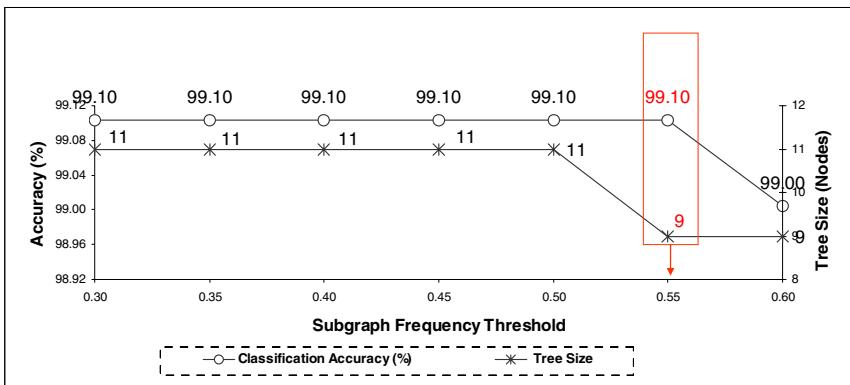


Fig. 5.7. C4.5 Classification Results for the Smart Approach with 100 nodes

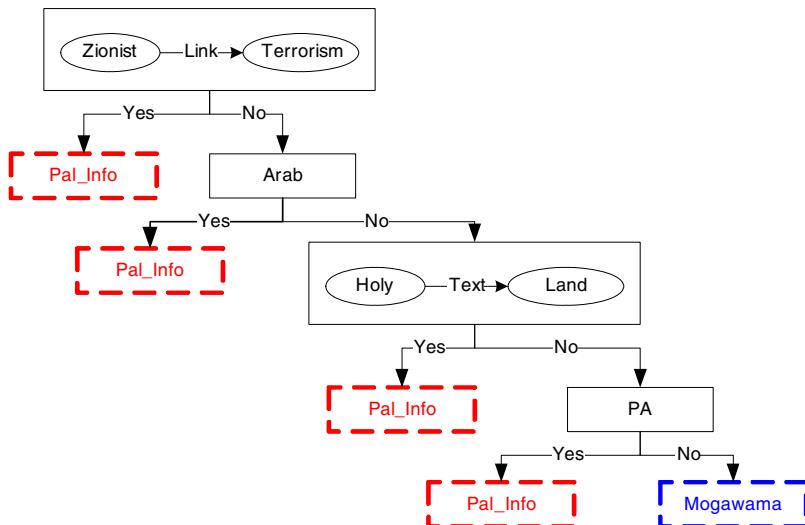


Fig. 5.8. C4.5 Decision Tree for Classification of Terrorist Web Pages in English

5.4 Conclusions

In this chapter we have presented a multi-lingual document classification methodology, which can help us to identify automatically terrorist content on the WWW. The proposed approach is utilizing the novel, graph-theoretic representation of web documents. The hybrid classification techniques were demonstrated on collections of real-world web documents in Arabic and English with the C4.5 classification algorithm. The results of initial case studies show that the hybrid document classification methods can be used for the fast and accurate detection of multi-lingual terrorist content on the Internet. Finding the optimal Graph Size N , the Minimal Subgraph Frequency

Threshold t_{min} and the Minimal Classification Rate Threshold CR_{min} is a subject for future research. Experimentation with other graph-based document representations and categorization algorithms can also be performed.

Acknowledgements

We are grateful to Dror Magal, an expert in Arabic, for his valuable help with analysis of Arabic web sites and to graduate research assistants Slava Kiselevich and Dani Alberg for their assistance with execution of experiments presented in this chapter.

References

1. Abbasi, A., Chen, H.: Applying Authorship Analysis to Arabic Web Content. In: Kantor, P., Muresan, G., Roberts, F., Zeng, D.D., Wang, F.-Y., Chen, H., Merkle, R.C. (eds.) ISI 2005. LNCS, vol. 3495, pp. 183–197. Springer, Heidelberg (2005)
2. Ahmed, A., James, C., David, F., William, O.: UCLIR: A Multilingual Information Retrieval tool. In: Proceedings of the Workshop on Multilingual Information Access and Natural Language Processing, pp. 89–96 (2002)
3. Aljayl, M., Frieder, O.: Effective Arabic-English Cross-Language Information Retrieval via Machine-Readable Dictionaries and Machine Translation. In: Tenth International Conference on Information and Knowledge Management (2001)
4. Corera, G.: Web Wise Terror Network. BBC NEWS: 2004/10/06 (2004), <http://news.bbc.co.uk/go/pr/fr/-/1/hi/world/3716908.stm>
5. della Sera, C.: (September 24, 2004)
6. Debat, A.: Al Qaeda's Web of Terror. ABC News (March 10, 2006)
7. Han, J., Kamber, M.: Data Mining Concepts and Techniques. Morgan Kaufmann, San Francisco (2001)
8. Harding, B.: 29 charged in Madrid train bombings, New York Times (April 11, 2006)
9. Kuramochi, M., Karypis, G.: An Efficient Algorithm for Discovering Frequent Subgraphs. IEEE Transactions on Knowledge and Data Engineering 16(9), 1038–1051 (2004)
10. Larkey, L.S., Ballesteros, L., Connell, M.E.: Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. In: Proceedings of the 25th Annual international ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2002, ACM Press, New York (2002)
11. Last, M., Markov, A., Kandel, A.: Multi-Lingual Detection of Terrorist Content on the Web. In: Chen, H., Wang, F.-Y., Yang, C.C., Zeng, D., Chau, M., Chang, K. (eds.) WISI 2006. LNCS, vol. 3917, pp. 16–30. Springer, Heidelberg (2006)
12. Lipton, E., Lichtblau, E.: Even Near Home, a New Front Is Opening in the Terror Battle. New York Times (September 23, 2004)
13. Lyall, S.: London Bombers Tied to Internet, Not Al Qaeda, Newspaper Says. New York Times (April 11, 2006)
14. Markov, A., Last, M.: A Simple, Structure-Sensitive Approach for Web Document Classification. In: Szczepaniak, P.S., Kacprzyk, J., Niewiadomski, A. (eds.) AWIC 2005. LNCS (LNAI), vol. 3528, pp. 293–298. Springer, Heidelberg (2005)
15. Markov, A., Last, M., Kandel, A.: Model-Based Classification of Web Documents Represented by Graphs. In: Proceedings of Web KDD 2006 Workshop on Knowledge Discovery on the Web at KDD 2006, pp. 31–38 (2006)

16. Mitchell, T.M.: Machine Learning. McGraw-Hill, Boston (1997)
17. Porter, M.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
18. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
19. Reid, E., Qin, J., Zhou, Y., Lai, G., Sagerman, M., Weimann, G., Chen, H.: Collecting and Analyzing the Presence of Terrorists on the Web: A Case Study of Jihad Websites. In: Kantor, P., Muresan, G., Roberts, F., Zeng, D.D., Wang, F.-Y., Chen, H., Merkle, R.C. (eds.) ISI 2005. LNCS, vol. 3495, pp. 402–411. Springer, Heidelberg (2005)
20. Ripplinger, B.: The Use of NLP Techniques in CLIR. In: Revised Papers from the Workshop of Cross-Language Evaluation Forum on Cross-Language Information Retrieval and Evaluation (2000)
21. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
22. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. Communications of the ACM 18(11), 613–620 (1971)
23. Schenker, A., Bunke, H., Last, M., Kandel, A.: Graph-Theoretic Techniques for Web Content Mining. Series in Machine Perception and Artificial Intelligence, vol. 62. World Scientific, Singapore (2005)
24. Schenker, A., Last, M., Bunke, H., Kandel, A.: Classification of Web Documents Using Graph Matching. International Journal of Pattern Recognition and Artificial Intelligence, Special Issue on Graph Matching in Computer Vision and Pattern Recognition 18(3), 475–496 (2004)
25. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(1) (March 2002), 1–47 (1999)
26. Talbot, D.: Terror's Server. Technology Review (2005),
http://www.technologyreview.com/articles/05/02/issue_feature_terror.asp
27. Yan, X., Han, J.: GSpan: Graph-Based Substructure Pattern Mining. In: IEEE International Conference on Data Mining (ICDM 2002) (2002)
28. Yang, Y., Slattery, S., Ghani, R.: A Study of Approaches to Hypertext Categorization. Journal of Intelligent Information Systems 18(2-3), 219–241 (2002)
29. Zhou, Y., Reid, E., Qin, J., Chen, H., Lai, G.: US Domestic Extremist Groups on the Web: Link and Content Analysis. IEEE Intelligent Systems, special issue on AI for Homeland Security 20(5), 44–51 (2005)

Online Resources

- Arabic Information Retrieval and Computational Linguistics Resources:
<http://www.glue.umd.edu/dlrg/clir/arabic.html>
- Cross-Language Information Retrieval Resources:
<http://www.ee.umd.edu/medlab/mlir/>
- Fighting Terror in Cyberspace Conference: <http://www.ise.bgu.ac.il/ftc/>
- Intelligence and Terrorism Information Center:
<http://www.terrorism-info.org.il>
- International Policy Institute for Counter-Terrorism, Interdisciplinary Center, Herzliya, Israel: [http://www.ict.org.il/](http://www.ict.org.il)
- Internet Haganah: <http://www.haganah.org.il/>

- Project for the Research of Islamist Movements (PRISM):
<http://www.e-prism.org/>
- SITE Institute: <http://www.siteinstitute.org/>

Questions for Discussions

1. Suggest alternative forms of graph-based representations for text and web documents in English.
2. Suggest a form of graph-based representation for a language, where word separation is not necessarily provided (such as German or Chinese).
3. Modify the FSG algorithm [9] for implementation with both Naïve and Smart sub-graph selection techniques.
4. Suggest pre-processing steps for detecting terrorist content in a language of your choice that is neither English nor Arabic. Discuss the main challenges associated with this language.
5. Suggest a methodological approach for detecting terrorist content in a multi-lingual document (e.g., containing both English and Arabic words).

Modeling Anticipatory Event Transitions

Qi He, Kuiyu Chang, and Ee-Peng Lim

School of Computer Engineering,
Nanyang Technological University, Singapore

Abstract. Major world events such as terrorist attacks, natural disasters, wars, etc. typically progress through various representative stages/states in time. For example, a volcano eruption could lead to earthquakes, tsunamis, aftershocks, evacuation, rescue efforts, international relief support, rebuilding, and resettlement, etc. By analyzing various types of catastrophic and historical events, we can derive corresponding event transition models to embed useful information at each state. The knowledge embedded in these models can be extremely valuable. For instance, a transition model of the 1918-1920 flu pandemic could be used for the planning and allocation of resources to decisively respond to future occurrences of similar outbreaks such as the SARS (severe acute respiratory syndrome) incident in 2003, and a future H5N1 bird-flu pandemic. In this chapter, we study the Anticipatory Event Detection (AED) framework for modeling a general event from online news articles. We analyze each news document using a combination of features including text content, term burstiness, and date/time stamp. Machine learning techniques such as classification, clustering, and natural language understanding are applied to extract the semantics embedded in each news article. Real world events are used to illustrate the effectiveness and practicality of our approach.

6.1 Introduction

Open Source Intelligence (OSI) plays a fundamental role in Intelligence and Security Informatics (ISI), accounting for as much as 80% of the overall intelligence. In fact, former US Joint Chiefs Chairman and former Secretary of State Colin Powell said: “I preferred the Early Bird with its compendium of newspaper stories to the President’s Daily Brief, the CIA’s capstone daily product”. Thus, the ability to constantly monitor and accurately track events from news sources all over the world is vital to ISI.

Major online portals like Google and Yahoo allows users to subscribe to news alerts by specifying a list of present/absent keywords to define a particular event that he or she is interested in. Unfortunately, current alert systems are not smart enough to figure out whether a news document containing all the user defined words positively actually confirms occurrence of the event. In fact, some service providers like Yahoo still entrust a human operator to approve system triggered news alerts, whereas others like Google prefer to use a completely automated approach, at the expense of generating many false alarms/alerts [9].

The *Anticipatory Event Detection* (AED) framework can uncover impending or anticipated events specified by a user. For example, it can be configured to monitor news streams for the occurrence of very specific events like “Taiwan declares

independence”, “Coup in Thailand”, “Osama bin Laden captured”, etc., which we called *anticipatory events* (AE). An AED news alert prototype has been previously reported by Chua, et al. [6].

One way to look at AED is to think of it as finding the transition between two adjacent events in an *event transition graph* (ETG). Events are represented by news articles reported before and after an *anticipatory event transition* (AET) has consummated [9, 23]. A user may only be interested in receiving a notification when a particular AET has fired, and not be bothered about the remaining AETs. If sufficient number of news articles can be collected for each of the events, it would be possible to detect any number of AETs. In order to learn a particular AET, a model will have to be trained to classify articles as occurring “before” or “after” the AET.

AED thus boils down to classifying sentences/documents into those that consume a predefined AE (hit) and those that do not. In this book chapter we present investigation of ETG modeling for AED, and also review some results on AED. The rest of this chapter is organized as follows. Sect. 6.2 surveys related work and compares AED to existing event detection tasks. In Sect. 6.3, we formally define the AED problem, types of AE detection, and subsequently propose the AED framework. We introduce various solutions for event representation suitable for AED in Sect. 6.4 and propose different classification approaches to learn the ETG in Sect. 6.5. Sect. 6.6 presents our experimental setup and results, and Sect. 6.7 concludes the chapter with a discussion of limitations and future work.

6.2 Related Work

AED falls under the broader family of problems collectively known as Topic Detection and Tracking (TDT), which hitherto includes New Event Detection (NED), Topic Tracking (TT), Retrospective Event Detection (RED), and Event Transition Graph (ETG) Modelling, etc. We shall examine each of these briefly in this section.

6.2.1 Topic and Event

The classical definition of *topic* from TDT 2004 is given as follows,

Definition 1. (Topic) *A topic is a seminal event or activity, along with all directly related events and activities.*

Note that a TDT topic has a much narrower scope than traditional IR topics or categories, and should be viewed more like a fine-grain news category. Likewise, a TDT *event* from TDT 2004 is defined as follows,

Definition 2. (Event) *An event refers to a particular incident occurring at a specific time and place, along with all necessary preconditions and unavoidable consequences.*

6.2.2 New Event Detection (NED)

NED, also known as *First Story Detection*, aims to detect the first story of a topic without reference to any seed news articles, i.e., it is unsupervised. The seminal paper

of Allan, et al. [1] empirically showed NED to be an inherently hard problem when tackled using only cosine similarity approaches. Later studies supported this viewpoint [4, 12, 18, 22], summarized below.

Brants, et al. [4] applied a combination of techniques to NED, including Hellinger distance, tiling for better document matching, etc., and reported modestly improved NED performance.

Kumaran, et al. [12] used text classification techniques and named entities to detect *all* new events of a particular category (using a model trained threshold, i.e., supervised). They reported a mixed bag of results for different categories both with and without using named entities. For example, using solely named entities resulted in better detection rates for the legal and science categories while on the other hand, excluding named entities helped the election and sports categories. Interestingly, named entities neither help nor worsened performances for the financial category. The last observation is in agreement with our earlier findings [9], which were evaluated primarily on financial news articles. Moreover, for our AED model, we found that combining named entity types with non-named entity terms worked better than each representation alone for financial category.

Stokes, et al. [18] proposed a composite document representation using both lexical chains and proper nouns for NED, which is a more sophisticated method of applying named entities to NED.

Yang, et al. [22] reported substantial performance gain by first classifying news articles into different topics, followed by applying one nearest neighbor to detect new events (NED). This approach makes intuitive sense since a similarity comparison within news events of the same topic works better than across the board. However, along with this new approach came two new problems, namely 1) accurately classifying a news article into one or more topics, and 2) setting a reliable outlier threshold for each topic. One of the main contributions of our work on AED is the bursty document representation, which helps improve the accuracy of document classification.

All in all, despite the numerous attempts to improve NED, none have yielded spectacular results so far. This is because NED is basically an ill-posed outlier detection problem where only one class of data is known before hand, thereby suffering from the fate that a new event may be too similar to an historical event, especially within the same topic. We believe that a performance breakthrough would require a supervised approach, one that involves higher order understanding of event domain semantics. In other words, unless domain knowledge is utilized, NED will remain a hard problem for the foreseeable future.

AED is not subject to the same problems faced by NED as it simultaneously define the topic and transition type that it should monitor. Since AED is well defined as a two state problem, documents of historically similarly events can be used to train it. As such, AED is a well-posed problem and therefore theoretically much simpler than NED.

6.2.3 Topic Tracking (TT)

Topic Tracking (TT) aims to monitor and identify news articles of a specific topic based on a few training stories.

Franz and McCarley [7] formulated TT as a NED problem by replacing the document-document similarity with the similarity between a document and a cluster centroid.

Carthy, et al. [5] benchmarked two different TT systems, one keyword-based and one using lexical chaining. They used lexical chaining to discover word co-references within a sentence, and found that it significantly outperformed keyword-based systems.

There is a concept of binary state for each AET; the anticipated transition can either take place or not. In general, TT will detect and return *all* new developments of a specific topic, whereas AED will detect and return any documents reported after the specified binary transition has fired. For example, on the topic of earthquakes, TT will detect *any* new developments pertaining to a specific earthquake. In contrast, AED will fire only when a state specified by the user has been reached. For example, the firing state could be “Earthquake strikes major Chinese city with heavy casualties”. In some ways AED can be considered as a special combination of NED and TT; its topic is constrained by keywords as in TT, and it tries to detect the first story after a fired transition (albeit properly defined by training documents instead of relying on outlier threshold) just like NED.

6.2.4 Retrospective Event Detection (RED)

Closely related to AED is RED, another NED derivative, which is concerned with detecting previously unidentified events from historical news corpus [21, 13].

Yang, et al. [21] first defined the RED problem and addressed it using document clustering. Li, et al. [13] attempted to identify events within a corpus of historical date-stamped news articles with the help of both time and content information. It assumes that the news event histogram of a particular event genre is Gaussian-distributed with peaks/bursts denoting a new event. This is related to our approach of using Kleinberg’s two-state automata [11] to represent term burstiness at different points in time.

The RED approach cannot be applied generally to solve the AED problem in practice since 1) it is constrained to detect generic events (such as *any* earthquake *anywhere*), and 2) it only works on historical events as it requires all (pre and post event) time information about the event in order to model it. Note that the second restriction also applies to our AED bursty model, which will be addressed in future work.

6.2.5 Event Transition Graph (ETG)

An Event Transition Graph (ETG), otherwise known as *Event Evolution Graph*, is a directed graph that models a set of events within a specific topic as nodes and edges. Specific states of the topic are represented by nodes, with the edges denoting possible transitions and associated firing conditions. Below we briefly review some previous work on ETG [14, 16, 23].

Makkonen [14] coined the term *event evolution* to denote the various time stages of a typical topic. In his work, an event is comprised of time-ordered related documents, and multiple events together constitute an event evolution graph.

Nallapati, et al. [16] defined *event threading* as the dependencies between events. They evaluated several candidate dependency graphs of clusters of news documents based on similarity, with the primary objective of analyzing the inherent structure of retrospective topics.

Yang, et al. [23] formally defined event evolution as the relationship between all events within a topic. Each relationship is in fact a transition that traverses in time from seminal events to terminal events. Their work depends heavily on the similarity metric between events, which are assumed to be comprised of well-clustered documents.

In our work, AED also assumes events to be a time-ordered sequence of documents. However, we make a simplifying assumption that a pre-existing ETG is readily available. We have yet to address the challenges of building an ETG from scratch. In principle, AED could use a combination of the above techniques to come up with a reliable ETG based on historically similar (to the current AE) events.

6.3 AED Model

6.3.1 Problem Definition

AED was originally motivated by the desire to deliver precise and customized SMS news alerts to the mobile phone subscribers [6]. As a push application with extremely high precision and recall demands, conventional keyword based alert systems simply did not cut it, and thus the birth of AED. The idea of AED is to allow a subscriber to receive only specific news alerts that he or she is interested in. A formal definition of AED is given as follows:

Definition 3. (AED) *The objective of AED is to detect and identify entities (messages or documents) that confirm the occurrence of a user specified anticipatory event transition (AET), which is also known as the user preference.*

The user preference is defined formally as follows.

Definition 4. (Anticipatory Event Transition (AET) or user preference) *A user preference or Anticipatory Event Transition (AET) corresponds to a single event transition selected from an event transition graph (ETG) of a given topic.*

Events belonging to the same topic (e.g., election of US President) often involve a common set of event transitions, e.g., nomination of party's Presidential candidates, nomination of party's Vice-Presidential candidates, election of party's Presidential team, election of Presidential team. These events collectively form an event transition graph, defined as follows.

Definition 5. (Event Transition Graph (ETG)) *An Event Transition Graph (ETG) G models multiple event transitions belonging to the same topic genre as a sequence of n events $E = [e_1, e_2, \dots, e_n]$ related by a set T of transition links between each pair of transitive events of the form*

$$T = \{t_{i,j} \mid \forall i, j \text{ if } \exists \text{ transition } e_i \text{ to } e_j, \text{ where } i < j \text{ and } e_i, e_j \in E\} \quad (6.1)$$

Thus, a user can select one amongst $|T|$ transitions as his AET or user preference.

6.3.2 AED on Document Streams

In practice, AED is usually applied to an online stream of news documents. Fig. 6.1 shows a global time-ordered sequence of documents, some on-topic and others off-topic with respect to an AET. Among the on-topic documents, only those that confirm the AET are considered hit documents, and should be identified by an AED system.

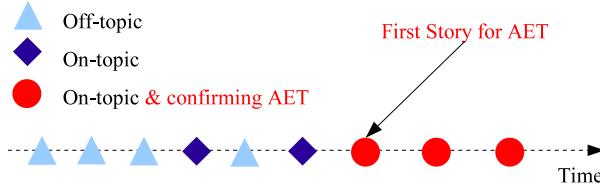


Fig. 6.1. AED for document streams

Ideally, a user should be allowed to specify any desired AET explicitly. However, this is not possible in practice due to the lack of a machine-understandable syntax to describe event semantics. One compromise is to present various known and trained ETGs to the user, from which he or she could pick a desired AET and specify the associated named entities. For example, if an ETG on the disposal of country leaders is available, a user could select from it an AET denoting resignation. The user should also supply a set of keywords such as “Taiwan President Chen Shui Bian” to indicate that he wishes to detect events confirming this particular resignation and not every resignation in the world.

Fig. 6.2 shows an ETG describing a common structure shared by all company acquisition topics. Suppose a user is interested in the event transition $t_{2,3}$ from event e_2 (“In talks to acquire”) to event e_3 (“Announces acquisition”). As multiple news articles could be associated with the “Announces acquisition” event, the earliest one will be the first story confirming the transition and is therefore the candidate document to be identified by an AED system.

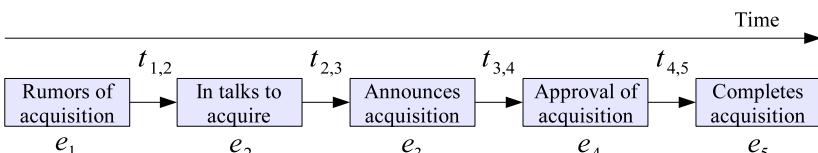


Fig. 6.2. An Event Transition Graph (ETG) for the “acquisition” topic genre

6.3.3 Types of AE Detection

Although the objective of AED is to detect the first story after the user specified AET, it may not always be successful in practice. Here, we formally define four types of AED scenarios.

Suppose we are given a set of N news articles $X = \{x_1, \dots, x_N\}$ about a topic, and a sequence of n events $E = [e_1, \dots, e_n]$ and its associated ETG. Each news article x_i has a

publication date/time represented by $t(x_i)$ and an event type in E represented by $e(x_i)$, the latter of which is also known as the true event of x_i .

We assume that all news articles in X are sorted in time ascending order, i.e., $t(x_i) = t(x_j) \forall i < j$, and all events in E are sorted in time ascending order, i.e., $t(e_i) = t(e_j) \forall i < j$.

By applying any AE detection technique on a news article x_i , we obtained its assigned event denoted by $s'(x_i)$. Given an anticipatory event transition $t_{k-1,k}$ as the user preference, the objective of AED is therefore to find the news article x_m that satisfies:

$$x_m = \arg \min \{t(x_i) \mid \forall x_i \text{ where } s'(x_i) = e_k\} \quad (6.2)$$

To make the time comparison easier between the detected first story x_m and the event e_k , we also define the *true time* of e_k , $t(e_k)$, as follows:

$$t(e_k) = \min \{t(x_i) \mid \forall x_i \text{ where } e(x_i) = e_k\} \quad (6.3)$$

Once the first story x_m of the anticipatory event e_k is determined by the AED classifier, all subsequent news articles, $x_j, j = (m+1), \dots, N$ will be assigned to event(s) e_k post $t_{k-1,k}$. Since the first story identified by AED may be premature, delayed, or undefined (never found), we define four types of AED scenarios as follows:

Accurate Alarm : $t(x_m) = t(e_k)$. First story of e_k found successfully.

Delayed Alarm : $t(x_m) > t(e_k)$. First story found was too late.

False Alarm : $t(x_m) < t(e_k)$. First story found was premature.

Miss : $t(x_m) = \text{undefined}$. No x_i in X has $s'(x_i) = e_k$. AED fails to even identify the event.

Fig. 6.3 graphically depicts each of the four types of AED scenarios. In practice, the preferred scenarios are ranked in descending order of preference as follows: accurate alarm, delayed alarm, false alarm, miss. Intuitively, a delayed alarm is preferred over a false alarm or a miss according to the age-old saying, “better late than never”.

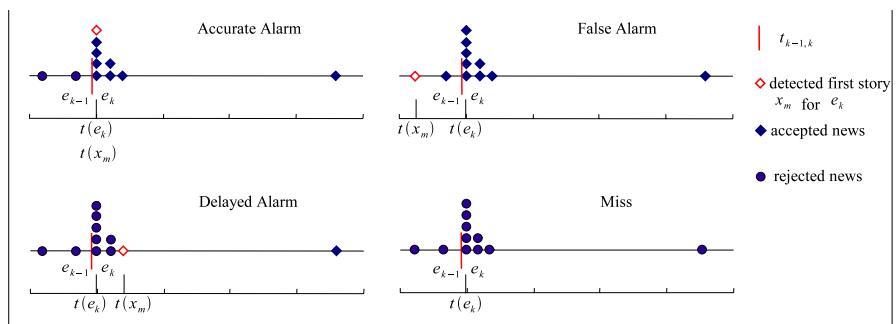


Fig. 6.3. The four AED scenarios

6.3.4 AED Prototype

Our AED prototype is shown in Fig. 6.4. Here, we make a number of simplifying assumptions.

- A reliable ETG is available, from which the user selects a desired AET and enters a set of related key words (usually named entities).
- The system retrieves a set of training documents based on the selected AET. The articles could be based on historically similar events.
- The training documents are manually annotated either at the sentence or document level as belonging to one of the two possible states of the AET, pre (-) or post (+).

Based on the above assumptions, the system trains a classifier for each AET in the ETG using the labelled/annotated documents. As long as two sets of training documents for both states of a transition are available, a classifier can be pre-trained in offline mode.

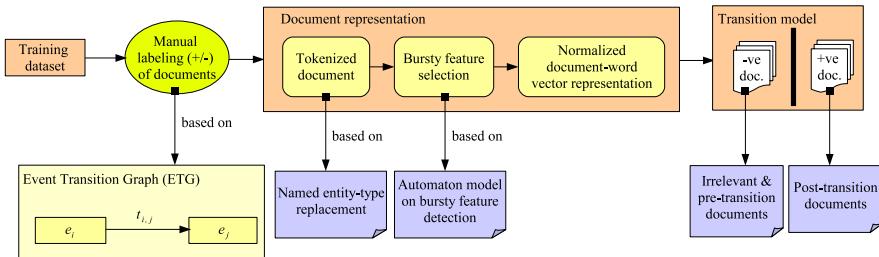


Fig. 6.4. Online AED system showing only the selected AET of the ETG

After training, the AED prototype operates online as follows:

1. The user inputs a set of keywords describing the desired event and selects the appropriate AET from an available ETG.
2. The system monitors an online news stream and filters off a set of candidate documents matching the user specified keywords.
3. The trained classifier corresponding to the user selected AET is applied to this candidate set, where each document is classified as negative or positive. Once a positive document is found, the AED system is deemed to have detected the anticipatory event.

6.4 Document Representation for AED

An essential portion of our AED system lies in the document representation format, as shown in Fig. 6.4. In this section, we describe various approaches to effectively represent event semantics, which can lead to better AED results.

6.4.1 Extracting Named Entities Types from Documents

In order to train a classifier on an AET using historically similar documents, it is very important to have the named entities replaced by named entity types [9]. For example,

consider the following statements referring to two different “announces acquisition” events, with the named entities in boldface:

“**China**’s biggest computer maker, **Lenovo Group**, said on **Wednesday** it has acquired a majority stake in **IBM Corp**’s personal computer business in a deal worth a total value of **US\$1.75 billion** (**\$2.86 billion**), one of the biggest **Chinese** overseas acquisitions ever.”

“**SBC Communications** on **Monday** announced plans to acquire **AT&T** in a **\$16 billion** deal, a move designed to bolster **SBC**’s sales to enterprise customers nationwide and give it new national and global networks.”

In order for the two statements to be representative of the “post” state of the transition, it is better to replace the named entities with their name entity types, as follows:

“**GPE**’s biggest computer maker, **ORGANIZATION**, said on **DATE** it has acquired a majority stake in **ORGANIZATION**’s personal computer business in a deal worth a total value of **MONEY (ORGANIZATION MONEY)**, one of the biggest **NATIONALITY** overseas acquisitions ever.”

“**ORGANIZATION** on **DATE** announced plans to acquire **ORGANIZATION** in a **MONEY** deal, a move designed to bolster **ORGANIZATION**’s sales to enterprise customers nationwide and give it new national and global networks.”

Clearly, after the replacement, the two examples become more similar to one other, which invariably helps the classifier learn the event transition better.

6.4.2 Factoring Burstiness into Document Representation

Motivation for Bursty Feature Representation

An up and coming topic is usually accompanied by a sharp rise in the reporting frequency of some distinctive features, known as “bursty features”. These bursty features could be used to more accurately portray the semantics of an evolving topic. Fig. 6.5 illustrates the effectiveness of using top bursty features to represent two separate topics. Had we used the usual feature selection and weighting scheme, the word features “Gingrich” and “Newt” frequent in both related but different topics would turn up nearly important for representing documents of these two topics.

Thus, the classical static Vector Space Model (VSM) [17] simply is not ideal in representing evolving trends in text streams, nor is it able to meaningfully model a transition from one semantic context to another. We therefore propose a new text

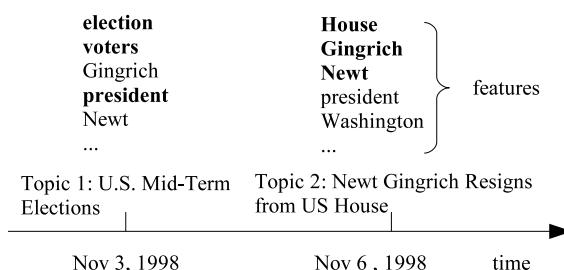


Fig. 6.5. Frequent features of two topics (bursty features shown in bold)

stream representation model, called bursty feature representation, which can emulate sophisticated temporal and topical behaviour via bursty features, as illustrated in Fig. 6.5. In our model, a burst corresponds to the definition as follows

Definition 6. (burst) A burst is a phenomenon in which a large amount of text content about the same topic is generated in a short time period.

Bursty Topic Representation

A bursty topic can be identified by modeling the document frequency (DF) with Kleinberg's two state automaton [11]. Likewise, the DF of every word can also be modeled, thereby uncovering bursty words. Fig. 6.6 shows an example of a bursty topic “Clinton’s Gaza Trip” from the TDT3 dataset.

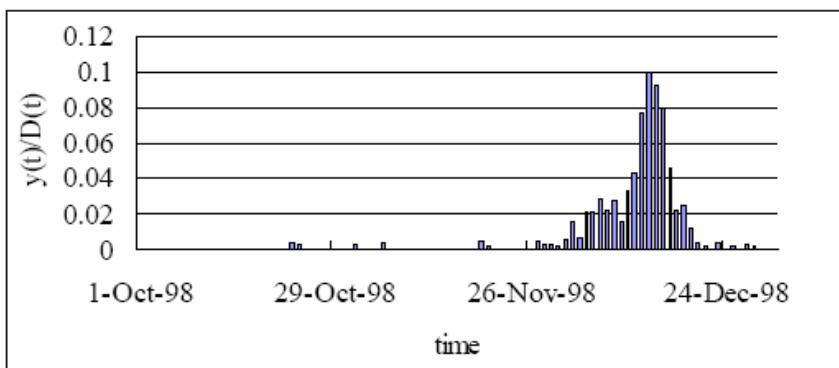


Fig. 6.6. A bursty topic (Clinton’s Gaza Trip) taken from TDT3, shown as a plot of the fraction of on-topic document frequency versus time.

Going a step further, we proposed representing a document with bursty features [10], which involves two steps: (1) identifying bursty features, and (2) representing documents using bursty features/weights.

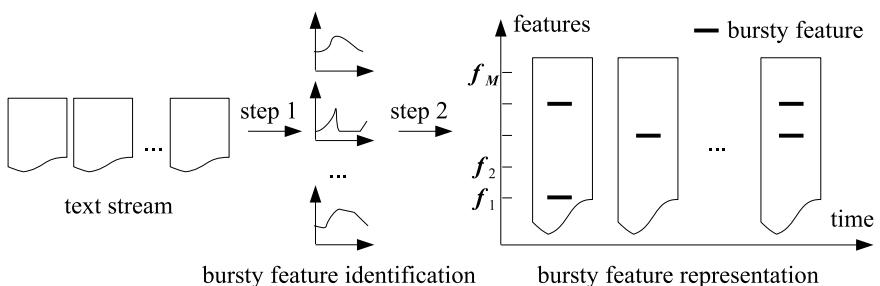


Fig. 6.7. An overview of bursty feature representation

Fig. 6.7 shows how a document is assigned bursty weights that are dependant on its time stamp t . The same raw document may have different bursty feature representations at two different time points $t_i \neq t_j$.

We now formally describe our bursty feature representation that combines burstiness with static feature weights [10]. Let $F = \{f_j; j = 1, 2, \dots, M\}$ be the static VSM feature space, and FP_{ij} indicates the static feature weight (i.e., binary weighting) of f_j in document \mathbf{d}_i . Let B represent the bursty feature space where $B \subseteq F$.

Definition 7. (Bursty Feature Representation) A document $\mathbf{d}_i(t)$ at time t has a bursty feature representation in the form

$$\mathbf{d}_i(t) = [d_{i1}(t), d_{i2}(t), \dots, d_{iM}(t)]^T \quad (6.4)$$

where

$$d_{ij}(t) = \begin{cases} FP_{ij} + \delta w_j & \text{if } f_j \in B \text{ and } t \in p_j \\ FP_{ij} & \text{otherwise} \end{cases} \quad (6.5)$$

where $\delta > 0$ is the burst coefficient, w_j is the bursty weight of a bursty feature f_j and p_i is the bursty period of f_i .

Here, the role of δ is to combine the sufficiency of the static VSM feature space with the discriminatory properties of bursty features. In other words, bursty features are enhanced or boosted by a factor of δw_j , whereas non-bursty documents will simply fall back to their static feature representation as explained in Fig. 6.8.

Under the general assumption that bursty features are representative and unique for each topic, we showed theoretically that the bursty feature representation will always improve the objective function of a clustering solution [10].

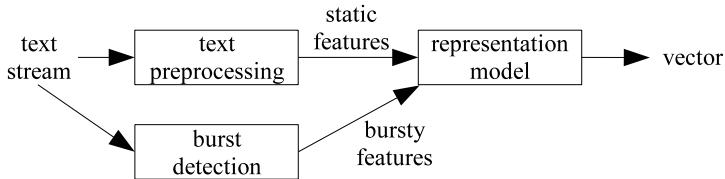


Fig. 6.8. Bursty feature representation

6.5 Modeling the AET

We have experimented with two different resolutions for training the transition model of Fig. 6.4, 1) sentence resolution, and 2) document resolution. Training at the sentence resolution is not only hard, but also extremely labour intensive because every sentence has to be manually annotated. In this section, we describe both approaches.

6.5.1 Sentence Classifier

A sentence classification model was initially built to model the AET [8]. We considered the sentence resolution based on the intuition that the most representative sentences from on-topic AET confirming articles typically provide a good summary of the transpired event transition.

In general, an event transition can be confirmed from a sentence, given enough contexts. For example, the following sentence would qualify as a “hit” sentence for the anticipatory event “win basketball match”.

1. Hit Sentence: “*The Knicks outscored Philadelphia 32-22 in the fourth quarter to secure the win.*”
2. User Preference (AET): “*win basketball match.*”

Single-Level and Two-Level SVM Sentence Classifiers

For the sentence classification model, we proposed a simple single-level support vector machine (SVM) sentence classifier and a more sophisticated two-level hierarchical SVM sentence classifier.

The single-level SVM sentence classifier simply classifies all sentences as either positive (i.e., on-topic and event confirming) or negative (i.e., on-topic but non-event confirming, and off-topic).

The two-level SVM classifier attempts to distinguish sentences about current events from those about historical events as these sentences could otherwise confuse the single-level sentence classifier that is also responsible for distinguishing on- and off-topic sentences. The sentences about current events and historical events are known as *positive* and *historical* sentences respectively. For example, “the rejuvenated Celtics have won three straight since then and six straight at home overall” is a typical historical sentence, which is considered as “on topic” by the single-level classifier but hard to identified as non-event confirming because the single level classifier is not trained to distinguish event confirming sentences from non-event confirming sentences. After applying two-level classification, this confusing case is easily solved.

Fig. 6.9 shows the structure of the two-level SVM classifier. The first level classifier aims to detect all on-topic sentences, which include both positive and historical

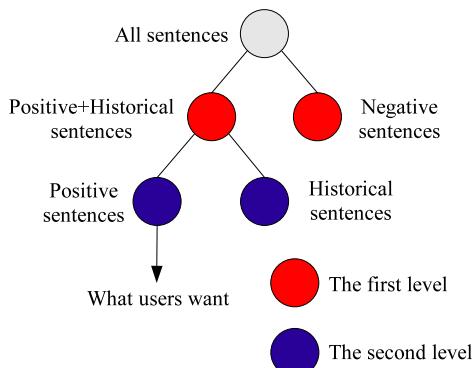


Fig. 6.9. 2-level SVM sentence classifier

sentences. The second level classifier performs a refinement on the on-topic sentences by further classifying them as positive or historical.

Sentence Classification Methods

We investigate various sentence retrieval strategies for AED, with a substantial focus on improving retrieval quality. In practice, the term weighting scheme used to represent a sentence vector has an enormous impact on the classification accuracy. The following methods using different term weighting schemes were compared in our experiments:

- Single-Level Classifier with {Standard TF, TFIDF, TFISF, TF+named entity features}
- Two-Level Classifier with {Standard TF, TFIDF, TFISF, TF+named entity features}

The standard TF scheme simply uses the raw frequency count of each term within a sentence. Another important factor to consider is the distribution of terms across a collection. Usually terms that are limited to a few sentences are useful for discriminating those sentences from the rest of the collection. This assumption leads to the introduction of ISF, called *inverse sentence frequency*. We also introduced the IDF, called *inverse document frequency*, at the sentence level to assume that terms appearing in a small number of documents are useful. The various term weighting schemes are summarized as follows:

$$\text{Standard TF} : f_{ij} \quad (6.6)$$

$$\text{TFIDF} : f_{ij} \times \log\left(\frac{N}{n_i}\right) \quad (6.7)$$

$$\text{IFISF} : f_{ij} \times \log\left(\frac{S}{s_i}\right) \quad (6.8)$$

where f_{ij} is the frequency of term i in sentence j , N is the total number of documents in the collection, S is the total number of sentences in the collection, n_i is the number of documents containing term i , and s_i is the number of sentences containing term i . Our proposed weighting scheme, TF with named entities is simply standard TF appended with some domain named entity features denoting the frequencies of them.

6.5.2 Document Classifier

Sentence retrieval is a very difficult problem [2] by itself. This is because a single sentence contains neither enough information (curse-of-dimensionality) nor context to form a meaningful model. Thus, we proposed modeling the AET transition at the document resolution [9].

Document Classification Methods

We tried three different feature representation methods and one classifier combining strategy to train the AET classifier, as follows:

<i>CONTENT</i>	: Entire news content as features.
<i>TITLE</i>	: Title as features.
<i>1SENT</i>	: First sentence as features.
<i>VOTING</i>	: Majority voting on the above three classifier outputs.

The TITLE and 1SENT representations were inspired by the observation that human experts can usually decide if a news is a hit simply based on its first sentence and/or title. Moreover, the TITLE and 1SENT representation of a news article may not always carry useful features, and the AED decision will have to fall back to the CONTENT representation. For example, the first sentence “*Signature Control Systems is off to a busy start in early 2006*” does not contain features really relevant to the “*acquisition*” event transition. VOTING was thus used as a simple and effective way to improve the overall accuracy.

6.6 Experiments

6.6.1 Dataset

Since AED is a relatively new area of research, we created three customized datasets, namely *basket100* to test the sentence AED model, *Google Acquisition* and *Acquisition7* to test the document AED model for the “mergers and acquisitions” topic genre.

To evaluate our bursty document representation, we created the *TDT3-Eng* set, which is comprised of documents from 116 topics extracted from the TDT3 collection.

TDT3-Eng Dataset

The TDT3 dataset includes 51,183 news articles collected during the 3 month period of October through December 1998. Among these, 37,526 English articles originated from 8 English sources, and 13,657 Chinese articles came from 3 Chinese sources. We extracted a subset of 8,458 on-topic English news articles covering 116 topics as *TDT3-Eng*.

After stop-word removal, 125,468 distinct features remained in *TDT3-Eng*. Among these, 2,646 were identified as bursty (set *B*) using the 2-state automaton model described in [10]. We independently selected another 2,646 features (set *F*) using the document frequency thresholding technique [20].

For a fair comparison, only bursty features in $F \cap B$ are used in our bursty feature representation. Finally we have 1,394 distinct bursty features ($|F \cap B| = 1,394$) with 1,863 bursts, averaging 1.34 bursts per bursty feature.

Basket100 Dataset

The *Basket100* collection comprises 100 documents returned by Google using the user preference “win basketball match”. In *Basket100*, 93 out of 100 documents are relevant, i.e., describes basketball games, and the remaining 7 are irrelevant. The collection contains 2,340 sentences, comprising 4,499 unique terms (words). The 2,340 sentences were manually annotated into 3 categories:

1. positive-current class for “current basketball result”
2. negative-historical class for “historical basketball results”
3. negative class for “irrelevant” or off-topic sentences

Table 6.1 shows the summary statistics for *Basket100*.

Table 6.1. Class distribution of Basket100

<i>Classes</i>	<i>Count</i>
Positive documents (class 1: win basketball event)	93
Negative documents (class 2: irrelevant)	7
Total	100
Positive sentences (class 1: current win basketball event)	189
Negative sentences (class 2: historical win basketball event)	117
Negative sentences (class 3: other irrelevant sentences)	2,034
Total	2,340

Google Acquisition Dataset

We would like to find a way of automatically retrieving the training dataset for event transition detection. Therefore, *Google Acquisition* dataset, which contains 346 as-it-happens news articles returned by Google News Alerts during the two-month period from Dec 19, 2005 to Feb 19, 2006 using the user preference “announces acquisition”, is created.

In *Google Acquisition*, 178 documents were labelled as positive and 168 as negative w.r.t the “announces acquisition” transition, which means that Google News Alerts returned 168 (48.6%) outright false alarms for the subscribed keywords “announces acquisition”. This is a typical result from a simplistic keyword-based news alert system.

Acquisition7 dataset

Another dataset, *Acquisition7*, which covers seven recent acquisition topics, was created as the test data for the document classification model. Each acquisition news topic in *Acquisition7* is comprised of 20 news articles returned by Google News, approximately half of each (10) were reported before and after “announces acquisition” transition.

The 7 acquisition news topics are listed in Table 6.2, where $t(e)$ refers to the true occurrence date for “announces acquisition” event.

6.6.2 Experimental Setup

Version 2 of the open source Lucene software was used to tokenize the news text content, remove stop words, and generate the document-word vector. In order to preserve time-sensitive past/present/future tenses of verbs, no stemming was done other than the removal of a few articles. SVM Cost factors [15] were used to deal with the highly unbalanced class sizes.

Table 6.2. Make up of the *Acquisition7* dataset.

Acquisition Topics	$t(e)$
Adobe acquires Macromedia	Apr 18, 2005
CNPC acquires PetroKazakhstan	Oct 26, 2005
eBay acquires Skype	Sep 12, 2005
Lenovo acquires IBM PC Division	Dec 08, 2004
Oracle acquires PeopleSoft	Dec 13, 2004
Oracle acquires Siebel	Sep 12, 2005
SBC acquires AT&T	Jan 31, 2005

In our experiments, we use BBN's Identifinder [3] to identify 24 types of named entities, including *Animal*, *Contact info*, *Disease*, *Event*, *Facility*, *Game*, *Geo-political entities*, *Language*, *Law*, *Location*, *Nationality*, *Organization*, *Person*, *Plant*, *Product*, *Substance*, *Work of art*, *Date*, *Time*, *Cardinal*, *Money*, *Ordinal*, *Percentages*, and *Quantity*. Extracted named entities are then replaced in line by one of the 24 named entity types.

6.6.3 Clustering TDT3-Eng

We applied K-means ($K = 116$) clustering to **TDT3-Eng**, which comprises 116 topics or classes. Since bursty features are identified based on *TDT3-Eng* itself, the bursty coefficient δ is set to 1 as suggested in [10].

Assume that K clusters are generated. Let $|k_j|_{C_i}$ denote the number of documents from topic C_i assigned to cluster k_j . Similarly, let $|C_i|_{k_j}$ denote the number of documents from cluster k_j originating from class C_i .

We evaluated our clustering results using three standard metrics: cluster purity, cluster entropy, and class entropy defined as follows:

Definition 8. (Purity) The purity of cluster k_j is defined by

$$purity(k_j) = \frac{1}{|k_j|} \max_i (|k_j|_{C_i}) \quad (6.9)$$

The overall purity of a clustering solution is expressed as a weighted sum of individual cluster purities

$$\text{cluster purity} = \sum_{j=1}^K \frac{|k_j|}{|D|} purity(k_j) = \frac{1}{|D|} \sum_{j=1}^K \max_i |k_j|_{C_i} \quad (6.10)$$

Definition 9. (Cluster Entropy) Cluster entropy measures the diversity of a cluster k_j and is defined as

$$\text{entropy}(k_j) = -\sum_i \frac{|k_j|_{C_i}}{|k_j|} \log \frac{|k_j|_{C_i}}{|k_j|} \quad (6.11)$$

The total entropy of a cluster solution is

$$\text{cluster entropy} = \sum_{j=1}^K \frac{|k_j|}{|D|} \text{entropy}(k_j) \quad (6.12)$$

Both cluster purity and entropy measure the homogeneity of a cluster, but neither of them measures the recall of each topic. Thus, we introduce class entropy as follows

Definition 10. (Class Entropy) The class entropy of a cluster is defined as:

$$\text{entropy}(C_i) = -\sum_j \frac{|C_i|_{k_j}}{|C_i|} \log \frac{|C_i|_{k_j}}{|C_i|} \quad (6.13)$$

The total class entropy of a cluster solution is

$$\text{class entropy} = \sum_{i=1}^K \frac{|C_i|}{|D|} \text{entropy}(C_i) \quad (6.14)$$

In general, a good clustering algorithm should have high cluster purity, low cluster entropy, and low class entropy.

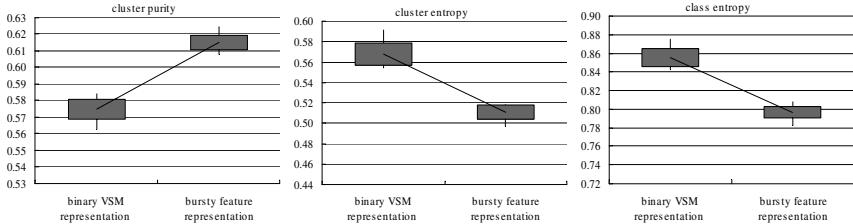


Fig. 6.10. Averaged clustering results for *TDT3-Eng* over 10 runs, showing the mean (end points of line joining the two box plots), spread (box), and range (vertical line).

Table 6.3 lists the 3 evaluation metrics averaged over 10 clustering runs for the binary VSM and bursty feature representations. The metrics are also plotted in Fig. 6.10, which shows the mean, spread (standard deviation) in each direction, and range.

Table 6.3. Averaged clustering results for *TDT3-Eng* over 10 runs

representation	cluster purity	cluster entropy	class entropy
binary VSM	0.5750	0.5682	0.8553
bursty feature	0.6149	0.5110	0.7971
Improvement	6.93%	10.06%	6.81%

From Table 6.3, we see that bursty feature produces clusters with on average 10.06% and 6.81% lower cluster and class entropies, respectively, and 6.93% higher cluster purity. Fig. 6.10 further highlights that bursty feature representation yields more consistent and stable clustering solutions with lower variance and smaller range in the three metrics.

The results are very encouraging considering that 1) many of the topics in *TDT3-Eng* are small (with just a few documents) and non-bursty, and 2) there is a fair amount of overlap in bursty feature space between the various topics, which clearly violated the non-overlapping assumption.

6.6.4 Modeling Transitions at the Sentence Resolution

Two classifiers using various term weighting schemes of the sentence model were applied to the *Basket100* dataset, with the goal of detecting a winning basketball event. We evaluate the sentence classification performance using the standard precision, recall and F-Measure metrics defined as:

$$Precision = \frac{\# \text{correct positive predictions}}{\# \text{positive predictions}} \quad (6.15)$$

$$Recall = \frac{\# \text{correct positive predictions}}{\# \text{positive samples}} \quad (6.16)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (6.17)$$

Single-level SVM Sentence Classifier

Fig. 6.11 shows the classification results of the single-level SVM. We see that the sentence classifier using our proposed weighting scheme yielded the best F-Measure of 0.69, leading the next competitor by 15%. Moreover, the recall of 0.63 is low by practical standards, despite it beating the nearest competitor (TF) by more than 20%.

The other methods fared significantly worse. TFISF performed worse than TF, probably due to the fact that there were too many negative (including historical) sentences, thereby distorting the ISF. Note that for single-level classification, the positive and historical winnings are labelled differently, despite them sharing a common vocabulary, e.g., “win”, “loss”, etc. TFIDF performed the worst, due to the large discrepancies between the importance of a term at the sentence and document level.

Two-level SVM Sentence Classifier

Fig. 6.12 shows the results of the two-level classifier. Since the first level classifier is only responsible for distinguishing on-topic sentences from off-topic ones, its performance was measured based on all on-topic sentences which included historical sentences.

Figs. 6.12(a)–(b) show that the precision values at both levels were not affected much by the different weighting schemes, unlike with the single-level classifier. This

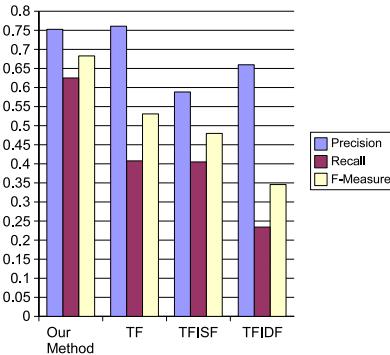
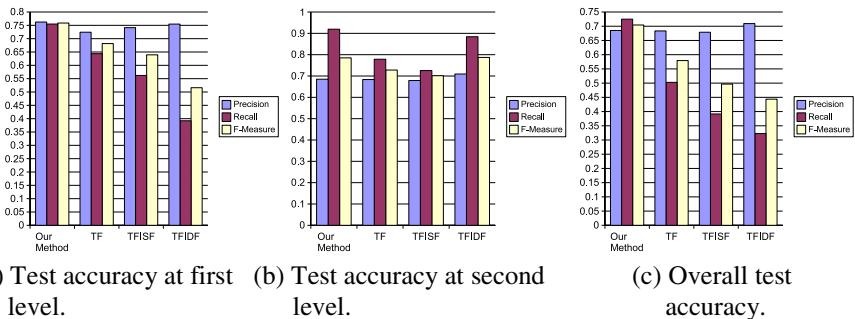


Fig. 6.11. Cross validated (10-fold) results of single-level SVM classifier

confirmed our previous suspicion that the similarity between positive and historical sentences was a large contributing factor to the low precision when inverse document and sentence frequencies come into play for the single-level classifier. The overall performances of the two-level classifier is shown in Fig. 6.12(c), with our method achieving the overall best result of 0.69 precision and 0.72 recall.



(a) Test accuracy at first level. (b) Test accuracy at second level. (c) Overall test accuracy.

Fig. 6.12. Cross validated (10-fold) results of two-level SVM classifier

6.6.5 Modeling Transitions at the Document Resolution

Before we test our trained transition model, we need to evaluate its raw cross-validated performance, to make sure it is a decent model. Afterwhich, we evaluated its AET performance on a given set of unseen AE by tallying the total number of *false alarms*, *delayed alarms*, *accurate alarms*, and *misses*.

Validating Google Acquisition Dataset

In order to validate the generic “announces acquisition” trained model, we conducted two-fold cross-validated experiments using the four text classification approaches of Sect. 6.5.2 on the *Google Acquisition* dataset. The dataset is first split along the timeline into two equal parts: 1) news articles dating from Dec 19, 2005 to Jan 19, 2006,

Table 6.4. Average test results on *Google Acquisition*. Best results are shown in bold.

Average	CONTENT	TITLE	ISEN	VOTING
False Alarms	22.5	15.5	17	13.5
Misses	9	24.5	15	10
Precision	0.7847	0.8110	0.8172	0.8571
Recall	0.9011	0.7308	0.8352	0.8901
F1	0.8389	0.7688	0.8261	0.8733

and 2) news articles dating from Jan 20, 2006 to Feb 19, 2006. One part was used for training with the other part used for testing and vice-versa.

The significance of this experiment shown in Table 6.4 is that it increased the precision of Google's returned news alerts from 51.4% to 85.7%, a more than 33% improvement! Furthermore, the high precision and recall figures confirmed that the *Google Acquisition* dataset is indeed suitable for modelling the transition into the “announces acquisition” event.

Testing the AET Model on Acquisition7 dataset

In this section, we test the generic AED classifier trained by *Google Acquisition* on the *Acquisition7* dataset. One AED outcome is shown in Fig. 6.13. Note that once the “first” story of “announces acquisition” event has been identified by AED, all subsequent news articles are labelled “positive”.

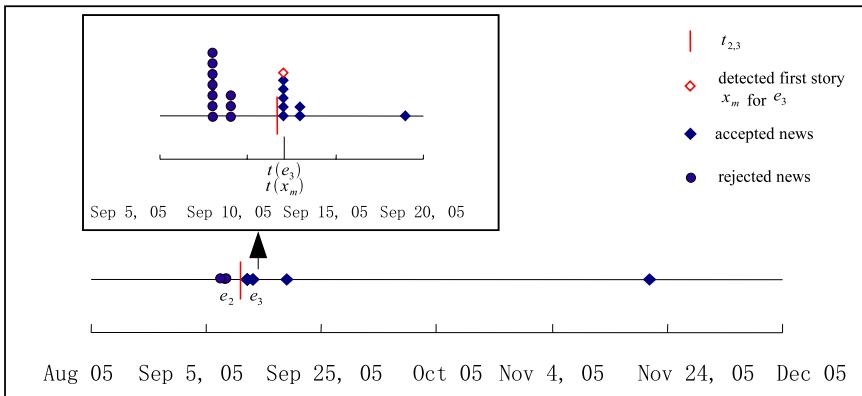


Fig. 6.13. Online AED of “eBay acquires Skype” found an accurate alarm, $t(x_m) = t(e_3)$. $transition_{2,3}$ is the “announces acquisition” transition in Fig. 6.2.

Table 6.5 gives a summary of the overall performances, which shows that AED based on the VOTING method generated 4 accurate alarms, 1 delayed alarm, 2 false alarms, and 0 misses. This means that the model trained by *Google Acquisition* was able to cover the main characteristics of all 7 acquisition topics. The results shown here is markedly better than methods based on cosine similarity, which failed for all except one of the 7 events [9].

Table 6.5. AED results on *Acquisition7* using the VOTING method

Alarms:	Accurate	Delayed	False	Miss
Adobe acquires Macromedia	✓			
CNPC acquires PetroKazakhstan	✓			
eBay acquires Skype	✓			
Lenova acquires IBM PC Division			✓	
Oracle acquires PeopleSoft			✓	
Oracle acquires Siebel	✓			
SBC acquires AT&T			✓	

6.7 Conclusions and Future Work

6.7.1 Conclusions

We proposed a new practical application called Anticipatory Event Detection (AED), which is a more refined and personalized form of event tracking and detection. We then investigated suitable document presentation and various classification methods to tackle the AED problem, with a substantial focus on improving the AED transition models, which were verified experimentally on restricted domains.

AED has an essential application in ISI news alerts system, which can help information analysts to monitor only relevant events. The holy grail of AED is to detect any number of AE transitions of arbitrary genres. This is akin to having a live assistant constantly scanning newsfeed monitoring a set of AEs. Our current contributions have achieved the goal of verifying the feasibility of training AE transitional models for homogenous future events, and investigated the burstiness nature of representative features for important events by improving the traditional IR document representation.

The main limitation of AED lies in its reliance on a pre-trained transition model for every user-specified anticipatory event. This means that in practice, a user is not allowed to specify any anticipatory event, but instead must choose from a list of available pre-trained anticipatory event transitions and ETGs, e.g., terrorist bombing, earthquake disaster, mergers and acquisitions, sports scores, etc. The flip-side of this is that accurate ETG can be built for topics that matter the most to ISI analysts, and extremely accurate detection rates can be achieved.

6.7.2 Future Work

For the foreseeable future, we envisage a real-time feedback AED system as a testbed for conducting experiments on different AED methods, by allowing a subscriber to refine his/her anticipatory event definition using similar historical events. For example, to define an anticipatory event such as “China attacks Taiwan”, the user can specify a similar transpired event like “Iraq invades Kuwait”, and manually supply the set of “pre” and “post” documents of the historical event, from which the AED system can learn the transition.

Secondly, we would like to introduce outlier detection for an event transition along with more sophisticated distribution of high-dimensional free text, semi-automatically

build ETGs for different topic types by applying clustering, and further investigate the burstiness properties of important events. With the above improvements, the AED system could very well become a truly reliable and personalized alert system that anyone can put to good practical use.

References

1. Allan, J., Lavrenko, V., Jin, H.: First story detection in TDT is hard. In: CIKM 2000, pp. 374–381 (2000)
2. Allan, J., Wade, C., Bolivar, A.: Retrieval and Novelty Detection at the Sentence Level. In: SIGIR 2003, pp. 314–321 (2003)
3. Bikel, D.M., Schwartz, R., Weischedel, R.M.: An algorithm that learns what's in a name. *Machine Learning* 34(1-3), 211–231 (1999)
4. Brants, T., Chen, F., Farahat, A.: A system for New Event Detection. In: SIGIR 2003, pp. 330–337 (2003)
5. Carthy, J.: Lexical Chains for Topic Tracking. PhD thesis, Department of Computer Science, National University of Dublin (2002)
6. Chua, K., Ong, W.S., He, Q., Chang, K., Kek, A.: Intelligent Portal for Event-triggered SMS Alerts. In: IEE Mobility (2005)
7. Franz, M., Ward, T., McCarley, J.S., Zhu, W.J.: Unsupervised and supervised clustering for topic tracking. In: SIGIR 2001, pp. 310–317 (2001)
8. He, Q., Chang, K., Lim, E.P.: Anticipatory Event Detection via Sentence Classification. In: IEEE SMC 2006, pp. 1143–1148 (2006)
9. He, Q., Chang, K., Lim, E.P.: A Model for Anticipatory Event Detection. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 168–181. Springer, Heidelberg (2006)
10. He, Q., Chang, K., Lim, E.P.: Bursty Feature Representation for Clustering Text Streams. In: SIAM Data Mining 2007 (2007)
11. Kleinberg, J.: Bursty and Hierarchical structure in streams. In: SIGKDD 2002, pp. 91–101 (2002)
12. Kumaran, G., Allan, J.: Text classification and named entities for new event detection. In: SIGIR 2004, pp. 297–304 (2004)
13. Li, Z.W., Wang, B., Li, M.J., Ma, W.Y.: A probabilistic model for retrospective news event detection. In: SIGIR 2005, pp. 106–113 (2005)
14. Makkonen, J.: Investigations on Event Evolution in TDT. In: HLT-NAACL 2003, pp. 43–48 (2003)
15. Morik, K., Brockhausen, P., Joachimss, T.: Combining statistical learning with a knowledge-based approach – a case study in intensive care monitoring. In: ICML 1999, pp. 268–277 (1999)
16. Nallapati, R., Feng, A., Peng, F., Allan, J.: Event Threading within News Topics. In: CIKM 2004, pp. 446–453 (2004)
17. Salton, G., Buckley, C.: Term-weighting Approaches in Automatic Text Retrieval. *Information Processing and Management* 24, 513–523 (1988)
18. Stokes, N., Carthy, J.: Combining semantic and syntactic document classifiers to improve first story detection. In: SIGIR 2001, pp. 424–425 (2001)
19. TDT04, TDT: Annotation Manual Version 1.2, August 4 (2004), <http://www.ldc.upenn.edu/Projects/TDT2004>

20. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: ICML 1997, pp. 412–420 (1997)
21. Yang, Y., Pierce, T., Carbonell, J.: A study on retrospective and on-line event detection. In: SIGIR 1998, pp. 28–36 (1998)
22. Yang, Y., Zhang, J., Carbonell, J., Jin, C.: Topic-conditioned Novelty Detection. In: SIGKDD 2002, pp. 688–693 (2002)
23. Yang, C.C., Shi, X.D.: Discovering event evolution graphs from newswires. In: WWW 2006, pp. 945–946 (2006)

Online Resources

1. Open Source Intelligence in Wikipedia:
http://en.wikipedia.org/wiki/open_source_intelligence
2. Google News with News Alerts subscription:
<http://news.google.com>
3. TDT: Topic Detection and Tracking Research:
<http://www.nist.gov/speech/tests/tdt/index.htm>
4. TDT 2004: Annotation Manual Version 1.2, August 4 2004:
<http://www.ldc.upenn.edu/Projects/TDT2004>
5. TDT3: Topic Detection and Tracking Dataset 3:
<http://projects.ldc.upenn.edu/TDT3>
6. Support Vector Machines for classifying text:
<http://svmlight.joachims.org>
7. Apache Lucene-Core 2.0.0: <http://lucene.apache.org>

Questions for Discussions

1. Does text summarization help AED?
2. Is it possible to represent a document-word vector using only bursty features for AED? If yes, how? If no, why?
3. What is the major difference between AED and an event driven search engine?
4. Why do we need to train the transition model for an AE? Could we simply use online unsupervised clustering?
5. How can we automatically train a ETG for certain topic genres? Does similarity comparison support multiple outcomes from one seminal event?
6. How can we statistically select a threshold to differentiate bursty feature from normal features?
7. (Scenario study) The CIA would like to monitor an underground extremist run discussion forum for any new terrorist attack plans. What should be the first step if CIA deploys an AED system for this?

Exploring Gray Web Forums: Analysis and Investigation of Forum-Based Communities in Taiwan

Jau-Hwang Wang¹, Tianjun Fu², Hong-Ming Lin¹, and Hsinchun Chen²

¹Central Police University, Taiwan

² University of Arizona, USA

jwang@mail.cpu.edu.tw

Abstract. Our society is in a state of transformation toward a “virtual society.” However, due to the nature of anonymity and less observability, internet activities have become more diverse and obscure. As a result, unscrupulous individuals or criminals may exploit the internet as a channel for their illegal activities to avoid the apprehension by law enforcement officials. This paper examines the “Gray Web Forums” in Taiwan. We study their characteristics and develop an analysis framework for assisting investigations on forum communities. Based on the statistical data collected from online forums, we found that the relationship between a posting and its responses is highly correlated to the forum nature. In addition, hot threads extracted based on posting activity and our proposed metric can be used to assist analysts in identifying illegal or inappropriate contents. Furthermore, a member’s role and his/her activities in a virtual community can be identified by member level analysis. In addition, two schemes based on content analysis were also developed to search for illegal information items in gray forums. The experiment results show that hot threads are correlated to illegal information items, but the retrieval effectiveness can be significantly improved by search schemes based on content analysis.

Keywords: Forum-based Community, Gray Web Forum, Forum Analysis & Investigation, Crime Lead Discovering, Content Analysis.

7.1 Introduction

Since the introduction of the first electronic computer in 1946, computers and their storage devices created a new capability to process and store information in digital format. This trend was further accelerated by the introduction of computer networks in 1969 and the World Wide Web (WWW) in 1992. Nowadays, computers and computer networks are everywhere and used in every facet of modern society. Computer networks are not only used as a tool for processing information, but also has become a new medium to share and access information online. For example, more than 2.2 billion messages were sent and received daily in the US, more than 3 billion indexed web pages on the world wide web, and more than 550 billion documents are available on-line [7]. Furthermore, bulletin board systems, internet relay chat systems, and I-phone systems, are all integrated within the WWW and provide various communication channels for individuals to exchange information beyond the limits of time and space. Consequently, our society is in a state of transformation toward a “virtual

society.” The culture which once was based mainly on physical interactions is transforming toward one where people’s daily activities, such as shopping, getting services, and especially sharing information, can be accomplished without face-to-face contact with others.

Although the internet has enabled global businesses to flourish, it has also become one of the major channels for criminal activities. Nowadays, the internet allows criminals to make acquaintance of and acquire their victims, and eventually commit crimes. For example, just a few years ago, a National Taipei University student was found dead and wrapped in a luggage box dumped on a street corner by his “net friend,” whom he met on a homosexual forum online. Today, many teenagers continue making friends through online activities, such as exchanging e-mails and playing internet video games, without having any physical interactions. The lack of physical interactions leaves little or no observable trails for their parents to make inferences and thus become less informed on their children’s social activities. Just recently, two teenagers in Taiwan who got acquainted through an internet chat room committed suicide together. The breaking news astonished both the two families as well as the Taiwan society. In both cases, law enforcement agencies could only remain passive in their investigations. This situation will get worse as more sex predators and cyber-stalkers exploit the internet to acquire their victims.

The recent advance of computer forensics research has shown that online activities do leave electronic trails, only that it is relatively easy to modify, highly volatile, more difficult to access, and much harder to find and recover [12]. For examples, bulletin board messages are stored in system archives, and e-mail messages are either stored in the mail server or client computers. Although the archives in private storage can be acquired for analysis only under proper authorization, the information stored in the public archives can be retrieved for analysis if necessary. After the 911 disaster, the FBI has shifted the reactive or post crime investigation paradigm to proactive investigation [9]. Thus, precrime investigation and data analysis are of critical importance to mitigate the negative effects of online activities. Although the collection and analysis of the “dark web¹” sites have been under intensive investigations ([16, 3, 6, 10, 14]), few researches addressed the issues of deriving crime leads or detecting symptoms of crimes from online archives. Wang, et al., conducted a study to gather illegal web sites, such as pornography and drug dealing using special searching mechanisms. They used domain experts to identify a collection of typical pornography and drug web pages, and extracted a list of vocabularies to further search the web for potential illegal web pages [13]. Dringus, et al., used data mining to discover and build alternative representations for asynchronous education forums. Temporal participation indicators are extracted to improve the instructor’s ability to evaluate the progress of a threaded discussion [5]. However, no research has been done on gathering crime leads from forum archives. This paper examines the “Gray Web Forums” in Taiwan. We develop an analysis framework for assisting analysis and investigation on the “gray” forum communities.

The organization of this paper is as follows: Sect. 7.2 reviews the literature and describes forum-based communities and their characteristics. Sect. 7.3 introduces the

¹ Refers to the internet used by terrorist or extremist groups for communication, recruiting, and propaganda.

Gray Web Forum concept and its implication and challenges on precrime investigation and analysis. Sect. 7.4 describes the framework for exploring the Gray Web Forums. Sect. 7.5 presents two content analysis search schemes for crime investigation on gray forums. Sect. 7.6 gives conclusions and the future work of this research.

7.2 Forum-Based Communities

A forum is defined as the *computer-mediated medium for the open discussion of subjects of public interest* [11]. Registered forum users or members basically have three types of operations:

1. View the existing postings.
2. Reply to an existing posting.
3. Start a new topic (also called a thread) of discussion.

“Postings” are messages that are sent to the forum for public viewing by members. Forum members may respond to an existing posting or create a new topic for discussion. A group of postings that are related to the same topic of discussion is called a “thread.” Users who do not register are called guests and are usually only allowed to view existing postings. The registration procedure usually requires the user to submit some form of identifications, such as a valid e-mail address. Forums can be classified into two categories: ‘public’ or ‘private’. Public forums can be viewed by anyone from any computer with internet access. However, private forums can only be accessed by members with proper authorization. Access to private forums is usually restricted by usernames and passwords. More detailed descriptions about online forums can be found in [11]. However, more sophisticated mechanisms can be used to control the access to forums. For example, besides a username and a password, some forums only allow a user to view more postings after the user has responded to a topic by typing or by selecting a prefabricated message. Furthermore, some forums may give credits for every single posting sent by members, and only after a member has accumulated enough credits and reached some privilege level can he/she be allowed to view postings of higher privilege. Early forums are mainly text based, i.e., postings are composed solely in text or a combination of symbols. However, recent forum software tools allowed users to upload multimedia objects as attachments of postings. Forum software is a set of computer programs, which provides the functionalities, such as registration, message posting, and grouping, etc, which are needed for forum administration and operation. There are many commercially available forum software packages as well as freeware. For a more detailed survey on forum software, see [15].

According to Wally Bock, communities are characterized by three things: common interests, frequent interaction, and identification [2]. Community members are not only looking for information, they are also looking for affiliation, support, and affirmation [4]. Internet forum members discuss on a certain topic or related topics, seek support from members, exchange information or opinions by posting messages, and are identified usually by e-mail address before being able to post messages. Therefore internet forums are perfect platforms to provide the necessary mechanisms for the formation of communities or virtual communities. However, due to the nature of anonymity and the lack of observability, internet activities have similar characteristics to

“night culture,” “which is characterized by...sparse and homogeneous population; chances for escape and opportunity; a wider range of tolerated behavior than in day-time life; ...decentralization of authority; lawlessness and peril...” [8]. Furthermore, as stated by Barlow: “Cyberspace, in its present condition, has a lot in common with the 19th Century West.... It is, of course, a perfect breeding ground for both outlaws and...” [1]. As a result, unscrupulous individuals or criminals may exploit the internet forum as a channel for their illegal activities to avoid the apprehension by law enforcement officials.

7.3 Gray Web Forum-Based Communities

The Gray Web Forum-based virtual communities are defined as: *communities formed through internet forums, which focused on topics that might potentially encourage biased, offensive, or disruptive behaviors and may disturb the society or threaten the public safety*. Such forums may include topics such as pornography, pirated CDs, suicide discussions, gambling, spiritualism, and so on. For example, a forum devoted to discussions on pornography may involve attachments of illegal or offensive contents or images; members of pirated CD forums may share music CDs or software without proper licensing; gambling forum members may provide hyperlinks to online gambling web sites, which is illegal in Taiwan; gang forums may spread information to its members and to recruit new members, which will result in issues of public safety; forums devoted to discussing spiritualism may teach teenagers misguided value systems and encourage disruptive behaviors (e.g., several teenagers who committed suicide together in Taiwan were traced back to chat rooms in such forums).

Investigations on Gray Web Forums are difficult due to the following reasons. Firstly, except the top level web pages, most forum postings are not indexed by major search engines, which make the internet search engine unsuitable for detection. Secondly, internet forums are highly distributed and the huge quantity of forums prohibits effective coverage by manual investigations. Thirdly, a thread of discussion may cause numerous postings or sub-threads and the duration of each thread can be very long, these make manual investigations very time consuming and thus highly inefficient. Finally, the forum access control mechanisms adopted by a forum may also become a major obstacle for effective and efficient investigation.

New research is necessary to better understand the Gray Web Forums in order to develop better methodologies or tools to assist in gathering and investigating illegal contents and potential crime leads. This research developed a framework for analyzing Gray Web Forums. We study web sites from Taiwan and identify several main types of Gray Web Forums as examples for the framework.

7.4 A Framework for Gray Web Forum Analysis

7.4.1 Research Design

The framework for Gray Web Forum analysis mainly consists of four components, as shown in Fig. 7.1:

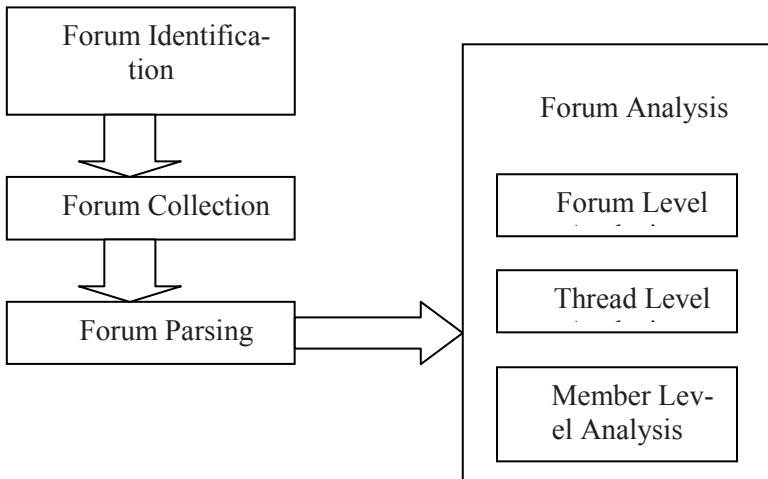


Fig. 7.1. The Gray Web Forum Analysis Framework

1. Forum Identification is used to identify Gray Web Forums based on the knowledge of domain experts. However, since many forums contain multiple boards, it is also important to filter out boards that are unrelated to Gray Web Forum-based communities.
2. Forum Collection is to spider the web pages in the forums or boards that have been identified. It is also necessary to take steps against forum access control mechanisms. For example, if messages in a forum could only be viewed by its members, the spider should be directed to use cookies with authorized user ID to access the forum; if the forum does not allow the opening of many links at the same time, the number of spider threads would also be limited.
3. Forum Parsing is needed to extract useful content, such as the number of threads, the number of forum members, the duration and distribution of threads, and the frequency as well as the volume of online postings, which are used for further analysis.
4. Forum Analysis can be further divided into three levels of analysis: forum level, thread level, and member level. Forum level analysis is to provide a broad overview for the forums; thread level analysis is to identify *hot or influential threads* for further investigation; and member level analysis is to segment the forum members into different categories and identify their roles in virtual communities.

In our analysis forum members can be further classified as *initiators*, *active members*, *followers*, and *visitors*, based on their activity data. An initiator is one who initiates a thread of discussion and has a large volume of postings at the beginning stage of the thread duration, but the frequency and volume of postings may decrease after the initial stage. Initiators may become active members if they continue to participate frequently and contribute many postings. Active members are those who have both

high frequency of participation and a large volume of postings. On the other hand, followers have high frequency of participation but a small volume of postings. Finally, visitors are those who have low frequency of participation and a small volume of postings. Based on posting statistics the forum members can be segmented into four clusters as shown in Fig. 7.2.

Such information is sometimes critical in crime investigation. For example, in terms of attributing responsibility for illegal activities, the active members should be charged with the major responsibility; while the initiators may only have to share a part of the responsibility. On the other hand, followers are most likely the victims and visitors are usually unrelated third parties and should not be charged.

In the following sections, selected Gray Web Forums in Taiwan as identified by selected domain experts (Taiwan public safety personnel) are used as examples for the analysis framework.

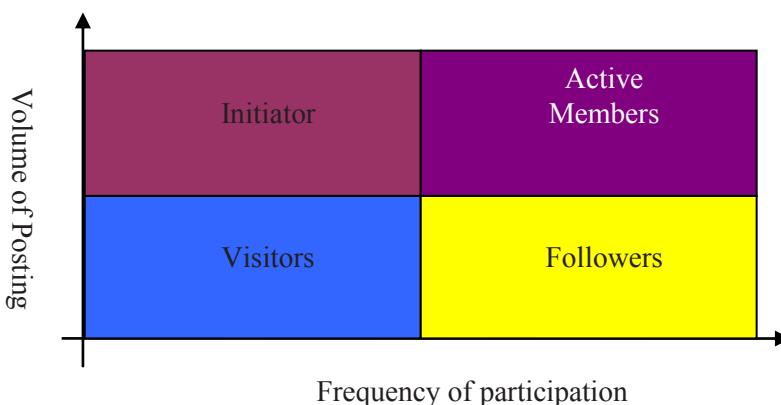


Fig. 7.2. Categories of Forum Community Members

7.4.2 Taiwan Gray Web Forum: Forum Level Analysis

Several representative types of the Taiwan Gray Web Forum identified are shown in Table 7.1.

Forum 1 is a platform that provides its members with illegal software and pirated CDs. Forum 2 releases gambling information. Forum 3 allows its members to share and discuss sentimental essays, some of which may encourage biased and disruptive behaviors. Forum 4 consists of threads discussing superstitious subjects. The results of forum level analysis provide an overview of each forum and are shown in Table 7.2.

Table 7.2 reveals several interesting features of each forum. Both Forums 1 and 2 adopt similar access control mechanisms, which only allow members to view the contents of a posting after they have replied a message to the posting. The first postings of many threads in these two forums often contain URL links to illegal software and pirated CDs or important gambling information, which cannot be viewed unless a viewer replies to them. Therefore, their members have to post short and simple message, such as “I want to see it,” “Thanks for sharing the information,” in order to view the information. Although both Forums 1 and 2 have a high post-per-thread value, the

Table 7.1. Selected Gray Web Forums in Taiwan

FID*	URLs	Type
1	http://bbs.a35.info/thread.php?fid=486	Pirated CD
2	http://oie.idv.tw/cgi-bin/bbs/forum.cgi?forum=1	Gambling
3	http://www.525.idv.tw/bbs/cgi-bin/forum.cgi?forum=9	Sentimentalism
4	http://www.helzone.com/vbb/forumdisplay.php?f=38	Spiritualism

*Forum Identification Number.

Table 7.2. Overview of the Gray Web Forum Collection

FID	Type	Threads	Postings	Size(Mb)	Members
1	Pirated CDs	252	7749	32.2	3013
2	Gambling	515	31128	292	539
3	Sentimentalism	1396	4452	62.8	463
4	Spiritualism	434	2415	41.4	228

average number of postings per member in Forum 1 is 2.6 (7749/3013), which is much less than 56.5 (31128/539) in Forum 2. The reason for this difference is that the members of these two forums are quite different. Gamblers do not give up any chance to make money so that they often reply to as many threads as possible; while people who search for pirated CDs or illegal software only reply to threads of interest.

Forum 3 has 1396 threads but each thread only contains 3.2 (4452/1396) postings on average. This may suggest that there are fewer discussions in that forum. On the contrary, the average number of postings for each thread in Forum 2 is 60.4 (31128/515). Again, this may be because gamblers tend to reply to many threads to gather more gambling information.

7.4.3 Taiwan Gray Web Forum: Thread Level Analysis

One of the major goals of our analysis is to identify *hot* or *influential* threads for further investigations. *Hot threads* are defined as threads which statistically have longer length of duration, more forum members, more active members, and more forum postings in numbers and volumes. The number of active members is a good indicator of the intensiveness of discussion. Based on this definition, we developed the following metric to calculate scores for every thread in each forum. The higher the thread score, the “hotter” the thread is.

$$\text{Thread Score} = F_{\text{norm}}(N_p) \times F_{\text{norm}}(V_t) \times F_{\text{norm}}(D_t) \times F_{\text{norm}}(N_{\text{am}}) \times F_{\text{norm}}(N_m) \quad (7.1)$$

Where N_p is the number of postings in a thread, V_t is the volume of postings in a thread, D_t is the duration of a thread, N_{am} is number of active members in a thread, and N_m is the number of members who have postings in a thread. The function $F_{\text{norm}}()$ is used to normalize each variable to a range of [0,1]. Note that in this case study, we classify members who have more than one posting in a thread as active members. The

hottest thread from each forum and their statistical information and topics are shown in Table 7.3.

Among these hot threads, Thread 4 has the potential to attract young depressed teenagers or students, who are contemplating suicide. Some robbery cases have been reported among people who were involved in “network love,” the hottest thread of Forum 3. We believe such “hot thread analysis” can be used to assist analysts in identifying illegal or inappropriate contents or activities. The hottest threads of Forum 1 and Forum 2 are both protected by forum access control mechanisms.

7.4.4 Taiwan Gray Web Forum Member Level Analysis

A further analysis of members can also help understand Gray Web Forums. Table 7.4 provides the percentage of members who have created new threads (i.e., initiators).

Table 7.3. The Hottest Threads from Each of the Four Forums

TID*	Type	Postings	Volume (char)	Duration (day)	Members	Active Members
1	Pirated CDs	469	7219	69	462	7
2	Gambling	211	4132	4**	186	25
3	Sentimentalism	91	9628	118	16	6
4	Spiritualism	88	5295	962	67	8
	TID Topics					
1	Pornography websites recommendation (reply needed to view the content)					
2	Lottery recommendation (made by an experienced member)					
3	A true story about “network love”					
4	What will you do if you know you will die tomorrow?					

* Thread Identification Number.

**4 days is in fact a long time for a typical thread in the gambling forum.

Table 7.4. Percentage of Members Who Have Created New Threads

FID	Type	Number of threads members created						
		0	1	2-9	10-29	30-49	50-99	>=100
1	Pirated CDs	99.3%	0.3%	0.2%	0.1%	<0.1%	0%	0%
2	Gambling	88.5%	4.5%	3.9%	2.6%	0.4%	0.2%	0%
3	Sentimentalism	10.6%	40.6%	44.1%	3.2%	0.9%	0.6%	0%
4	Spiritualism	77.6%	11.4%	7.9%	2.2%	0.4%	0%	0.4%

Table 7.4 reveals great differences among the four types of Gray Web Forums. The percentages of members who have never created a new thread in Forum 1 (99.3%) and Forum 2 (88.5%) are higher than those of the other two forums. It is because only a small portion of members in Forum 1 and Forum 2 have valuable information or

resources, such as URLs to pirated CDs or lottery analysis results. Therefore, most members in gambling forums and pirated CD forums are only information receivers or followers. On the contrary, members in sentimentalism and spiritualism forums tend to publish their own thoughts or raise questions. Consequently the percentage of posting initiators is higher. The table also shows that the percentage of Forum 3 has a unique distribution. Most forum members are able and willing to share their deep feelings with strangers online. Besides the fact that sentimentalism is a more common topic, the internet forums are also becoming another popular venue for people to share their feelings with strangers because of the anonymous nature of the internet. Forum 4 is similar to Forums 1 and 2; however, the percentages of members who have created 1 thread (11.4%) and who have created 2-9 threads in Forum 4 (7.9%) are much higher.

7.5 Crime Investigation on Forum-Based Communities

Although we were able to learn more about forum communities through forum level, thread level, and member level analysis based on the forum statistics analysis as discussed in Sect. 7.4, we are still far from being able to effectively deduce clues for crime investigation. The hot topics may indeed contain illegal items, but it is also quite possible that a hot thread becomes popular only because its contents being attractive. According to our study on <http://bbs.a35.info/>, a forum devoted to software sharing and consists of 1068 topic postings² (downloaded on April 18, 2006), the top 375 hot threads³ contain 157 suspicious topic postings⁴, which is 40.2% in term of precision. The analysis shows that hot threads are indeed related to suspicious activities. However, it is still not very effective to investigate gray forums based on the hot thread analysis. As a result, better analytical techniques are needed to assist law enforcement officials in order to investigate forums efficiently as well as effectively, i.e., the solution should be able to assist law enforcement officials to gather related information quickly and to pinpoint to threads containing illegal items as well. Although the hot thread analysis gave some interesting results in online forum investigation, the analysis based on the statistics of forum postings did not use semantic information of the data. We developed two search schemes based on content analysis techniques to extract inappropriate information from gray forums. Experiments were also conducted to evaluate the effectiveness of the proposed search schemes.

7.5.1 Thread Analysis Using Full Text Search Based on Key Phrases Identified by Domain Expert

In the first experiment the top 375 hot topic postings from <http://bbs.a35.info/> were collected and six key phrases were identified by the domain expert, which are SN,

² A thread consists of a group of postings that are related to the same topic of discussion. The posting which defines the topic is called “topic posting”. Usually the topic posting is the first posting (head) of a thread.

³ Due to time limit, only the top 375 hot topic postings were analyzed.

⁴ The suspicious postings were identified by the domain expert, one of MIS students at Central Police University, who has been trained and spent more than six months in studying the gray forums.

序號(SN),註冊(registration),破解(break),驗證(authentication), and key. These key phrases were then used to retrieve topic postings which have one or more of these key phrases using the full text matching algorithm from the collection. 121 postings were retrieved and classified as suspicious. Further manual analysis on the top 375 hot topic postings and the 121 retrieved postings shows that the recall⁵ is 63.7% (100/157), the precision is 82.6% (100/121), the false positive rate is 17.3% (21/121), the false negative rate is 36.3% (57/157), and the F-measure⁶ is 71.9. The false positive is due to the following two factors: firstly, although some postings did explicitly use some of these key phrases, the discussions are mainly of question asking and answering activities and in some cases there is a negation ahead of these key phrases, such as “有破解版嗎？” and “不含破解版”；secondly, a few response postings may use some of the six key phrases, but their contents are not related to any illegal item. The factors contribute to the high false negative rate are: firstly, some postings contain only image or video data of inappropriate items, or use rare terms such as “正式版”(usually “破解” is used to represent “破解正式版”); secondly, some postings were about specific illegal items and they never explicitly used any of these six key phrases, but their titles have illegal content; thirdly, some postings are related to illegal contents but they do not contain any of the six phrases, due to typing errors, such as “註冊”->“注冊”. Note that the 82.6% precision is much better than that (40.2%) of the hot thread analysis.

7.5.2 Threads Analysis Based on Automatic Key Phrase Extraction and Vector Matching

The second study downloaded 100 topic postings from p2p101.com, a forum devoted to software sharing. Since most responses of this forum are short, only topic postings were used for the experiment. The types of items shared in this software sharing forum include: Freeware, Shareware, Trial Version, and Proprietary Software. The first three types are considered to be legal, while the last one is illegal. The types of postings in the forum can be divided into two categories: *requesting for information items* and *offering information items*, or simply *question asking* and *question answering*. Although question asking and answering are major activities in such type of forums, many in times forum members may request for or upload an information item to share with other members. However, if an information item protected by the *Intelligence Property Act* is uploaded, it may become a legal issue if a complaint is filed against the case. The forum investigation problem is portrayed in Fig. 7.3.

The search problem for illegal postings could be solved if a set of protected items are available. For items protected by the Intelligence Property Act, such sets do exist

⁵ In the area of crime investigation, only those classified as suspicious are of interest. So the recall is calculated by divided the number of suspicious postings retrieved by the total number of suspicious postings in the collection and the precision is the number of suspicious postings retrieved divided by the total number of retrieved postings.

⁶ $F = \frac{2 \times P \times R}{P + R} \times 100\%$, where P refers to the precision, and R refers to the recall. In this study, precision and recall are considered to be equally important.

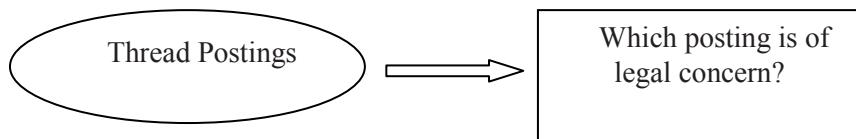


Fig. 7.3. Crime Investigation on Forums

and can be downloaded from the web, such as the PC Home web site (http://ec2.pchome.com.tw/exh/00000134/main_list.htm) or web site of Business Software Alliance (www.bsa.org). We devised a scheme for gathering illegal forum postings based on key word extraction and vector matching techniques. The match search scheme is depicted in Fig. 7.4.

Although postings may also contain multimedia contents, only text contents were analyzed in this experiment. The development of multimedia analyzing techniques is left as the future work of this work. In this experiment 100 topic postings from p2p101.com were downloaded and 35 of them were identified to be illegal by manual analysis. Phrasing was carried out by submitting the texts of each posting to an online phrasing tool, which has accuracy of 98%, provided by the Academia Sinica⁷, Taiwan. The TF-IDF formula⁸ was then used to weight each phrase and extract key phrases for each posting. Terms with frequency less than 2 and terms with weight below the 25% percentile of each posting were removed to achieve high F-measure. The key phrases extracted are used to index postings, each of which is represented by a term vector consisting of a set of key phrases. The vectors of each posting are matched with the term vector derived from the list of proprietary software items using the Vector Space Model. A non-zero match score is considered as match and the corresponding posting is classified as illegal. Among the 100 postings, 41 of them were classified as illegal. Further manual analysis shows that the recall is 91.43%⁹ (32/35) with precision at 78.5% (32/41), and the F-measure is 84.21%. The factors affecting the recall are: firstly, some illegal items are presented in images instead of text; secondly, the spelling or translation of some software items of some topic postings is not consistent with the proprietary software listing. The low precision rate is due to the fact that some postings only contain discussions on proprietary software, however there are no sharing activities involved.

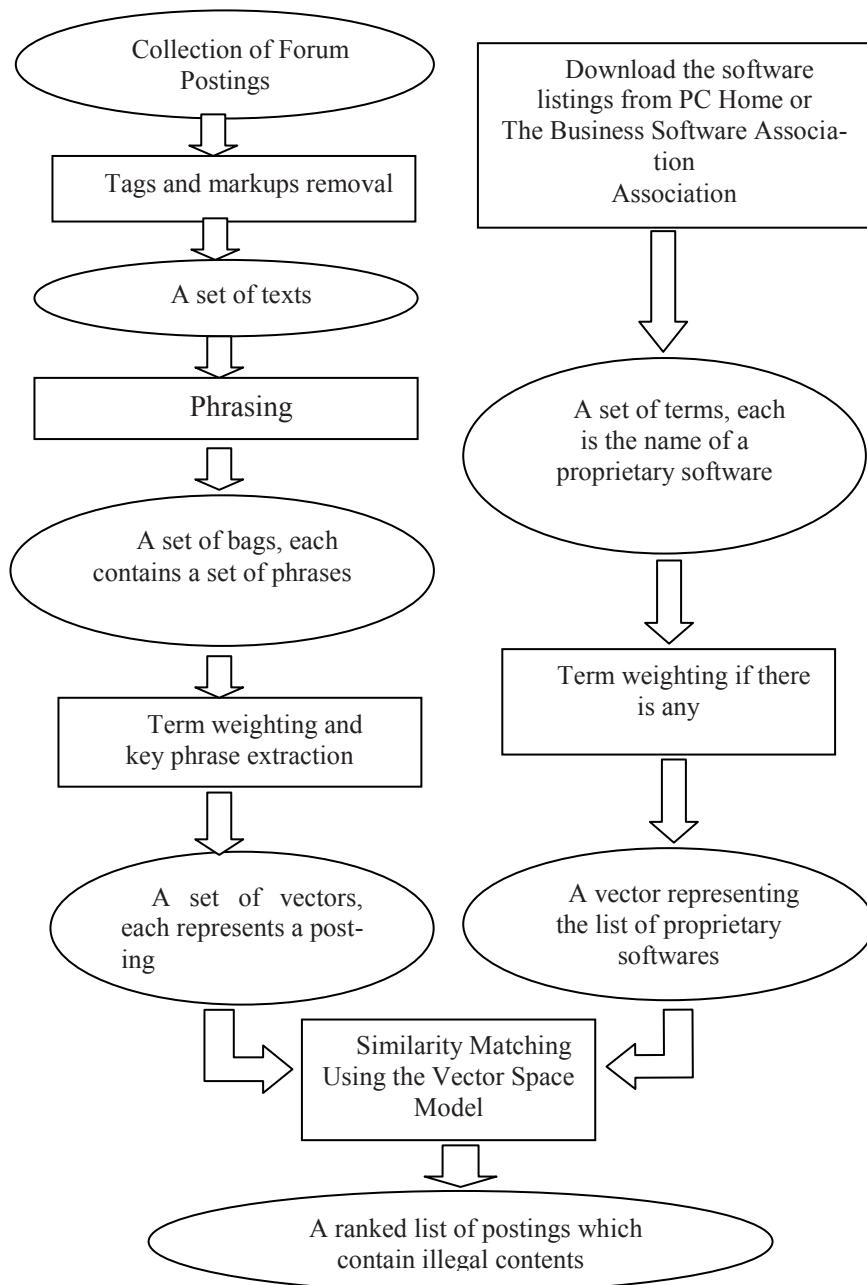
7.6 Conclusions and Future Work

This paper introduced the concept of Gray Web Forums. We developed a framework to analyze Gray Web Forums in three different levels: forum level, thread level, and member level. Based on the posting data collected from the selected Gray Web Forums, we were able to identify some interesting results. The relationship between a

⁷ <http://ckipsvr.iis.sinica.edu.tw/>.

⁸ $W_{ij} = (n_j / n_{all}) \times \log_2 (N/df_j)$, where W_{ij} is the weight of term j in document i , n_j is the frequency of term j in document i , n_{all} is the number of key phrases in document i , N is the total of documents, and df_j is the number of documents containing term j .

⁹ Some illegal postings were missed because their illegal contents were contained in images.

**Fig. 7.4.** An Information Retrieval Model for Forum Investigation

posting and its responses is highly correlated to the forum nature. For example, gambling forum members reply to as many threads as possible to gather gambling information; while people who search for pirated CDs or illegal software reply only to related threads of interest. Hot threads extracted based on posting activity and our proposed metric can be used to assist in manual analysis to further identify illegal or inappropriate contents. In addition, member level analysis can be used to identify a member's role and his/her activities in a virtual community.

We also developed two search schemes based on content analysis techniques for assisting investigation on illegal items in information sharing forums. The experiments show that hot threads are indeed related to illegal information items, in particular about 40.3% in precision in our case study. However, the thread analysis using full text search based on key phrases identified by domain expert can significantly improve the retrieval effectiveness and the performance is much better than that of the hot thread analysis based on forum statistics. Furthermore, the performance of the thread analysis based on automatic key phrase extraction and vector matching is also very promising. However, to being able to pinpoint specific threads or postings for illegal or offensive contents remains challenging in general. Firstly, a list of illegal information items may not be available and it may be difficult to assign a standard set of search phrases for other gray forums. Secondly, analyzing techniques for multimedia contents are also needed to be developed in order to broaden the scope of content analysis and to improve the recall of the search. In light of the importance and the need of precrime investigation in Gray Web Forums, more efforts are still needed in this research area.

References

1. Barlow, J.P.: Crime and puzzlement. Whole Earth Review, 45–57 (1990)
2. Bock, W.: Christmas, Communities, and Cyberspace (2001),
<http://www.bockinfo.com/docs/community.htm>
3. Chen, H.: The Terrorism Knowledge Portal: Advanced Methodologies for Collecting and Analyzing Information from the Dark Web and Terrorism Research Resources, Sandia National Laboratories, August 14 (2003)
4. Donath, J.S.: Inhabiting the Virtual City: The design of social environments for electronic communities. MIT, Cambridge (1996)
5. Dringus, L.P., Ellis, T.: Using Data Mining as a Strategy for Assessing Asynchronous Discussion Forums. Computer & Education 45, 141–160 (2004)
6. Elison, W.: Netwar: Studying Rebels on the Internet. The Social Studies 91, 127–131 (2000)
7. Marcella, A.J., Greenfield, R.S.: Cyber Forensics: A Field Manual for Collecting, Examining, and Preserving Evidence of Computer Crimes. Auerbach Publications (2002)
8. Melbin, M.: Night as Frontier. Free Press, Glencoe (1987)
9. Mena, J.: Investigative Data Mining for Security and Criminal Detection. Butter-worth Heinemann (2003)
10. Tsafati, Y., Weimann, G.: Terror on the Internet. Studies in Conflict & Terrorism 25, 317–332 (2002). <http://www.terrorism.com>
11. Spaceman (2001),
<http://www.vbulletin.com/forum/showthread.php?t=32329>

12. Wang, J.H.: Cyber Forensics – Issues and Approaches. In: Managing Cyber Threats: Issues, Approaches and Challenge. Kluwer Academic Publishers, Dordrecht (2005)
13. Wang, J.H., et al.: A study of Automatic Search System on Cyber Crimes. Special Research Report to Telecommunication Bureau, Ministry of Traffic and Communication, Taiwan (1999)
14. Weimann, G.: How Modern Terrorism Uses the Internet. Special Report 116, U.S. Institute of Peace (2004), <http://www.terrorism.net>, <http://usip.org/pubs>
15. Wooley, D.R.: Forum Software for the Web: An independent guide to discussion forum & message board software (2006), <http://www.thinkofit.com/webconf/forumsoft.htm>
16. Zhou, Y., Reid, E., Qin, J., Lai, G., Chen, H.: U.S. Domestic Extremist Groups on the Web: Link and Content Analysis. IEEE Intelligent Systems, Special Issue on Artificial Intelligence for National and Homeland Security 20(5), 44–51 (2005)

Questions for Discussions

1. The Internet has become one of the major channels for criminal activities. The anonymity and less observability of internet activities allow criminals to make acquaintance of and acquire their victims, and eventually commit crimes. The modus operandi based on the Internet have introduced new challenges for crime investigation. Discuss how can the Internet and information technology be also used to counter the high technological crime problem?
2. According to Wally Bock, communities are characterized by three things: common interests, frequent interaction, and identification. Identify Internet Technologies which can be used as platforms for the formation of virtual communities.
3. Although the “hot thread analysis” can be used to assist analysts in identifying illegal or inappropriate contents in some forums, content search methods seem to perform much better. Discuss how the “hot thread analysis” can/can’t be enhanced to improve its performance in identifying inappropriate contents? Justify your answers?
4. Although the proprietary software listing from merchant websites can be used to automatically derive the standard search strings for retrieving illegal or inappropriate postings in a “pirated-CD forum”, it is not trivial to derive the set of standard search strings for other types of gray forums, such as spiritualism. Give feasible methods, which can be used to derive the set of standard search strings for other types of forums, such as the gambling forum and spiritualism forum. Justify your answers.

Identifying Interesting Networks of Criminal Activity

Byron Marshall

Department of Accounting, Finance, and Information Management,

Oregon State University, College of Business, USA

byron_marshall@bus.oregonstate.edu

<http://oregonstate.edu/~marshaby/>

Abstract. While Electronic Records Management Systems (RMS) make it possible for investigators to quickly query large volumes of Law Enforcement (LE) data, much more can be done to effectively use police records to support investigational processes. Detectives frequently identify interesting networks of association and draw them up in link charts to help generate leads, focus investigations, and facilitate communication. A variety of multi-jurisdictional data sharing initiatives facilitate access to increasingly large data sets. Methodologies that display, analyze, or help create useful representations of criminal networks are needed to help sift through massive quantities of available associational data. This chapter discusses a model for analyzing criminal data which employs obtainable, sharable datasets in support of realistic investigational tasks, and illustrates the key issues by demonstrating how a cross-jurisdictional dataset can be used to identify a network of interesting criminal associations. Models of this kind are needed to guide the development of policies, procedures, and technical components appropriate for investigational tasks as agencies work to move beyond administrative efficiency towards investigational effectiveness.

8.1 Introduction

The information needed to effectively track criminals is spread across local, metropolitan, regional, and national agencies. Thus, handling local crimes often requires data from multiple law enforcement (LE) agencies [24]. This growing concern is demonstrated in the development of sharing and analysis systems by the FBI (Virtual Case File), the Department of Defense (TIA and LInX), the Department of Justice (Global JXDM), the Department of Homeland Security (BorderSafe), and other law enforcement agencies (ARJIS, FINDER). Please see the electronic references at the end of the chapter for more information on these systems. These efforts persist despite policy concerns, high costs, and some highly publicized failures. Existing policies and techniques allow for the routine exchange of criminal justice data such as warrants, convictions, and incarceration records, but many agencies are facing significant challenges as they attempt to share more information because useful investigational datasets are frequently inconsistent or ambiguous and often contain sensitive or restricted information.

System developers face at least 4 key challenges as they seek to enhance cross-jurisdictional data sharing and analysis: infrastructure limitations, policy concerns,

data federation and integration, and development of effective analysis methodologies. Solving these problems will require contributions from a variety of sources: legislators and administrators need to continue to invest in infrastructure; government officials, scholars, and advocacy groups need to work together to assure appropriate handling of private and sensitive data; information system practitioners need to engineer massive data sharing capacity; and researchers need to adapt and apply existing techniques to facilitate effective analysis. Although it is tempting to divide this monumental task into smaller pieces, the community needs to recognize that the challenges we face are intertwined. In particular, information system engineers cannot design effective solutions without addressing the privacy and security rules that must apply to shared data. And, both sharing and analysis methodologies need to account for inconsistency and ambiguity because of the heterogeneous infrastructure of existing data sources.

A number of research and development efforts have addressed the integration, organization, and analysis of LE data. Several agency consortiums have applied various integration methodologies to improve administrative and investigational communication. Significant advancements have been made in regional consortiums of LE agencies who have hammered out template memorandums of understanding (MOU) to support shared access to selected data. Social network analysis techniques have been studied in a criminal activity context for many years and other analysis approaches have been proposed to help investigators sift through the massive quantity of data available to law enforcement personnel. For more information on this important foundational work please see the literature review below. To guide and effectively exploit continued advances in each of these areas, developments should be evaluated based on their usefulness to investigators in a heterogeneous, cross-jurisdictional environment.

The case study presented later in this chapter describes a proposed importance flooding methodology intended to help investigators effectively create link charts by identify interesting networks of criminal association [13]. We focus on an important investigational task (link chart creation) using realistically sharable datasets (relationships found in police records) and a spreading activation-style analysis algorithm (importance flooding). Link charts depict criminal associations to help generate leads, focus investigations, facilitate communication, and can develop evidence of criminal conspiracy. Fig. 8.1 is a link chart used in our case study. The resolution of the picture is intentionally low due to privacy and security concerns. Key individuals are listed and lines are drawn to show important associations. The methodology developed in this study allows for somewhat ambiguous input. Its data structures support sharing of data which omits many sensitive details. And, the algorithm allows an investigator to inject additional data and personal judgment into the analysis process. While this is not the only promising methodology for integrated LE data sharing, the description and discussion presented here should highlight key issues in this important domain of investigation.

8.2 Literature Review

This section describes previous work relevant to the development and use of cross-jurisdictional investigation support systems. It briefly addresses several questions:

How do LE agencies integrate data? What does previous research say about extracting and analyzing criminal activity networks? And, what can we learn from research in other fields? We begin by touching on how large datasets are integrated, with a special emphasis on the entity matching problems crucial in LE analysis. While relatively simple queries are commonly used by investigators to try to find data in a collection of LE records, network-based analysis has a long history in criminal investigation. The second half of this section reviews previous criminal activity network analysis research, providing a foundation for the importance flooding algorithm used in our case study. We do not spend much time on the important issue of LE data sharing policy. Please see [1] for more information.

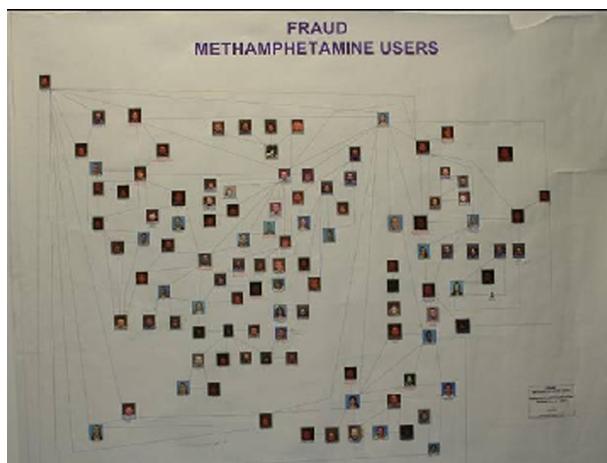


Fig. 8.1. This Fraud/Meth Link Chart was manually drawn, in 2003, by an experienced crime analyst, in six weeks, using Tucson Police Department and Pima County Sheriff's Department records. Beginning with a few investigational targets, a network of associations was identified to help with investigations involving fraud and methamphetamine trafficking.

8.2.1 How Do LE Agencies Integrate Data?

Combining data from independently-developed sources is a challenging task. Several approaches are currently in use in the LE world. One approach aims to create a single, integrated system. For example, the FBI, in attempting to replace its aging Automated Case Support system, invested \$170 million in a new virtual case file system which was never successfully deployed [6]. Without commenting on the details of this particular attempt, it is clear that creating large integrated systems covering multiple regions and addressing many administrative and investigational purposes is very difficult. Another approach ties several systems together so that queries can be processed against multiple data sources from a single interface. For example, the Automated Regional Justice Information System (ARJIS – <http://www.arjis.org/>) has connected 50 local, state, and federal agencies in the San Diego region using “mediators” which translate queries as appropriate for each data source. The Florida Integrated Network for Data Exchange and Retrieval (FINDER) system uses software run at each agency

to manage FINDER “Nodes” which work together to provide integrated query responses. In a third approach, local agencies continue to operate independent Records Management Systems (RMS) but implement automatic processes that reorganize the records and store them in a data warehouse. For example, the COPLINK system (www.coplink.com) is used by a number of agencies across the United States to reorganize their data in support of a variety of analysis tasks. For sharing, agencies can combine information in a single COPLINK node or allow their individual nodes to interoperate to support cross-jurisdictional queries. Effective systems recognize the need for each agency to retain some appropriate level of control over “their” data while supporting some level of shared query functionality.

Two things are needed to integrate a collection of databases: a common schema (a schema says what kinds of things are listed in a database and what attributes are recorded) and a way of matching entities between the datasets (is John Jones in Tucson the same as John Jones in San Diego?). The Global Justice XML Data Model supported by the U.S. Department of Justice (DoJ) Office of Justice Programs (OJP) <http://www.it.ojp.gov/jxdm/> is emerging as a standard schema for data sharing amongst the justice and public safety communities. Many of the previously listed systems provide some level of support for Global JXDM, which specifies entities and attributes appropriate for encoding and sharing criminal justice data (e.g., locations, vehicles, charges, arrests, cases, weapons, drugs, and incidents). The schema includes entities and attributes useful in a wide variety of criminal justice applications but is not specifically focused on investigational tasks. While supporting a widely adopted schema can help format data for sharing (schema-level matching) it does not resolve the entity matching problem.

Entity matching is a key problem in cross-jurisdictional analysis. Even within a single LE data set there are serious entity matching issues. Existing matching processes identified in the data integration literature can be categorized as using (1) key equivalence, (2) user specified equivalence, (3) probabilistic equivalence, or (4) heuristic rules [11]. To clarify the problem, consider the need to match up all references for a single individual within and across a set of police records. In key equivalence matching, a unique number or code identifies each individual. Officers try to accurately identify the individuals involved in an incident with a driver’s license number or social security number but in many cases only names, birthdates, and addresses are collected. This approach is problematic in LE because many individuals lie about their identity, errors are made in recording or transcribing numbers, and no single code applies to everyone. User specified equivalence is commonly applied. When a new police report is recorded, clerks are encouraged to manually match the people in the report to existing records. For a variety of reasons, records are not always matched correctly. Probabilistic equivalence methods use partial matching to identify records which may involve the same person (e.g., Jon T. Fox and Jon Timothy Fox). Some systems try to triangulate by checking other attributes such as birthdates, addresses, or descriptions [20]. In investigational situations professional judgment and heuristics or rules of thumb are generally applied (e.g., manually filtering records for all individuals with a last name of Fox and a first name starting with Jon). Appropriately identifying entity matches strongly affects the effectiveness of analysis techniques.

In [14] we proposed a relationship-oriented framework for usefully combining police records to support investigational tasks, categorizing investigational data as base,

supplemental, and query-specific data. Several problems arise because police reports, as stored in most jurisdictions, are not formatted or organized for sharing and automatic analysis. Narrative descriptions occasionally contain private information which should not generally be shared even with other law enforcement agencies (e.g., a victim's health concerns or home address). This level of detail raises privacy concerns and it is difficult to process information stored in free-text. However, basic incident information is easier to extract and share. The identity (subject to the problems listed in the last paragraph) and role (e.g., victim, suspect, and arrestee) of individuals are carefully entered into appropriate boxes in most electronic incident reports along with details such as date, time, and crime type. These few bits of relatively simple, relatively sharable information can be extracted in relations. The relations can then be built into an association network by sequentially connecting individuals through reported incident data. The base network can then be annotated with facts (this person's car crossed the border) and sensitive information (these two people live together).

8.2.2 Network-Based Analysis of Criminal Activity

Network analysis has a long history in criminal investigation [3, 4, 10]. In [18], Sparrow highlights the importance of social network analysis techniques and identifies several network structure measures with investigational implications. He points out that “ ‘who is central to the organization?’, ‘which names in this data-base appear to be aliases?’, ‘which three individuals’ removal or incapacitation would sever this drug-supply network?’, ‘what role or roles does a specific individual appear to play within a criminal organization?’ or ‘which communications links within a international terrorist fraternity are likely to be most worth monitoring?’ ” (p. 252) are all social network analysis questions.

Some of the ideas presented by Sparrow have been explored in LE research and development projects, resulting in three generations of criminal network analysis tools [10]. First generation tools allow investigators to manually depict activity in an association network. Second generation systems (e.g. Netmap, Analyst's Notebook, Watson, and the COPLINK Visualizer [2, 8, 9]) provide various levels of interaction and pattern identification, representing the data using various visual clues and algorithms to help the user understand charted relationships. Third generation tools implement advanced analytical capabilities. This class of tool has not yet been widely deployed although techniques and methodologies are explored in the research literature. [4] introduces genetic algorithms to implement subgraph isomorphism and classification via social network analysis metrics for intelligence analysis. Network analysis tools measure centrality, detect subgroups, and identify interaction patterns in [22], and [14] studies the topological characteristics of cross-jurisdictional criminal networks.

Because investigators often focus on close associates of target individuals, shortest path measures have received particular attention. One form of analysis identifies the shortest path between target individuals. Identifying close associates and short paths is important for link chart creation. CrimeLink Explorer employed relation strength heuristics to support shortest-path analysis [17]. Guided by domain experts, they weighted associations by: crime-type, person-role, shared addresses or phones, and incident co-occurrence. An algorithm for shortest path analysis for criminal networks was implemented and tested in [23].

The case study we present later in this chapter uses network paths to help identify “interesting” subsets of large criminal activity networks. Identifying interesting (or important) relationships is a problem addressed in the association rule mining field [7]. Measures of interestingness can be classified into two categories: objective measures and subjective measures [19]. Objective measures are generally statistical and include confidence and support. Subjective measures, on the other hand, can be classified into two groups: actionable and unexpected. Beliefs play a role in identifying interesting associations [15]. Results can be filtered by encoding user beliefs (e.g., unexpected or potentially actionable relationships or patterns) using some “grammar” and comparing extracted relationships to that grammar [16]. In the LE domain, finding a way to incorporate the “beliefs” (e.g., the target individuals must be working with both drug pushers and check washers) is important if convincing, actionable leads are to be generated by an analysis system.

Notions of interestingness have received special attention in network-based analysis. Some researchers emphasize that interestingness is relative. For example, a “root set of nodes” within a larger network enhances relevance searching in [21], which describes a general class of algorithms that use explicit definitions of relative importance. This approach implements two notions: 1) two nodes are related according to the paths that connect them, and 2) the longer a path is, the less importance is conferred along that path. Using a scalar coefficient, White and Smyth pass smaller amounts of importance as the distance between a pair of nodes increases. This notion of relative importance aligns well with the cognitive model described by investigators who often begin with some target suspect(s) and look for close associates to identify leads.

In [12] novel network paths (not just nodes or links) are identified to reveal interesting information. This was a new way of analyzing the HEP-Th bibliography dataset from the Open Task of the 2003 KDD Cup [5]. Bibliographic citation data was analyzed to answer questions such as “which people are interestingly connected to C.N. Pope?” The basic notion of their analysis was to detect interesting short paths through a network rather than to detect interesting nodes. They categorized link types and used multiple node types in their network. So, for instance, universities were associated with authors who had published a paper while affiliated with the university, and authors were associated with their co-authors. Without putting in specific rules defining “interesting”, their algorithm discovered that Mr. H. Lu. was the most interesting person relative to C.N. Pope because he interacted with Pope along a variety of network paths. These paths take the following form:

[Lu]-writes-[Paper1]-cites-[Paper2]-written_by-[Pope]
 [Lu]-authors-[Paper1]-authored_by-[Pope], and
 [Lu]-authors-[Paper1]-authored_by-[Person1]-authors-[Paper2]-authored_by-[Pope].

This notion that interestingness is path-based rather than node-based is applicable to criminal investigations. For example, the crime analyst who created the chart shown in Fig. 8.1 noted that she was more interested in people who sold methamphetamines (pushers) and were associated both with people who committed aggravated assault (enforcers) and people who committed fraud (check-washers). This kind of association pattern is a short path through the criminal activity network.

8.3 Case-Study: Helping Create a Fraud/Meth Link Chart

8.3.1 The Task: Link Chart Creation

As in many other jurisdictions, methamphetamine trafficking in the Tucson area results in many crimes besides drug sales and drug use. The fraud unit of the Tucson police department assigned a crime analyst the job of organizing available data to help identify, arrest, and prosecute criminal conspirators who were involved in both drugs and fraud in the Tucson area. Because conspirators receive longer jail terms, investigators hoped to get a group of dangerous individuals off the street for a long time. Beginning with a few target individuals, the analyst systematically queried Tucson Police Department (TPD) and Pima County Sheriff's Department (PCSD) records, then manually pasted pictures and drew lines in a software package to create the link chart shown in Fig. 8.1. Pima County surrounds the city of Tucson and houses the county jail facility, thus the two jurisdictions often have data about the same individuals. The chart was intended to generate leads, focus several investigations, and facilitate communication. This process took 6 weeks, involved review of thousands of cases, and leveraged the experience of the veteran analyst.

Link charts such as this are clearly valuable and it would save time if a computer system could automatically analyze the records to help with this complex task. This case study describes our attempts to leverage TPD and PCSD records to support the link chart creation process. Although the two agencies use different RMS systems to manage their very different operations, both have invested in COPLINK data warehousing installations which convert their records into a common analysis-oriented data structure. Each of the datasets includes more than a million people and millions of incidents recorded in the last decade.

The link charting task calls upon the analyst to evaluate available records. Simplistically, the analyst looks up all the incidents a particular person is involved in and lists every other person associated with each incident. Employing expert judgment, they decide which people to add to the chart. Because the identities of individuals in the records are less than precise, the analyst has to consider partial identity matches, querying for known aliases and similar name spellings. To evaluate a possible identity match she might look at related birth dates, addresses, phone numbers, and physical descriptions. Relationships derived from the incident reports go beyond simply following the “known associates” manually recorded by previous investigators. The analyst weighs available evidence to determine if a connection exists. Perhaps two individuals show up in the same domestic disturbance report, one is suspected of assaulting the other, or case notes list one as the other’s domestic partner.

8.3.2 Our Analysis Methodology

Our analysis methodology imitates and supports the manual link chart creation process. First the data is prepared for analysis, next analysis parameters are identified, then, beginning with the investigational targets, an importance flooding algorithm sifts through the data to generate suggestions. For this case study, we formed an association network beginning with a target set of individuals and following links found in police reports. Individuals were considered for evaluation if they are found within two

“hops” of a selected person; investigators told us they would not look any farther than that. Then we automatically evaluate the general relevance of each individual in the network based on their incident records and the records of known associates. Using an importance flooding computation, we suggest individuals and compare the suggestions to the list of people selected by the investigator. We expand the network to include associates of correct suggestions and adjust the rankings to reduce the importance of rejected people. The methodology is designed to use data that is readily available from police records and reasonably shareable despite privacy and security concerns.

Preparing the Data for Analysis

Representing such data for automatic analysis is no easy task. Emulating the basic steps in the manual process, we extracted relations consisting of two people connected by an incident. People are represented as network nodes and associations between the individuals are recorded as links between the nodes. A person (node) was identified as having a unique first name, last name, and date of birth. Actually, we are quite sure that other correct matches existed, but for this analysis we did not manually adjust the matches. In a production system the user should be able to override the matching system to help improve accuracy. The analysis system also ignores relationships drawn from the case notes. For example, the electronic records do not record family relationships. These additional relationships might have improved on the results we report and can easily be included using our methodology. This preparation results in an interconnected network of individuals.

Identifying Analysis Parameters

To guide the analysis process we addressed two main questions: “what constitutes a strong link?” and “what are we interested in finding?” From an association-rule mining perspective, we are looking for a useful way to encode our investigational “beliefs”. Our first task was assigning a weight to individual associations found in the extracted network. Adapting techniques from previous research [17], we implemented a procedure for assigning an association weight between 0 and 1 for any two individuals listed together in at least one police report.

1. If the individuals appear in 4 or more reports together, assign a final weight of 1 to the association. Otherwise:
 2. Accumulate weights based on individual incidents:
- Assign an initial score to each report involving both individuals. If both are suspects: .99, if only one is a suspect: .5, if neither is a suspect: .3. Intuitively, suspect-to-suspect relations indicate strong criminal relationships while suspect-to-victim or witness-to-victim are weaker indicators.
 - Multiply the highest report score times 3/5 and any others by 1/5.

For example, a pair of individuals appearing together in 6 reports is assigned an association weight of 1, while individuals arrested together in one incident and both witnessing another receive $(.99 \times 3/5) + (.3 \times 1/5) = .694$.

Our methodology goes beyond traditional social network analysis by including a notion of initial importance. Instead of asking a question such as “who is the leader of the group?” we want to ask “who is important to this particular investigation?” As a

starting point, we evaluate past behavior based on investigator input. For the Fraud/Meth investigation, a set of rules was developed by talking with the crime analyst. Fig. 8.2 depicts three types of importance rules: activity-based group rules, multi-group membership rules, and path rules. The use of path rules builds on the previously cited work by Lin and Chalupsky [12]. Including a path-based notion of importance allows us to steer the analysis towards potentially interesting clusters of individuals. In this case the analyst said that she was more interested in a “drug dealer” if that dealer also had identified associations with “leg-breakers” and “check-washers.” Searching for such information in existing LE tools would require multiple queries and many steps of iterative follow-up. Each selection rule is assigned an importance value (1–9). Rules used in this case focused on fraud, drug sales, and aggravated assault.

1. Individuals suspected or arrested for crimes in any two of those categories had their importance score increased by 3.
2. If they were involved in all three categories we added 5.
3. If they participated in a “path” where one person with a fraud crime is connected to another person with a drug crime, they received an additional score of 3.
4. If they participated in an association path connecting a fraud perpetrator to a drug seller and the seller to the actor in an aggravated assault, we added 5 more.

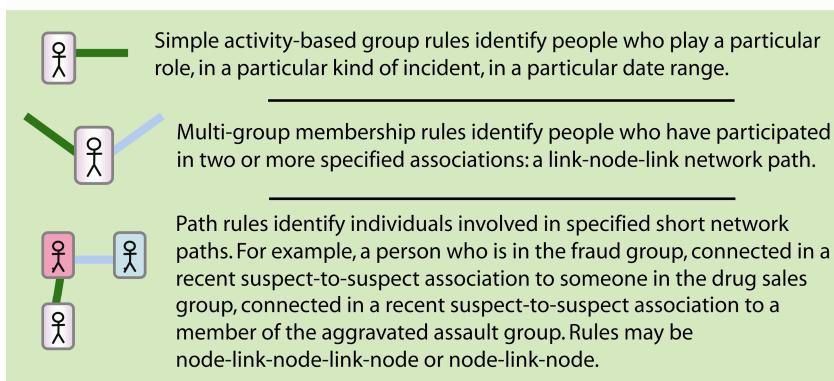


Fig. 8.2. Three Types of Initial Importance Rules are used in this case study

The Importance Flooding Algorithm

The initial importance scores are passed along the weighted network links to identify potentially interesting people. For readers interested in technical details, the importance flooding computation is described in Fig. 8.3. It implements three notable mechanisms:

1. Important nodes pass importance to nearby nodes.
2. Closely connected nodes (strong weights and short paths) share more importance than weakly connected nodes (small association scores or distant connections).
3. Evaluation is focused on the investigational targets.

```

Main Process:
Initialize all nodes N1 in N: N1.PREV= 0, N1.ADD = 0
For each iteration
  For each node N1 in N      // Call recursive path tracing
    PassAmt = N1.PREV + N1.INIT
    PathList = N1.ID, PathLen = 1
    pathTrace (PassAmount, PathList, PathLen)
  For each node N1 in N // Normalize and re-initialize
    N1.PREV = (N1.PREV + N1.INIT + N1.ADD) / MAXVAL
    N1.ADD = 0
  // reinforce the importance investigational targets
  For each node T1 in the TargetNode List: T1.PREV = 1

Recursive Path Tracing:
pathTrace (PassAmount, PathList, PathLen)
  PassingNode = The last node included in PathList
  NumOfAssoc = The # of nodes associated with PassingNode
  For each node Na associated with PassingNode
    if Na is not already included in PathList
      RELWGT = the relation weight for the pair [PassingNode,Na]
      DECAYRATE = the decay coefficient for PathLength
      PASSONAMT = PassAmt * RELWGT * DECAYRATE * (1 / NumOfAssoc)
      Na.ADD = Na.ADD + PASSONAMT
      if PathLen < DDD // traverse paths to length DDD
        pathTrace (PASSONAMT, PathList + Na.ID, PathLen + 1)

```

Fig. 8.3. Importance Flooding Pseudo Code. Initial importance (INIT) and association weights have already been assigned as described in the previous section. Importance is passed to associated nodes by following weighted links. In these experiments we used a scalar coefficient of [.5, .25] which defines a path length (DDD) of two hops. Thus passed importance is reduced by .5 for direct connections and then by .25 for transitive links. For example, given N1 with an initial value of 1, associated only with N2 and N3 with association weights of 1 and .694 respectively, N3 would receive an importance increase of .1735 -- 1 (current importance) x .694 (association weight) x ½ (because there are two associated nodes) x .5(the decay rate for the first hop). If N4 is the only other node associated with N3 (association weight of .5) it would receive .010844 -- .1735 (from the last computation) x .5 (association weight) x ½ (N3 is associated with 2 nodes) x .25 (the decay rate for the second hop). Importance accumulates for each node along a number of paths from a number of sources. When all nodes have passed along their importance, the scores are normalized to a value between 0 and 1 and the cycle is repeated. We used 4 iterations in this case study. Before each iteration, the nodes for the investigational targets are always reset to 1 to reinforce the importance of the investigational targets.

8.3.3 Experimentation and Results

This approach was designed to help a crime analyst build a link chart more efficiently. To test the effectiveness of our approach we measured its ability to help an analyst create a link chart like the one shown in Fig. 8.1. Starting from the same individuals used by the crime analyst, we prepared the data and defined the rules as described in the previous section. We applied the importance flooding methodology and compared its suggestions to two other algorithms: Breadth First Search and Closest Associate. The Breadth First Search approach looks up all associations for the 4 starting individuals in random order and puts them in a list. Then, it suggests all the associates of the people on the list. This is similar to the manual process where an analyst looks at each target person's records and searches for all identified associates. The second approach applies the kind of shortest path analysis proposed in previous research to

the link chart creation task. In essence, the Closest Associate algorithm looks through all the associates of previously suggested individuals and selects those with the highest association strength first. An algorithm is considered to be more effective when it suggests a higher proportion of “correct” individuals. A suggestion is said to be correct if the analyst included the suggested individual in the original link chart.

Fig. 8.4 and Table 8.1 report on the accuracy of each of the three algorithms. Importance flooding substantially out performed the other approaches. Fig. 8.4 shows that for any number of suggested nodes the list prepared using similarity flooding included more of the nodes that were chosen by the analyst. When 318 nodes had been suggested, 70 (22%) of them were nodes that the analyst had selected for inclusion in the manually-drawn link chart. Only half that number of correct nodes had been suggested by the closest associate algorithm (35). Table 8.1 lists the results until 2,000 suggestions had been made.

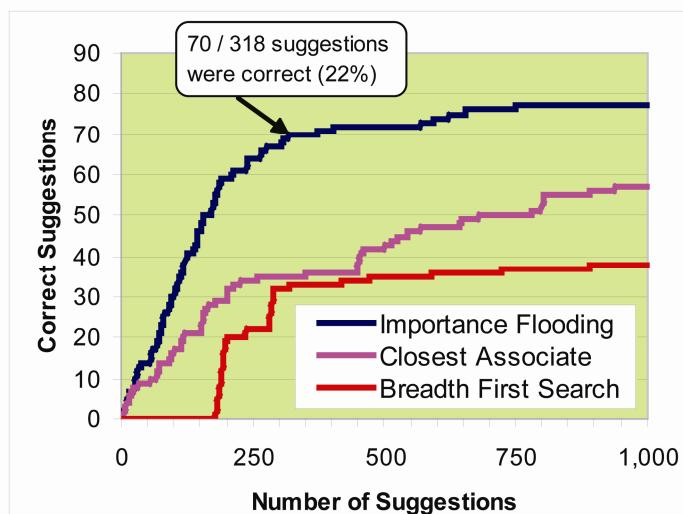


Fig. 8.4. In our experiment, the importance flooding algorithm made better suggestions as compared to the closest associate and breadth first search methodologies.

Table 8.1. Number of Correct Nodes per Suggestion

Number of Suggestions	Correct Suggestions					
	Importance Flooding		Closest Associate		Breadth First Search	
	#	%	#	%	#	%
100	30	30%	16	16%	0	-
250	64	26%	34	14%	22	9%
500	72	14%	54	9%	35	7%
1,000	77	8%	57	6%	38	4%
2,000	93	5%	83	4%	45	2%

To put these results into perspective, consider the impact this methodology might have on operations. Creating a link chart is somewhat expensive and requires the time of experienced crime analysts. Thus, it is only used in the most important or high-profile cases. An analyst using the technique might find more important individuals while reviewing fewer case files. This could result in cost saving to the agency or give the agency a chance to apply the technique in more investigations.

8.4 Discussions and Conclusions

We believe our results demonstrate the promise of our methodology. Although investigators know that association networks are important in supporting investigations, standard query interfaces do not really leverage associational data. Some advanced systems do allow the user to follow links, but they generally do not include the kind of heuristic link and importance analysis employed in our technique. Solutions which ask the user to manually filter regular search results will become increasingly problematic as agencies share more data. Importance flooding and similar techniques should help the investigator intelligently sift quantities of associational information. It is also important to consider the privacy and policy implications of various cross-jurisdictional data sharing techniques. The importance flooding tool presented here uses relatively shareable datasets which avoid many widely recognized policy and privacy problems.

The methodology outlined here introduces several new ideas for law enforcement research. First of all, it focuses on the task of link chart creation. Previous work has focused on analyzing criminal groups or identifying crime trends. It integrates previous shortest-path evaluation into a larger investigational process. Secondly, it employs both association closeness and a measure of importance based on previous records in a network context. Previous work has represented networks of criminals using association closeness to perform social network analysis but has generally not combined link weights with other relative measures to support analysis. Finally, importance flooding recognizes that interestingness is a path-based phenomenon. Path-based heuristics guide the analysis process. This is an adaptation of previous work which noted that interestingness algorithms should identify interesting paths not just individual nodes or links. Please see [13] for more evaluation of the importance of these path-based heuristics. Each of these innovations is appropriate for use in a cross-jurisdictional context.

Applying a spreading activation model such as importance flooding in the law enforcement domain makes sense. One of the key problems in law enforcement data is missing information. Two criminals may work together closely without affecting available data if they are not caught. However, sometimes other associations can make a connection. For example, consider an investigation centering on person A. Persons A and B have substantial individual records and a strong but unknown association. The connection might be made transitively. That is, if person C, a relatively unknown individual, is identified in one incident with A and another incident with B, the importance flooding algorithm would pass importance from both A and B to C, guiding the investigator to more quickly check on B's importance. Making this kind of inference manually can be a tedious process. While some existing system can

render a visual link chart by retrieving associations, the importance flooding approach employs user-supplied heuristics to help an investigator to more quickly focus on the most important targets.

Several factors might improve on the effectiveness of this approach in an ongoing investigation. To avoid potential experimental bias, we used only data taken directly and automatically from the police records. This data could be improved as the analyst looks at the data. Extra data such as co-habitation and family ties could be added to improve network accuracy. Additional entity matches might be identified to tie in more incidents that were left out of our analysis due to typographical errors or because the individuals lied to officers. In addition, it may be that other individuals “should” have been included on the chart but the manual search method used by the crime analyst did not identify them.

The importance flooding algorithm involves a number of parameters which must be set to drive the analysis. For example, various combinations of the scalar coefficient and number of iterations could be used. The results presented here were achieved with a path length of two, a scalar coefficient of [.5, .25], and with 4 computational iterations. In ongoing work we have tested several other combinations and our preliminary results suggest that while results may improve somewhat when different combinations are chosen, the general pattern is the same: the importance flooding approach suggests more of the correct answers. The link weight values are also somewhat subjectively determined. While it seems that the link weight methodology is reasonable given previous research, it may be that different weights produce better results in certain types of investigations. It is also not yet clear how transferable the importance heuristics are. For example, it may be that there is an identifiable set of heuristics that performs well in identifying potentially dangerous individuals associated with an investigation. If so, these heuristics could be routinely applied to help create a “dangerous face” book to help protect investigators working on a case. On the other hand, heuristics useful in the Fraud/Meth case might not be very useful in tracking terrorists, robbers, or even cocaine traffickers. More study is needed to help guide practitioners.

This chapter discusses and highlights many of the key challenges agencies face when trying to employ cross-jurisdictional data sets to support investigational processes. Although we have tested importance flooding on another large link chart with similar results, the scope of our testing is still quite limited. Hopefully, additional research will continue to identify useful methodologies to reduce the number of costly system failures and increase the effectiveness of our investigational agencies while respecting important privacy concerns.

References

1. Atabakhsh, H., Larson, C., Petersen, T., Violette, C., Chen, H.: Information Sharing and Collaboration Policies within Government Agencies. In: Proceedings of the 2nd Symposium on Intelligence and Security Informatics, Tucson, AZ (2004)
2. Chabrow, E.: Tracking The Terrorists: Investigative skills and technology are being used to hunt terrorism's supporters. In: Information Week (2002)
3. Coady, W.F.: Automated Link Analysis - Artificial Intelligence-Based Tool for Investigators. Police Chief 52, 22–23 (1985)

4. Coffman, T., Greenblatt, S., Marcus, S.: Graph-Based Technologies for Intelligence Analysis. *Communications of the ACM* 47, 45–47 (2004)
5. Gehrke, J., Ginsparg, P., Kleinberg, J.: Overview of the 2003 KDD Cup. *SIGKDD Explor. Newsl.* 5, 149–151 (2003)
6. Goldstein, H.: Who Killed the Virtual Case File. *IEEE Spectrum* 42 (2005)
7. Hilderman, R.J., Hamilton, H.J.: Evaluation of Interestingness Measures for Rank-ing Discovered Knowledge. In: Cheung, D., Williams, G.J., Li, Q. (eds.) *PAKDD 2001*. LNCS (LNAI), vol. 2035, pp. 247–259. Springer, Heidelberg (2001)
8. I2, I2 Investigative Analysis Software 2004 (2004)
9. KCC, COPLINK from Knowledge Computing Corp. 2004 (2004)
10. Klerks, P.: The Network Paradigm Applied to Criminal Organizations: Theoretical nitpicking or a relevant doctrine for investigators? Recent developments in the Netherlands. *Connections* 24, 53–65 (2001)
11. Lim, E.P., Srivastava, J., Prabhakar, S., Richardson, J.: Entity Identification in Data-base Integration. *Information Sciences* 89, 1–38 (1996)
12. Lin, S.D., Chalupsky, H.: Using Unsupervised Link Discovery Methods to Find Interesting Facts and Connections in a Bibliography Dataset. *SIGKDD Explor. Newsl.* 5, 173–178 (2003)
13. Marshall, B., Chen, H.: Using Importance Flooding to Identify Interesting Net-works of Criminal Activity. In: Mehrotra, S., Zeng, D.D., Chen, H., Thuraisingham, B., Wang, F.-Y. (eds.) *ISI 2006*. LNCS, vol. 3975. Springer, Heidelberg (2006)
14. Marshall, B., Kaza, S., Xu, J., Atabakhsh, H., Petersen, T., Violette, C., Chen, H.: Cross-Jurisdictional Criminal Activity Networks to Support Border and Transportation Security. In: Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems, Washington D.C. (2004)
15. Padmanabhan, B., Tuzhilin, A.: Unexpectedness as a Measure of Interestingness in Knowledge Discovery. *Decision Support Systems* 27, 303–318 (1999)
16. Sahar, S.: On Incorporating Subjective Interestingness into the Mining Process. In: Proceedings of the IEEE International Conference on Data Mining (ICDM 2002) (2002)
17. Schroeder, J., Xu, J., Chen, H.: CrimeLink Explorer: Using Domain Knowledge to Facilitate Automated Crime Association Analysis. In: Chen, H., Moore, R., Zeng, D.D., Leavitt, J. (eds.) *ISI 2004*. LNCS, vol. 3073. Springer, Heidelberg (2004)
18. Sparrow, M.K.: The Application of Network Analysis to Criminal Intelligence: An Assessment of the Prospects. *Social Networks* 13, 251–274 (1991)
19. Silberschatz, A., Tuzhilin, A.: What Makes Patterns Interesting in Knowledge Discovery Systems. *IEEE Transactions on Data and Knowledge Engineering* 8, 970–974 (1996)
20. Wang, G., Chen, H., Atabakhsh, H.: Automatically detecting deceptive criminal identities. *Commun. ACM* 47, 70–76 (2004)
21. White, S., Smyth, P.: Algorithms for Estimating Relative Importance in Networks. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, D.C. (2003)
22. Xu, J., Chen, H.: Untangling Criminal Networks: A Case Study. In: Chen, H., Miranda, R., Zeng, D.D., Demchak, C.C., Schroeder, J., Madhusudan, T. (eds.) *ISI 2003*. LNCS, vol. 2665. Springer, Heidelberg (2003)
23. Xu, J., Chen, H.: Fighting Organized Crime: Using Shortest-Path Algorithms to Identify Associations in Criminal Networks. *Decision Support Systems* 38, 473–487 (2004)
24. Zhao, J.L., Bi, H.H., Chen, H., Zeng, D.D., Lin, C., Chau, M.: Process-driven collaboration support for intra-agency crime analysis. *Decision Support Systems* 41, 616–633 (2006)

Online Resources

A Growing Standard for Law Enforcement Data Sharing

The Department of Justice sponsored Global Justice XML Data Model (Global JXDM) is “... intended to be a data reference model for the exchange of information within the justice and public safety communities.” <http://www.it.ojp.gov/jxdm/>

Global JXDM includes a set of XML schemas appropriate for encoding and sharing criminal justice data. For example, locations, vehicles, charges, arrests, cases, weapons, drugs, and incidents can be described using Global JXDM attributes. Some of the codes and taxonomies used in the model are maintained by other agencies such as the American Association of Motor Vehicle Administrators, the U.S. Census Bureau, the Department of Defense, and the U.S. Postal Service.

Links to Several Law Enforcement Data Sharing Initiatives

As of July, 2006, the Florida Integrated Network for Data Exchange and Retrieval (FINDER) system reports that it is used by 139 agencies in Florida. The system supports person, vehicle, and pawn ticket queries. Developed at the University of Central Florida’s Public Safety Technology Center, FINDER is planning to support data sharing with other agencies in the Global JXDM schema. See <http://www.finder.ucf.edu/> for more information and a demo system.

The Naval Criminal Investigative Services’ LInX system lists some 100 agencies which use its data warehouse system to support investigations. <http://www.ncis.navy.mil/linx/index.html>

The Automated Regional Justice Information System (ARJIS) supports a secure network with “data on the regions’ crime cases, arrests, citations, field interviews, traffic accidents, fraudulent documents, photographs, gang information and stolen property”. <http://www.arjis.org/>

BorderSafe (funded by the Department of Homeland Security) includes several cross-jurisdictional data sharing initiatives. <http://ai.arizona.edu/research/coplink/bordersafe.htm>

Reports

The Law Enforcement Information Technology Standards Council (LEITSC) is a consortium of the International Association of Chiefs of Police (IACP), National Organization of Black Law Enforcement Executives (NOBLE), National Sheriffs’ Association (NSA), and Police Executive Research Forum (PERF). Their mission is to “...foster the growth of strategic planning and implementation of integrated justice systems.” <http://www.leitsc.org/About LEITSC.htm>

They suggest that records management systems (RMS) acquired by law enforcement agencies meet important guidelines such as compatibility with data sharing initiatives. This report includes recommended specifications for such a system. http://it.ojp.gov/documents/LEITSC_Law_Enforcement_RMS_Systems.pdf

The CLEAR system promotes data sharing in Illinois. Its scope and impact (2003) is reported in:

http://www.cops.usdoj.gov/html/cd_rom/tech_docs/pubs/

PolicingSmarterThroughIT.pdf

Questions for Discussions

1. What are some of the main issues facing law enforcement agencies who want to share collected data?
2. Contrast the approach described in this chapter to familiar computerized searches tools. What are some of the difficulties preventing the federal government from creating a useful nationwide crime data search tool?
3. Many data mining applications in other domains train selection algorithms using previously classified examples to identify interesting items. For example, the IRS (Internal Revenue Service) could label fraudulent tax returns in previous filings and use a data mining approach to identify other returns that should be audited. Compare that approach to the technique discussed in this chapter.
4. Why is it difficult to match up people in criminal data sets from different jurisdictions? How can this problem be addressed?
5. How is the network-based search system described in this chapter different from most existing information retrieval systems which either support complex and specific queries (like an electronic card catalog for a library) or accept simple keywords (like several popular internet search portals)? Can you think of other applications where a network-based search algorithm would be valuable?

Name Matching in Law Enforcement Database

Olcay Kursun¹, Michael Georgioupolos², and Kenneth Reynolds³

¹ Department of Computer Engineering,
International Black Sea University, Georgia
{okursun@ibsu.edu.ge, okursun@mail.ucf.edu}

² Department of Electrical and Computer Engineering,
University of Central Florida, USA
michaelg@mail.ucf.edu

³ Department of Criminal Justice and Legal Studies,
University of Central Florida, USA
kreynold@mail.ucf.edu

Abstract. As it is the case with all database systems that collect and store data, data input errors occur resulting in less than perfect data integrity, or what is common referred to as the “dirty-data” problem. American investigators are not familiar with many foreign names such as Zacarias Moussaoui. If the first or last name is spelled incorrectly during a query, the person record could be missed. Individuals who are chronic offenders and those who are attempting to evade detection use alias. Moussaoui is also known as Shaqil and Abu Khalid al Sahrawi. Unless smart analytical tools are available for effective name matching where data integrity conditions, challenging name spellings, and deliberate obfuscation are present, the likelihood of missing a critical record is high. This paper addresses some of the problems stemming from unreliable and inaccurate law enforcement data. Although the ideas proposed are using “*name data*” as an illustration of how to deal with dirty data, the proposed approaches will be extended to other types of dirty data in the law enforcement databases, such as addresses, stolen item/article names/descriptions/brand names, etc.

9.1 Introduction

With the advances in computer technologies, large amounts of data are stored in data warehouses (centralized or distributed) that need to be efficiently searched and analyzed. With the increased number of records that organizations keep the chances of having “dirty data” within the databases (due to aliases, misspelled entries, etc.) increases as well [5, 13]. Prior to the implementation of any algorithm to analyze the data, the issue of determining the correct matches in datasets with low data integrity must be resolved. In this paper, we focus on the problem of searching proper nouns (first and last names) within a database. The application of interest to us is in law enforcement; however, there are many other application domains where availability of accurate and efficient name search tools in large databases is imperative, such as in medical, commercial, or governmental fields [2, 15].

There are two main reasons for the necessity of techniques that return fuzzy matches to name queries: (1) the user does not know the correct spelling of a name; (2) names are already entered within the database with errors because of typing errors,

misreported names, etc [12]. For example, record linkage, defined as finding duplicate records in a file or matching different records in different files [17, 9], is a valuable application where efficient name matching techniques must be utilized.

The main idea behind all name matching techniques is comparing two or more strings in order to decide if they both represent the same string. The main string comparators found in the literature can be divided in phonetic and spelling based. *Soundex* [11] is used to represent words by phonetic patterns. Soundex achieves this goal by encoding a name as the first letter of the name, followed by a three-digit number. These numbers correspond to a numerical encoding of the next three letters (excluding vowels and consonants h, y, and w) of the name [18]. The number code is such that spelled names that are pronounced similar will have the same soundex code, e.g., “Allan” and “Allen” are both coded as “A450”. Although soundex is very successful and simple, it often misses legitimate matches, and at the same time, detects false matches. For instance, “Christie” (C623) and “Kristie” (K623) are pronounced similarly, but have different soundex encodings, while “Kristie” and “Kirkwood” share the same soundex code but are entirely different names. In addition, even though the phonetic-based methods are really fast, the search times for these methods exhibit high variability since the phonetic-based density of certain names is higher than others, a property also known as *name dependency*.

On the contrary, spelling string comparators check the spelling differences between strings instead of phonetic encodings. One of the well-known methods that is used to compare strings is by measuring their “edit distance”, defined by Levenshtein [6]. This can be viewed as the minimum number of characters that need to be inserted into, deleted from, and/or substituted in one string to create the other (e.g., the edit distance of “Michael” and “Mitchell” is three). Edit-distance approaches can be extended in a variety of ways, such as taking advantage of phonetic similarity of substituted characters (or proximity of the corresponding keys on the keyboard) or checking for transposition of neighboring characters as another kind of common typographical error [4] (e.g., “Baldwin” vs. “Badlwin”). The name-by-name comparison by edit distance methods throughout the entire database renders the desired accuracy, at the expense of exhibiting high complexity and lack of scalability.

The technique of n-grams can be viewed as a simple variance of the edit-distance approach. An n-gram is a substring of length n . If two strings are compared with respect to their n -grams, the sets of all distinct substrings of length n will be calculated for both strings. Next, the similarity of the two strings will be computed based on the number of n -grams occur in both of the sets. In the literature, n -grams are reported to perform merely comparable to the other edit-distance approaches [12, 18].

There are some other name matching approaches that are different than the string comparisons mentioned so far. One such interesting method, called reverse edit-distance [3], generates a list of words by transforming the query string with one (or more) of the typographical errors mentioned earlier. This approach is not suitable for our task as will be explained in Sect. 9.5. Another approach employs stemming algorithms for reducing words to their stem forms by removing the suffixes and transforming the words to their infinitive forms [12]. This method is interesting for verbs, nouns and adjectives, but it is not appropriate for name searching. For a comparative review of different name matching algorithms and their efficiency and performance, the reader is also referred to [12] and [4], and to the more recent [2].

In this paper, we propose a string-matching algorithm, named ANSWER (Approximate Name Search With ERrors), that is fast, accurate, scalable to large databases, and exhibiting low variability in query return times (i.e., robust). This string comparator is developed to establish the similarity between different attributes, such as first and last names. In its application to name matching, ANSWER is shown to be even faster than phonetic-based methods in searching large databases. It is also shown that ANSWER's accuracy is close to those of full exhaustive searches by spelling-based comparators.

The organization of the paper is as follows. In Sect. 9.2, we briefly describe *FINDER* (the Florida Integrated Network for Data Exchange and Retrieval). *FINDER* is the operational application environment for our proposed name searching algorithms. This data domain is a real world database system used by law-enforcement agencies. In Sect. 9.3, we describe PREFIX, which forms the basis for our proposed name matching technique (ANSWER). The PREFIX method is a full-search tool based on a tree structure called *prefix-dictionary*. In Sect. 9.4, we fully describe the ANSWER algorithm. In Sect. 9.5, we present our experimental results and comparisons with the top-level well-known approaches. The comparisons are based on a variety of aspects of the methods such as precision and recall, scalability, and means and standard deviations of run-times. We discuss DBMS implementation issues in Sect. 9.6, and finally, summarize our conclusions in Sect. 9.7.

9.2 The Operational Environment -- **FINDER**

One of the major advantages of our research is that we have a working test-bed to experiment with (*FINDER* – the Florida Integrated Network for Data Exchange and Retrieval). *FINDER* (see Fig. 9.1) has been a highly successful project in the state of Florida that has addressed effectively the security and privacy issues that relate to information sharing between various law enforcement agencies/users. It is operated as a partnership between the University of Central Florida and the law-enforcement agencies in Florida sharing data – referred to as the Law Enforcement Data Sharing Consortium.

The *FINDER* software is a web-based software that uses a web service interface to respond to queries from participating agencies and send query results back to a requesting agency. The software is capable of sending queries to multiple agencies simultaneously and compiling the results into a format that is useful to the requesting law enforcement officer. This approach has allowed agencies that participate to meet the political and organizational constraints of sharing data, such as (1) they retain local control of their own data; (2) they know who has accessed their own data (through log files), and furthermore it has been of low cost approach to the participating agencies. As of October 2006, more than 130 agencies have executed contracts with the University of Central Florida (UCF) to share data using the system, and many more have access to the data through guest accounts at neighboring agencies. Detailed information about the organization of the data sharing consortium and the *FINDER* software is available at <http://finder.ucf.edu/>.

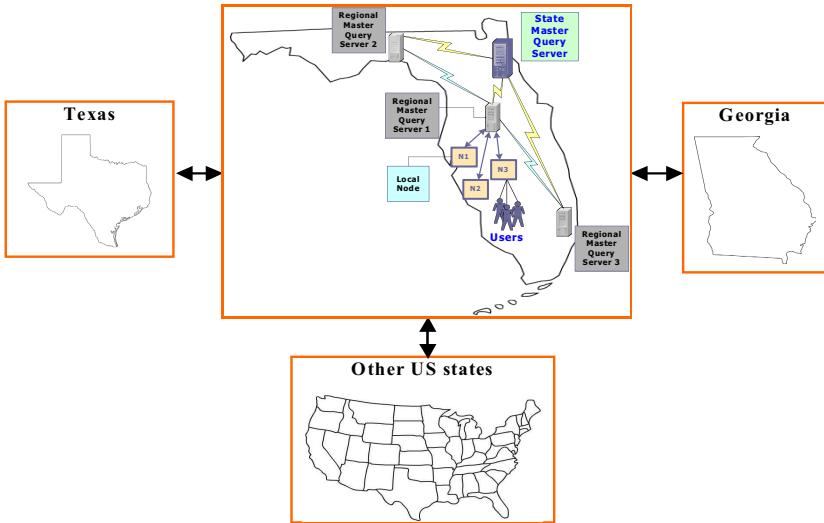


Fig. 9.1. The general overview of the FINDER network in Florida and expanded to other states

Part of the constraints of the FINDER system and also most law enforcement records management systems is that once the data has entered into the system it must remain intact in its current form. This includes data that have been erroneously entered, and consequently they contain misspellings. This problem was identified by the FINDER team and has also been substantiated in the literature [5, 16, 7]. A simple illustration related to name matching, utilizing dirty data available in the FINDER system, is shown in Table 9.1, which emphasizes both the level of data integrity and the challenges of using standard SQL queries to retrieve records from a law enforcement database. In Table 9.1, we are depicting the results of an SQL query on “Joey Sleischman”. An SQL query will miss all the records but the first one. The other records could be discovered only if we were to apply an edit distance algorithm on all the existing records in the database, an unsuitable approach though, due to its high computational complexity, especially in large databases. In particular, the rest of the records (besides the exact match), shown in Table 9.1 were identified by comparing the queried record (“Joey Sleischman”) against all records in the database (by applying the edit distance approach). The Last Name, First Name, DOB (Date of Birth), and Sex were used as parameters in this search. In order to detect the matching records, we assigned weights to the fields: Last Name (40%), First Name (20%), DOB (30%), and Sex (10%). We used the edit distance algorithm [6] for determining the degree of match between fields. In order to convert the edit distance to a match score, we have used the formula given in Eq. 9.1. The overall match scores were obtained by using Eq. 9.2.

$$match(s_f, r_f) = 1 - \frac{LD(s_f, r_f)}{\max(\text{length}(s_f), \text{length}(r_f))} \quad (9.1)$$

where s_f and r_f represent the strings in field f of record s and r , respectively, and LD represents Levenshtein’s edit distance [6] reviewed in Sect. 9.1. LD computes the

minimum number of characters that need to be inserted into, deleted from, and/or substituted in s_f to get r_f :

$$\text{match_score} = \sum_f \text{match}(s_f, r_f) * \text{weight}(f) \quad (9.2)$$

Table 9.1. Example of the Data Integrity Issues within the FINDER data

Last Name	First Name	DOB	Sex	Match
<u>INPUT QUERY:</u>				
SLEISCHMAN	JOEY	1/21/1988	M	≥ 85%
<u>MATCHING DISTINCT RECORDS:</u>				
SLEISCHMAN	JOEY	1/21/1988	M	100%
SLEICHMAN	JOEY	7/21/1988	M	91%
SLEISCHMANN	JOSEPH	1/21/1988	M	88%
SLEISCHMANN	JOSPEH	1/21/1988	M	88%
SLEISHMAN	JOEY		M	87%
SLEISCHMANN	JOEY		M	87%
SLEISHCHMANN	JOSEPH	1/21/1988	M	86%
SLESHMAN	JOEY		M	85%

As it can be seen in Table 9.1, the edit distance algorithm provides an excellent level of matching, but the algorithm requires a full table scan (checking all records in the database). This level of computational complexity makes it unsuitable as a technique for providing name matching in applications, such as FINDER, where the number of records is high and consistently increasing. In the next sections, we are discussing in detail a name matching approach that alleviates this computational complexity.

Quantifying the dirtiness of the data elements in a database is an important diagnosis step prior to engaging any data integrity tools to alleviate the problem of dirtiness in the data. Our task here is to assign a cleanliness value to the database, based on the data, and according to some appropriately defined rules, which are database specific. This measure of database cleanliness can be extended to a variety of databases by modifying the cleanliness criteria/rules, as needed.

As an exemplary case, we chose to demonstrate the dirtiness of “person-related” fields. A table with a distinct combination of the selected person related fields: first name, last name, DOB, sex is first created. Then, to define distinct persons we have to first assign different weights to each field describing a distinct person (such as last name, first name, DOB, etc). These weights can best be determined after discussions with police investigators are conducted. For instance, two records with matching last names, sex, and DOBs but with somewhat different first names may pass the similarity threshold, and be considered that they correspond to the same person. However, if the last names were somewhat different, then these two records may not pass the similarity threshold, and thus be considered that they refer to two different people. It is important to be recognized here that in order to come up with an accepted set of

weights for a database considerable interaction with the users of this database is needed, and in our case these users are the FINDER law enforcement agencies.

Once the issue of persons being identical or not has been resolved a measure of dirtiness of each data field, related to a person's record, needs to be addressed. One suggestion is to define as measure of dirtiness for each field to be the average ratio of the number of variations in that field to the number of records for that identity. For instance, the cleanliness of *lastname* field, suppose that the identity "SLEISCHMAN, Joey" has been recorded 100 times in the database. As seen in Table 9.1, the last name "SLEISCHMAN" has one correct and 5 other distinct variations, then the last name data is said to be 5% dirty for this identity. The dirtiness will then be computed for all the individuals (or a subset of the individuals) and the average dirtiness will be reported as the dirtiness factor of the *lastname* field of the database.

The dirtiness of every field related to a person's identity can be determined. The measure of dirtiness of each field related to a person's identity is useful information for any database user, and it is also useful information to any data-analysis approach that is proposed, which utilizes information from these data fields.

9.3 The PREFIX Algorithm

In order to reduce the time complexity of the full-search of partially matching names in the database (of crucial importance in homeland security or medical applications), we propose a method that constructs a structured dictionary (or a tree) of prefixes corresponding to the existing names in the database (denoted PREFIX). Searching through this structure is a lot more efficient than searching through the entire database.

The algorithm that we propose is dependent on a maximum edit distance value that is practically reasonable. Based on experimental evidence, it has been stated that edit distance up to three errors performs reasonably well [8]. For example, "Michael" and "Miguel" are already at an edit distance of three. Let k represent the maximum number of errors that is tolerated in the name matching process. Using a minimal k value that works well in the application at hand, would make the search maximally fast. Setting k to zero would equal to an exact search which is currently available in any query system. Increasing k increases the recall (i.e., it will not miss any true matches), even though this implies a decrease in the precision (i.e., it will return many more false positives).

PREFIX relies on edit distance calculations. Its innovation though lies on the fact that it is not searching the entire database to find names that match the query entry but accomplishes this goal by building a dictionary of names. One might think that it would not be very efficient to have such a dictionary due to the fact that we would still need to search the whole dictionary, as the spelling error could happen anywhere in the string, such as "Smith" vs. "Rmith". However, our algorithm can search the dictionary very fast, using a tree-structure, by eliminating the branches of the tree that have already been found to differ from the query string by more than k .

There are two key points to our approach: (1) Constructing a tree of prefixes of existing names in the database and searching this structure can be much more efficient than a full scan of all names (e.g., if "Jon" does not match "Paul", one should not consider if "Jonathan" does); (2) such a prefix-tree is feasible and it will not grow

unmanageably big. This is due to the fact that many substrings would hardly ever be encountered in valid names (e.g., a name would not start with a “ZZ”); consequently, this cuts down significantly the number of branches that can possibly exist in the tree. Other data structures that rely on such key points are also proposed in the literature [1, 10].

9.3.1 Constructing a Dictionary of Prefixes of Names

The PREFIX algorithm creates a series of prefix-tables $T_1, T_2, T_3\dots$, where T_n will link (index) T_{n+1} . T_n will contain all n -symbol-long prefixes of the names in the database. These tables correspond to the levels of the prefix-tree. The reason that we use tables is to facilitate the implementation of this approach in any database system. T_n will have the following fields: current symbol (the n^{th} symbol), previous symbol ($n-1^{st}$ symbol), next symbol ($n+1^{st}$ symbol), its links (each link will point to the index of an entry in the prefix-table T_{n+1} with this current entry as the prefix, followed by the symbol in the “next symbol” field), and a field called Name that indicates whether or not the prefix itself is already a name (e.g., Jimm is a prefix of Jimmy but it may not be a valid name). Note that in the links field we cannot have more than 26 links because there are only 26 letters in the English alphabet. Also note that the first prefix-table (T_0) will not utilize the previous symbol field.

Suppose that our database contains “John”, “Jon”, “Jonathan”, “Olcay”, “Jim”, “Oclay”, and “Jimmy”. After building the prefix-dictionary shown in Fig. 9.2, it can be used as many times as needed for subsequent queries. It is very simple to update the dictionary when new records are added (the same procedure explained above, when creating the tables in the first place, is used to add records one by one). Each level i in Fig. 9.2 is a depiction of the prefix-table T_i (for example the third table consists of JOH, JON, OLC, JIM, OCL). The dark-colored nodes in Fig. 9.2 are the prefixes that are actually valid names as well.

Essentially, we create a list of all possible prefixes in order of appearance (no sorting of the tables is needed). Despite the fact that the prefix tables are not sorted, each one is implicitly indexed by the entries of the preceding table. This is an important advantage, because without this hierarchical structure, the search time would be linear due to the need of checking all the names in the database, one by one. Such a full scan of all the records is a very costly database query. Alternatively, if we used a simple sorted list of all distinct names, this list would still require a full-scan (i.e., each distinct name should be fetched from this list and compared with the query name); our approach, however, can simply prune a node and its offspring nodes if they do not match the queried name (within the required threshold of allowed errors). Moreover, such a sorted list of names would also require continuous updates to keep it sorted as more records are entered into the database.

The advantage of PREFIX is that when we search for approximate name matches, we can eliminate a sub-tree of the above-depicted tree (a sub-tree consists of a node and all of its offspring nodes and branches). Suppose that we search for any similar names with no more than one edit-error to the name “Olkay”. When the algorithm examines level two of the tree, (i.e., the prefix-table T_2), it will find that the node JI is already at a minimum edit distance of two from “Olkay”. Therefore any node that extends from JI-node should not be considered any further. That is, any name that starts

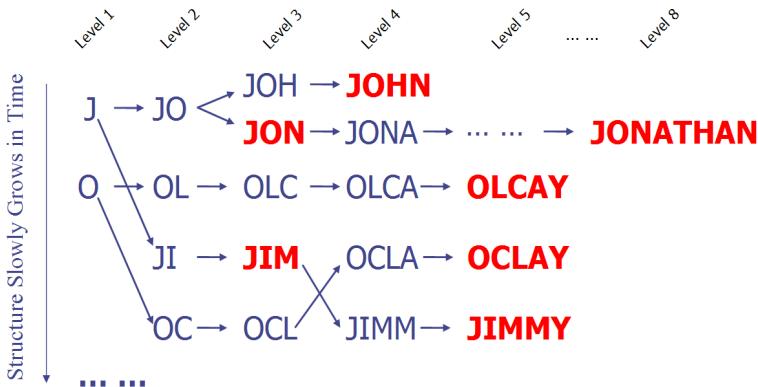


Fig. 9.2. The Name Dictionary Tree obtained by processing “John”, “Jon”, “Jonathan”, “Olcay”, “Jim”, “Oclay”, and “Jimmy”. This tree provides a fast and complete search of all records for approximate matches to queried names. The growth of the dictionary will be much slower compared to the number of records entering the database, and the updates will be very cost-efficient because no sorting will be required when new names are entered into the database.

with a JI is not going to be within the allowable error margin. Consider the best possible matches to “Olkay” that may be pruned by PREFIX: “Jikay” and “Jiolkay”. Neither one looks reasonably similar to the query name “Olcay”. Note that some exceptional prefixes to this end such as “Jr.”, or “Mr.” might be handled separately by preprocessing (e.g., “Jrolkay” may stand for “Jr. Olkay”).

Notice that there are some crisscrosses (e.g., JIMM-node and OCLA-node) in Fig. 9.2, which is also observable in the prefix-table T_3 . It is important to understand why this happens because it will reinforce the comprehension of the PREFIX algorithm. The reason the crisscrosses happen is obvious: when the name “Jim” is entered into the database, it resulted in creation of JIM-node in level 3 of the tree. Then, OCL and OCLA nodes are created in level 3 and 4 of the tree, respectively, when “Oclay” is entered into the database. Then, when “Jimmy” is entered, it created JIMM node in level 4, which falls after OCLA (because the prefix-tables are never sorted as its not needed). The crisscross occurs because JIMM-node is connected to JIM-node, which was created before OCL-node which links to OCLA-node.

It is worth mentioning, that despite the fact that theoretically there is no limit on the length of a name and the number of levels of the tree (prefix-tables) that our method can construct, in practice the length of a given name is less than about 15–20 characters long (this can be a bound established by the creator of the database). After examining the first m characters of the names, we would get a good bound (estimate) on the actual number of edit errors. This would be sufficient to decide if the current node should be eliminated or not. Therefore, to reduce the extra storage by PREFIX, one should consider creating a smaller number of levels of the tree.

9.3.2 Searching the Prefix-Dictionary

In this section we are describing an incremental approach for calculating the edit distance between a query name and the names that are already residing in the database.

To do so, consider the computation of the edit distance of JON to JOHN. In this example, suppose JON is the query string. In our algorithm the edit distance between JON (the query string) and JOHN (the name in the database) is found in an incremental way: JON vs. J, JON vs. JO, JON vs. JOH, and finally JON vs. JOHN. This approach of calculating the edit distance computes a matrix of edit distances, column by column, in a way that a child in the prefix-table T_n does not need the whole matrix; it can simply inherit the previous column computed by its parent in the prefix-table T_{n-1} . In other words, as shown in Table 9.2, once we compute the first column corresponding to the edit distance of JON vs. J, this column can be used in computing the edit distance between JON vs. JO (second column of the matrix shown in Table 9.2), it can be used, as well, in computing JON vs. JI, or JON vs. any name that starts with a J for that matter. Thus, our approach allows us to minimize the number of the costly column computations required by the edit-distance procedure used in PREFIX. Otherwise, the column computations by the edit-distance would be unnecessarily replicated. In Table 9.2, the key observation is that the minimum in a column is no more than the actual distance and it is non-decreasing. The cell with the actual edit distance is marked by a star symbol.

Table 9.2. Edit Distance Computation in PREFIX

	T_1	T_2	T_3	T_4
	J	O	H	N
J	0	1	2	3
O	1	0	1	2
N	2	1	1	1*
Min	0	0	1	1

Let us now assume that the query is to find “Olkay” with $k=1$ (up to one error is allowed). In the first level of the tree given in Fig. 9.3 (representing the prefix-table T_1), there are two nodes: J and O. The J-node has a minimum possible edit distance of one (because there is no J anywhere in “Olkay”), whereas the O-node has a minimum possible edit distance of zero (i.e., it is possible that the O-node leads us to an exact match). Then, we visit all the children of nodes of J and O. The prefix-table T_2 (level 2 of the tree) has 4 nodes: JO, OL, JI, OC. The minimum possible edit distances are 1, 0, 2, and 1, respectively. Therefore, we can eliminate the JI node and all the nodes that are branching off this node from further consideration, since their edit distances from “Olkay” will simply be equal to or greater than the edit distance calculated at the JI node. Now let us examine Level 3 (the prefix-table T_3) for the children of JO, OL, and OC. These children are JOH, JON, OLC, and OCL. Minimum possible edit distances are 2, 2, 0, and 1, respectively. Note that the minimum possible edit distance between OCL-node and the query “Olkay” is only one because OCL-node might possibly have OCLKAY as a descendant node, which would be at an edit-distance of one to “Olkay”. Therefore, JOH and JON nodes are pruned in conjunction with their descendant nodes. Note that pruning the sub-tree that contains the nodes from JON to JONATHAN implies that these nodes will not be even read from the dictionary. That

leaves only the children of OLC and OCL for further consideration. The only child of OLC is OLCA and the only child of OCL is OCLA. Their minimum possible edit distance to the query string “Olkay” is 0 and 2, respectively. Therefore, the node OCLA is pruned, leaving out the only surviving node OLCA and the child of OLCA is OLCAY which is the only node that is also a valid name (not only a prefix). Thus, one of the (in fact, the only) matching names is “Olcay”. Hence, as Fig. 9.3 shows, the PREFIX algorithm in this example will find a close match and in doing so it will not need to visit the nodes that are light-colored.

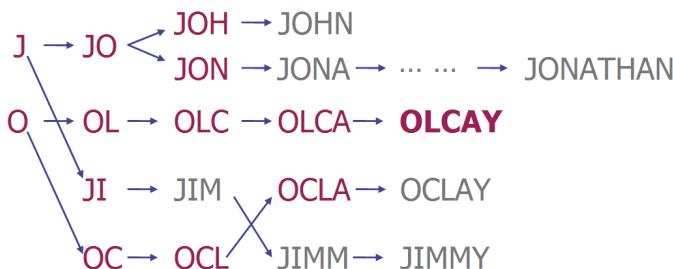


Fig. 9.3. Searching the prefix-dictionary for “Olkay” with $k=1$, where “Olcay” is a valid match

In Fig. 9.4, we are depicting two curves. The linearly increasing curves show the number of prefixes of certain length, if all the prefixes were possible prefixes for names. The saturating curve shows the actual number of name prefixes in the

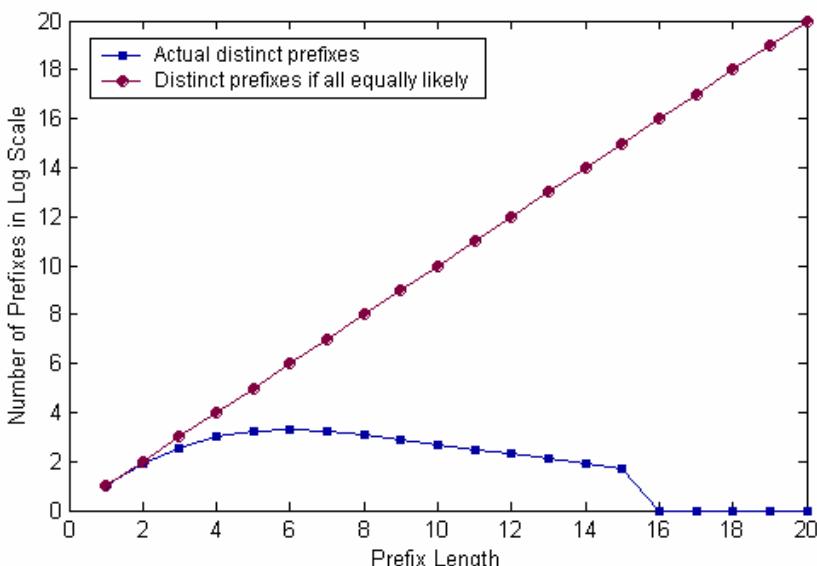


Fig. 9.4. The number of distinct prefixes in the last names available in the FINDER database as a function of the length of the prefix

FINDER database (consisting of about half a million records of names, some of which happen to contain typos and spelling errors). As shown in Fig. 9.4, the number of distinct prefixes actually goes down as the length of the prefixes increase. This phenomenon is due to the fact that there are not many long person names in the FINDER database (or in any database that contains names for that matter). The curve of Fig. 9.4 indicates that even for short prefixes (e.g., of length up to 5 or 6), not all the possible prefixes are encountered in a collection of names, much unlike the case for randomly generated strings. As common names (e.g., “Michael” and “Johnson”) repeatedly occur in the database and very little bizarre permutations of letters (e.g., WXY or RRR) can be found, the number of entries in each level of the prefix-dictionary is expected to grow only gradually, thus increasing the search time only slightly.

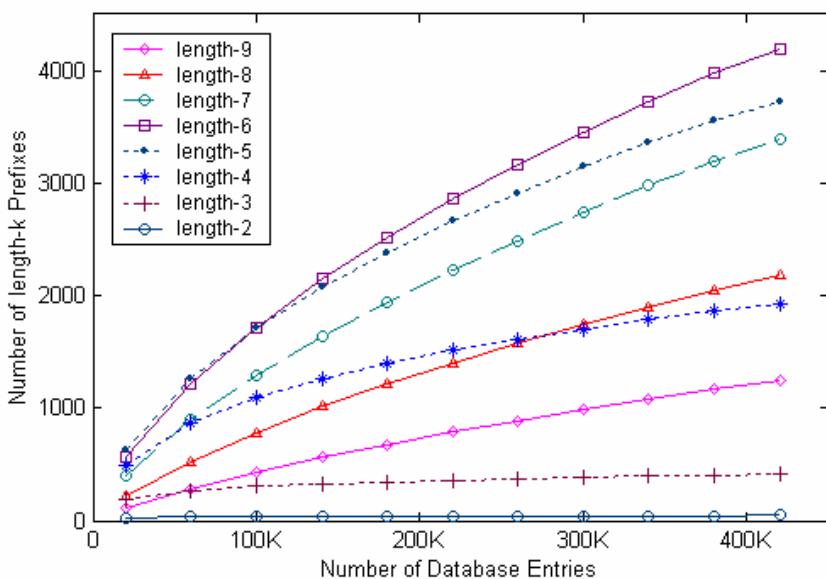


Fig. 9.5. The number of distinct prefixes in the FINDER database as a function of the number of records in the database

Fig. 9.5 shows the increase in the number of prefix table entries as the number of records in the FINDER database grows. Note that the curves depicting very short prefixes saturate very quickly, as all possible prefixes are exhausted fairly quickly. The curves depicting the number of the longer prefixes have a small increasing slope, which causes minor increase in the search time, as more records are entered into the database. Note that for person names, we expect these curves to reach saturation (plateau) due to the implausibility of new incoming names having completely new prefixes compared to name prefixes of already existing names in the database (e.g., we do not expect names to have prefixes such as ZZ or XYZ). As a result, we anticipate that as the number of entries grows in the FINDER database, the number of prefixes will not grow as fast. The growth rates for the distinct prefixes can be expected to be

higher for a database of company names including names such as “AAATOWING”, “AACLEANING”, “ABCFINANCIAL”, “ABCNEWS”, and so on, because the company names are less structured and restricted with respect to person names.

To use the PREFIX algorithm for a full name query rather than a single string query (such as a last name or a first name only), we apply the following steps: (1) build prefix-dictionary for the last names; (2) for a given full name query, search the tree for similar last names; (3) apply edit-distance algorithm on the returned records to obtain the ones that also have matching first names. In step 1, we could have built a prefix-tree for first names and in step 2, we could have obtained matching first names by scanning this tree; however, it would not have been as efficient as the stated PREFIX algorithm because first names are, in general, less distinct; consequently, by using first names at the beginning of the search process would have reduced our capability of filtering out irrelevant records.

9.4 The ANSWER Algorithm

The PREFIX algorithm offers a very efficient search of names. Nevertheless, it does not provide any direct way of utilizing a given first name along with the last name of a query because it does not use the first name information during the search of the tree.

We propose the ANSWER (Approximate Name Search With ERrors) algorithm for fast and still highly accurate search of full names based on the PREFIX idea. In the process of building the prefix-dictionary, ANSWER takes every full name in the database, and using the PREFIX algorithm, it creates required nodes and links for the last names in the tree. It also augments each node in the tree by 26 bits, each bit representing whether any last name on that branch has an associated first name starting with the corresponding letter in the alphabet. For example, if the last name “Doe” could be found in the database only with the first names “Jon”, “John”, and “Michael”, the corresponding nodes in the “Doe” branch in the tree would be “linked” with “J” and “M”, meaning that the last name “Doe” can only have first names starting with “J” or “M”.

This architecture allows early (before the edit-distance exceeds the predefined threshold k) pruning of tree nodes based on the first letter of the first name of the query. For example, if the query name was “John Doe”, ANSWER would prune, say the F-node, if there were no last names starting with letter “F” associated with a first name that starts with “J”, the first letter of “John”. Based on our preliminary experiments and what we deduced from the literature [12, 18], it is unlikely that both first name and last name initials are incorrect (e.g., “Zohn Foe” is not an expectable match for “John Doe”). On the other hand, PREFIX would not prune the F-node right away because it does not take into consideration the first name at all, and there could be a last name similar to DOE that starts with “F” (e.g., “Foe”). Thus, PREFIX would scan more branches and take longer than ANSWER. Moreover, even though ANSWER is not an exhaustive search algorithm, it exhibits high hit rate as explained in the following section.

9.5 Experimental Results

In order to assess the performances of our exhaustive search engine PREFIX and its heuristic version ANSWER, we conducted a number of experiments. After creating the prefix-dictionary tree, we queried all distinct full names available in the FINDER database and measured the *time* taken by PREFIX and ANSWER in terms of the number of columns computed in the calculation of edit-distance calls (how edit-distance computation works was explained in Sect. 9.3.2). This way, the effect of factors such as operating system, database server, programming language, are alleviated. Furthermore, we compared PREFIX’s and ANSWER’s performance with other name matching techniques. In particular, we compared PREFIX and ANSWER with two other methods: (1) filtering based soundex approach applied only last name (SDXLAST); (2) filtering based soundex approach applied on first or last names (SDXFULL).

In our experiments, we also measured the *precision* and the *recall* of the compared methods. Recall (also known as “hit rate”) is the ratio of the true positives identified versus the total number of actual positives. Precision is the ratio of the number of true positives identified versus the number of candidate records that are selected by a method.

SDXLAST is a simple method that is based on the commonly used soundex schema that returns records with soundex-wise-matching last names, and then applies the edit-distance procedure (just as in our methods, the edit-distance calls terminate the computation once the maximum allowable edit errors k is exceeded) to the last names to eliminate the false positives, and applies the edit-distance procedure once more on the first names of the remaining last names, in order to obtain the final set of matching full names.

It is worth noting though, that the hit rate obtained by using only the soundex-matches for the last names is insufficient due to inherent limitations of the soundex scheme [18]. For example, searching for “Danny Boldwing” using SDXLAST would not return “Danny Bodlwing” because the soundex code for “Boldwing” does not match the soundex code for “Bodlwing”. Therefore, we devised an extension of SDXLAST in order to enhance its hit rate. We called this new method SDXFULL. SDXFULL selects records with soundex-wise-matching last names *or* soundex-wise-matching first names. As a result, if “Danny Boldwing” is the input query, SDXFULL would return not only “Danny Bodlwing” and “Dannie Boldwing” as possible true positives, but it would also return many false positives such as “Donnie Jackson” or “Martin Building”. These false positives will be eliminated (by applying edit distance calculations to all these returned records) as in the SDXLAST method. Thus, it is expected that SDXFULL has a higher recall (true positives) than SDXLAST but longer run-time since it also returns a larger number of false positives). The low recall rate of soundex is the reason for not comparing our methods with other phonetic-type matching methods, such as *Phonix* [18]. Phonix assigns unique numerals to even smaller groups of consonants than soundex, and it is thus expected to have an even lower recall rate than the already unacceptable recall rate observed in SDXLAST [18].

In comparing the computational complexity of the soundex-based methods we ignored the time that is required to query records that match soundex of the first names or soundex of last names in our database. Hence, our comparisons are unfair to our

proposed methods, but not to the point that it would change our conclusions from this comparison. This is due to the fact that the time to run the soundex query is almost negligible compared to the time required to calculate edit distances of the names returned by the query.

Our database contains about half a million (414,091 to be exact) records of full names, out of which 249,899 are distinct. In order to evaluate the behavior of these four name matching methods as the number of records in the database increases, we have applied each one of the aforementioned methods (PREFIX, ANSWER, SDXLAST, and SDXFULL) to the database at different sizes. For our experiments, we have chosen 25% (small), 50% (medium), 75% (large), and 100% (x-large) of records as the working set sizes (see Fig. 9.6). Note that a different prefix-dictionary is used for different set sizes, as the number of records in the database expand from the small to x-large sizes. We used the PREFIX algorithm as the baseline for our algorithm comparisons, since it performs an exhaustive search. For our experiments we used a maximum number of allowable edit-distance of 2 ($k=2$), for both last and first names. Thus, for every query by the exhaustive search, we have selected from the database all the available full names of which neither the last nor the first name deviates by more than two errors from the last and the first names, respectively, of the query.

Fig. 9.7 plots the graph of average run-times for queries of each approach as a function of the database size. Note that in some applications, the hit-rate of the search can be as important as (if not more important than) the search time. Therefore, in order to quantify the miss rate, we have also computed the average hit-rates for these methods (see Fig. 9.8). SDXLAST is the fastest search; however, it has the lowest hit-rate amongst all the algorithms. Furthermore, SDXLAST's hit-rate is unacceptably low for many applications [12, 18]. The ANSWER search is the next fastest for large

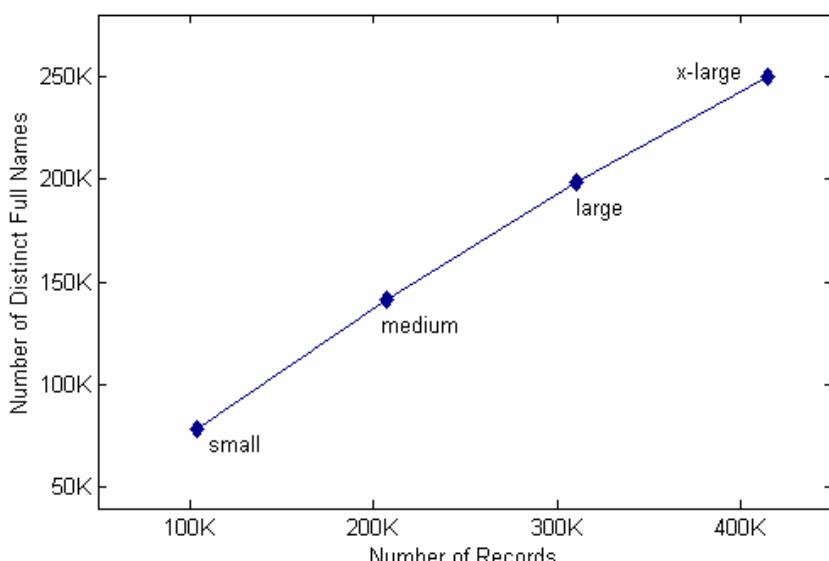


Fig. 9.6. The various sizes of the databases used for the experiments

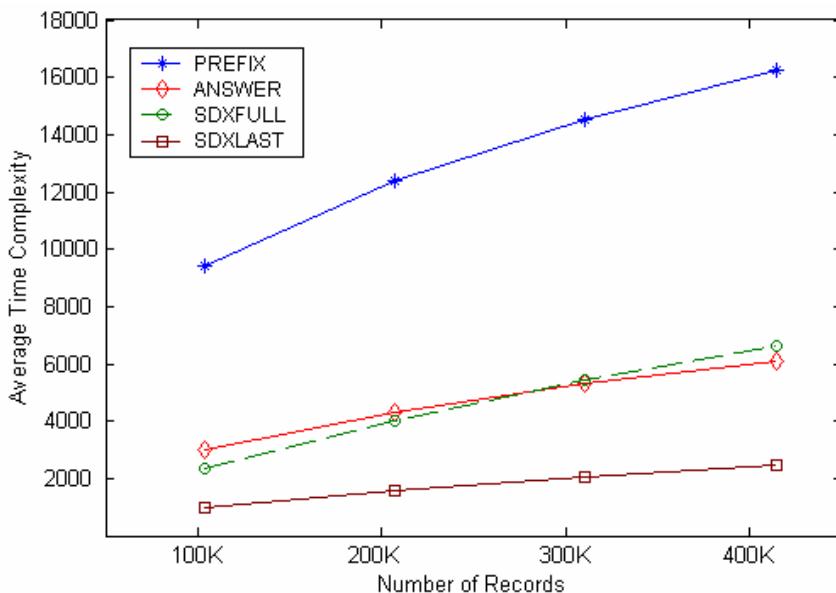


Fig. 9.7. Run-times of the compared name searching approaches versus the database size

Table 9.3. Exemplary queries for which ANSWER returns legitimate matches, which SDXFULL misses

FIRST NAME STRING	SDX	LAST NAME STRING	SDX	
SAMANTHA	S553	ETCHEESEN	E322	QUERY
SAMATHA	S530	ETCHEENSON	E325	MISSED
YOUSAF	Y210	KODYXR	K326	QUERY
YOUSE	Y200	RODYXR	R326	MISSED
KOLLINS	K452	LISA	L200	QUERY
KOLLION	K450	TINA	T500	MISSED

databases (except for SDXLAST, which has a small hit rate). ANSWER is also significantly more accurate than the SDXFULL search. SDXFULL executes a simple search that fails when there are errors in both the last and the first names (this happens in increasingly more than 15% of the records). For instance, some of the records that are found by ANSWER but missed by SDXFULL are “Samantha Etcheeson” versus “Samatha Etcheenson” or “Yousaf Kodyxr” versus “Youse Rodyxr”, as shown in Table 9.3. Note that the slopes of the curves in the Fig. 9.7 and Fig. 9.8 are important because they emphasize the scalability of each name searching approach as the database expands.

Fig. 9.9 shows the precision of each method (the ratio of the number of true positives to the number of candidate records). For PREFIX and ANSWER, the candidate records are the distinct full names with matching last names (found in the tree). For

SDXFULL, they are the distinct records with soundex-wise-matching last names or soundex-wise-matching first names. For SDXLAST, they are the distinct records with soundex-wise-matching last names. The problem with very low precision is two-fold: (1) it requires that a large number of records ought to be examined by calculating their edit distance from the query name and this contributes to the needed run-time (CPU time); (2) too many records selected by a query keep the server's memory engaged to the task. This can be detrimental in a shared database environment where multiple users might be sending queries simultaneously (typical of the FINDER environment, as well as other applications in law enforcement and homeland security). ANSWER's precision is superior to all the other techniques because the last names of all the candidate records satisfy two strict conditions: (1) small edit-distance to query last name, (2) existence of at least one association of a name that starts with the initial letter of the query first name. ANSWER uses these two conditions in a very efficient way by using the prefix-tree that eliminates irrelevant records quickly, while returning candidates that are very likely to be true positives.

Effectiveness [14] is a frequently used combination value of P (precision) and R (recall). It is computed as $2 \cdot P \cdot R / (P+R)$. Fig. 9.10 shows the plot of effectiveness as a function of the database size. ANSWER is the most effective solution to name matching, and the curves in Fig. 9.10 indicate that it will remain to be the most effective as the database grows further.

Another advantage of ANSWER is that it has a much smaller standard deviation of search time, whereas the execution time of SDXFULL can be erratic, as shown in Fig. 9.11. One of the reasons for the latter approach having increasingly higher standard

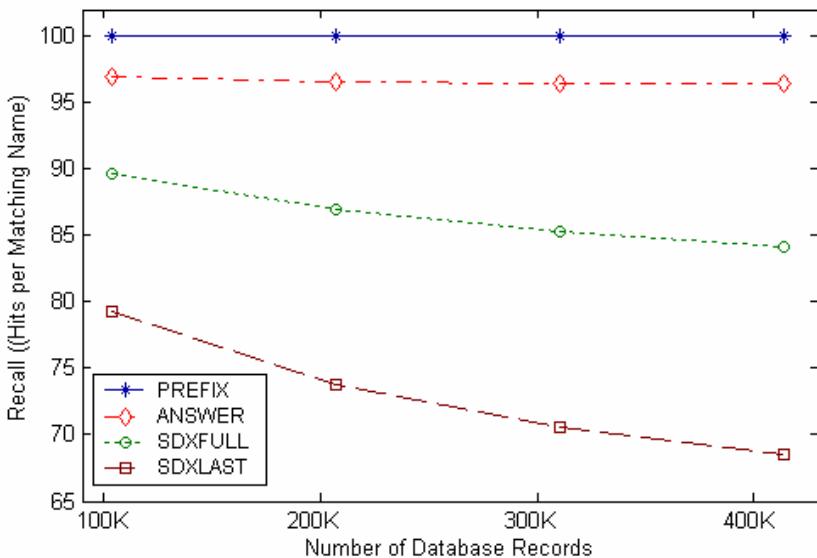


Fig. 9.8. Recall rates of the SDXLAST, SDXFULL, PREFIX and ANSWER versus the database size

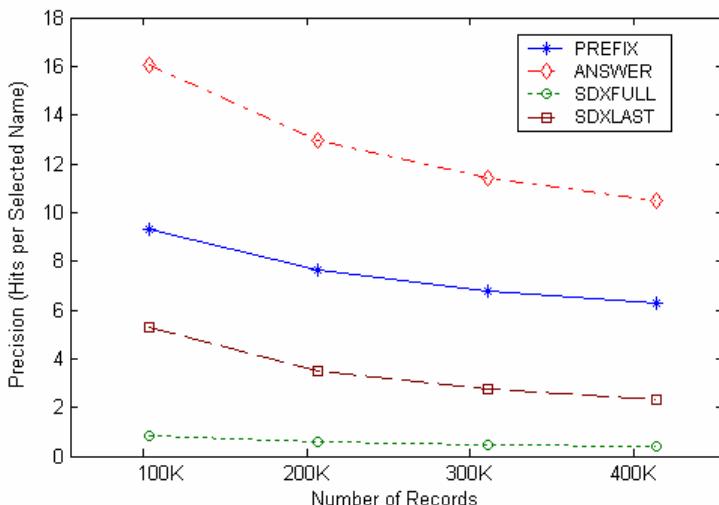


Fig. 9.9. Precision rates of the compared name searching approaches versus the database size

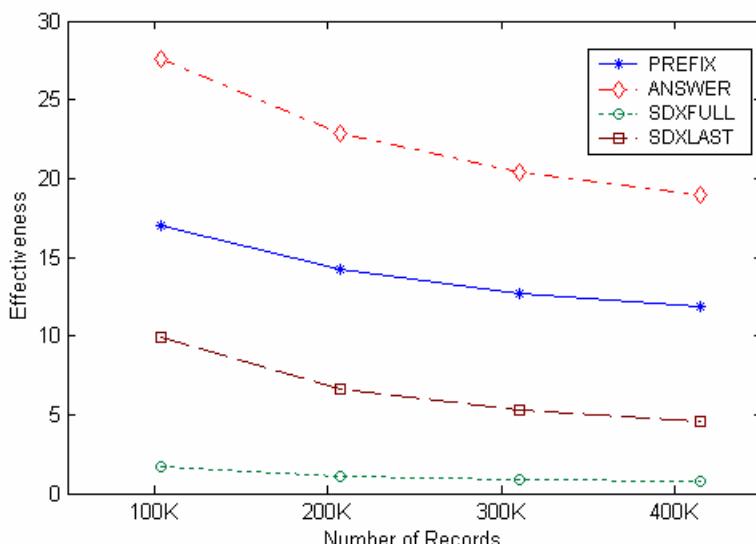


Fig. 9.10. Effectiveness of the compared name searching approaches versus the database size

deviation as the number of records increase is that it returns progressively more records due to the first name matches. Consider, for example, the search for “Michael Desansi”. There are going to be progressively more records in the database for which the soundex of the first name matches the soundex of “Michael”. This will cause SDXFULL to return many “Michael”s (or any record with some other first name with the same soundex code) with possibly completely incongruent last names. The

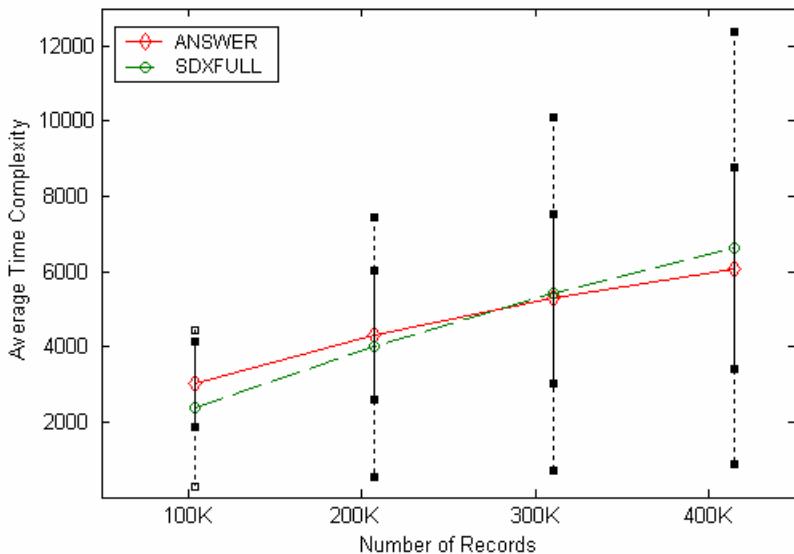


Fig. 9.11. Means and standard deviations of run-times of the ANSWER and SDXFULL

performance collapses with the search of short names. Soundex of “Joey” will match many short names starting with a “J”, which will cause many records to be selected even though their last names are completely different than the last name of the queried entry.

9.6 DBMS (Database Management System) Implementation

ANSWER offers a very efficient search of names. Its database system implementation is not as fast, however it still remains to be a crucial tool of querying because it is a full search tool. Other techniques we could use are either partial searches with 60%–70% recall rates (such as soundex or phonix), or very slow (e.g. one pass over all the distinct full names with Levenshtein comparison takes about 10 minutes in database implementation). Soundex takes about half a second to query a name. However, it misses matching records due to its weak heuristicity.

This does not come as a surprise because it is a problem in general that algorithms implemented offline that use data outside the database system can employ efficient structures that reside in memory and a minimal number of database scans, and thus exhibit better performance than the equivalent database implementations. From the performance perspective, data mining algorithms that are implemented with the help of SQL are usually considered inferior to algorithms that process data outside the database systems. One of the important reasons is that offline algorithms employ sophisticated in-memory data structures and can scan the data as many times as needed without much cost due to speed of random access to memory 18. Our initial experiments with early efforts of DBMS implementation resulted that the run time for ANSWER for $k=1$ is under a second. For $k=2$, the search time is in order of 5 seconds.

Disk-access is a very costly operation in database systems. Therefore, we will have to reduce the number of database accesses needed for searching the tree. One idea is to use the breadth-first search algorithm. For a query with a searched name of length n and MaxError tolerable edit distance, the upper bound of the number of database accesses is, therefore, $n + \text{MaxError}$.

In order to further reduce database access, when we import the names into prefix tables, we can load the names in partial order so that the similar names are stored together in prefix name tables. Names whose initial letters are “AA” are firstly imported, then those with “AB”, “AC”, and until “ZZ”. This way, when we query names, database server will automatically load and cache the data blocks with similar prefixes, thus we can facilitate I/O accesses by reducing the number of memory blocks to be retrieved.

9.7 Summary, Conclusions and Future Work

The advantages of introducing effective and efficient analytical tools that compensate for inherent data integrity conditions and address the processing of unreliable and inaccurate data within law enforcement databases are obvious. Increasing the probability of discovering police records that may be obscured by erroneous data is critical. The ability to obtain information about an individual on a terrorist watch-list that has numerous aliases or a challenging foreign name has major implications for practice. The matching tool will find potential matches for any record string thereby reducing the risk of missing a key record if the social security number is entered incorrectly, a birth date, race or sex. Data integrity tools that deal with “dirty data”, residing in federated distributed systems like FINDER increase confidence that critical “dots” will not be missed.

Moreover, dirty data is a necessary evil in large databases, which are used in a variety of application fields such as homeland security, medical, among others. In that case, a search for specific information by a standard query fails to return all the relevant records. The existing methods for fuzzy name matching attain variable levels of success related to performance measures, such as speed, accuracy, consistency of query return times (robustness), scalability, storage, and even ease of implementation.

Name searching methods using name-by-name comparisons by edit distance (i.e., the minimum number of single characters that need to be inserted into, deleted from, and/or substituted in one string to get another) throughout the entire database render the desired accuracy, but they exhibit high complexity of run time and thus are non-scalable to large databases. In this paper, we have introduced a method (PREFIX) that is capable of an exhaustive edit-distance search at high speed, at the expense of some additional storage for a prefix-dictionary tree constructed. We have also introduced a simple extension to it, called ANSWER. ANSWER has compared very favorably in comparison with a variety of soundex-based methods. Even though the soundex methods can be faster and easy to implement, our experiments as well as studies in the literature (see Sect. 9.1 for a review) have shown that they produce many false-positive and false-negative matches. In addition, the search time for the soundex methods can be erratic as the phonetic-based density of certain names is higher than others (name dependant).

The ANSWER method returns results in a much more consistent search time (i.e., the variability in search time is lower) than the soundex-based methods, and turns out to be faster when applied to large databases. In particular, the ANSWER name search has run-time complexity comparable to soundex methods, and it maintains robustness and scalability, as well as a comparable level of accuracy compared to an exhaustive edit distance search. In light of the experimental evidence presented in this paper, we believe that ANSWER's advantages will hold true compared to any other phonetic-based name searching approach (e.g., Phonix). ANSWER has been tested, and its advantages have been verified, on real data from a law-enforcement database (FINDER). Furthermore, it should be evident that ANSWER's potential extends beyond the field of law enforcement, to other application fields where a name searching algorithm with low search time, high accuracy, and good precision is needed.

A number of related issues such as records with last and first names switched, street addresses entered into the name fields by mistake, middle names etc. will also be the focus of our work. Extension of ANSWER to other types of dirty data in the law enforcement databases, such as addresses, stolen item/article names/descriptions/brand names, etc, will require building separate dictionaries for each one of these data-types.

We also propose to extend the ANSWER algorithm to distributed environment as well. This will be accomplished by creating a disparate name dictionary tree for each agency's data. When a query is broadcasted, each agency will search its own name dictionary and send back the matching names. This brings up the special query method called *subscribed name queries*. When a name is searched but not found by a subscribed name query, it will place a trigger in the name dictionary and whenever that name is entered into the dictionary, it will send the information to the investigator that subscribed the query. This trigger method is much more efficient and timely than querying the whole database for the name at *regular* intervals (e.g., at midnight).

Acknowledgements

The authors wish to acknowledge helpful discussions with detectives from Orange County Sheriff's Office. This work was supported, in part by, the Orange County Sheriff's Office, the State of Florida, and National Institute of Justice grant 2004-IJ-CX-K030.

References

1. Aoe, J., Morimoto, K., Shishibori, M., Park, K.: A Trie Compaction Algorithm for a Large Set of Keys. *IEEE Transactions on Knowledge and Data Engineering* 8(3), 476–491 (2001)
2. Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., Fienberg, S.: Adaptive name matching in information integration. *IEEE Intelligent Systems* 18(5), 16–23 (2003)
3. Durham, I., Lamb, D., Saxe, J.: Spelling correction in user interfaces. *Communications of the ACM* 26(10), 764–773 (1983)
4. Jaro, M.A.: UNIMATCH: A Record Linkage System: User's Manual. Technical Report. U.S. Bureau of the Census, Washington, DC (1976)

5. Kim, W.: On Database Technology for US Homeland Security. *Journal of Object Technology* 1(5), 43–49 (2002)
6. Levenshtein, V.L.: Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics, Doklady* 10, 707–710 (1966)
7. Maxwell, T.: Information, Data Mining, and National Security: False Positives and Unidentified Negatives. In: Proceedings of the 38th Hawaii International Conference on System Science (2005)
8. Mihov, S., Schulz, K.U.: Fast Approximate Search in Large Dictionaries. *Journal of Computational Linguistics* 30(4), 451–477 (2004)
9. Monge, A.E., Elkan, C.P.: An Efficient Domain-Independent Algorithm for Detecting Approximately Duplicate Database Records. In: Proceedings of the ACM-SIGMOD Workshop on Research Issues on Knowledge Discovery and Data Mining, Tucson, AZ (1997)
10. Navarro, G.: A Guided Tour to Approximate String Matching. *ACM Computing Surveys* 33(1), 31–88 (2001)
11. Newcombe, H.B., Kennedy, J.M., Axford, S.J., James, A.P.: Automatic linkage of vital records. *Science* 3381, 954–959 (1959)
12. Pfeifer, U., Poersch, T., Fuhr, N.: Searching Proper Names in Databases. In: Proceedings of the Hypertext - Information Retrieval – Multimedia (HIM 1995), vol. 20, pp. 259–276 (1995)
13. Taipale, K.A.: Data Mining & Domestic Security: Connecting the Dots to Make Sense of Data. *The Columbia Science & Technology Law Review* 5, 1–83 (2003)
14. van Rijsbergen, C.J.: *Information Retrieval*, 2nd edn. Butterworths, London (1979)
15. Wang, G., Chen, H., Atabakhsh, H.: Automatically detecting deceptive criminal identities. *Communications of the ACM* 47(3), 70–76 (2004)
16. Wilcox, J.: Police Agencies Join Forces To Build Data-Sharing Networks: Local, State, and Federal Crimefighters Establish IT Posse. *Government Computer News* (September 1997)
17. Winkler, W.E.: The state of record linkage and current research problems. In: Proceedings of the Section on Survey Methods of the Statistical Society of Canada (1999)
18. Zobel, J., Dart, P.: Finding approximate matches in large lexicons. *Software-Practice and Experience* 25(3), 331–345 (1995)

Online Resources

A list of online resources that are relevant to the topic, e.g. Web sites, open source software, datasets, testbeds, demos, ontologies, benchmark results, golden standards, online publications, and reports, etc.

The Florida Integrated Network For Data Exchange And Retrieval:

- <http://finder.ucf.edu>
- <http://druid.enrgr.ucf.edu/datassharing/PresentationMay2005.ppt>
- <http://www.pstc.ucf.edu>

Description and Algorithms For Calculating The Edit Distance:

- <http://www.merriampark.com/ld.htm>
- http://www.cut-the-knot.org/do_you_know/Strings.shtml
- http://en.wikipedia.org/wiki/Levenshtein_distance

String-matching Demos:

- <http://www.cs.unm.edu/~luger/ai-final/demos.html>
- <http://www-igm.univ-mlv.fr/~lecroq/seqcomp/node2.html>

Exemplary Criminal Justice Websites Relating To Name Matching:

- <http://www.ncjrs.gov>
A sample grant opportunity involving name matching: Information-Led Policing Research, Technology Development, Testing, and Evaluation
<http://www.ncjrs.gov/pdffiles1/nij/sl000769.pdf>
- <http://www.cops.usdoj.gov>
A sample need for name matching: “Imprecise victim names. Detecting re-victimization of individuals involves matching recorded victim names. But victims may alter their name, move after being victimized, be victimized in different locations, …”
<http://www.cops.usdoj.gov/txt/pop/e07055803.txt>
- <http://www.ojp.usdoj.gov/nij>
A sample product with partial name matching capabilities: “Person Name: Enter a complete or partial name (using the appropriate wildcards) button executes the search and displays any incidents that match the search”
www.ojp.usdoj.gov/nij/maps/pdf/facilitycop.pdf

Links to some companies offering data/name matching solutions:

- <http://www.identitysystems.com>
- <http://www.datactics.com>
- <http://www.namethesaurus.com>

Questions for Discussions

1. What are the main sources of dirty data in databases?
2. Quantifying dirtiness of a database, say if we learn that 10% of our data entries have typos in them, is a critical task especially in compound queries. Discuss why. If two databases are both 10% dirty, then what is the expected level of dirtiness of the compound (merged) database?
3. What are the advantages and disadvantages of using a dictionary of entered names rather than a dictionary of all known names?
4. How can subscribed name queries be implemented in a database environment such that it can notify the subscriber immediately as soon as the searched data become available?
5. What kind of data-entry errors cannot be detected by Soundex-like methods?
6. In this section we addressed the searching for partial matches, however, cleaning the data is another issue. What are the main difficulties in automatic cleaning of the dirty data?

Discovering Investigation Clues through Mining Criminal Databases

Patrick S. Chen

Center for Strategic and Industrial Studies,
Dept of Information Management, Tatung University, Taiwan
chenps@ttu.edu.tw

Abstract. The law enforcement holds a large quantity of data that are records of official operations or private materials of the people. These data can be used for increasing benefits of the people or enhancing the efficiency of governmental operations. Data mining is one of the emerging techniques to manipulate huge amount of data. In this paper we will apply to this technique to the data of stolen automobiles to explore the unknown knowledge hidden in the data and provide this knowledge to transportation, insurance as well as police agencies for decision supports. The data we use are abstracted from 378 thousand records of stolen automobiles in the past eleven years in Taiwan. After constructing a data warehouse, we apply to the technique of classification, clustering, association rule, prediction, data generalization and summarization-based characterization to discover new knowledge. Our results include the understanding of automobile theft, possibility of finding stolen automobiles, intrigue in theft claims, etc. The knowledge we acquired is useful in decision support, showing the applicability of data mining in public affairs. The experience we gathered in this study would help the use of this technique in other public sectors. Along with the research results, we suggest the law enforcement to consider data mining as a new means to investigate criminal cases, to set up a team of criminal data analysis, to launch a new program to crack down automobile thefts, and to improve the quality of criminal data.

Keywords: Data Mining, Data Warehousing, Databases, Criminal Data, Stolen Automobiles.

10.1 Introduction

Law enforcements around the world have accumulated a large quantity of data from their daily operations and crime investigations. They have also taken advantages of various information technologies to investigate all kinds of crimes for years. A lot of database technologies, management information systems, and expert systems have been used in criminal-justice data analysis. In addition, a few agencies have implemented intelligent systems and high-level data analysis techniques in this direction, but the utilization of criminal databases is no by means fully explored.

The effectiveness of utilizing database techniques in criminal investigation is also recognized. The use of database systems for crime-specific cases, such as gang-related incidents, and serious crimes, such as homicide, aggravated assault, and sexual crimes, has been proven to be highly effective [16, 26; 38]. Most law enforcements have a lot of database system to store various kinds of criminal data, and many police

officers have queried criminal databases to obtain useful information to investigate crimes. These data often contain implicit knowledge or unveiled information that is helpful in decision-making. Recent researches show that law enforcements are more and more aware of the value of these data. Though, till now, the data are prepared mostly for simple queries, many attempts are conducted to arrange them for higher level processing to implement an intelligence analysis.

Addressing the conversion of information to useful and easily understandable knowledge is a powerful aspect of data mining that will become a common practice in law enforcement some day.

There is a newly developed information technology, *data mining*, in the past few years. Data mining is a data analysis method that extracts meaning information from a large quantity of data to establish effective business statistics. Most successful applications of data mining can be found in logistics, financial businesses, and retail businesses. In this chapter we are going to introduce data mining techniques to crime investigation. But, in order to achieve effective use of these techniques, we have to consider in the following aspects:

- (1) What techniques will be applied in this research?
- (2) Why do we need these techniques?
- (3) When can these techniques be used?
- (4) To what extent are these techniques effective?
- (5) What theoretical foundation is to be considered?

All these topics are to be discussed in this chapter. We will illustrate our arguments accompanied with an example, so the reader will gain a concrete impression on the subject. The example is an application of data mining on automobile thefts. The data we used in this research is a collection of 378 thousand records of automobile thefts¹ recorded in Taiwan. They were collected in the time span from January 1991 to December 2001. The sense made by this example is automobile theft represents a major crime problem in Taiwan. In 2004, estimated 217 thousand automobiles (incl. motorcycles) are stolen, 65 percent of the total theft cases². The dark figure, arising from that victims are unwilling to report the cases to police, is believed to be higher than the number of reported cases [14]. Automobile thefts bring about a lot of social problems: besides monetary loss, victims suffering from inconvenience and loss of work time. Personal injuries often incur while the thieves conduct joyriding or attempting to evade police. The stolen automobiles are often the transportation tools for other criminals. Insurance companies suffer from big loss from automobile thefts annually, in which deceptions are involved.

10.2 Adequacy of Using Data Mining Techniques in Discovering Investigation Clues

We sometimes distinguish between incidental crimes and serial crimes. While incidental crimes are difficult to control, serial crimes can be more easily observed. The

¹ We thank for the support from National Police Agency in Taiwan for providing the data for research purpose under strict secrecy protection regulations.

² <http://www.npa.gov.tw> accessed in October, 2005.

illegal committing multiple crimes, known as serial criminals or *career criminals*, are a major threat to society. Understanding the behavioral patterns of these career criminals and comprehending them is an important task for law enforcement. According to the experience of criminal investigators, the *modus operandi* (MO) of a criminal can be identified because he used to commit crime in a fixed manner and using similar technique. Due to the limitation of human imagination, a criminal normally does not have many behavior templates. Therefore, we can observe criminal incidents with the similar temporal, spatial, and *modus operandi* features, which possibly come from the same criminal. It is possible to identify the serial criminal by associating these similar incidents. Therefore, analyzing large databases to discover behavioral patterns for identifying certain criminal suspects is reasonable. The criminals often develop networks in which they form groups or teams to carry out illegal activities. This indicates that we are able to develop data mining tools to identify behavioral patterns of crimes and, simultaneously, the law enforcement is able to find effective solutions to prevent those crimes in advance.

Even for the incidental crimes, they often happen in certain environment under certain condition. Understanding this environmental situation we are able to take steps to eliminate vulnerabilities and prevent crimes. Data mining techniques can be used to find environments and conditions favoring criminal behaviors.

10.3 Criminological Foundations

We may obtain results by performing various analysis techniques on large databases, but do these results make sense? Whether the information obtained by data mining are meaningful, can be checked against empirical research results. In this place, theories from criminology are to be referenced. Another purpose of studying criminology in depth is to find related variables to guide our research directions. We review related works and find aspects considered in the pertaining domain. We have to find out all variables involved in a crime that serve in turn as attributes of interest for data mining. Furthermore, we shall find to know to what extent intelligence analysis and data mining can be applied in law enforcement and try to figure out where data mining can be located in criminological study.

As a first example, we noticed that theft has been the most frequently happening crime among others. Criminals often develop networks in which they form groups or teams to carry out illegal activities. This indicates we are able to develop data mining tools to identify behavioral patterns of theft and, simultaneously, the law enforcement is able to find effective solutions to prevent those crimes in advance.

Next example is concerning fraud. Internet fraud has become a serious problem: credit card fraud has been a serious problem for long. Spyware is now an often-seen tool used in business espionage, or Internet phishing. Many conventional crimes are undertaken along with all sorts of disguises. Therefore, an in-depth study on deception theories is important.

In the following we will discuss the theories about (automobile) theft and deception in more details.

10.3.1 Theories on Thefts

A well-known theory about is the *rational choice theory* in criminology. A criminal evaluates the benefit and the risk for committing an incident and makes a ‘rational’ choice to maximize the ‘profit’. Whether a thief conducts his act depends on his judgment of benefit and risk: if the benefit surpluses the risk, a crime may happen. Rational choice theory [15] claims that whether an offender determines to commit a crime or not depends on the use of effort. The definition of the Effort is the use of energy to do something. Rational choice is a decision process in which the offender calculates the risks, costs, and benefits of committing a particular crime in a particular time and place. Empirical work has also represented that offenders commit crimes close to his dwelling [29, 30, 31, 36]. The rationale is that it takes time, money, and effort to travel and overcome large distances [2]. According to routine activities theory [13], opportunity exists when there is a convergence of a suitable target, a motivated offender, and a lack of a capable guardian (Fig. 10.1). These three factors converge in time and space according to the routine activities of individuals as they go about their daily lives. Opportunity is therefore a necessary, though not sufficient, cause of crime. Furthermore, Potchak, et al. [27] argues that effort is a combination of distance and relevant opportunity, and find that when a person is considering and weighing the costs, benefits, and risks of stealing an auto at a particular place and time [15], she or he will need to travel to the corresponding points of opportunity and tries to minimize that distance.

In the study of thefts there is a theory, the *routine activity theory*, in which a crime is considered as the product of an interactive process of three key elements: a ready criminal, a suitable target, and lack of effective guardians [13]. Routine activity theory claims that crime occurs where there are attractive targets (opportunities), motivated offenders, and an absence of capable guardianship. The environment sends out some cues (physical, spatial, cultural, etc.) about its characteristics, and the criminal uses these cues to evaluate the target and make the decision. A crime is usually an outcome of a decision process involving a multi-staged search in the place where the criminal is situated. During the search phase, the criminal associates these cues with his intended ‘target’. The cues form a template of the criminal, providing us a basis for systematic research. Once the template is built, it is self-reinforcing and relatively enduring. Previous work of a routine activity – rational choice approach has shown the importance of weak guardianship and ample opportunity [2, 11]. Routine activity theory also suggests that crime will occur where daily activities create the most numerous opportunities for the most profitable crime and the least chances of detection or arrest [22].

As far as automobile theft is concerned, there is a second theory, the *social disorganization theory*, in which three exogenous factors—poverty, racial and ethnic heterogeneity, and residential mobility—are hypothesized to result in a withdrawal in community social control activities and an increase in delinquent and criminal activities, including automobile theft [2, 11]. The application of social disorganization theory to automobile theft is especially interesting because contrary to other street crimes, automobile theft offending has long been believed to be concentrated among the socially advantaged and thus seems to negate the hypothesis of social disorganization theory. Sanders [32] typified what has commonly been believed when he wrote,

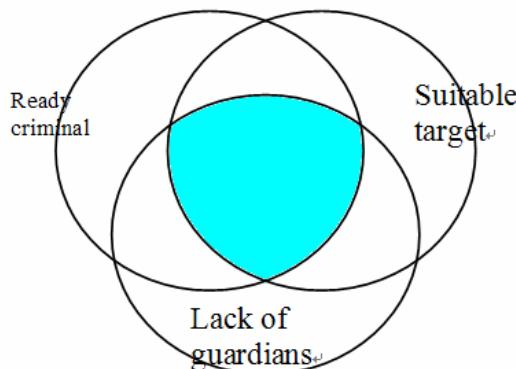


Fig. 10.1. Three key elements of the routine activity theory (The shadowed area denotes conditions for the occurrence of crime)

“Automobile theft is generally committed by white middle-class youths in groups of two or more, largely for ‘kicks’” [10, 33]. The high recovery rate of stolen automobiles also suggests that most theft occurs for recreational and short-term use rather than for profit [12, 23, 25, 35]. Furthermore, Clarke and Harris [12] found strong evidence that “car chopping” and “parts marketing” of stolen automobiles are relatively rare events because patterns of automobile theft are not dependent on the interchangeability of parts, the demand for specific parts, or the market price of parts.

10.3.2 Theories of Deception

Deception is a common, but non-normative event unfolds over time. When engaging in deception, the deceiver must manage to achieve his intention and to be alert to signs of detection from other people. Deception is a continuous process, rather than a one-time event. [39] argued there were several theories worthy of mentioning: (1) *media richness theory*: a deception is often associated with a variety of media [17]. In an environment of fabulous combination of the media and intensive information a person can hardly tell the truth from the falsity; (2) *interpersonal deception theory*: due to the dampening effect of suspicion of the counterpart, a deceiver used to conduct a series of interaction with his partner (target) in order to achieve deception [5]; (3) *interpersonal adaptation theory*: it clarifies and describes the interaction pattern of reciprocity and compensation in dynamic interaction [37] ; (4) *Theories based on cognitive complexity* (e.g., [6]): the complexity of environment or situation is beyond control of a person. This often leads to a successful deception, and the victim is often unaware of the cheating. .

10.4 Systems for Mining Criminal Data

There are currently a number of systems that make use of intelligence analysis and data mining tools for law enforcement. Traditionally, they use information retrieval techniques to collect criminal information. It is especially adequate for mining the

Web; *e-Detective* [7] is a successful example. Other researchers use neural networks to solve problems by developing associations between information objects and being trained to solve problems by comparing known objects with unknown objects [19]. The Timeline Analysis System (TAS) utilizes visualization and time analysis to help analyst visually examine large amounts of information by illustrating cause-and-effect relationships. This system graphically depicts relations found in the data, resulting in trends or patterns [26]. Expert systems that use rule-based information have also been developed to assist in knowledge-intensive applications [3, 4]. For instance, COPLINK has been developed at the University of Arizona's Artificial Intelligence Lab in collaboration with the Tucson Police Department (TPD) and the Phoenix Police Department (PPD). They developed intelligent systems and data mining technologies for capturing, accessing, analyzing, visualizing, and sharing law enforcement-related information in social and organizational contexts [1]. In Taiwan, Criminal Police Bureau has set up a very large database for criminal cases in 2002, and it is a knowledge management system with intelligent queries and full-text mining for case narratives. These systems attempt to utilize all kinds of intelligence analyses and data mining tools to improve information retrieval and knowledge finding capabilities, aiming to assist investigation more effectively and efficiently for law enforcement.

10.5 Methodology in Mining Criminal Data

10.5.1 Research Process

This research relies on the following steps to implement the whole data mining process (Fig. 10.2):

1. *Establishing the Mining Goals:* At first, we establish the mining goals and directions according to domain knowledge of a certain crime and the needs of investigation of the law enforcement. In our demonstrative example, assistance in investigating automobile thefts is intended.
2. *Load Data and Data Transfer:* Data of interest are usually collected by many organizations over a long period of time. They are normally stored in different formats by different kinds of machines. Necessary preprocessing should be done before they can be used. In our example, the data is a collection of 378 thousand records of automobile thefts that cover a ten-year time span from January 1991 to December 2001. They are collected and stored in a NEC database system. The NEC database, a dedicated system developed twenty years ago, is unable to support all kinds of modern data mining tools. We have to export the original data to an open relational database, where we export these data into text files, and then load to Sybase database management system. At the same time we transfer the data attributes.
3. *Create Relational Database and Data Warehouse:* As mentioned, criminal data are to import to relational database management systems for higher-level data analysis. Moreover, we are to build data warehouses. In our example, after that the original data has been transferred into Sybase relational database and saved in table format, we rebuild the relation between these tables. In addition, we

create multi-dimensional structure for automobile theft data to support OLAP³ analysis. The criteria used in the analysis are time, place, automobile type, automobile model, and so on.

4. *Data Preparing and Preprocessing*: Works in this stage include data filtering, attributes reduction, data cleaning⁴, data integration, data transformation, and data reduction.
5. *Data mining implementation*: This research uses various data mining techniques such as classification, clustering, association rule, prediction regression, generalization and summarization-based characterization to explore patterns of the automobile thefts and discover knowledge to prevent related crimes. We make use of training data, test data and evaluation data to create and form best patterns or models.
6. Finally, we have to evaluate the mining results upon domain knowledge of criminology. The results are to be judged according to empirical studies.

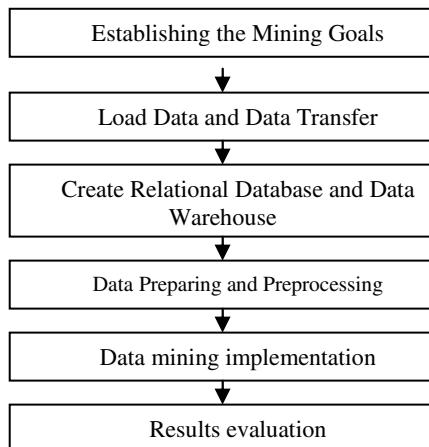


Fig. 10.2. Process of mining criminal data

10.6 Crime Data Mining Techniques

Techniques used for discovering knowledge are information retrieval, classification, clustering, association rule, regression analysis, generalization and summarization, and visualization. We will give them a brief description in this section and use them to mine the stolen automobile data later.

1. *Information retrieval* originated from library science and can be used for finding investigation clues from criminal data.

³ Online analysis and processing.

⁴ Missing values, noisy data and inconsistent data are to be cleaned. Data cleaning is the most labor-intensive work in data mining.

2. *Classification* the technique finds common properties among different automobile thefts and organizes them into predefined classes. We use it to decide which stolen automobiles could be recovered. Often used to predict crime trends, classification can reduce the time required to identify crime entities. However, the technique requires a predefined classification scheme. Classification also requires reasonably complete training and testing data because a high degree of missing data would limit prediction accuracy [9].
3. *Cluster analysis* it is a set of methodologies for automatic classification of samples into a number of groups using a measure of association, so that the samples in one group are similar and samples belonging to different groups are not similar [21]. For example, to map stolen automobiles according some properties into different classes automatically, every stolen automobile in the same class has similar characteristics. These techniques do not have a set of predefined classes for assigning items.
4. *Association rule mining* the technique discovers interesting association or correlation relations among a large set of data items. A typical example is market basket analysis, it runs against transactions database is find sets of items, or itemsets, that appear together in many transactions [21]. We use each label of each attribute as an item, and find association rules among items—for example, a item that represents a stolen automobile reported to the police within a week is frequently occurred with another item that represents a stolen automobile recovered, hence there is a association between the two items.
5. *Prediction Regression* The prediction of continuous values can be modeled by a statistical called *regression*. The objective of regression analysis is to determine the best model that can be related the output variable to various input variables. More formally, regression analysis is the process of determining how a variable Y is related to one or more other variables $X_1, X_2, X_3, \dots, X_n$ [21]. We can apply this technique to find what independent variables (for example, make, model, and year for stolen automobile) can predict depend variable (for example, whether the case can be broken or not).
6. *Data Generalization and Summarization-Based Characterization* data and object often contain detailed information at primitive concept levels. It is useful to be able to summarize a large set of data and present it at a high conceptual [18]. For example, we can use some attribute-oriented induction approach or OLAP to summarize the top 10 models of automobile are frequently stolen in 2005, which can be very helpful for insurance and law enforcement managers.
7. *Visualization:* The purpose of visualization is two-fold: one is to present mining results in a media-rich form, so we might understand the actual state of crime better. Another purpose is to express the relationships among criminals, gangs, their activities, their locations, or their environments. Map with criminal hot spots is a common tool used by law enforcements. Advanced visualization tools support presentation with timeline, geographic information systems, or 3-dimensional graphs.

10.7 Experiments and Mining Results

After having constructed data warehouses we are in the position to analyze their data. The analysis results will be compared with those of empirical studies to examine the reasonability of the mining results.

The experiments made in our case study is as follows: starting from case descriptions such as car type, car manufacturing year, case reporting date, etc. we try to find investigation clues to break the case or to find back the stolen automobile. With this purpose in mind, we present our mining results in this section.

10.7.1 Crime Facets

According to the Facet Analysis Method (FAM) suggested by Chen, a crime can be described in several aspects, *facet*. For example, a crime of cautioning pirated compact discs (CD) may consist of four facets: materials, advertisement, order, and copyrights. Knowing the facets about a crime helps us to understand a crime and, consequently, to know how to investigate it. An entity should have at least a facet, and a facet should consist of one attribute; the structure of a facet is defined as:

$$\text{Entity} = \{ \text{facet_name}_{\{\text{attribute}\}} \}^+$$

In our example, every report of automobile theft is associated with a narrative description, providing us a possibility of performing information retrieval. Automobile thefts involve the following aspects with attributes:

$$\begin{aligned} \text{automobile_theft} = & \{ \text{auto_type}_{\{\text{auto_make, auto_type, ...}\}}, \\ & \text{color}_{\{\text{red, green, silver, ...}\}}, \\ & \text{age}_{\{\text{1-year, 2-year, ...}\}}, \\ & \text{time}_{\{\text{theft-time, report-time, ...}\}}, \\ & \text{place}_{\{\text{city_area_street, ...}\}}, \\ & \text{motive}_{\{\text{fun-riding, money, accessories, ...}\}}, \\ & \text{modus_operandi}_{\{\text{unlocking, destruction, ...}\}} \} \end{aligned}$$

The crime, *automobile_theft*, can thus be described in 7 facets: *auto_type*, *color*, *age*, ..., and *modus_operandi*, in which has *auto_type*, *auto_make*, etc. as attributes. The other facets are defined correspondingly. With the facets we are able to imagine related aspects od a crime.

10.7.2 Association Rule Analysis

- Facts with High Occurrences

With association rule analysis, when we set the minimum support 20 % and minimum confidence 90 % in the association mining of our example. We found a rule

$$\begin{aligned} & \text{"auto type='sedan'" } \longrightarrow \text{"auto category = 'small private car'" } \\ & \quad \text{with support = 70.05\% and confidence=97.38\%.} \end{aligned}$$

Table 10.1. The association between reporting date and case clearance

Association Rule	Support	Confidence
Reported on the day when automobile was stolen —> Case broken	48.80%	68.97%
Reported on the day when automobile was stolen —> Case not broken	21.96%	31.03%
Reported some days after automobile was stolen —> Case broken	18.83%	64.63%
Reported some days after automobile was stolen —> Case not broken	10.42%	35.63%

This shows both attributes *auto type*=‘sedans’ and *auto category* = ‘small private automobile’ are highly related to automobile theft. We further analyze these cases and find that 72% of the stolen automobiles are sedans and 96% of the stolen automobiles are small private automobiles, leading to bias in association rule analysis. We will exclude these two factors to search for other meaningful rules.

- The association between reporting date and case clearance

An automobile theft had better be reported immediately on the day when it was stolen, since the case could be more easily cleared. From Table 10.1 we know the association of same-day-reporting with case clearance has 48.80% support and 68.97% confidence. This is much higher than that of later-day-report followed by case clearance with support=18.83% and confidence=31.03%. This implies the earlier an automobile theft could be reported, the more likely it could be cleared or the stolen automobile could be recovered, but, this is not very significant.

- The relation of car make and loss recovery

Because the proportion of any car make of the total stolen automobiles is not high, we may set minimum support at 0.1% and minimum confidence at 60%. This analysis will reflect which lost cars could likely be recovered (Table 10.2).

- The association of car make and lost car not recovered

When we set minimum support = 0.1% and minimum confidence = 50%, we can find which make of automobiles could not be recovered if the car is lost in Table 10.3.

- The association of car production year and loss recovery

Here we set minimum support = 7% and minimum confidence = 50%, and try to find in which year a car was manufactured that could be found when it is lost. From Table 10.4 we learn that the older the stolen car was, the easier it could be recovered.

Table 10.2. The association car make and loss recovery

Association Rule	Support	Confidence
Make='SATURN' —> lost automobile recovered	0.38%	86.96%
Make='DODGE' —> lost automobile recovered	0.36%	80.90%
Make='CHRYSLER' —> lost automobile recovered	1.83%	78.87%
Make='DAIHATSU' —> lost automobile recovered	2.26%	78.68%
Make='YTDALIHATATS' —> lost automobile recovered	0.24%	76.50%
Make='YULON MOTOR' —> lost automobile recovered	13.46%	75.51%
Make='CHINA MOTOR' —> lost automobile recovered	11.47%	74.45%
Make='DOMINCO' —> lost automobile recovered	0.65%	72.39%
Make='MAZDA' —> lost automobile recovered	0.79%	70.43%
Make='FORD LIO HO MOTOR' —> lost automobile recovered	12.03%	70.08%
Make='NISSA' —> lost automobile recovered	2.42%	68.16%
Make='LANCIA' —> lost automobile recovered	0.18%	67.93%
Make='ISUZU' —> lost automobile recovered	0.26%	67.81%
Make='SANYANG MOTOR' —> lost automobile recovered	7.61%	64.13%
Make='BUICK' —> lost automobile recovered	0.14%	62.30%
Make='BENZ' —> lost automobile recovered	1.83%	61.37%
Make='OPEL' —> lost automobile recovered	0.19%	60.37%
Make='PONTIAC' —> lost automobile recovered	0.29%	60.22%

- The association of car category and loss recovery

When we set minimum support = 0.001% and minimum confidence = 50%, we can find which category of automobiles could be easily recovered when it is lost. As indicated in Table 10.5, we can know that the business passenger cars were frequently recovered.

Table 10.3. The association of car make and lost car not recovered

Association Rule	Support	Confidence
Make=‘SAAB’ —→ lost automobile not recovered	0.152	56.973
Make= ‘VOLVO’ —→ lost automobile not recovered	0.364	56.399
Make= ‘SUZUKI’ —→ lost automobile not recovered	0.397	54.685
Make= ‘TOYOTA’ —→ lost automobile not recovered	1.433	51.966
Make= ‘CITROËN’ —→ lost automobile not recovered	0.134	50.099

Table 10.4. The association of the car manufacturing year and loss recovered

Association Rule	Support	Confidence
Manufacturing year = 1994-1993 —→lost automobile recovered	12.95%	77.90%
Manufacturing year = before 1992 —→ lost automobile recovered	20.76%	75.16%
Manufacturing year = 1996-1995 —→ lost automobile recovered	12.84%	69.42%
Manufacturing year = 1998-1997 —→ lost automobile recovered	12.71%	59.21%
Manufacturing year = 2000-1999 —→ lost automobile recovered	8.36%	52.95%

10.7.3 Decision Trees Analysis

Decision tree is a particularly efficient method for producing classifiers from data, and every path to the leaf in the decision tree represents a classification rule (Fig. 10.3). This research use decision trees to decide which stolen automobile could be recovered. We use the stolen automobile data set of 2001 year as training data for implementing decision trees. After clustering these data, we find they are divided into two classes automatically: one is the stolen automobiles recovered, the other the stolen auto mobile not recovered. We use the two classes as left nodes of the decision tree.

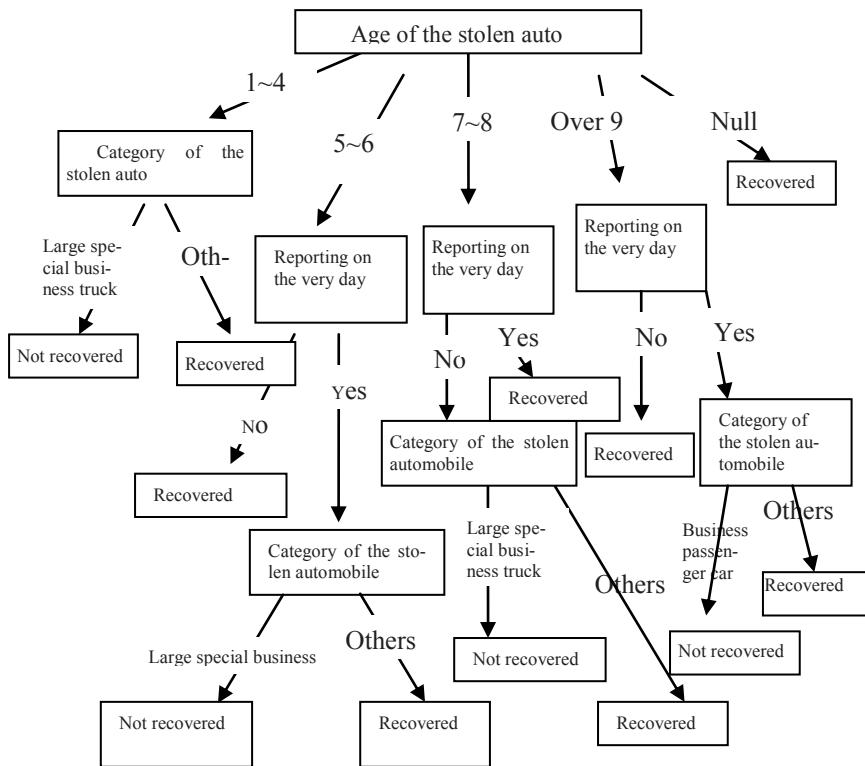
We use ARNAVAC method to find which attributes are highly related to the case of loss recovery in order to implement decision tree. Furthermore we find the manufacturing year of the automobile is highly related to the lost automobile recovered. In

Table 10.5. Association rule analysis of category of stolen automobile and this car recovered

Association Rule	Support	Confidence
Category = ‘business passenger car’ —→ lost automobile recovered	1.74%	91.92%
Category = ‘general business bus’ —→ lost automobile recovered	0.002%	87.50%
Category = ‘special business bus’ —→ lost automobile recovered	0.012%	83.64%
Category = ‘business pick-up’ —→ lost automobile recovered	0.14%	81.55%
Category = ‘private bus’ —→ lost automobile recovered	0.76%	75.68%
Category = ‘general business large truck’ —→ lost automobile recovered	0.50%	70.60%
Category = ‘private passenger car’ —→ lost automobile recovered	64.28%	67.10%
Category = ‘special business large truck’ —→ lost automobile recovered	0.22%	53.22%

addition, we also rely on domain knowledge to add two attributes that are “*category of the stolen automobile*” and “*the theft reporting on the very day*”. We use the above-mentioned three attributes and the two predefined classes — the lost car recovered and not — to construct the decision tree. The tree generated has 6 levels, 15 non-leaf nodes and 45 leaves. In addition, we also implement a transformation of a decision rules, for example, if the manufacturing year of the automobile < 2 years and category of the stolen automobile = ‘*the private passenger car*’ and ‘*the theft reporting on the very day = true*’ then classification = ‘*the lost car recovered*’. Total classification error is 32.32%, the rest of the information obtained is shown in Table 10.6.

The interpretation of Table 10.7 is as follows: The vertical dimension indicates the predicted outcome of a stolen automobile whether it could be recovered; the horizontal dimension indicates the real outcome. Furthermore, Class ‘No’ represents that a lost car could not be found, and Class ‘Yes’ represents a successful recovery. From Table 10.7 we learnt that there are 657 lost cars that were predicted not to be recovered and they are actually not recovered. There are 255,235 lost automobiles found as predicted. On the other hand, there are 121,759 lost automobiles that are mistakenly predicted to be recovered, but in vain. 444 lost automobiles are predicted not to be found, but actually recovered. In addition, undefined cases are zero.

**Fig. 10.3.** A decision tree**Table 10.6.** Decision tree for estimating whether a stolen automobile could be recovered

Depth of constructed tree:	5
Number of non-terminal nodes:	15
Number of leaves:	45
Total classification error:	32.32%
Percentage of undefined prediction cases:	0.00%
Classification probability:	67.68%
Classification error for class "the theft not recovered"	99.46%
Classification error for class "the theft recovered"	0.17%

Table 10.7. The count of each category of decision tree for stolen automobile recovered

Real Predicated	No	Yes	Undefined
No	657	121759	0
Yes	444	255235	0

10.7.4 Prediction

This research use 3 attributes (“*the manufacturing year of the automobile*”, “*category of the stolen automobile*”, and “*the theft reporting on the very day*”) in the previous section to implement linear regression analysis. We produce prediction rules in Table 10.8.

If the above-mentioned value of the regression equation is greater than 0.4894, we predict that a stolen automobile could be recovered; else it is predicted not to be recovered. Total classification error is 32.32%, and the percentage of undefined prediction cases is 0.01%. P-Value is zero ($<10^{-7}$), showing that explanatory capability of the model of regression equation is sound. The rest of the information obtained is shown in Table 10.9.

Table 10.8. Prediction rule table for whether a stolen automobile to be recovered

Prediction Result

IF the prediction rule (see below) > 0.4894
 THEN the stolen automobile recovered
 ELSE the stolen automobile not recovered

Prediction rule

$0.4894 < (0.619283 + 0.0615684 * (\text{age of the automobile}) - 0.190913 * \text{if}(\text{category} = \text{the private passenger car}) - 0.18414 * \text{if}(\text{category} = \text{the general business bus}) + 0.0764537 * \text{if}(\text{category} = \text{the business passenger car}) - 0.380972 * \text{if}(\text{category} = \text{the special business large truck}) - 0.160599 * \text{if}(\text{category} = \text{the private bus}) + 0.0658527 * \text{if}(\text{theft reporting on the very day}))$

The interpretation of Table 10.10 is similar Table 10.7. The vertical dimension indicates the predicted outcome of a stolen automobile whether it could be recovered; the horizontal dimension indicates the real outcome. Furthermore, Class ‘No’ represents that a lost car could not be found, and Class ‘Yes’ represents a successful recovery. From Table 10.10 we learn that there are 650 lost cars that were predicted not to

Table 10.9. Prediction for a stolen automobile whether it could be recovered

Total classification error:	32.32%
Classification error for Class “the theft not recovered”	99.47%
Classification error for Class “the theft recovered”	0.17%
Percentage of undefined prediction cases:	0.01%
Classification probability:	67.68%
P-value:	0

be recovered and they are actually not recovered. There are 255,225 lost automobiles found as predicted. On the other hand, there are 121,761 lost automobiles that are mistakenly predicted to be recovered, but in vain. 439 lost automobiles are predicted not to be found, but actually recovered. We find the results of prediction are similar that of decision tree, this shows that we can verify the results by using the two techniques simultaneously.

Table 10.10. The count of each category of Prediction for stolen automobile recovered

Real predicated \ No	No	Yes	Undefined
No	650	12761	5
Yes	439	255225	15

Table 10.11. The Parameter of prediction analysis for stolen automobile to be recovered

Attribute name	coef.	std dev.	F-Ratio	part sum of sq
“the manufacturing year of the automobile”	0.06157	0.000519	1.41E+04	0.03321
If (category = ‘the private passenger car’)	-0.1909	0.01721	123	0.002516
If (category = ‘the general business car’)	-0.1841	0.01934	90.64	0.0004543
If (category = ‘the business passenger car’)	0.07645	0.01803	17.98	0.002858
If (category = ‘the special business large truck’)	-0.381	0.02068	339.3	0.0008677
If (category = ‘the private bus’)	-0.1606	0.01874	73.41	0.0001841
If (the theft reported on the very day)	0.06585	0.001643	1606	0.004057

In addition, we analyze F-ratio, and find that two attributes including “the manufacturing year of the automobile” and “the theft reporting on the very day” have a significant impact to dependent variable. This is represented in Table 10.11.

10.7.5 Data Generalization and Summarization-Based Characterization

We implement Generalization and Summarization-Based Characterization and conduct online analysis and processing (OLAP) to find some meaningful knowledge and investigation clues.

- Rate of stolen automobile recovered and time interval from theft reporting date to the day of recovery

We summarize the total rate of stolen automobile recovered and the time interval from the date of automobile theft reported to the day when it was recovered. The data are collected from 1991 to 2001 as shown in Table 10.12. Total recovery rate of stolen automobiles was 49.4% in 1991, it grew into 73.30% in 2001. The highest was 74.91% in 2000. Furthermore, there was a growing trend year by year in terms of stolen car recovery. Similar tendency was found in the analysis in which a lost car was recovered in the first week and in the first month. For example, rate of case cleared within a month was 68.68% in 1991, it grew into 90.49% in 2001. In addition, rate of case cleared within a week was 37.53% in 1991, it grew into 61.72% in 2001. This shows the rate of stolen automobile recovered was growing, and the needed time to find a stolen automobile was reducing.

Table 10.12. Total Rate of stolen automobile recovered and time interval from theft reporting date to the day of recovery

Year \ Time interval	Total recovery rate	Within one month	Within one week	In two weeks	In three weeks	In four weeks	In five weeks	Over five weeks
Year	Total recovery rate	Within one month	Within one week	In two weeks	In three weeks	In four weeks	In five weeks	Over five weeks
1991	45.01	60.88	25.88	16.26	10.21	6.83	5.38	35.40
1992	51.54	65.70	27.14	18.11	11.28	7.43	5.31	30.69
1993	51.70	64.44	27.78	17.31	10.54	7.10	5.24	32.00
1994	53.20	63.27	27.46	16.68	10.27	7.13	5.31	33.12
1995	47.35	65.73	29.60	17.24	10.50	6.85	4.89	30.89
1996	52.15	72.11	35.21	18.20	10.37	6.79	4.76	24.63
1997	52.84	71.20	35.28	17.75	10.16	6.50	4.65	25.63
1998	54.98	72.72	36.03	18.16	10.22	6.83	4.70	24.04
1999	56.42	70.91	33.30	18.29	10.68	7.05	4.94	25.71
2000	54.46	71.48	34.20	18.22	10.53	6.97	4.87	25.18
2001	51.18	80.31	44.22	18.54	9.93	6.19	4.37	16.73
Average	51.91	69.44	32.85	17.75	10.40	6.86	4.91	27.19

- Owners whose automobiles are frequently stolen

If an owner has his automobiles frequently stolen, it might be a clue of crime. For example, when an auto rental company rents its car to a customer, the employee of the company could steal this automobile in order to cheat compensation for the customer.

We find an owner who has 126 automobiles stolen at most. In addition, there are 1,268 cases in which each owner has more than 40 automobiles stolen. They occupy 0.34% of total automobile thefts as shown in Table 10.13.

Table 10.13. Owners whose automobiles are frequently stolen

Number of auto-mobiles	Owner number	Case number	Rate of total automobile thefts
126	1	126	0.03%
98	1	98	0.03%
71	1	71	0.02%
70	1	70	0.02%
55	1	55	0.01%
52	1	52	0.01%
49	1	49	0.01%
48	2	96	0.03%
47	3	141	0.04%
46	3	138	0.04%
44	1	44	0.01%
43	1	43	0.01%
42	2	84	0.02%
41	1	41	0.01%
40	4	160	0.04%
total	24	1268	0.34%

- Analysis of the makes of stolen automobiles

In Table 10.14, we listed total amount of the ten most frequently stolen auto makes from 1991 to 2000. In addition, it is to note that the auto companies of YUELOON, FORDLH, CHINA, SANYANG, and KUOZUI are the five largest sharer of the domestic auto market, and TOYOTA, BENZ⁵ and BMW are the three largest foreign auto importers. In addition, NISSAN and DAIHATSU have also a higher marketing

⁵ Daimler-Chrysler.

share of sedans, both of them are also famous foreign auto makes. With the aforementioned figure in mind, we know that an auto make which has a high marketing share would have also a high stolen rate, and the domestic auto makes have much higher stolen rates than foreign auto makes. There are two possible reasons for this reality: Firstly, thefts of domestic autos have much higher profit than foreign autos because the valuable parts of the domestic automobiles have a higher demand than foreign ones. This explanation meets the assumption of routine activity theory. Secondly, an auto model which has a high market share would have a high stolen rate even if most autos were stolen just for fun, for short-term use or for conducting other crimes. This phenomenon confirms the assumption of the social disorganization theory. Therefore, we may come to a conclusion that routine activity theory and social disorganization theory do not conflict each other regarding to auto theft, they are complementary instead [24]. Disregarding which reason it might be, the larger market share a car model has, the more susceptible to theft it is.

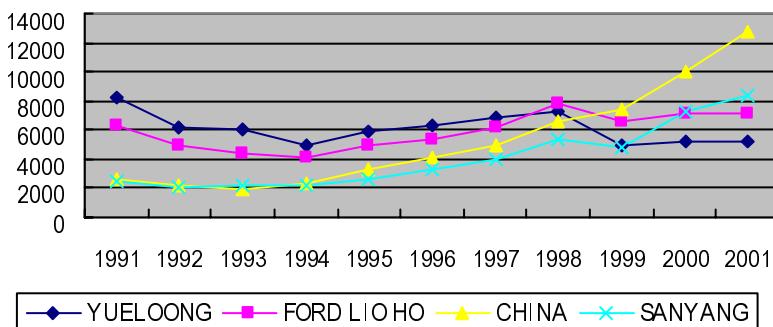
Table 10.14. Amount of stolen autos listed according to car makes

Make	Case numbers	Proportion
YUELOONG	67377	17.82%
FORD LIO HO	64908	17.17%
CHINA	58234	15.40%
SANYANG	44880	11.87%
KUOZUI	17876	4.73%
NISSAN	13442	3.56%
BENZ	11278	2.98%
DAIHATSU	10863	2.87%
TOYOTA	10426	2.76%
BMW	10368	2.74%

In addition, we find that the domestic makes of YULON MOTOR, FORD LIO HO MOTOR, CHINA MOTOR and SANYANG MOTOR ranked the four most frequently stolen makes, and much higher than others. The case numbers of stolen autos of the four makes from 1991 to 2001 as shown in Fig. 10.4.

- Analysis of reporting time of auto thefts

It is interesting for law enforcement to know what time autos are susceptible to theft. From Table 10.15, we learnt that most frequently reporting time is ‘4~8 a.m.’. It occupies 37.44%, over one third, of the total cases. A British crime survey also states that auto thefts often occur in the time from 6.00 p.m. and 6.00 a.m. next day [20]. Our research finds that most people are in deep sleep in the time between 0.00 a.m. and 6.00 a.m., and most auto thefts also happened in this time. They cannot find their autos stolen and report to the police until they wake up for work. Therefore, the category ‘4~8 a.m.’ has the highest reporting rates. Other categories have no significant difference.

**Fig. 10.4.** Yearly stolen amount of the four most frequently stolen makes**Table 10.15.** Number of the stolen autos according to reporting time

Time	Case numbers	Proportion
0~4 a.m.	46130	12.20%
4~8 a.m.	141577	37.44%
8~12 a.m.	60933	16.12%
0~4 p.m.	35758	9.46%
4~8 p.m.	43571	11.52%
8~12 p.m.	50126	13.26%

- Analysis of geographical locations of stolen autos

From Table 10.16, we find that there are 171,516 cases happened in north Taiwan, occupying 45.36 percent. It is much higher than the other geographical locations of Taiwan. On the other hand, the inhabitants in north Taiwan is about ten millions. It is about 45% of the total population, similar to the proportion of the stolen autos in

Table 10.16. Number of stolen autos according to geographical locations

Location	Case numbers	Proportion
North	171516	45.36%
Center	131203	34.70%
South	66674	17.63%
East	8525	2.25%
Off-shore islands	177	0.05%

Table 10.17. Number of stolen autos according to the time needed for case clearance

Needed days	Case numbers	Proportion
2	27203	10.64%
3	21823	8.54%
4	18155	7.10%
1	16798	6.57%
5	14844	5.81%
6	12432	4.86%
7	10571	4.13%
8	9379	3.67%
9	7984	3.12%
10	6972	2.73%
11	6043	2.36%

Table 10.18. Number of stolen autos according to their categories

Auto category	Case numbers	Proportion
Small business truck	645	0.17%
Small business passenger car	7139	1.89%
Small private passenger car	362180	95.79%
Private bus	3804	1.01%
General business bus	8	0.002%
Special business bus	55	0.015%
Large general business truck	2680	0.71%
Large special business truck	1584	0.42%

north Taiwan. This implies that the occurrences of auto theft are highly related to density of population and urbanization. The phenomenon is also observed in the research results of Clark and Harris [12, pp 15].

- Analysis of time needed for case clearance

From Table 10.17, we learned that most recovered autos are found in the second day after they are stolen. It occupies 10.64 percent of total recovered autos. Furthermore, about 60 percent of the recovered autos are found within ten days, and there is a lower possibility to recover the stolen autos over 10 days.

- Analysis of categories of stolen autos

As shown in Table 10.18, we learned that most stolen autos are small private passenger cars. It is about 95.79% of total stolen autos. However, small private passenger cars occupy only 70 percent of total motor vehicles in Taiwan, much lower than their stolen rate. From another point of view, we might conclude that there should be a higher profit to steal small private passenger cars, because the demands of their parts are very high. On the other hand, small private passenger cars are often stolen for short-term use.

10.7.6 Visualization and Presentation

Graphical tools can effectively help us in investigation. For example, to link groups of gangs, to link the activity places of an individual criminal, to show the relationship of a crime and its environment. Fig. 10.5 shows where automobile thefts often happened and monitoring systems can be installed to control the order in the area.

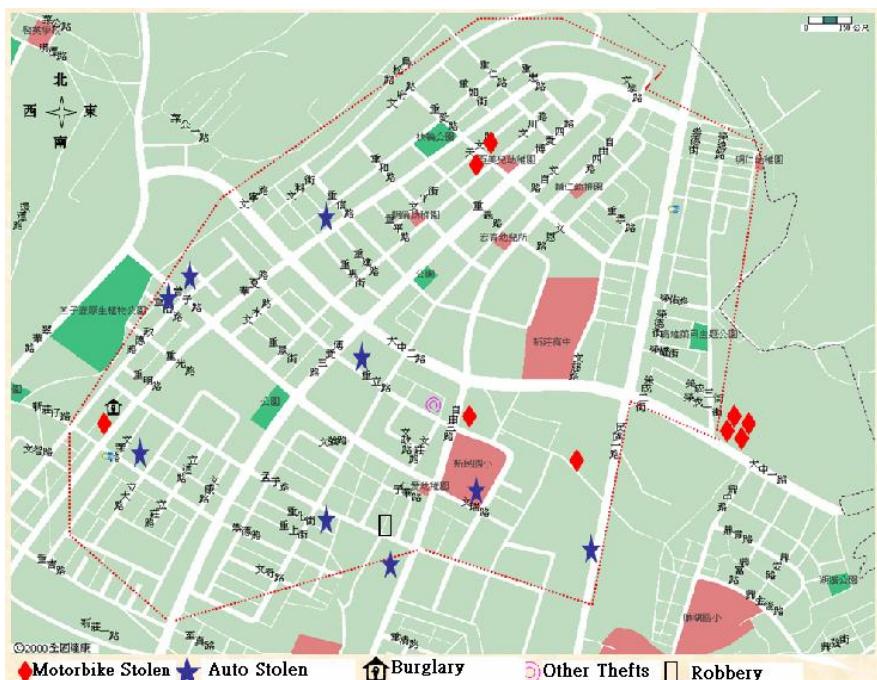


Fig. 10.5. Visualizing the distribution of automobile thefts in an urban area

10.8 Discussions

Through the experiment we made, we have demonstrated the usefulness of data mining in crime investigation. But, there are several works left for further study:

1. We have mined historic data and obtain a lot of investigation clues. But, if we want to control crimes in real time, we have to add newest data so that the database will reflect the reality and we can obtain the current-state investigation clues.
2. Since there are some automobiles stolen repeatedly, it provides us a hint to observe certain persons closely. This would help us to prevent from crimes and damages.
3. We did not mention the relationship between automobile theft and other crimes. It is believed that some thieves steal autos for short-time use not merely for fun, but for conducting other crimes, but what are they? An in-depth study is necessary.
4. To use geographical and environmental information in analysis, e.g., to check whether streetlamps and monitoring cameras deployed in communities would reduce theft cases.
5. These research topics are left for further study to implement more complete data mining.

10.9 Conclusion

In this chapter we have introduced the use of data mining techniques to find investigation clues. With the help of our case study, we have demonstrated data mining techniques including association rule, decision tree classification, clustering, prediction analysis, generalization and summarization-Based characterization, and visualization to mine the criminal data. It would assist law enforcement and its police officers to find some meaningful knowledge of automobile thefts. Meanwhile, it suggests the law enforcement to adopt effective strategies to prevent crimes.

The experience gained in the case study is valuable: In a similar way, we may take a further step to use data mining techniques to build intelligent systems to mine other serial crimes, for example, other kinds of thefts (e.g., burglary, pickpocket), sexual crime, fraud, arson, gang offenses, drug offenses, and cybercrime. Using data mining techniques, law enforcement can often get important clues, for instance, modus operandi of serial criminals, time and place of a crime, and criminal tools. These can assist police officers to determine investigative directions and search possible suspects. For the above-mentioned reasons, we suggest the law enforcement to intensively use data mining as a new means to investigate criminal cases, to set up a team of criminal data analysis and to launch a new program to crack down crimes

References

1. Atabakhsh, H., Schroeder, J., Chen, H., Chau, M., Xu, J., Zhang, J., Bi, H.H.: COPLINK knowledge management for law enforcement. In: Text analysis, visualization and collaboration. National Conference on Digital Government, Los Angeles, CA, May 2001, pp. 21–23 (2001)
2. Barclay, P., Buckley, J., Brantingham, P.J., Brantingham, P.L., Whinn-Yates, T.: Preventing Automobile theft in Suburban Vancouver Commuter Lots: Effects of a Bike Patrol. In: Clarke, R.V. (ed.) Crime Prevention Studies, vol. 6, pp. 133–162. Criminal Justice Press, NY (1996)

3. Bowen, J.E.: An Expert System for Police Investigators of Economic Crimes. *Expert Systems with Applications* 7(2), 235–248 (1994)
4. Brahan, J.W., Lam, K.P., Chan, H., Leung, W.: AICAMS: Artificial Intelligence Crime Analysis and Management System. *Knowledge-Based Systems* 11, 355–361 (1998)
5. Buller, D.B., Burgoon, J.K.: Interpersonal Deception Theory. *Communication Theory* 6, 203–242 (1996)
6. Burgoon, J.K., Blair, J.P., Moyer, E.: Effects of Communication Modality on Arousal, Cognitive Complexity, behavioral Control and Deception Detection during Deceptive Episodes. In: The Annual Meeting of the National Communication Association, Miami (2003)
7. Chen, P.S.: An Automatic System for Collecting Crime Information on the Internet. *Journal of Information, Law and Technology* (3) (2000)
8. Chen, P.S., Chan, P.P.: Using Text Mining in Finding Criminal Data on the Web. In: Proceedings of the 8th World Multiconference on Systemics, Cybernetics and Informatics (SCI), Orlando, July 18–21 (2004)
9. Chen, H., Chung, W., Xu, J., Wang, G., Chau, M.: Crime Data Mining: A General Framework and Some Examples. In: IEEE Computer Society, April.2004, pp. 50–56 (2004)
10. Chilton, R.J.: Middle Class Delinquency and Specific Offence Analysis. In: Vaz, E.W. (ed.) *Middle-Class Juvenile Delinquency*, pp. 91–101. Harper & Row, New York (1967)
11. Clarke, R.V.: Theoretical Background to Crime Prevention through Environmental Design (CPTED) and Situational Prevention. In: Geason, S., Wilson, P. (eds.) *Designing Out Crime: The Conference Papers*, pp. 13–20, Australian Institute of Criminology, Canberra (1989)
12. Clarke, R.V., Harris, P.M.: Automobile theft and Its Prevention. In: Tonry, M. (ed.) *Crime and Justice: A Review of Research*, vol. 16, University of Chicago Press, Chicago (1992)
13. Cohen, L.E., Felson, M.: Social Change and Crime Rate Trends: A Routine Activities Approach. *American Sociological Review* 44, 588–608 (1979)
14. Coleman, C., Moynihan, J.: *Understanding Crime Data*. Open University Press, Milton Keynes (1996)
15. Cornish, D.B., Clarke, R.V. (eds.): *The reasoning criminal: Rational choice perspectives on offending*. Springer, New York (1986)
16. Fazlollahi, B., Gordon, J.S.: CATCH: Computer Assisted Tracking of Criminal Histories System. *Interfaces* 23(2), 51–62 (1993)
17. George, J.F., Carlson, J.R.: Group support systems and deceptive communication. Presented at HICSS-32. In: Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences (1999)
18. Han, J., Kamber, M.: *Data Ming: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco (2001)
19. Hauck, R.V., Chen, H.: Coplink: a case of intelligent analysis and knowledge management. In: Proceedings of the 20th international conference on Information Systems, January 1999, pp. 15–27 (1999)
20. Hope, T.: Residential aspects of auto crime. In: *Research Bulletin*, vol. 23, pp. 28–33, Home Office, London (1987)
21. Kantardzic, M.: *Data Mining: Concepts, Models, Methods, and Algorithms*, November 2002. IEEE Press & John Wiley (2002)
22. Massey, J.L., Krohn, M.D., Bonati, L.M.: Property Crime and the Routine Activity of Individuals. *Journal of Research in Crime and Delinquency* 26, 378–400 (1989)
23. McCaghy, C.H., Giordano, P.C., Henson, T.K.: Automobile theft: Offender and Offense Characteristics. *Criminology* 15, 367–385 (1977)

24. Miethe, T.D., Meier, R.F.: Crime and Its Social Context: Toward an Integrated Theory of Offenders, Victims, and Situations. State University of New York Press, Albany (1994)
25. Ogrodnik, L., Paiement, R.: Motor Automobile Theft. In: Jurist at Service Bulletin, Canadian Centre for Justice Statistics, vol. 12. Minister of Industry, Science, and Technology, Ottawa, Canada (1992)
26. Pliant, L.: High-technology Solutions. *The Police Chief* 5(38), 38–51 (1996)
27. Potchak, M.C., McGloin, J.M., Zgoba, K.M.: A Spatial Analysis of Criminal Effort: Auto Theft in Newark, New Jersey. *Criminal Justice Policy Review* 13(3), 257–285 (2002)
28. Rice, K.J., Smith, W.R.: Socioecological Models of Automobile Theft: Integrating Routine Activity and Social Disorganization Approaches. *Journal of Research in Crime and Delinquency* 39(3), 304–336 (2002)
29. Rhodes, W.M., Conly, C.: Crime and mobility: An empirical study. In: Brantingham, P.J., Brantingham, P.L. (eds.) *Environmental criminology*, Prospect Heights, Waveland, IL, pp. 167–188 (1981)
30. Rossmo, D.K.: A methodological model. *American Journal of Criminal Justice* 11(2), 1–16 (1993)
31. Rossmo, D.K.: Overview: Multivariate spatial profiles as a tool in crime investigation. In: Block, C.R., Daboub, M., Fregly, S. (eds.) *Crime analysis through computer mapping*, pp. 65–98. Police Executive Research Forum, Washington, DC (1995)
32. Sanders, W.B.: *Juvenile Delinquency*, Praeger, New York, p. 94. Praeger, New York (1976)
33. Schepses, E.: Boys Who Steal Cars. *Federal Probation*, 56–62 (March 1961)
34. Taiwan: Crime Statistics Yearly Report, National Police Agency, Taiwan (2004)
35. U.S. News and World Report, p. 14 (January 15, 1996)
36. van Koppen, P.J., Jansen, R.W.J.: The road to robbery: Travel patterns in commercial robberies. *British Journal of Criminology* 38, 230–246 (1998)
37. White, C.H., Burgoon, J.K.: Adaptation and commutative design: Patterns of Interaction in truthful and deceptive conversation. *Human Communication Research* 27, 9–37 (2001)
38. Wilcox, J.: IT-armed Officers Fight Gangs. *Government Computer News: State and Local* 3(12), 19–20 (1997)
39. Zhou, L., Burgoon, J.K., Twitchell, D.P.: A Longitudinal Analysis of Language Behavior of Deception in E-mail. In: Chen, H., Miranda, R., Zeng, D.D., Demchak, C.C., Schroeder, J., Madhusudan, T. (eds.) *ISI 2003. LNCS*, vol. 2665, pp. 102–110. Springer, Heidelberg (2003)

Online Resources

- Center for Strategic and Industrial Studies:
<http://www.csistaiwan.org>
- Coplink:
<http://ai.bpa.arizona.edu/research/coplink/index.htm>
- Criminal Investigation Bureau:
<http://www.cib.gov.tw/english/index.aspx>
- National Police Agency:
<http://www.npa.gov.tw/>

Questions for Discussions

1. What data-mining techniques can be used to discover unknown investigation clues? How do you evaluate the applicability of these techniques?
2. The outcomes of data mining are sometimes bizarre or unheard before. Do they always make sense? How can we compare them with the results of empirical studies or experts' experiences?
3. In general, thefts are highly related to economical situation (of the thieves). Can this postulate be applied to automobile thefts? In our case study, we find both routine activity theory and social disorganization theory support our mining results. Do you think our analysis controversial?
4. What is the difference between statistical analysis and data mining? Are there any kinds of problems can better be analyzed by data mining?
5. Before we undertake data mining, we have to construct a data ware house with various dimensions. The dimensions are chosen from attributes of data sets. Upon what criteria shall you select these attributes? Justify your decision.

Automated Filtering on Data Streaming for Intelligence Analysis

Yiming Ma, Dawit Yimam Seid, and Sharad Mehrotra

Dept. of Information and Computer Science

University of California, USA

{maym, dseid, sharad}@ics.uci.edu

Abstract. Intelligence analysis involves routinely monitoring and correlating large amount of data streaming from multiple sources. In order to detect important patterns, the analyst normally needs to look at data gathered over a certain time window. Given the size of data and rate at which it arrives, it is usually impossible to manually process every record or case. Instead, automated filtering (classification) mechanisms are employed to identify information relevant to the analyst's task. In this paper, we present a novel system framework called FREESIA (Filter REfinement Engine for Streaming Information) to effectively generate, utilize and update filtering queries on streaming data.

11.1 Introduction

Intelligence analysis involves routinely monitoring and correlating large amount of data streaming from multiple sources. In order to detect important patterns, the analyst normally needs to look at data gathered over a certain time window. However not all data is relevant for the analysts task; the relevant set of data needs to be selected from the streaming data. The task of monitoring involves a combination of automated filtering system to identify candidate cases and human analysis of cases and their related data. The filtering system is typically part of a data aggregation server to which transaction data are fed from numerous agencies in near real time. An analyst stores his task or goal specific filters that are matched to incoming data as it flows. Multiple filters may be needed to extract information from different sources.

Formulating the right filtering queries is an iterative and evolutionary process. Initially the analyst may draw from his domain knowledge to express a filter. But this filter needs to be refined based on how well it performs. Besides, it needs to be refined to capture the changes over time in the emphasis given to various attributes. In this paper we consider how to enable the filtering system to perform automatic query refinement based on minimal and continuous feedback gathered from the user. Below we give examples drawn from two intelligence related tasks that illustrate how such a system can be employed:

Example 1 (Intelligence Report Monitoring). Massive amount of incident reports are continuously generated by law enforcement agencies which are monitored by analysts at

different levels to detect trends and correlations. For instance, an analyst at federal level may want to continuously filter and analyze all relevant incident reports from local agencies that relates to multi-housing (e.g. rental apartment or condominium) and lodging (e.g. hotels or motels) facilities that have national monuments in their proximity. The analyst may also express detailed preferences on the attributes related to the suspects described in the incident report. To achieve this, the analyst draws from his domain knowledge to specify an initial (imprecise) filter to the data reporting server. The server matches the filter with incoming reports. In cases where matching reports can be large, it will be useful if the system can also rank the reports based on how strongly they match the given filter. To refine both the classification and ranking capabilities of the filter over time, the system offers the analyst a feature to provide feedback on the relevance of the reports. Based on the feedback the system automatically refines the filter.

Example2 (Intelligence Information Dissemination). Large amount of intelligence information is gathered everyday from various sensors. For instance, US custom services use various sensing technologies (e.g., cameras, finger-print reader) to gather passenger information from airports and seaports. Different feature extraction tools are used to extract features from these data. Data matching given multi-feature criteria, watch-lists or archived data must be disseminated to analysts in different agencies for further processing. Analysts register filtering queries to the central system that gathers the data which then disseminates relevant information in a prioritized manner to analysts. Similar to the previous example, feedback from the analyst can be used to automatically adjust filtering queries stored in the system.

Technically, filtering queries can be considered as *classifiers* since their purpose is to classify each incoming data item as relevant (i.e. belong to the target class) or non-relevant. However, the following three important requirements distinguish our filtering queries from traditional classifiers:

1. Ranking and Scoring. For the purpose of filtering data instances belonging to a target class from massive volumes of streaming data, classifiers that merely make binary decisions are inadequate. The classifiers need to also score and rank records based on how strongly they match the filters. Ranking is useful for two reasons: (1) it enables the analyst to prioritize the processing of records, and (2) in cases where rigid binary partitioning of relevant and non-relevant data is undesirable, it facilitates the prioritization of records that are highly likely to be in the target class while at the same time not eliminating records. The latter issue is particularly important due to the fact that in most situations the filters are not crisp rules but rather fuzzy and approximate. This makes classifiers that score and rank data instances more appropriate than classifiers that only make binary decisions on class membership.

2. Incorporating Analyst's Domain knowledge. In a great majority of intelligence applications, analyst's domain knowledge (e.g. about features of suspects, etc.) forms a critical component. Hence, it is imperative that the system provides a mechanism to readily incorporate domain knowledge-induced filtering rules. However, while filtering rules representing domain knowledge are normally vague and imprecise, current database systems on which much of data filtering is carried out require crisp expressions. In order to express filtering rules on such systems, analysts are forced to convert their rules to very complex crisp expressions. To avoid this problem, the filtering system needs to allow direct execution of inexact filtering queries.

3. Interactive Refinement. As illustrated in the above examples, allowing the analyst to refine filters through relevance feedback (a.k.a. supervised learning) is an important requirement. This becomes necessary when the rules expressed by the analyst fail to capture the desired domain knowledge, or rules change over time. An important issue to notice here is that unlike traditional approaches where a classifier is learned and then applied in distinct phases, here the classifier needs to be incrementally refined using a feedback loop. Also notice that human domain knowledge is incorporated in two ways: first, through submission of domain knowledge in the form of initial filtering queries, and second, through feedback on the classified records.

In this chapter, we introduce a framework called FREESIA (Filter REfinement ENgine for StreamInG InformAtion) that meets the above requirements. FREESIA achieves ranking of streaming data by representing filtering queries (classifiers) as multi-parametric similarity queries which allow the analyst to express his imprecise filtering rules. Then, in the course of data analysis, the analyst can refine and update these filters through example-based training so as to achieve required accuracy and meet evolving demands. To efficiently support such dynamic adaptation of filters, FREESIA provides a set of algorithms for refining filters based on continuous relevance feedback.

11.2 Definitions and Preliminaries

11.2.1 Data Model

Filters in FREESIA assume a structured multi-dimensional data. However, originally the data can be either a set of relational tables or in any unstructured/semi-structured format. If the data is unstructured, data extraction tools¹ can be first applied to extract relevant values (e.g. names, places, time, etc.). The extracted data is then represented in the form of attribute-value pairs and fed into filtering modules.

11.2.2 Filtering Query Model

In this section we define a flexible query model that is powerful enough to capture human supplied filters and domain knowledge in addition to enabling incremental refinement. A filtering query or rule, henceforth simply referred to as *filter* or *classifier*, consists of four components: a set of similarity predicates structured in DNF form (Disjunctive Normal Form), a set of weights assigned to each similarity predicate, a ranking function and a cut-off value.

Definition 1. A filter (classifier) is represented as a quadruple $\langle \rho, \omega, \phi, \alpha \rangle$ where ρ is a conditional expression, ω is a set of weights, ϕ is a ranking function and α is a cut-off value. Below we give a brief description of these four elements.

Conditional Expression: A conditional expression, ρ , is a DNF (Disjunctive Normal Form) expression over similarity predicates. Formally, an expression

¹ For example, Attensity's Extraction Engines: www.attensity.com

$Q = C_1 \vee C_2 \vee \dots \vee C_n$ is a DNF expression where $C_i = C_{i1} \wedge C_{i2} \wedge \dots \wedge C_{in}$ is a conjunction, and each C_{ij} is a similarity predicate. A *similarity predicate* is defined over the domain of a given data type (attribute type). A similarity predicate takes three inputs: (1) an attribute value from a data record, t , (2) a target value that can be a *set* of points or ranges, and (3) a similarity function, f , that computes the similarity between a data value and the target value. A similarity function is a mapping from two data attribute values, v_1 and v_2 , to the range $[0,1]$, $f = v_1 \times v_2 \rightarrow [0,1]$. The values v_1 and v_2 can be either point values or range values. Similarity functions can be defined for data types or for specific attributes as part of the filtering system.

DNF Ranking Functions, Weights and Cut-off: A *DNF ranking function*, ϕ , is a domain-specific function used to compute the score of an incoming record by aggregating scores from individual similarity predicates according to the DNF structure of ρ and its corresponding set (template) of *weights* that indicate the importance of each similarity predicate. The template of weights, ω , corresponds to the structure of the search condition and associates a weight to each predicate in a conjunction and also to each conjunction in the overall disjunction.

A DNF ranking function first uses *predicate weights* to assign aggregate scores for each conjunction, and it then uses *conjunction weights* to assign an overall score for the filter. A conjunction weight is in the range of $[0, 1]$. All predicate weights in a conjunction add up to 1 while all conjunction weights in a disjunction may not add up to 1. We aggregate the scores from predicates in a conjunction with a weighted L₁ metric (weighted summation). Using weighted L₁ metric as a conjunction aggregation function has been widely used in text IR query models where a query is typically expressed as a single conjunction [12, 10]. To compute an overall score of a query (disjunction), we use the *MAX* function over the weighted conjunction scores. *MAX* is one of the most popular disjunction aggregation functions [4].

11.2.3 Filter Refinement Model

The similarity conditions constituting a filter are refined using relevance feedback that is used as real-time training example to adapt the predicates, condition structure and corresponding weights to the information needs of the analyst. More formally, given a filter, Q , a set R of the top k records returned by Q , and relevance feedback F on these records (i.e., a triple $\langle Q, R, F \rangle$), the refinement problem is to transform Q into Q' in such a way that, when Q' is used to filter future streaming information or is re-executed on archival information, it will return more relevant records. Sect. 11.3.2 will discuss in detail the types of feedback that are gathered by the system and how they are represented.

11.3 Our Approach

In this section, we present our proposed approach for applying and refining filters on streaming data. We first present an overall architecture of our system FREESIA followed by a description of how the analyst interacts with FREESIA (i.e. the feedback

loop). We then propose algorithms that implement the classifier refinement and scoring/ranking model refinement components of FREESIA.

FREESIA's schematic design is depicted in Fig. 11.1. The following four main components constitute the system.

Filter Processing Component. When a new data group is received by the system, the filters that are represented as similarity queries are executed by the Filter Processing Component in order to score and filter relevant target records. This component can be implemented in any commercial database system using common similarity query processing techniques (e.g. [5, 1]). To readily apply the similarity queries in this context, we use an SQL equivalent of the weighted DNF query defined in Sect. 11.2.2.

If the similarity query has been modified (refined) since its last execution, the system will also evaluate it on the *archived data store* which is used to store the *unseen* (but matching) records as well as *filtered out* records. Re-evaluating the query on the archive allows the identification of previously excluded records that match the current filter. The scored list of records that results from the filter processing component is passed to the ranking component.

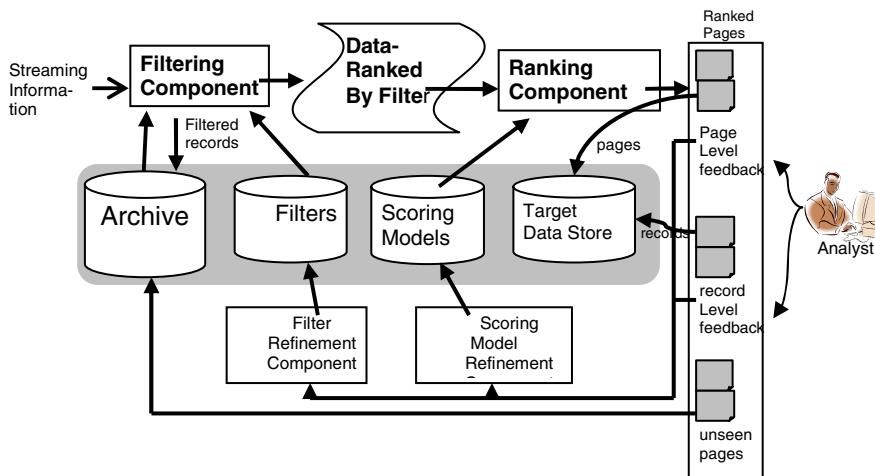


Fig. 11.1. FREESIA system overview

Example 3. Consider the incident report analysis application from *Example 1*. For simplicity suppose that a data instance consists of only the location coordinates, incident type, location type and number of suspects. Then one possible query that filters potential analyst is given below.

```

SELECT Location, IncidentType, LocType, NumSuspects,
        RankFunc(w1,s1, w2, s2, w12) AS S,
FROM IncidentReports
WHERE LocNear(location, National_Monument, s1)
        AND LocTypeLike(LocType, {multi-housing,lodging}, s2)
ORDER BY S desc
  
```

The label “*National_Monument*” stands for a set of national monuments stored separately. *LocNear* takes a given location and computes its distance from the nearest national monument. *LocTypeLike* implements heuristic techniques to match similarity of a location type to a classification hierarchy of places.

Ranking Component. This component applies scoring rules to produce a ranking of data instances. We employ two types of scoring methods. The first is the similarity scoring rule (i.e. the ranking function defined in Sect. 11.2.2) that is used by the Filter Processing Component. This represents the *long-term* filter of the analyst. In addition to this scoring rule, FREESIA also incorporates other scoring models that represent *short-term* (or special-case) rules that may not participate in the filtering process (i.e. are not evaluated as similarity queries) but are used for ranking. This, for instance, allows the analyst to temporarily force the system to rank reports that fulfill a given complex criteria at the top. Also, such rule can be specified by giving a record as a sample and asking “give me records like this”. Many data mining methods can be used (e.g. [11]) to model such samples to produce scores for incoming records (more details on this will be given in Sect. 11.3.3). Given the scores from the similarity match and the scoring rules, the *Ranking Component* applies a combination method to produce the final ranking. As we mentioned in Sect. 11.2.3, the resulting ranked list of records is partitioned into pages for presentation to the analyst.

Filter Refinement Component. As discussed before, it is often necessary to interactively refine the analyst’s initial filtering queries. This is achieved in FREESIA by collecting relevance feedback on the outputs of a filter. Upon seeing the ranked list of records, the analyst can submit feedback on the relevance (or otherwise) of the records - i.e. whether the records belong to the target class or not. Based on this feedback, the Filter Refinement Component refines the similarity queries. Sect. 11.3.3 will give details on the refinement process.

Scoring Model Refinement Component. In addition to its use for filter refinement, the feedback from the analyst is also utilized to refine the additional scoring rules. Sect. 11.3.3 will give details of this refinement process.

11.3.1 Gathering and Representing Feedback

Various types of feedback are gathered in FREESIA. One type of feedback is what we call *record-level feedback* where the analyst provides feedback on particular records. However, in cases where the analyst receives large amount of matching reports, FREESIA also provides feature to provide group feedback. For this we exploit the fact that the system presents the results in pages (groups) of a certain size which enables *page-level feedback*. Often, when an analyst looks at a page, he can tell whether most of the records in the page are relevant in his initial scanning of the results. If some level of error in a page is tolerable, the analyst may want to accept all the records in a page for further processing. On the contrary, in cases where the analyst determines that a page contains only some relevant records, she may want to give record-level feedback. For feedback gathering purposes, we distinguish three types of pages:

- Highly relevant pages: almost all the records in these pages are relevant. In other words, the analyst will use *all* the record in these pages for further actions despite the fact that there could be a few records in these page which are not relevant.
- Relevant pages: only some of the records in these pages are relevant. For these pages, the analyst provides feedback on each record.
- Unseen pages: these are the pages returned by the filter but are not viewed by the analyst. We assume that these are deemed to be non-relevant

Despite the availability of page-level feedback, providing record-level feedback may still be a time consuming operation in some cases. To deal with this, FREESIA provides a parameter to specify the number of pages the analyst wants to give record-level feedback on. The remaining pages are considered unseen pages.

11.3.2 Filter Refinement

Feedback Processing

The refinement strategies used by the Query Refinement and the Scoring Model Refinement components require two sets of data: contents of records on which the analyst gave relevance feedback (for e.g. to modify target values in predicates), and the feedback itself. We initially capture these two types of information in the following two tables:

1. A *Result Table* contains the ranked list of records returned by the filter as well as the score assigned to each by the system.
2. A *Feedback Table* contains the relevance feedback given by the analyst on records that are a subset of those in the Result Table. Particularly, this table contains record-level feedback given on *Relevant Pages*. Table 11.1 shows a sample feedback table from the intelligence report monitoring example.

Since data attributes can have complex and non-ordinal attributes, performing query refinement directly on the *result* and *feedback* tables is difficult as this will require specialized refinement method for each attribute type. To circumvent this problem, we transform the diverse data types and similarity predicates defined on them into a homogeneous similarity space on which a single refinement method can operate. We refer to the resulting table as *Scores Table*. It contains the following five columns that store statistical information useful for the refinement process:

1. Entry Identifier: This identifier is a triple $\langle Attribute\ ID, Value\ ID, Conjunction\ ID \rangle$. The first two entries show the attribute-value pair. The Conjunction ID, which comes from the filter, identifies the conjunction that is satisfied by the attribute-value pair. Since we use a DNF representation, a conjunction contains one or more predicates.
2. Counts of relevant records having the value in this entry.
3. Count of non-relevant records having the value in this entry.
4. Proximity to other values (of same attribute) of relevant records.
5. Proximity to other values (of same attribute) of non-relevant records.

For every distinct value v_i in the scores table, we compute its weighted proximity to other relevant values of the same attribute *in the scores table* using the following formula:

$$v_i.\text{RelevantCount} + \sum_{j=1}^{k-1} (v_j.\text{RelevantCount} \times \text{sim}(v_i, v_j)) \quad (11.1)$$

where $v_i.\text{RelevantCount}$ is the count of relevant records (second column in the scores table), k is the total number of distinct values of the attribute corresponding to v_i that also have the same *conjID*, and $\text{sim}(v_i, v_j)$ is the similarity between v_i and v_j as computed by the similarity function corresponding to the attribute. In the same fashion, we compute proximity to non-relevant values using the above formula with $v_i.\text{nonRelevantCount}$ and $v_j.\text{nonRelevantCount}$ values. The intuition behind the proximity values is to bolster the exact counts of every distinct attribute-value pair in the scores table with the counts of other values of the same attribute weighted by their similarity to the attribute value at hand. This in essence allows us to capture the query region which the user is giving an example of. Table 11.2 shows an example scores table with one conjunction (*C1*) of one predicate on the attribute *location*.

Table 11.1. Example feedback table (I=Irrelevant, R=Relevant)

ID	Location	Incident Type	LocType	#Suspect	FB
4	Irvine	Photographing	Retail center	2	I
1	Irvine	Bomb threat	Hotel	0	R
7	LA	Request for Building blueprints	Apartment	1	R
10	LA	Unauthorized access	Hotel	2	R
60	LA	Arson	Inn	5	I
2	San Diego	Suspicious package delivery	Office	1	I
3	San Diego	Larceny	Fed. Building	5	I

Table 11.2. Example scores table

Obj ID	Rel Count	Irrel Count	AggRel	AggIrrel
<Loc, Irvine, C1>	1	1	1+2*0.8	1+1*0.8+2*0.2
<Loc, LA, C1>	2	1	2+1*0.8	1+1*0.8+2*0.2
<Loc, SD, C1>	0	2	0+1*0.2+2*0.2	2+1*0.2+1*0.2

Refinement Algorithms

In principle, the feedback given by the analyst can potentially result in one of three types of filter refinement. A filter that is too specific can be made more general (called *filter expansion*), a filter that is too general can be made more specific (called *filter contraction*) and finally the target values of predicates can be shifted to a new value (called *filter movement*).

ComputeNewQuery()

Input: Filter (Q), Scores_table(ST), NumCase (N), NumRelCase (NR), HACThreshold (τ)

Output: NewFilter

1. $ST_{pruned} = \text{pruneInsigEntries}(N, NR)$
2. **foreach** Conjunction (C_i) in Query
3. **foreach** Predicate (P_j) in C_i
4. $ST_{pj} = \text{filterScoreTable}(ST_{pruned}, P_j, \text{attribute}, C_j)$
5. $\text{Cluster}_{pj} = \text{computeHACCluster}(ST_{pj}, \tau)$
6. **foreach** Cluster (Cl_k) in Cluster_{pj}
7. **if** $P_j.\text{isNewPoint}(Cl_k.\text{centroid}, P_j)$
8. $P_j.\text{addQueryPoint}(Cl_k.\text{centroid})$
9. **endif**
10. **endfor**
11. $P_j.\text{weight} = \text{averageConf}(P_j.\text{queryPoints})$
12. **endfor**
13.
$$C_i.\text{weight} = \frac{\sum_{j=1}^{|C_i|} P_j.\text{weight}}{|C_i|}$$
14. NewFilter.addConjunction(C_i)
15. NewFilter.normalizeWeight()
16. **endfor**

Fig. 11.2. ComputeNewFilter Algorithm

The refinement algorithm in Fig. 11.2 performs all three kinds of refinements and adjusts weights. The algorithm starts by pruning insignificant predicates (entries) from scores table using the *pruneInsigEntries* function. Due to limited space, we skip the detailed discussion of this function. In short, it uses statistical method to measure the performance of each candidate predicate, and deletes the useless ones. The output of this function is a subset of the scores table, ST_{pruned} , whose entries are used as *candidates* to refine the filter.

Using the pruned scores table, ST_{pruned} , the algorithm next tries to determine whether the filter should be updated (line 3 to line 13). For each predicate in each conjunction, the algorithm first extracts the relevant entries, ST_{pj} . This is performed by matching the *conjunction ID* and attribute name in the filter with ST_{pruned} entries. This

matching may result in many candidate predicates. Hence, we need a mechanism to select those that represent the right values to which we should move the filter. We do this selection by first clustering the candidate predicates and then choosing the cluster centroid as a representation of the new target value of P_j .

For this, we use hierarchical agglomerative clustering (HAC) [2] method where the distance measure is computed based on the similarity between predicates in ST_{pj} .

Once we get a set of candidate target points using HAC clustering, the algorithm tests whether each candidate is actually a new target value (line 7). The *isNewPoint* function determines the closeness of each candidate to each of the existing filter predicate. If a candidate is not near any of the existing target points, we add it as a new target value of the current predicate (line 8).

Next, the algorithm updates the weights of the predicates and conjunctions in the query. The predicate weight is computed as the average confidence level of the query points in the updated predicate. The confidence value of a predicate is computed based on its proximity values stored in the scores table as:

$$\frac{\text{ProximityTo Relevant}}{\text{ProximityTo Relevant} + \text{ProximityTo Irrelevant}}.$$

The weight for a conjunction is computed as the *average* of the weights of its constituent predicates.

Refining the Scoring Model

The primary task of FREESIA's ranking component is to assign an accurate ranking score to each record. This component uses the maximum of the scores from the Filtering Component and the score from the short-term scoring model to be a record's final ranking score. When the analyst provides samples to form the scoring rules, many general incremental learning methods can be applied. In FREESIA, we use a pool based active learning method [11] which is suited to streaming data and is able to capture sample based short-term user model. It is a three-step procedure:

- Train a Naive Bayes classifier -- *short-term* model -- using sampled feedback records.
- Apply the *short-term* model to score the records returned by the Filter Component.
- Merge the scores from *long-term* model (i.e., filter score) and from *short-term* model.

11.4 Experiments

In this section, we present the results of the experiments we conducted to evaluate the effectiveness of our refinement method.

We used four real-life datasets from the UCI machine learning repository [8]. The datasets are a good ensemble to some intelligence data. They are reasonable in size and have predefined target (classes); they also cover some portions of US census data (adult), environmental data (covtype), disease data (hypo) and scientific analysis data (waveform21). Table 11.3 shows the characteristics of the datasets. There are two types of attributes in the datasets (viz. continuous and discrete). We manually generate 20 initial and target query pairs for each dataset.

Table 11.3. Data set descriptions and parameters

Dataset	#Cases	#Cls.	# cont. attrs.	#disc. Attrs.	Page Size	Data Group size
Adult	32,561	2	6	8	40	1,000
Covertype	10,000	7	10	40	40	1,000
Hypo	3,163	2	7	18	20	316
Waveform21	5,000	2	21	0	20	500

Our evaluation process closely follows the FREESIA architecture(Sect. 11.3.1). Table 11.3 shows two of the parameters we used for each dataset, namely page size and data group size. Page size specifies the number of records in each page. Data group size shows the number of records streaming into the system at each iteration. In addition, we set two more parameters, namely precision threshold and record-level feedback page threshold. Precision threshold shows that if the precision in a page is higher than this number, the page will be treated as a highly relevant page. We use 80% for all data sets. For all data sets, record-level feedback is gathered for 2 pages.

The initial query is executed in the first iteration. The system then presents the data in pages. If a page is a highly *relevant page*, the system continues fetching the next page, and no feedback will be given to the system. This is to simulate the page-level feedback. If a page is not a highly relevant page, then in reality record-level feedback will be given on this page. Since we are using a pre-labeled dataset, we simulate this feedback process by assigning the respective true label of each record in the page. If the number of record-level feedback pages is beyond the page limit of record-level feedback specified above, the system will dump the remaining pages to the data archive (i.e. they are unseen pages).

Tested Strategies. Four approaches were compared in our experiments:

1. Baseline method (Q). This uses only the initial query.
2. Query and Scoring Model Refinement (QM+). This refines the scoring model, but the similarity query is not refined. This, in effect, simply makes the query more specific.
3. Query Refinement (Q+). This refines the similarity query only. This performs all three types of refinement.
4. Query Refinement and Scoring Model Refinement (Q+M+). This refines both the query and the scoring models. This also refines all three types of refinement but is capable of producing much more focused queries.

11.4.1 Results

Figs. 11.3 to 11.6 show the precision and recall measures across different refinement iterations. In the first two datasets (*adult* and *hypo*), we show results where the desired refinement of the initial queries is achieved in the first few iterations (around two or three iterations). Moreover, the system was able to maintain the high precision and recall measures across the subsequent iterations. As can be clearly seen in these two figures, the two algorithms that perform similarity query refinement (i.e. *Q+* and *Q+M+*) have much better performance compared to the other two which do not

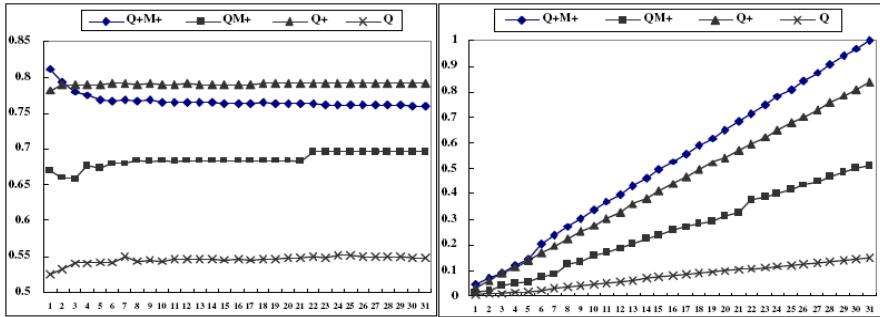


Fig. 11.3. Adult: Prec-Rec over 31 iters

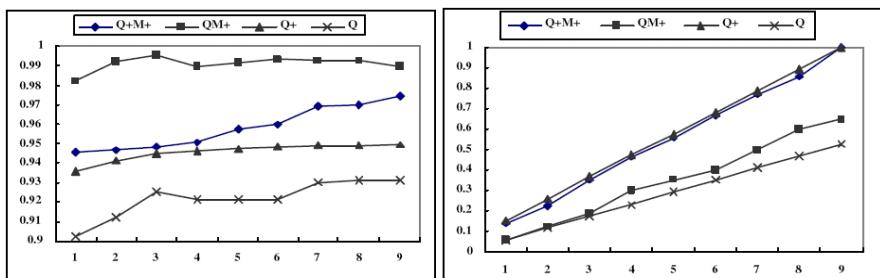


Fig. 11.4. Hypo: Prec-Rec over 9 iters

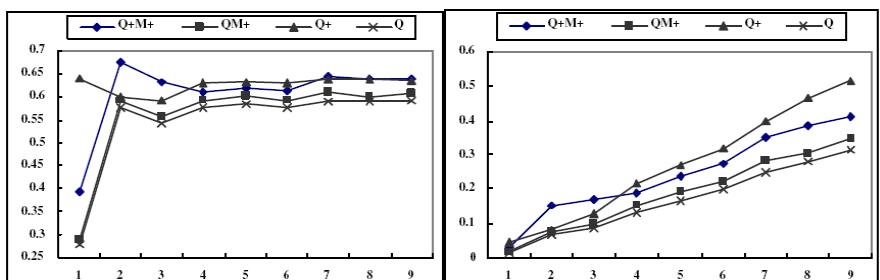


Fig. 11.5. Wave21: Prec-Rec over 9 iters

perform query refinement. For the dataset (*waveform21*), to achieve the desired refinements, more refinement iterations are required compared to the above two datasets (see the recall graph). Here as well $Q+M+$ and $QM+$ achieved the best precision and recall. The last dataset (*covertype*) shows cases where the initial query is very different from the desired target. As shown in the graph, precision declines as more iterations are needed (i.e. relatively more non-relevant records are retrieved). Still, $Q+M+$ performs better than the rest. The above results clearly show the effectiveness of FREESIA's refinement algorithms.

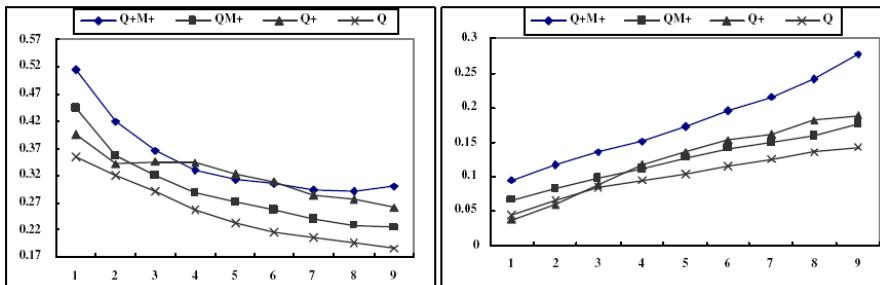


Fig. 11.6. Covertype: Prec-Rec over 9 iters

11.5 Related Work

The filtering process studied in this paper is related to target data selection techniques proposed in data mining on static data warehouses. However, unlike data mining on data warehouses (where a relevant subset of the database is filtered out for data mining tasks by carrying out as much refinement on the filters as required), in streaming data filtering has to be done continuously to allow data mining to occur as soon as the data arrives. There has been some research to address the problem of target subset selection from static data using classifiers [7, 9]. This body of research, however, only dealt with the problem of automatic classifier generation and the data considered were static. Recently, [3, 6] have considered the problem of data mining on streaming data. These works considered dynamic construction and maintenance of general models in a precise data environment. Whereas, our work deals with user predefined imprecise selection filters, and exploits the user knowledge to improve the accuracy of the filtering process.

11.6 Conclusions

In this paper, we have proposed a novel filtering framework called FREESIA, which enables analysts to apply the classifiers directly on database systems (in the form of similarity queries) to filter data instances that belong to a desired target class on a continuous basis. We believe our system can be used in many intelligence related tasks.

References

1. Chaudhuri, S., Gravano, L.: Evaluating top-k selection queries. In: Proc. of the Twenty-fifth International Conference on Very Large Databases (VLDB 1999) (1999)
2. Day, W., Edelsbrunner, H.: Efficient algorithms for agglomerative hierarchical clustering methods 1(1), 7–24 (1984)
3. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Knowledge Discovery and Data Mining, pp. 71–80 (2000)

4. Fagin, R.: Combining Fuzzy Information from Multiple Systems. In: Proc. of the 15th ACM Symp. on PODS (1996)
5. Fagin, R., Lotem, A., Naor, M.: Optimal aggregation algorithms for middleware. In: PODS 2001, Santa Barbara, California, May 2001, pp. 83–99 (2001)
6. Lambert, D., Pinheiro, J.C.: Mining a stream of transactions for customer patterns. In: Knowledge Discovery and Data Mining, pp. 305–310 (2001)
7. Ling, C., Li, C.: Data mining for direct marketing: problems and solutions. In: Proceedings of ACM SIGKDD (KDD 1998), pp. 73–79 (1998)
8. Merz, C.J., Murphy, P.: UCI Repository of Machine Learning Databases (1996), <http://www.cs.uci.edu/ml/mlrepository.html>
9. Piatetsky-Shapiro, G., Masand, B.: Estimating campaign benefits and modeling lift. In: Proceedings of ACM SIGKDD (KDD 1999), pp. 185–193 (1999)
10. Rocchio, J.: Relevance feedback in information retrieval. In: Salton, G. (ed.) The SMART Retrieval System: Experiments in Automatic Document Processing, pp. 313–323. Prentice Hall, Englewood Cliffs (1971)
11. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: Proceedings of ICML 2001, pp. 441–448 (2001)
12. Yates, R.B., Neto, R.: Modern information retrieval. ACM Press Series. Addison Wesley, Reading (1999)

Online Resources

There are number of online resources that are related to the automated filtering.

1. FREESIA has been incorporated as a system component to the filtering and dissemination framework in the RESCUE project at: <http://www.itr-rescue.org/index.php>

The goal of the RESCUE project is to radically transform the ability of responding organizations to gather, manage, use, and disseminate information within emergency response networks and to the general public. Depending upon the severity of the crisis, response may involve numerous organizations including multiple layers of government, public authorities, commercial entities, volunteer organizations, media organizations, and the public. These entities work together to save lives, preserve infrastructure and community resources, and to reestablish normalcy within the population. The efficacy of response is determined by the ability of decision-makers to understand the crisis at hand and the state of the available resources to make vital decisions. The quality of these decisions in turn depends upon the timeliness and accuracy of the information available to the responders. The RESCUE project is an interdisciplinary effort that brings computer scientists, engineers, social scientists, and disaster science experts together to explore technological innovations in order to deliver the right information to the right people at the right time during crisis response.

2. The datasets used in this chapter are available from UCI machine learning repository at: <http://www.ics.uci.edu/~mlearn/MLRepository.html>
3. General classification and ranking system CBA can be downloaded from: <http://www.comp.nus.edu.sg/~dm2/>

CBA (v2.1) is a data mining tool developed at School of Computing, National University of Singapore. Its main algorithm was presented in KDD-98.

The paper is entitled “Integrating Classification and Association Rule Mining” (KDD-98). Further improvements were made from the ideas in the papers presented at KDD-99 and KDD-00. CBA originally stands for Classification Based on Associations. However, it is not only able to produce an accurate classifier for prediction, but also able to mine various forms of association rules.

Questions and Discussions

1. From intelligence analysis perspectives, other than the real-life scenarios illustrated in Sect. 11.1, can you think about a few more real-life examples, for which filtering targeted information from data streams are useful to the analysts?
2. In this chapter, we discuss the information filtering mechanisms mainly from structured (relational) data sources. Can the techniques be applied to the unstructured or semi-structured data sources? For instance, can we execute the same or similar queries using general search engines like Google?
3. Most of the database management systems (DBMS) like ORACLE, DB2, and SQL server support user defined types (UDT) and user defined functions (UDF). Can you choose one of the systems; implement a similarity search query that is similar to the one in Example 3?
4. Relevance feedback is of major importance to the refinement systems like FREESIA. In this chapter, we introduce the page-level feedback as a way to enabling fast information feedback cycle and reducing the burden from the user in providing the relevance feedback. However, within a page-level feedback, there may be inconsistencies. For example, majority (e.g., 90%) of the records in a page level feedback are relevant, but there are still irrelevant records. How does FREESIA handle these inconsistencies?
5. In this chapter, we focus on the 2-level feedback – relevant and irrelevant, can more levels of feedback help the refinement process? Can you convert a three-level feedback system (e.g., relevant, neutral, irrelevant) to a two-level feedback system? If you could, what are the advantages and disadvantages?
6. Without differentiating short-term and long-term profiles, a system could be easily trapped by the short-term information need. What are the problems when a system is too bias towards short-term information? How does FREESIA balance the two kinds of information need?

Antonio Badia

Computer Engineering and Computer Science Department,
Speed School of Engineering, University of Louisville, USA
abadia@louisville.edu
<http://date.spd.louisville.edu/badia>

Abstract. The Intelligence Cycle is one of information processing: data is gathered, organized, analyzed, and summarized. In the current environment, there is a large amount of data available for Intelligence Analysts (IAs); therefore, a considerable amount of time is spent in the gathering and organizing phases, leaving little time for analysis and summarization. Thus, software tools that help with acquiring and categorizing new data are extremely useful to IAs.

The concept of Personal Information Management (PIM) is currently a hot topic of research, although its basic ideas have a long history. In this chapter we argue that many of the goals of PIM research are highly relevant to tasks that gather and organize large amounts of data; in particular, to Intelligence tasks. Hence, the intelligence community should pay attention to the developments in this area. After providing some basic background on the Intelligence Cycle, we examine the concept of PIM, point out some issues that are relevant to intelligence work, and discuss some areas of research that PIMs should address in order to be even more relevant to the Intelligence Community.

12.1 Introduction

The concept of a Personal Information Management (PIM) system is currently a hot topic of research [21]. While there is no universal definition of PIM, several research groups are working under this label on related projects [14]. Informally, PIM refers to methods and procedures to acquire, store, organize (classify, connect), search and retrieve personal information in order to serve certain needs. Some additional comments help sharpen this otherwise vague definition. First, the term “personal information” is used broadly to encompass any informational item, from an email to a book to a photo album, that an individual owns or controls. Hence, it is not limited to a certain media or a certain format. Second, the different manipulations of information are all considered in the context of “information needs”, that is, they serve a certain purpose -they support a certain activity, or are necessary to take certain decisions [14]. Thus, in PIM such manipulations do not exist in the abstract, but always in a certain context. Finally, in many cases PIM research has given a concrete form to this definition by focusing on day-to-day activities carried out in a certain environment, for instance the workplace. This is what gives PIM a distinctive character, making it different from work in Information Retrieval or other areas.

Many of the ideas considered in PIM research have a long history; for instance, search tools rely heavily on Information Retrieval techniques and show similarities with Web search engines. As another example, the need for information integration has a deep tradition and is a complex issue that has been investigated in the past in other contexts [10]. While not a completely new idea, PIMs are the point of encounter in the evolution of several trends: tools (appointment managers, to-do lists, calendars); hardware (new digital gadgets: PDA, smart phones); ideas (“record everything”/“digital memories”, “compute anywhere”); and, most importantly, needs: in a data-intensive, information-rich world, we sometimes fail to find what we’re looking for, even though we know it’s in there, somewhere.

The idea of PIM is not directed to a specific set of users, and may be beneficial to a large segment of the population, since it emphasizes simple, intuitive access to information. In fact, many PIM projects clearly have in mind non-expert users. However, professionals that deal with information processing tasks stand to benefit the most from the ideas proposed. In this sense, Intelligence Analysis can benefit considerably from some of the proposals currently being developed. In this position chapter, we argue that indeed PIM ideas and tools can carry valuable support for Intelligence Analysis tasks. In order to develop this idea, we introduce some basic concepts about the Intelligence cycle, and some common characteristics of PIM systems. We then show that said characteristics are well fit to the steps of the Intelligence Cycle. Therefore, the Intelligence Community should pay close attention to developments in the area, as PIMs may become useful tools for Intelligence analysts (IAs). The benefit may be mutual, as IAs may provide PIM researchers with ideas and challenges based on their experience. At the same time, we argue that some special characteristics of Intelligence Analysis are not well covered by current research and therefore specific topics need to be addressed in order to make valuable contributions to Intelligence tasks. We propose a list of such topics and discuss their impact. Finally, we close with some conclusions.

12.2 Background

Before we can discuss the main issue of this chapter (whether PIM research is of relevance for Intelligence Tasks) we need to review a few basic concepts from the involved fields. In the next sections, we introduce some ideas that will be used later in the discussion.

12.2.1 The Intelligence Cycle

The ultimate goal of intelligence analysis is to provide a customer, military or civilian, with the best possible information to help in taking policy, strategic and tactical decisions that affect national security. In this task, “intelligence” is used to refer to knowledge and information, the basic end product of the analysis. Such analysis is carried out by highly trained analysts, who work in a continuous process involving the following steps (the following is summarized from [1, 2, 17, 23]):

- *Needs.* Customers (policymakers and others) make requests that the analyst must translate to specific requirements and tasks, in order to make sure that the final

product answers the needs of the customer. Customer demands often need interpretation or analysis before they can be expressed as an intelligence requirement. The customer may have additional constraints on the intelligence product; the request may have time constraints (short-term versus long-term) or scope (broad or strategic versus narrow or tactical).

- *Collection.* This refers to the gathering of raw (uninterpreted) data. Nowadays, there is an abundance of data, due to the variety and richness of sources: Signal Intelligence (SIGINT), which includes information from radar, telemetry, and intercepted communications; Imagery Intelligence (IMINT), which refers to images delivered by electronic means, mostly satellites; Measurement and signature intelligence (MASINT), or data produced from sensors (chemical, acoustic...) other than SIGINT and IMINT; Human-source intelligence (HUMINT) which refers to data provided by informants, either through clandestine means, or through official contacts with allied nations, or through diplomatic missions. Finally, we have OSINT, open source information, which refers to publicly available information (radio, television, newspapers, commercial databases, etc.). This last category stands in contrast with all previous sources, which are usually classified and not open. The importance of OSINT has grown tremendously in the past few years as more and more relevant information has become available online. For instance, some groups are using the Web for propaganda, recruiting and instructional purposes (the so-called “Dark Web”). In order to monitor such groups, the Dark Web has become an invaluable source of information.
- *Processing and Exploitation.* On this stage, the raw data is converted to a form suitable for further analysis. This includes translation of documents in foreign languages, analysis of sensor data, decoding of messages, etc. These tasks consume a large amount of resources from intelligence agencies, since many of them are labor-intensive, and specialized personnel is needed to carry them out. Moreover, in this phase evaluation of the accuracy, reliability and meaning of the raw data (which continues in the next step) gets started.
- *Analysis and Production.* On this stage the processed data is integrated, interpreted, and evaluated. On this crucial phase, the analyst must assess (1) how reliable and complete the data pieces are; (2) how distinct pieces of data can be interpreted; and (3) how they fit in an overall picture. The first task is needed since many times the sources of information are not trustworthy, and an adversary may leave indications that actually mislead an intelligence agency, in order to disguise real intentions. The second task is needed since raw data is rarely unambiguous; the same act (for instance, buying fertilizer) may signal completely different intentions depending on the context (to work on a farm, or to prepare explosives). The last task is needed since data is rarely complete; after all collection is done, analysts usually have only fragmented and sometimes unrelated evidence. Finally, even after some conclusion is reached, there are two tasks left: first, analysts try to verify their work by correlating finished intelligence with data from other sources, looking for supporting evidence and/or inconsistencies. Because the process is far from exact, and is based on partial, tentative evidence, all conclusions reached are by necessity also tentative, best-estimate interpretations. Note that in this step we go from facts to interpretation and judgment; hence, it is in this step that the danger is greater for presumptions, biases and other problems to arise. In the second and final task, the

created intelligence must be tailored to the customer, and an effort must be made to make sure that the product answers the customer's needs. In particular, the information produced must be relevant to the original answer, as accurate as possible (and, if uncertain, accompanied by some measure of its certainty), objective, usable (i.e. actionable) and timely.

- *Dissemination.* This is the process of delivering the finished product to the consumer. Sometimes, this is followed by the consumers providing feedback to the intelligence analyst so that the process can be improved.

The environment in which intelligence services operate has changed radically in the last few years. In contrast with the information scarceness of the Cold War, we live in a situation of information overload, as the amount of available raw data has grown considerably. This growth is due to several reasons: more sophisticated signal intelligence, open sources (e.g. the Dark Web mentioned above), and others. However, this data are not always reliable, almost never complete, and the truly interesting facts are usually buried in the middle of large amounts of irrelevant facts. As a result, it is estimated that currently between 60% and 80% of an IA's time is spent in the *Collection and Processing and Exploitation* phases, leaving little time for other phases (notably, *Analysis and Production*). Clearly, IAs could benefit from software that facilitates gathering and organizing pieces of information, ideally without regard for questions like format, etc. which are of no interest to the analyst [19].

12.2.2 Personal Information Management (PIM)

There is no uniform, universal notion of PIM. However, most research in the area can be seen as focusing on the study of information-related activities: gathering (through search or other means), organizing (in directories or otherwise), maintaining and retrieving information [14]. There is an emphasis on activities that are related to everyday tasks, that is, the information-related activities are performed with a particular goal in mind, and are carried out in the context of day-to-day processes. For instance, organization and retrieval of e-mail messages are important activities.

Another important aspect is the integration of heterogeneous data. Current window-based interfaces depend on the binding of documents to applications, causing different pieces of information (say, those in a spreadsheet, versus those in a document, or those in a database) to be accessed through different applications. Relating data across applications, although supported on a primitive level by current tools (cut and paste), is cumbersome and does not allow for cross-document analysis. The overall goal of PIM is to “always have the right information in the right place, in the right form, and of sufficient completeness and quality to meet our current needs” [14]. Clearly, to meet this goal, relevant information must be managed regardless of what application handles it.

Some of the key concepts in PIM are [13, 14, 21]:

- The system should be able to manipulate any kind of data, no matter what application created it or how it was acquired (cross-application scope). The system should be able to deal with data that evolves over time, either in content or structure (“life-long data management”). This implies a need to deal with cumulative collections

that may grow considerably in size. This in turn generates a need to store efficiently and flexibly.

- Data items do not exist in isolation, they should be associated (linked, correlated) with others. Ideally, the system creates associations of different types for the user. At least some of these associations support data integration; that is, identifying identical or overlapping data items as such. Note that if the system wants to connect items for the user, and the data changes and evolves, the system should ideally adapt to and learn about the user, perhaps using machine learning and statistical techniques.
- Keeping everything is not enough (may in fact make things worse); making everything visible is what matters [9]. The ultimate goal is to find it, no matter what we remember about it, when and how we stored it (even if we forgot about it!) This generates the need for powerful and flexible search that allows access to data by multiple features: by recollection, by correlation, by detail, by metadata. As an example, when trying to locate an old picture or email, we may search by content (we remember that the photo contains a sunset, or that the email is about lunch today), metadata (the photo was taken in town, the email is from 2 days ago), context (the photo was taken in Bermudas, the email was a communication in the context of a project), association (the photo was taken while in vacation, the email is from the same person who called yesterday). The most advanced research deals with Proactive Information Gathering (also called “Finding Without Searching” in [9]). The idea is to analyze the current user context, identify her information needs, and automatically generate queries. Note that this approach may help find information that the user forgot about.

While not all systems deal with all these characteristics, most researchers seem to agree that these are all part of the desirable functionality for a PIM [12].

12.3 Case Studies

We briefly describe three PIM systems that exhibit some of the characteristics described above: Haystack, Semex and Stuff-I’ve-Seen. For each one, we analyze possible uses as an Intelligence Analysis tool.

Haystack is a project developed at MIT. Haystack is based on a simple domain model of objects with properties and relationships among them. Each object is given a Uniform Resource Identifier (URI) so it can be named and connected to other objects. An arbitrary set of properties and relationships can be associated with each object. Internally, Haystack stores all information in RDF (Resource Description Framework [25]); a set of extractors is used to import data, transforming the original format into RDF.

Haystack provides a very flexible user interface, in which hard-coded menus are substituted by *view prescriptions*, descriptions of how information should be presented, which are fully customizable. In this interface, all objects can be browsed (by simply clicking on them) and connected (by dragging and dropping one object into another). Because it works across applications, Haystack allows linking an email with a document, a picture, or a spreadsheet. In the Haystack interface, operations on objects can be naturally invoked as the system automatically sorts out what is the type of an object and what operations are possible in that type. For instance, right-clicking on

a picture opens up a menu with operations that are possible in (and typical of) pictures, like cropping, zooming, etc.

Search is the most important operation in Haystack, and is supported in a variety of ways: simple keyword search, full database-like queries (against the underlying RDF database), and browsing/navigating through the different views, are all allowed.

It is obvious that a tool like Haystack has some positive aspects that would make it useful to an IA: the ability to store information, regardless of format; the ability to create relationships between objects easily; the ability to transcend applications would all facilitate the data gathering task. On the other hand, data in Intelligence comes many times with a question about its validity: being able to assign different levels of trust or confidence to the data is a must. There is no specific ability to do this in Haystack (although the RDF model could certainly support such an extension). Also, gathering evidence from different sources requires that sometimes data is merged, or at least contrasted and examined side-by-side. Again, there is no direct support in Haystack for integrating information. Finally, most analysts follow some flow of reasoning for their task: they put together evidence to prove or disprove a point; they make inferences from the available data (and their background knowledge); they generate hypothesis, and test them. There is no support for reasoning in Haystack. Obviously, Haystack (and the other systems we describe here) were not developed with the goal of supporting Intelligence tasks, and therefore cannot be faulted if they miss some of the components that an IA would find useful. Our purpose in pointing out the lack of certain features is to highlight potential research issues that could turn existing PIM prototypes in highly useful tools for Intelligence analysis.

Semex (SEMantic Explorer) is a system developed at the University of Washington. The emphasis in SEMEX is in allowing flexible and powerful search of data items; towards this end, it offers search-by-association. The basic intuition is that people naturally organize information about entities (people, projects, etc.) and connect them by association. To support the approach, the system automatically builds a database of objects and associations among the objects. Objects can be, for instance, people, publications, etc., while associations could be any type of relation, like AuthoredBy (between a document and a person), Friend (between persons), etc. The system obtains associations by examining the objects in the user's desktop; for instance, analyzing email messages easily leads one to obtain information about associations like Sender, Receiver, and MentionedIn. Also, the user is allowed to define objects and associations from existing ones: thus, a CoAuthor would be defined from the AuthoredBy association, which in turn could be extracted from documents by an extractor that understood the document's layout (PDF, Word, etc.). An important point is that SEMEX updates the information in the database by crawling the desktop periodically.

One issue that SEMEX must deal with is the fact that (real world) objects may be referred to by more than one name; therefore, *reference reconciliation* (determining when two or more references are to the same object) is needed to keep the database consistent and free of redundancy. SEMEX achieves this through the use of several techniques derived from machine learning and statistics.

SEMEX offers an interface to the user for browsing the object database. In this interface, the user can do keyword search or selection search. In the latter, a series of available fields are offered to the user to fill in with values, just like a Web form; a

filled field acts as a filter that any object must satisfy. For any search, SEMEX retrieves not only the set of objects that contain the keyword or pass the query filters, but also objects that are strongly related to those ones: these are objects that are related to several of the directly retrieved objects, usually by a single association or by a chain of associations that are deemed to have a significant weight (the weight of an association is based on its type or given by the user).

Semex has, overall, strong similarities with Haystack –an expected outcome since both systems aim at very similar goals. Therefore, the list of strengths and weaknesses of Semex for Intelligence tasks is quite similar to that of Haystack. Semex could be considered stronger for accumulating and organizing data thanks to its emphasis in reference reconciliation –a feature missing in Haystack. However, Haystack’s interface seems a bit more flexible than Semex.

The Stuff-I’ve-Seen (SIS) system is being developed at Microsoft. The main motivation behind SIS is to allow people better control of their desktop and, in particular, to control information overload. It is very easy to accumulate data in the digital domain; the hard part is to organize it and retrieve relevant portions as needed. Compared with Haystack and Semex, SIS takes an extra step in the search part. The system tries to learn the contexts in which the user asks for information (when writing a report, for instance), and to anticipate requests for information next time the user is in the same context. The ability to proactively find task-relevant information [9] has led to the development of an independent module of SIS, the Implicit Query prototype. The system indexes objects (currently, it seems restricted to email messages) in order to later help generate queries automatically that will retrieve needed information without the user having to explicitly ask for it. An interesting aspect of this research is that it may lead to users finding information that they had forgotten about, an obvious asset when our capability to accumulate data surpasses our ability to organize and index it.

Compared to Haystack and Semex, SIS may not possess as rich an interface, and may not be as strong in supporting inter-application access, but the IQ prototype gives it a powerful tool which could indeed be very useful for an IA. As stated above, an IA will go through a certain process when carrying out a task (gathering data, integrating, evaluating, analyzing, summarizing). Similarities across these steps could help IQ to generate the queries that the analyst itself would, thus saving time and effort –always in short supply in Intelligence work. However, it is unclear if the IQ prototype should be extended with a more powerful notion of task (similar to those of workflow models) before it can be used in such an environment.

12.4 Discussion: PIM for Intelligence Analysis

The main issue of this chapter is: how relevant is PIM to Intelligence tasks? We argue that work on PIM is *very relevant* to IAs.

To a large extent, the task of an IA can be described as the gathering, analysis and evaluation of data [1, 2, 15]. The gathering step has been greatly facilitated by advances in networking and tools like Web search engines. However, other steps have not evolved as rapidly; the ability to capture and store data outstrips the capability to

analyze and evaluate it. To make things worse, not only the amount of data, but also the degree of heterogeneity has increased markedly. Heterogeneity in this context refers to the differences in structure or meaning of the data: the same information may be structured in several ways (for instance, the same fact may be stated in a sentence in a natural language in a document and as an entry (row) in a database). Even if structured in the same way, the fact may be presented in different ways: two different documents may refer to the same event using different vocabulary, or describe two (perhaps overlapping) aspects of the same event. Also, it is possible to have related information in different media: a narrative describing an event, and pictures (or video) of the same event. This diversity greatly complicates the analysis and evaluation steps. In intelligence work, it is often necessary to check facts, determine their source and judge their relevance, completeness and trustworthiness. Having evidence for the same event scattered across multiple sources adds to the challenge. Finally, when evaluating (which includes summarizing the distilled information, interpreting it and assigning it a degree of credibility), one should be able to integrate all bits and pieces of information necessary, regardless of format or origin, to make an overall assessment (since that assessment must take into account the number and quality of sources).

It is already established that there are information technology tools and techniques that are beneficial to Intelligence Tasks [18,19, 22]. However, it is also clear that some needs of Intelligence Analysis are not (adequately) covered by current tools. Even though the Intelligence Community has invested heavily in information technology, and has several systems available (some customized to its needs), there is still ongoing research on better, more intelligent and flexible tools. One of the challenges is that most analysts have developed their own work routines; only tools flexible enough to be used without causing undue change to analysts' habits can be successful [4, 24].

Because of this situation, the concept of PIM is especially well suited to intelligence analysis, for several reasons. First, the tasks that PIM seeks to support (gathering, organizing, maintaining information in support of everyday activities) are a perfect fit for intelligence analysis. Second, PIM aims at providing support to these tasks through various ideas: the emphasis of data integration helps in dealing with heterogeneity, as does the idea of cross-application scope; flexible search helps in the analysis phase, both in finding the right bits and in making connections (the famous "connect the dots" phase). Finally, PIM systems stress simplicity and ease of use; they do not require difficult setups or a steep learning curve. Thus, one can conclude that a PIM system (or a suite of PIM tools) would be highly valuable for Intelligence analysis. Indeed, the goal of having the right information in the right place at the right time should be very appealing to the Intelligence analyst.

At the same time, it must be acknowledged that current state of the art tools still does not provide functionality that could be considered necessary to support Intelligence work. In particular, we put forth the following needs:

- *Support for external sources:* currently, most work on PIMs focus on integrating information in a person's desktop. However, more and more we rely not only on internal sources (documents in one's computer, received email), but also on external sources (documents from the Web, databases to which a user has access). No

computer is an island, and therefore there is a need to support search and integration of information not currently in the desktop. Note that one could argue that tools to search such internal sources are enough: once some relevant artifact (document, email, etc.) is found, it can be downloaded and is now part of the desktop. However, this creates an artificial division between search and other tasks. Many times, in order to determine if an artifact is indeed relevant, some level of analysis beyond pure search (i.e. keyword appearance) must be carried out. It is not sensible (or a good use of resources) to download everything that could be useful. Instead, analysis tools should be extended to deal with data in external sources.

- *Support for querying as an iterative, interactive process:* most PIM concentrate on providing search capabilities, leaving aside traditional querying of data sources, i.e. using SQL against a database. However, it is clear that both actions are part of a continuum: both are tools that support data analysis. Thus, both activities should be fully integrated. In doing so, lessons from early research must be applied -and the most important lesson, probably, is the fact that search/querying is an iterative, interactive process. It is rare that users start with a clear and crisply defined goal in mind, or that they know exactly how to search for what they want (i.e. they know the exact keywords, or they know the exact directory or data source that contains all relevant data). Thus, as searching is expanded to mix with querying, we must make sure that we respect the characteristics of this process. While one may argue that this is an important but generic capability, we argue that it is especially important in Intelligence work, since many times data analysis implies an exploration of the data. During this process, hypothesis may be developed, tested, discarded or changed; the analyst may pursue several possible scenarios (what-if analysis), and explore alternative explanations for some situations. This is not to say that this is not a useful capability in other fields -but it is central to Intelligence work.
- *Support for ongoing monitoring:* a large part of warning intelligence rests on keeping track of an ongoing situation, finding clues that will set the alarm before the crisis occurs [17]. Thus, as new data and information are acquired, one would like the new to be connected to the old and any changes to be reflected in events under scrutiny. PIMs do aim at linking data and seamlessly incorporating new data into a general repository [11]. However, what is missing is the ability to define arbitrary events that the system should keep an eye on, as well the ability of having the system, on its own, determine that some events are of interest. Current research on stream analysis and mining [7, 8] is very relevant here, but it is still in a state of fast change and has not been integrated into PIMs yet.
- *Support for groupwork:* analysts rarely work on total isolation. Groups must be formed, many times dynamically, for complex tasks. Some initial research is reported in [11], where the concept of PIM is extended to GIM (Group Information Management). This is a step in the right direction; however, much work remains to be done. In particular, in the context of Intelligence analysis one must support several ways to collaborate, from selective sharing of information (i.e. the analyst decides what can be shared) to total search (i.e. search engines are allowed full access to all the information from all the participants in a group). Moreover, support for information integration, already mentioned above as a key component of PIMs, must be leveraged to the group level. This adds another degree of difficulty to this already complex task.

- *Support for processes (workflow):* Intelligence analysts try to follow processes in their work. In Intelligence work, it is not only necessary to reach some conclusions, but also be able to show how the conclusions were reached, what evidence supports them, and what other alternatives there may be; and given that data may be incomplete, unreliable and even contradictory, it is necessary to examine available evidence in a methodical manner. Thus, analysts may start by browsing through the data until some hypothesis emerges. At that point, evidence for and against the hypothesis may be sought. Depending on the results of this search, the evidence may become the conclusion, or alternatives may be explored. As steps are taken, it is necessary to keep the links between data, hypothesis, supporting evidence, and conclusions. Drilling down to detail, backward and forward analysis through the chain of evidence, structuring of arguments, all should be supported. Note that we are talking at the individual level; however, as pointed out in the previous point, many times the analyst is working as part of the team. In such a situation, there usually is a division of concerns, each participant working on a certain role that is related to others; this must also be supported by workflow tools, in order to facilitate teamwork and collaboration. Thus, workflow tools are needed at two levels, team and individual.
- *Support for security:* security is used here to refer to the overall need to protect sensitive and classified material while at the same time making sure that the right people have access to the right information. This is a delicate and difficult balancing act, with which the Intelligence Community is constantly struggling. Even though there has been research in this area [15], more remains to be done before a system can be deployed in real intelligence work.

While PIMs may not have all the characteristics needed by a good Intelligence analysis tool, what makes the concept so promising is that the goals that guide the development of PIMs are very similar to those of Intelligence analysis. The PIM emphasis on across-application capabilities, data linking, and flexible organization and search provide the right foundation for a highly useful Intelligence system. While some current tools may do some things well (some special-purpose software for Intelligence analysis, for instance, has support for Intelligence workflow), they are proprietary and usually cannot talk to each other. Hence, most of what they lack may be very hard to integrate into the system. PIMs, on the other hand, make minimal commitments to a particular data model and minimal assumptions about the data itself, and therefore are more open to expansion and customization.

12.5 Conclusion

We have examined the concept of PIM and argued that many of the ideas involved are relevant to Intelligence tasks. We have then proposed some further research directions that could make this concept even more relevant to the Intelligence Community. By examining points of contact between PIM and Intelligence analysis, we hope to stimulate research in the area that will be useful to the Intelligence Community, as well as involve the Intelligence Community in the research and development of the concept.

References

1. A Consumer's Guide to Intelligence, Washington, DC, CIA Public Affairs Staff (1995)
2. A Compendium of Analytic Tradecraft Notes, Washington DC, CIA, Directorate of Intelligence (1997)
3. Badia, A.: Knowledge Management and Intelligence Work: A Promising Combination. In: Encyclopedia of Knowledge Management., Idea Group, USA (2005)
4. Brooks, C.C.: Knowledge Management and the Intelligence community. Defense Intelligence Journal 9(1), 15–24 (2000)
5. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, ACM Press (1995)
6. Cuttrell, E., Dumais, S., Teevan, J.: Searching to Eliminate Personal Information Systems. In: Personal Information Management, vol. 49(1), ACM Press, New York (2006)
7. Demers, A., Gehrke, J., Riedewald, M.: The Architecture of the Cornell Knowledge Broker. In: Proceedings of the Second Symposium on Intelligence and Security Informatics (ISI-2004) (2004)
8. Dobra, A., Garofalakis, M., Gehrke, J., Rastogi, R.: Processing Complex Aggregate Queries over Data Streams. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data. ACM Press, New York (2002)
9. Dumais, S., Cuttrell, E., Sarin, R., Horvitz, E.: Implicit Queries for Contextualized Search. In: Proceedings of the International Conference on Research and Development in Information Retrieval. ACM Press, New York (2004)
10. Elmagarmid, A., Rusinkiewicz, M., Sheth, A. (eds.): Heterogeneous and Autonomous Database Systems. Morgan Kaufmann, San Francisco (1999)
11. Erickson, T.: From PIM to GIM: Personal Information Management in Group Contexts. In: Personal Information Management. Communications of the ACM, vol. 49(1). ACM Press, New York (2006)
12. Franklin, M., Halevy, A., Maier, D.: From Databases to Dataspaces: A New Abstraction for Information Management. SIGMOD Record 34(4), 27–33 (2005)
13. Halevy, A.: Semex: A Platform for Personal Information Management and Integration (2005), <http://www.cs.washington.edu/homes/alon/>
14. Jones, W.: A Review Of Personal Information Management, IS-TR-2005-11-01, The Information School Technical Repository, University of Washington (2005)
15. Karat, C.M., Brodie, C., Karat, J.: Usable Privacy and Security for Personal Information Management. In: Personal Information Management. Communications of the ACM, vol. 49(1). ACM Press, New York (2006)
16. Karger, D., Jones, W.: Data Unification in Personal Information Management. In: Personal Information Management. Communications of the ACM, vol. 49(1), ACM Press, New York (2006)
17. Krizan, L.: Intelligence Essentials for Everyone. Occasional Paper n. 6, Joint Military Intelligence College, Washington DC (1999)
18. Mena, J.: Homeland Security: Techniques and Technologies. Charles River Media (2004)
19. Popp, R., Armour, T., Senator, T., Numrych, K.: Countering Terrorism Through Information Technology. Communications of the ACM, special issue on Emerging Technologies for Homeland Security (2004)
20. Sageman, M.: Understanding Terror Networks. University of Pennsylvania Press (2004)
21. Teevan, J., Jones, W., Bederson, B.: Personal Information Management. In: Communications of the ACM, vol. 49(1). ACM Press, New York (2006)

22. Thuraisingham, B.: Web Data Mining and Applications in Business Intelligence and Counter-Terrorism. CRC Press, Boca Raton (2003)
23. Various, A Consumer's Guide to Intelligence. Office of Public Affairs, Central Intelligence Agency (1998)
24. Waltz, E.: Knowledge Management in the Intelligence Enterprise. Artech House Information Warfare Library (2003)
25. Resource Description Framework (RDF) <http://www.w3.org/RDF>

Online Resources

1. Resource for Personal Information Management:
<http://pim.ischool.washington.edu>
2. Haystack project:
<http://haystack.csail.mit.edu/home.html>
3. Semex project:
<http://www.cs.washington.edu/homes/alon/>

Questions for Discussions

1. What characteristics of the Intelligence Cycle are similar to those of Business Intelligence? Which ones are unique to the Intelligence Cycle (and hence require customization of software solutions)?
2. What facilities (if any) are provided by modern Database Management Systems (DBMS) to support PIM functionality? Could a PIM system be based on a current DBMS? If not, what is missing?
3. Do you know of any systems, besides the 3 ones mentioned, that could be called a PIM system? What characteristics do they have that makes them a PIM system?
4. In the chapter, a list is given of characteristics that PIM systems are currently missing and that would be beneficial for Intelligence tasks. For each of the characteristics, analyze how much effort, in your estimate, it would take to add the characteristic to a PIM system. Try to identify system functionality that would make the task easier.
5. Many tools that IAs use are based on *social network theory* and tend to model and display data as a graph. What special requirements do such tools present in order to be integrated with other tools? What can PIM systems do to incorporate such tools?

A Data Miner's Approach to Country Corruption Analysis

Johan Huysmans¹, Bart Baesens^{1,2}, and Jan Vanthienen¹

¹ Department of Decision Sciences and Information Management,
Katholieke Universiteit Leuven, Belgium

FirstName.LastName@econ.kuleuven.be

² School of Management,
University of Southampton, United Kingdom

Abstract. Corruption is usually defined as the misuse of public office for private gain.¹ Whereas the practice of corruption is probably as old as government itself, the recent emergence of more detailed measures has resulted in a considerable growth of empirical research on corruption. Furthermore, possible links between government corruption and terrorism have attracted an additional interest in this research field. Most of the existing literature discusses the topic from a socio-economical perspective and only few studies tackle research on corruption from a data mining point of view. In this chapter, we apply various data mining techniques onto a cross-country database linking macro-economical variables to perceived levels of corruption. In the first part, self organizing maps are applied to study the interconnections between these variables. Afterwards, various predictive models are trained on part of the data and used to forecast corruption for other countries. Large deviations for specific countries between these models' predictions and the actual values can prove useful for further research. Finally, projection of the forecasts onto a self organizing map allows a detailed comparison between the different models' behavior.

13.1 Introduction

The amount of empirical research concerning corruption is impressive. An overview can be found in [12, 19]. Most of existing literature tries to find causal relations between some explanatory variable and the perceived level of corruption. For example, in [4] the influence of democracy on the perceived level of corruption is tested while other studies focus on the influence of religion [29], colonial heritage [29], abundance of natural resources [20] or the presence of women in parliament [27]. Other studies focus on the consequences of corruption: does corruption lead to a decrease of GDP, foreign investments or aid [1, 21]? The main problem in all these empirical studies is to make the transition from 'highly correlated' to 'causes': many variables are highly

¹ This chapter is a revised and extended version of 'Country Corruption Analysis with Self Organizing Maps and Support Vector Machines', International Workshop on Intelligence and Security Informatics, Springer Verlag, Lecture Notes in Computer Science 3917, pp.103-114, 2006.

correlated with the perceived level of corruption, but it is difficult to derive causal relations from it. For example, in [19, 24] is reported that GDP per head and corruption are reported to be highly correlated in most studies but that there is a general agreement that there is no causality involved.

The general approach in the majority of these studies (e.g., [4]) is to regress a variable representing corruption on a number of independent variables for which the influence is tested with the possible inclusion of some control variables. In this paper, we apply a different technique to study corruption. We use self organizing maps (SOMs), also known as Kohonen maps, to gain deeper insight in the causes of corruption. This technique stems originally from the data mining community and allows a clear and intuitive visualization of high-dimensional data. In the second part of the paper, we apply CART regression trees and support vector machines (SVMs) to forecast the perceived levels of corruption. Support vector machines have proven to be excellent classifiers in other application domains (e.g., credit scoring [3]) and are able to capture nonlinear relationships between the dependent and independent variables.

In the next part, the machine learning techniques that were applied during this study are presented and afterwards, we describe in detail the different data sets that were used. The next section discusses the application of SOMs on this data whereby special attention is paid to the visualization possibilities that these models offer. In the final section, we use decision trees and least-squares support vector machines [26] to forecast the perceived levels of corruption. Input selection will be used to select the most significant variables. The main contribution of this paper is the projection of SVM predictions onto a SOM to gain more insight in the SVM model and the use of multi-year data sets to study evolutions in the perceived level of corruption.

13.2 Overview of Data Mining Techniques

In this section, we discuss the data mining techniques that were adopted in this study. First, we present the basic ideas behind Self Organizing Maps, an unsupervised technique that was adopted to visualize the correlations in high-dimensional data. Afterwards, we discuss briefly the two predictive algorithms that were used in this study, besides the well-known OLS regression. First, we discuss CART, an algorithm to learn decision trees. Afterwards, we present the basics of performing regression with Least-Squares Support Vector Machines (LS-SVM). Readers that are familiar with these techniques can skip the next paragraphs.

13.2.1 Self Organizing Maps

SOMs were introduced in 1982 by Teuvo Kohonen [16] and have been used in a wide array of applications like the visualization of high-dimensional data [31], clustering of text documents [13], identification of fraudulent insurance claims [6] and many others. An extensive overview of successful applications can be found in [17] and [20]. A SOM is a feedforward neural network consisting of two layers. The neurons from the output layer are usually ordered in a low-dimensional grid. Each unit in the input layer is connected to all neurons in the output layer with weights attached to each of these connections. This is similar to a weight vector, with the dimensionality of the input space, being associated with each output neuron. When a training vector x is

presented, the weight vector of each neuron c is compared with \mathbf{x} . One commonly opts for the euclidian distance between both vectors as the distance measure. The neuron that lies closest to \mathbf{x} is called the ‘winner’ or the Best Matching Unit (BMU). The weight vector of the BMU and its neighbors in the grid are adapted with the following learning rule:

$$W_c = W_c + \eta(t) \Lambda_{\text{winner},c}(t) (\mathbf{x} - w_c) \quad (13.1)$$

In this expression $\eta(t)$ represents the learning rate that decreases during training. $\Lambda_{\text{winner},c}(t)$ is the so-called neighborhood function that decreases when the distance in the grid between neuron c and the winner unit becomes larger. Often a gaussian function centered around the winner unit is used as the neighborhood function with a decreasing radius during training. The decreasing learning rate and radius of the neighborhood function result in a stable map that does not change substantially after a certain amount of training.

From the learning rule, it can be seen that the neurons will move towards the input vector and that the magnitude of the update is determined by the neighborhood function. Because units that are close to each other in the grid will receive similar updates, the weights of these neurons will resemble each other and the neurons will be activated by similar input patterns. The winner units for similar input vectors are mostly close to each other and self organizing maps are therefore often called topology-preserving maps.

13.2.2 CART

A very popular tree induction algorithm is **CART**, short for Classification and Regression Trees [5]. We will only discuss the version of CART used for induction of regression trees. A CART regression tree is a binary tree with conditions specified next to each non-leaf node. Classifying a new observation is done by following the path from the root towards a leaf node, choosing the left node when the condition is satisfied and the right node otherwise, and assigning to the observation the value below the leaf node. This value below the leaf nodes equals the average y-value of training observations falling into this leaf node. Variants [14] of CART allow the predictions of the leaf nodes to be linear functions of the input variables.

Similarly to a classification tree, the regression tree is constructed by iteratively splitting nodes, starting from only the root node, so as to minimize an impurity measure. Often, the impurity measure for regression problems of a node t is calculated as:

$$R(t) = \frac{1}{N} \sum_{x_n \in t} (y_n - \bar{y}(t))^2 \quad (13.2)$$

with (\mathbf{x}_n, y_n) the training observations and $\bar{y}(t)$ the average y-value for observations falling into node t . The best split for a leaf-node of the tree is chosen such that it minimizes the impurity of the newly created nodes. Mathematically, the best split s^* of a node t is that split s which maximizes:

$$\Delta R(s, t) = R(t) - p_L R(t_L) - p_R R(t_R) \quad (13.3)$$

with t_L and t_R the newly created nodes and p_L and p_R the proportion of examples sent to respectively t_L and t_R . Pruning of the nodes is performed afterwards to improve generalization behavior of the constructed tree. Pruning in CART is performed by a procedure called ‘minimal cost complexity pruning’ which assumes that there is a cost associated with each leaf-node. We will briefly explain the basics behind this pruning mechanism, more details can be found in [5, 15]. By assigning a cost α to each leaf node, we can consider the total cost of a tree to consist of two terms: the error rate of the tree on the training data and the cost associated with the leaf nodes.

$$\text{Cost(Tree)} = \text{Error(Tree)} + \alpha \text{ NumberLeafNodes(Tree)} \quad (13.4)$$

For different values of α , the algorithm first looks for the tree that has a minimal total cost. For each of these trees, the algorithm estimates the error on a separate data set that was not used for training and remembers the value of α that resulted in the tree with minimal error. A new tree is then constructed from all available data and subsequently pruned based on this optimal value of α . In case there is only limited data available, a slightly more complex cross-validation approach is followed, for which the details are provided in [5].

13.2.3 LS-SVM

In this section, an introduction to the ideas behind **Support Vector Machines** [30] is given. More specifically, it is explained how regression is performed with SVMs [27]. For more details on classification with SVMs, we refer to the tutorial by Burges [7].

Given a training data set of N observations $\{x_i, y_i\}_{i=1}^N$ with $x_i \in \Re^d$ and $y_i \in \Re$, the goal of regression is to find a function $f(x)$ that predicts as accurate as possible the output y for some input vector x . In ε -SV regression, this function is estimated by looking for a function $f(x)$ that has at most ε deviation from the actual output y_i for each of the training observations, and that at the same time is as flat as possible. For simplicity, we will start with the assumption that $f(x)$ is a linear function that can be written as:

$$f(x) = w^T x + b \quad (13.5)$$

with $w \in \Re^d$ and $b \in \Re$ the parameters to estimate. We want the function f to be as flat as possible. One possibility to achieve this is to minimize the norm of the weight vector, written as $\|w\|^2$. In summary, the problem to solve is the following

$$\begin{aligned} & \text{minimize } \frac{1}{2} \|w\|^2 \\ & \text{subject to } |y_i - w^T x_i - b| \leq \varepsilon \quad i = 1, \dots, N \end{aligned} \quad (13.6)$$

It is however possible that there is no solution to the above problem because the constraints are too strict. In order to allow for some errors, slack variables ξ_i are introduced. We can then reformulate the optimization problem as:

$$\begin{aligned}
 & \text{minimize} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \\
 & \text{subject to} \left| y_i - w^T x_i - b \right| \leq \varepsilon + \xi_i \quad i = 1, \dots, N \\
 & \quad \xi_i \geq 0
 \end{aligned} \tag{13.7}$$

whereby the relative importance of the slack variables can be regulated by changing the value of the positive constant C . A large value for C gives more importance to small slack variables, but the resulting function might fail to generalize to new observations. The constrained optimization problem of Eq. 13.7 can often be more easily solved in its dual formulation. We will only cover the main ideas here, a detailed description of the mathematical derivation of the dual formulation is given in [25]. The general idea is to construct a Lagrangian function for the constrained optimization problem and to partially differentiate this function with respect to w , b and ξ_i . From this differentiation, it follows that the function $f(x)$ of Eq. 13.5 can be rewritten as:

$$f(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) (x_i^T x) + b \tag{13.8}$$

with α_i and α_i^* being Lagrange multipliers. Observe that the function is rewritten as the weighted dot-product of each of the training observations with the new observation x . Moreover, it can be shown that for the training observations lying within ε -distance of the function f , the weights α_i and α_i^* will be zero. The dot-product must therefore only be calculated for those training observations that have non-zero coefficients. These observations are called the **support vectors**.

Until now, we have assumed that the function f is a linear function. Extending the theory to non-linear functions is however straightforward. The input patterns x_i can be preprocessed by mapping them from the original input space into a high dimensional feature space. Construction of a Lagrange function and derivation of this function with respect to the primal variables, allows us to rewrite Eq. 13.8 for the non-linear problem as follows:

$$f(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) (\varphi(x_i)^T \varphi(x)) + b \tag{13.9}$$

with φ representing the mapping from the original input space to the high-dimensional feature space. We can see from Eq. 13.9 that the mapping φ needs not to be explicitly known as only the dot-product of two mapped vectors is calculated. This dot-product can therefore be replaced by a Kernel function $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$. Some frequently used kernels functions are the linear kernel and the radial basis function (RBF):

Linear Kernel: $K(x_i, x_j) = x_i^T x_j$

Radial basis function: $K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{\sigma^2}}$.

By making small adaptations to the formulations from the previous section, we can obtain a solution that is computationally less expensive. This technique is called ‘Least Squares Support Vector Machines’ [26] and in this section, we discuss in detail how they can be applied for regression purposes [9].

The formulation of the regression problem is very similar to the approach of normal support vector machines, but the inequality from Eq. 13.8 is replaced by an equality. Additionally, the cost function to minimize includes the sum of the squared error terms instead of just their sum. So, the constrained optimization problem to solve becomes:

$$\begin{aligned} \min_{w,b,e} \mathfrak{J}(w, e_i) &= \frac{1}{2} \|w\|^2 + \frac{\gamma}{2} \sum_{i=1}^N e_i^2 \\ \text{s.t. } e_i &= y_i - w^T \varphi(x_i) - b \quad i = 1, \dots, N \end{aligned} \quad (13.10)$$

The solution to this optimization problem is found by the construction of following Lagrangian:

$$\ell(w, b, e_i, \alpha_i) = \mathfrak{J}(w, e_i) - \sum_{i=1}^N \alpha_i (e_i + w^T \varphi(x_i) + b - y_i) \quad (13.11)$$

where α_i represent the Lagrange multipliers associated with the equality constraints. After derivation of the Lagrangian with respect to w , b , e_i and α_i , one obtains the following equations:

$$\left\{ \begin{array}{l} \frac{\partial \ell}{\partial w} = 0 \Leftrightarrow w - \sum_{i=1}^N \alpha_i \varphi(x_i) = 0 \\ \frac{\partial \ell}{\partial b} = 0 \Leftrightarrow \sum_{i=1}^N \alpha_i = 0 \\ \frac{\partial \ell}{\partial e_i} = 0 \Leftrightarrow \gamma e_i - \alpha_i = 0 \quad i = 1, \dots, N \\ \frac{\partial \ell}{\partial \alpha_i} = 0 \Leftrightarrow e_i + w^T \varphi(x_i) + b - y_i = 0 \quad i = 1, \dots, N \end{array} \right. \quad (13.12)$$

After elimination of w and e_i , one obtains the following system of equations:

$$\left\{ \begin{array}{l} \sum_{i=1}^N \alpha_i = 0 \\ b + \sum_{i=1}^N \alpha_i \varphi(x_k)^T \varphi(x_i) + \frac{\alpha_i}{\gamma} = y_i \quad i = 1, \dots, N \end{array} \right. \quad (13.13)$$

The values for the $N+1$ unknowns (α_i, b) can be derived by solving this system of $N+1$ equations. From the first line of System 12, we obtain $w = \sum_{i=1}^N \alpha_i \varphi(x_i)$. Therefore the function $f(x)$, can be rewritten as

$$f(x) = w^T \varphi(x) + b = \sum_{i=1}^N \alpha_i \varphi(x_i)^T \varphi(x) + b = \sum_{i=1}^N \alpha_i K(x_i, x) + b \quad (13.14)$$

where we again use the Kernel function $K(x_i, x_j)$ to replace $\varphi(x_i)^T \varphi(x_j)$.

In summary, the general approach for performing LS-SVM regression is to first choose appropriate values for the hyperparameter γ and any kernel parameters, like σ^2 in the case of an RBF-kernel. Afterwards, we solve System 12 to obtain values for the α_i 's and b . Finally, forecasts for new observations are obtained from Eq. 13.14.

Selecting suitable values for the parameters γ and σ^2 (in case of a RBF-kernel) is of crucial importance for LS-SVM regression. An inappropriate choice for these parameters can significantly decrease the generalization behavior of the model. A large value of γ increases the importance of the second term in Eq. 13.10. It will give very small errors for the training data, but the resulting model may fail to generalize. A too small value for γ on the other hand, gives too much importance to the first (regularization) term in Eq. 13.10. The resulting function is flat and the LS-SVM is unable to learn anything. Several possibilities are available for the selection of appropriate values for these parameters. In this paper, we use an iterated gridsearch similar to the procedure used in [26].

13.3 Description and Preprocessing of the Data

For this study, data from three different sources was combined. Demographic information, for example literacy and infant mortality rate, was retrieved from the CIA Factbook [8] together with macro-economical variables, like GDP per capita and sectorial GDP information.

Information concerning the corruption level in specific countries was derived from Transparency International [28] under the form of the Corruption Perceptions Index (CPI). This index ranks countries according to the degree to which corruption is perceived to exist among public officials and politicians. The CPI is created by interviewing business people and country analysts and gives a score between 0 (highly corrupt) and 10 (highly clean) to each country. In this study, data concerning the years 1996, 2000 and 2004 was used. In the index of 1996, 54 countries received a corruption score. We select only these countries and omit from the more elaborated 2000 and 2004 indices all other countries, resulting in a total of 162 observations: three observations from different years for each of the 54 countries.² We use ISO 3166 codes, like BEL for Belgium or FRA for France, to refer to the individual countries whereby capitalization is used to indicate the year of the observation. Uppercase codes (e.g., BEL) indicate that the observation is from 2004, lowercase codes (e.g., bel) are used

² Pakistan and Bangladesh received a CPI rating in 1996 and 2004, but not in 2000. These two countries are not removed from the data set.

to refer to observations from the year 2000 and codes in proper case (only the first letter capitalized e.g., Bel) refer to 1996 observations.

Information about the democracy level in each country was obtained from Freedom House [11]. Each country is assigned a rating for political rights and a rating for civil liberties based on a scale of 1 to 7, with 1 representing the highest degree of freedom present and seven the lowest level of freedom. Similarly to the ‘level of corruption’, the ‘level of democracy’ in a country is a rather subjective concept and therefore difficult to express in only two indices. We refer to [4] for some critiques on the inclusion of these indices. An overview of all variables that were used in this study is given in Table 13.1.

13.4 SOM Analysis

13.4.1 Exploring the Data

We started by training a self-organizing map of 15 by 15 neurons: with this size it can be expected that each neuron will be the BMU for at most a few observations and this allows a clear visualization of the map. All available variables, including the CPI-scores, were used to create the map of Fig. 13.1. We can see that European countries are likely to be projected on the upper right corner of this map, while the other corners are dominated by respectively African, South-American and Asian countries. A second point that draws the attention is the fact that for most countries the observations

Table 13.1. Variables included in data set

Corruption Perceptions Index (CPI)
Civil Liberties (CL)
Political Rights (PR)
Arable Land (%)
Age Structure: % Population 0-14 years (Age 0-14)
Age Structure: % Population 15-64 years (Age 15-64)
Age Structure: % Population 65 years or over (Age > 65)
Population growth rate (PGR)
Birth rate (BR)
Death rate (DR)
Net migration rate (NMR)
Total Infant mortality rate (IMR)
Total Life Expectancy at birth (TLE)
Total Fertility rate (children born/women) (TFR)
GDP per capita
GDP agriculture
GDP industry
GDP services
Number of International Organisations the Country is Member Of (NIO)
Literacy (% of Total Population)

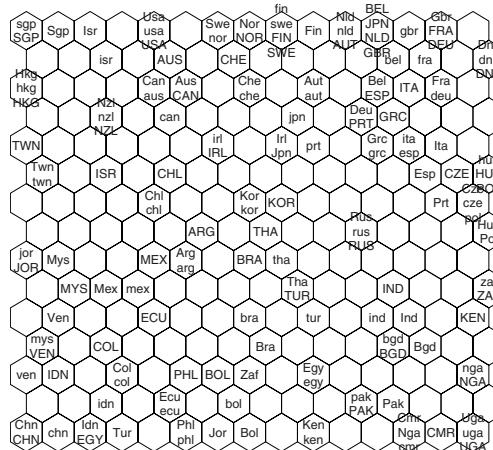


Fig. 13.1. Map of 15 by 15 Neurons

from the three years lie close to each other. The observations are either projected on the same neuron (e.g., Uganda (UGA), Denmark (DNK), Hong Kong (HKG)) or on adjacent neurons (e.g., Mexico (MEX), Australia (AUS)). The map also shows that for most of the countries for which the position changed over time, the capitalized 2004 observation lies closer to the upper right corner than the 1996 and 2000 observations. This is the case for Mexico (MEX), Brazil (BRA), Thailand (THA), Argentina (ARG), Chile (CHL) and Ecuador (ECU). It seems that these countries are in transition towards a “more European” model.

While the map of Fig. 13.1 provides general indications about the degree of similarity between countries, it does not allow us to obtain detailed information about corruption. To overcome this limitation, component planes were used to gain deeper insight in the data. Component planes can be created for each input variable and show the weights that connect each neuron with the particular input variable. The component plane for the Corruption Perceptions Index is shown in Fig. 13.2. In this figure, light and dark shades indicate respectively ‘non corrupt’ and ‘highly corrupt’ countries. We can observe that the lower right corner contains the countries perceived to be most corrupt (e.g., Pakistan (PAK), Nigeria (NIG), Cameroon (CMR) and Bangladesh (BGD)). At the opposite side, it can easily be noted that the North-European countries are perceived to be among the least corrupt: they are all situated in the white-colored region at the top of the map. Remember that most European countries were also projected on the upper-half of the map indicating a modest amount of corruption and that several countries seemed to be in transition towards a more European-less corrupt-model.

Component planes for other variables are shown in Fig. 13.3. The first component plane provides information about the literacy of the population. The dark spot indicates the countries where most of the population is illiterate. The second component plane shows Freedom House’s index of Political Rights. The light colored spots indicate the regions on the map with the countries that score low on ‘political freedom’. The resemblance between these two component planes and the component plane of

the corruption index is remarkable. There is a significant correlation between ‘corruption’, ‘literacy’ and ‘political freedom’. The third component plane (Fig. 13.3(c)) shows the number of international organizations that each country is member of. This component plane can be used to test the hypothesis that corrupt countries are less likely to be member of international organizations because they are either not welcome or not willing to participate. We can see that this hypothesis can not be confirmed based on the component plane. Countries in the corrupt region of the map do not differ from most European countries. Only some countries or regions close to the upper left corner (Hong Kong (HKG) and Taiwan (TWN)) seem to participate in fewer international organizations.

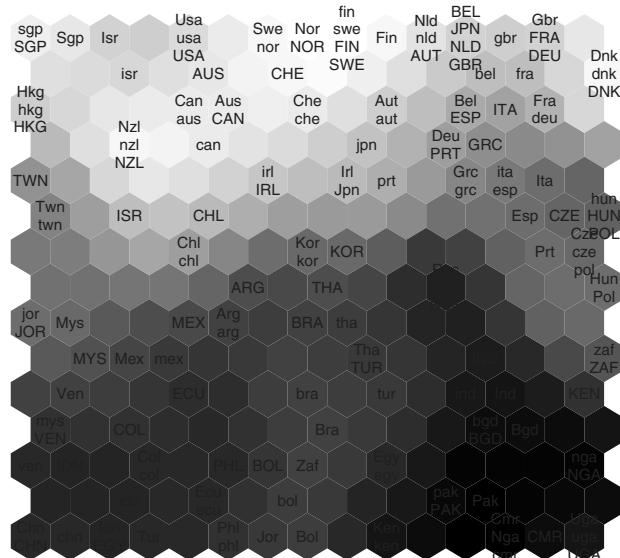


Fig. 13.2. Corruptions Perceptions Index

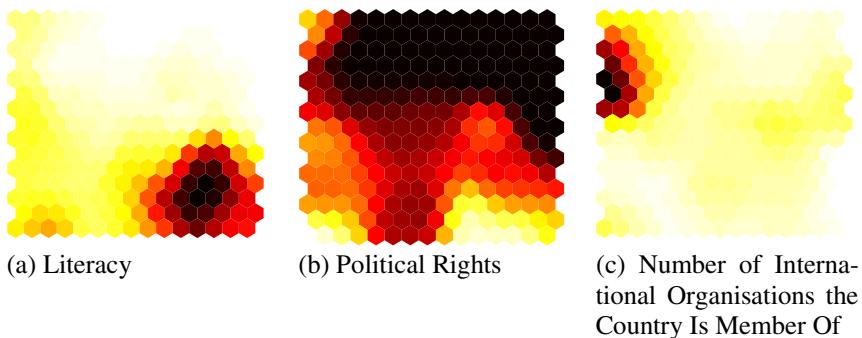


Fig. 13.3. Component Planes

The same kind of analysis can be performed for each of the other input variables. Several of these component planes, like ‘GDP per capita’ or ‘Total Life Expectancy’, show a high degree of correlation with the CPI component plane. Others, like ‘Birth Rate’ or ‘% GDP in agricultural sector’ are inversely correlated with the CPI component plane, indicating that in those countries corruption goes hand in hand with a young population and little industrialization.

13.4.2 Clustering of the SOM

In the preceding section, we have shown that SOMs are a suitable tool for the exploration of data sets. If the SOM grid itself consists of numerous neurons, analysis can be facilitated by clustering similar neurons into groups [32].

We performed the traditional k-means clustering algorithm on the trained map of 15 by 15 neurons. The result of this procedure, assuming 5 clusters are present in the data, is given in Fig. 13.4. Several unsupervised labelling techniques have been proposed to identify those variables that characterise a certain cluster (e.g., [2, 18, 23]). We briefly discuss the method presented in [2].

After performing a clustering of the trained map, the unlabelled training examples are projected again onto the map and it is remembered which observations are assigned to each cluster. Consequently, for each cluster the so-called salient dimensions are sought. These are defined as the variables that have significantly different values for the observations belonging and not-belonging to that cluster. Finally, a human domain specialist can manually interpret the salient dimensions of each cluster and assign a descriptive label.³

The above algorithm was executed on the clusters found by the k-means algorithm and the results are also shown in Fig. 13.4. In each cluster we show the variables that were found to be salient. A $>>$ ($<<$) sign behind the variable name indicates that countries within the cluster have on average significantly larger (smaller) values for this variable than countries outside the cluster. We can see that the analysis of the component planes can be largely automated by this procedure. For example, the procedure clearly indicates that countries from the cluster in the upper left corner seem to participate in few international organizations and score high on the CPI (low corruption). The cluster in the lower right corner on the other hand, is characterized by a low level of literacy, high fertility rate and a large percentage of GDP obtained from agriculture. A ‘high level of corruption’ is not preserved as a salient characteristic of this cluster because only the most salient features were included on this map and features like ‘literacy’ and ‘agricultural GDP’ were found to be more significant.

³ We have made one important change to the algorithm described in [2]. The original algorithm calculates difference factors based on the following formula $\frac{\mu_{in}(k,v) - \mu_{out}(k,v)}{\mu_{out}(k,v)}$ and uses these difference factors to find the salient dimensions (=dimensions with large or small difference factors). This has the disadvantage that for $\mu_{out}(k,v)$ close to zero the difference factor becomes very large even if the difference between $\mu_{out}(k,v)$ and $\mu_{in}(k,v)$ is small. To avoid this problem we add 1 in both numerator and denominator when calculating the difference factors.

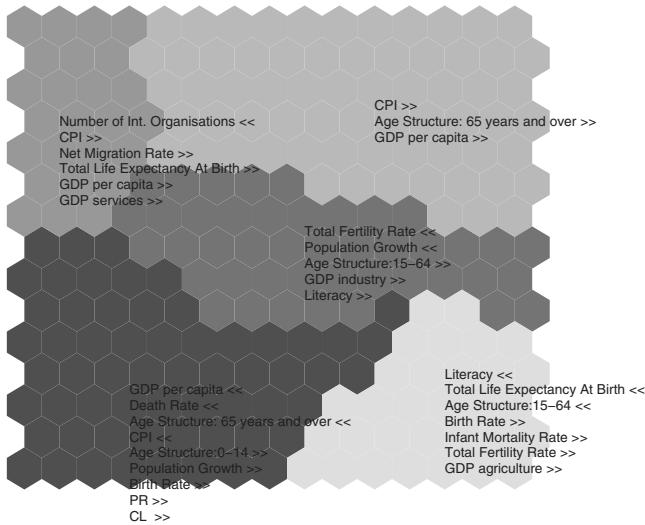


Fig. 13.4. Labelled Clusters

13.5 Regression Analysis

In this section several linear and nonlinear corruption forecasting models are created and evaluated. The models that are constructed can be used for different goals. First, the variables selected in these models provide indications about which variables are relevant when studying corruption. Additionally, for countries or regions that have not yet been assigned a CPI-score, the models can be adopted to obtain an estimate of the corruption in that country. For those countries that have already received a CPI-score, the differences between the forecasts of the models and the actual scores can prove useful for further investigation. Studying the reasons behind large deviations for specific countries provides more insight into the possible causes of corruption.

13.5.1 Absolute Level of Corruption

In this part of the paper, the information that is available in the data of 1996 is used to make forecasts of the corruption level in 2000 and 2004. Thus, we use the 54 observations of 1996 for training a predictive model and use this model to predict the corruption levels in 2000 and 2004. Additionally, the data sets of 2000 and 2004 are expanded by including those countries for which a 2000 or 2004 CPI value was available but a 1996 CPI value was not available. The result is a test data set consisting of 231 countries with 125 of them being observations from countries that are not present in the 1996 training data set. We will refer to these 125 observations as the ‘new’ (unseen) test data, while the other 106 observations are referred to as ‘old’ (unseen, but the same country is present in the training data of 1996).

First, a linear OLS regression model was constructed whereby input selection was performed on the variables of Table 13.1. Feature selection was performed with a

backwards procedure, but instead of using the traditional t-statistics to decide which variable to remove, we use the leave-one-out cross validation error on the training data set. The complete feature selection procedure occurs as follows:

1. Build a linear model containing all available inputs and measure the leave-one-out error.
2. Remove each of the variables one at a time and measure the leave-one-out error of the resulting model.
3. Select the model with the smallest leave-one-out error and go back to the previous step until a model is obtained with only one variable left.

Afterwards, one selects from all models created in step 3 the model with the smallest overall leave-one-out error. This model is then used to create predictions for the 2000 and 2004 test observations.

If the above procedure is performed on the corruption data sets, the model with 5 variables is selected as the best performing model. An overview of the model (with corresponding t-statistics between brackets) is given in Eq. (13.15):

$$\begin{aligned} \text{CPI} = & 3.005 + 1.066 * 10^{-4} \text{GDP per capita} - 0.88722 \text{ CL} - 0.015074 \text{ IMR} \\ & (2.4399) \quad (-3.3354) \quad (-1.5189) \\ & + 0.40462 \text{ PR} + 0.052143 \text{ GDP services} \\ & (1.9103) \quad (2.0301) \end{aligned} \quad (13.15)$$

Observe that the signs of the equation parameters correspond mostly to what one could expect based on common sense [24]. Increases in ‘GDP per capita’ and ‘GDP services’ increase the CPI score (corresponding to a decrease in corruption) and vice versa for the ‘Infant Mortality Rate’. The two variables approximating the democracy level in a country, Civil Liberties (CL) and Political Rights (PR), are also among the features selected by the backwards procedure. The negative sign for the CL-index indicates that an increase of the CL-index (less freedom) decreases the CPI-index (higher corruption). The positive sign of the PR-index seems counter-intuitive: an increase of the PR-index (less political rights) results in an increase of the CPI index (less corruption). This result might however be explained by research from Montinola and Jackman [22]. They found that the level of corruption is typically lower in dictatorships than in countries that have partially democratized, but once past a threshold, democratic practices inhibit corruption. If this non-linearity is indeed present, we should observe significant performance improvements when using a nonlinear LS-SVM model. This LS-SVM model is trained on the same five variables and suitable parameters for regularization and the RBF-kernel are selected by a gridsearch procedure as described in [26]. Additionally, a CART tree was also constructed on the same variables.

The constructed models are tested on the remaining data of 2000 and 2004. The results are shown in Table 13.2. It can be observed from the large R^2 values that all three models are able to explain most of the variance in the corruption index. The small values for the Mean Absolute Error (MAE) confirm this: on average the predictions differ only 0.78, 0.86 and 0.81 units from the actual observed values on the training data. The results on the test data are also shown in Table 13.2, where overall performance is indicated together with a breakdown by category. Whereas the MAE

is similar for ‘old’ and ‘new’ test data, there are huge differences in R^2 . The main reason for this deviation is due to the fact that the observations that are added to the CPI for the first time in 2000 or 2004 are on average more corrupt than the countries that were already present in the 1996 data set. Therefore, the ‘new’ test data have a smaller mean and variance of the CPI than observations from the ‘old’ test data, which causes the observed differences in the R^2 measure.

The three models are also compared with two naive forecasting methods. The first naive method provides always the same forecast, 5.34, which is the average CPI value of the training observations. The second naive model is slightly more advanced and provides its forecasts as the mean value of the training CPI’s for those countries that belong to the same region. We make a distinction between 7 different regions: Europe, Africa, Middle-East, Asia, North America, South America and Oceania. For example, from the 54 training countries there are 19 European countries with an average CPI of 7.13. The second naive model will use this value as its forecast for the European countries in the test data set and the same approach is adopted for each other region.

In Fig. 13.5, an overview of the performance of all models is provided. It shows the percentage of test observations for which the forecast error is smaller than the value specified on the X-axis. For example, we can observe that the LS-SVM model provides for approximately 55% of the observations a forecast that deviates less than 1 CPI unit from the actual value and for almost no observations the difference is larger than 3 units.

Just as for ROC-analysis, the performance of a model will be better if the corresponding line lies close to the upper-left corner. We can observe that all three models (LS-SVM, Linear and CART) perform significantly better than the naive approaches.

From Table 13.2 and Fig. 13.5, one can observe that the LS-SVM model provides a better forecasting accuracy than both other models. However, the LS-SVM has a serious disadvantage: it is very difficult to understand the motivation behind this model’s decisions due to its complexity. To relieve this opacity restriction and to gain more insight in the model’s behavior we will project its forecasts and the forecasting errors onto a self organizing map (Fig. 13.6).

Table 13.2. Overview Model Performance

	LS-SVM Model			Linear Model		CART Model	
	Mean	R^2	MAE	R^2	MAE	R^2	MAE
Training Data	5.35	0.84	0.78	0.81	0.86	0.82	0.81
Test Data	4.45	0.71	0.98	0.67	1.07	0.58	1.14
New	3.62	0.42	0.97	0.34	1.05	0.18	1.09
Old	5.43	0.75	0.98	0.73	1.09	0.66	1.19
Europe	6.03	0.64	1.17	0.60	1.24	0.42	1.52
Middle East	4.44	0.16	1.05	0.24	1.06	0.70	0.62
South America	3.70	0.15	1.12	0.10	1.13	-0.18	1.24
Asia	3.52	0.70	0.80	0.59	1.06	0.40	1.14
Africa	3.17	0.20	0.86	0.20	0.85	-0.02	0.90

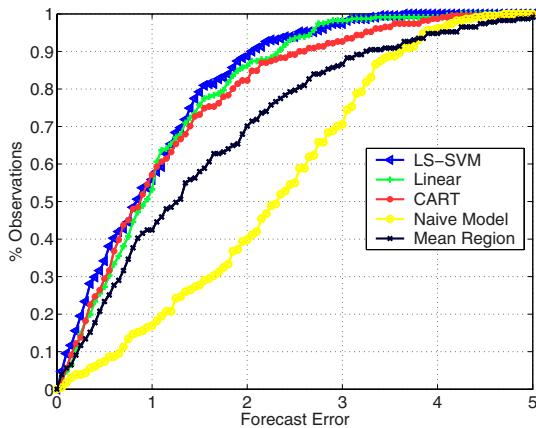


Fig. 13.5. Forecasting Errors

To create Fig. 13.6(a), a SOM is constructed from the 1996 training data, without use of the CPI, and on this trained map the observations from the test data are projected. Afterwards, each neuron of this map is assigned a color based on the average LS-SVM forecasts of the test observations projected onto that neuron. The neurons that were never the BMU are assigned the average color of the surrounding neurons. The same method was used to create Fig. 13.6(b), but with the colors based on the forecasting errors instead of the actual forecasts. From Fig. 13.6(a), it can be observed that observations projected on the upper half of the map are predicted to be among the least corrupt countries.

From Fig. 13.6(b), we learn that the model errors are evenly divided over the map. There are no regions where the LS-SVM model systematically over- or underestimates the perceived level of corruption. Both figures were also created for the linear model (not shown) and this allowed us to find out why this model was performing

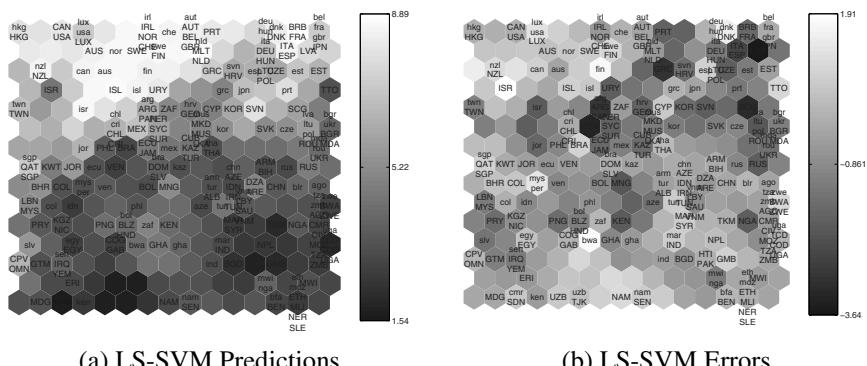


Fig. 13.6. Projection of Support Vector Machine onto Self Organizing Map

worse. The linear model's forecasts were very similar to the ones of the LS-SVM model, except for the lower right corner. In this region the linear model was systematically overestimating corruption, i.e. the actual corruption was less than predicted. Visual inspection also learned that both linear and LS-SVM model made similar forecasting errors for particular observations. For ARG (Argentina), GRC (Greece) and MEX (Mexico) the actual level of corruption is significantly higher than predicted by both models whereas the opposite is valid for bwa (Botswana) and fin (Finland). Further research is necessary to reveal the reasons for these deviations.

13.5.2 Relative Level of Corruption

Whereas in the previous section, we have shown that the constructed models are able to accurately predict the absolute level of corruption, we believe that it is also of major importance to understand why certain countries have a corruption level that is so drastically different from other countries in the same region. For example, why is Italy perceived to be more corrupt than its neighbors and what makes a country like Singapore stand out positively in the Asian region? To study these questions, we will create a second set of predictive models. The goal of these models is not to provide forecasts of the absolute CPI values, but to predict the CPI value relative to the other countries in the same region. For example, the mean CPI over all European training countries is 7.1. To let the predictive models learn the relative corruption level vis-a-vis the neighboring countries, we subtract this value from the original CPI scores of all European countries and similarly for each of the other regions discussed above. Countries that are more corrupt than the average of the region will therefore receive a negative score whereas the less corrupt countries will receive a positive score. An overview of the distribution of this relative CPI measure is shown in Fig. 13.7. The same procedure is also applied for each of the predictive variables in Table 13.1, i.e. the mean value for that variable within each region is subtracted from the observations to obtain relative values.

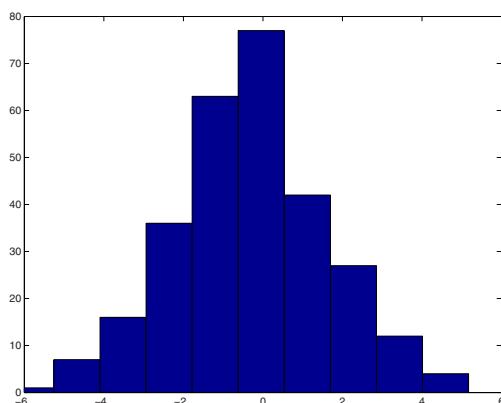


Fig. 13.7. Relative CPI Score

Afterwards, the modified data sets are used to learn predictive models. The procedure that was followed is similar to the approach discussed above. We first use backward linear input selection to decide which variables should be included in the models and afterwards apply OLS regression, LS-SVM regression and CART regression trees on the modified data sets.

The linear model that was obtained is shown in Eq. (13.16) (corresponding t-statistics are given between brackets; all variables, except NIO, are significant at 1% level). Remember that all variables in this equation represent relative values versus the mean of the region. The coefficients in this equation should therefore be interpreted as follows: If the GDP per capita of a country is 1 USD larger relative to the other countries in the same region, then we can expect the CPI to be 0.0002 units larger than the mean CPI in that region.

$$\begin{aligned} \text{CPI} = & 0.0002 \text{ GDP per capita}-0.4758 \text{ CL}-0.6430 \text{ Age} > 65 \\ & (4.22) \quad (-3.02) \quad (-4.00) \\ & + 0.3853 \text{ TLE} + 0.6219 \text{ DR} + 0.0337 \text{ NIO} \\ & (3.48) \quad (3.23) \quad (1.26) \end{aligned} \quad (13.16)$$

Other variables included in the equation are the Civil Liberties index (CL), the proportion of the population older than 65 years, Total Life Expectancy (TLE), Death Rate (DR) and the Number of International Organisations that the country is member of (NIO). We did not include dummy variables representing the regions in the final estimated model as the weights associated with these dummies were found to be very close to zero and therefore not significant. The signs of the other parameters in the model correspond mostly, but not always, to what we expected based on common sense. For example, we had expected a positive sign for the 'Age > 65'-variable, as we can imagine that countries with an older population, which might be considered a sign of wealth and/or good health policy, to be less corrupt. Multicollinearity between the explicatory variables seems a possible explanation for the counter-intuitive signs as some of the variables are highly correlated, e.g. GDP per capita, Age > 65 and TLE.

We also tested the predictive performance of all three models on the independent data set. The results are given in Fig. 13.8 and Table 13.3. Although, a large discrepancy between training and test set performance can be remarked, we do not believe that it is due to overfitting because the difference can also be observed for the rather

Table 13.3. Overview Model Performance (Relative CPI)

		LS-SVM Model		Linear Model		CART Model	
	Mean	R ²	MAE	R ²	MAE	R ²	MAE
Training Data	0.0	0.78	0.59	0.71	0.74	0.78	0.64
Test Data	-0.37	0.35	1.18	0.48	1.10	0.26	1.27
New	-0.72	0.32	1.19	0.49	1.08	0.26	1.30
Old	0.05	0.33	1.18	0.41	1.12	0.19	1.23

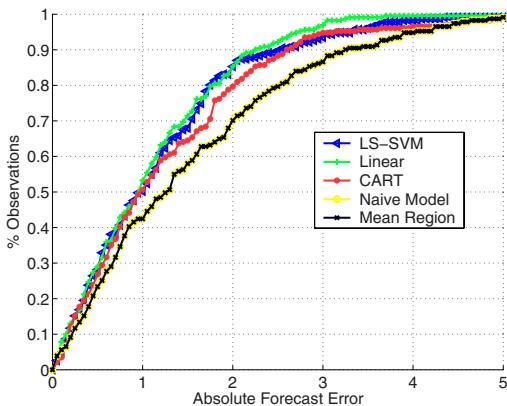


Fig. 13.8. Forecasting Errors

simple linear model. As explained above, the difference in forecasting performance is mainly caused by the differences in the characteristics of the countries included in the training and test data sets. Whereas the models are trained on a data set that consists mostly of industrialized countries with little corruption, the test data set contains a decent amount of countries with a higher level of corruption. Nevertheless, although performance improvements should be possible by resampling the training data as to increase the number of countries with a higher level of corruption, the models already perform better than the naive approaches. This can be observed in Fig. 13.8, where the line that corresponds with the naive approach of always predicting a constant value, the training target's average, always lies below the constructed models. The mean absolute error of this naive approach on the test observations equals 1.53.

13.6 Conclusion

In this chapter, we presented some techniques that can be adopted to study corruption and its causes. Whereas the amount of literature on corruption is impressive, the use of data mining techniques to study corruption is not so widespread. In the first part of this chapter, the powerful visualization possibilities of self organizing maps were used to study the interconnections between various macro-economical variables and the perceived level of corruption. The use of multi-year data sets allows the visualization of the evolution of corruption over time for specific countries. In the second part of this chapter, it was shown that forecasting models can be constructed that allow analysts to predict the level of corruption for countries where this information is missing. Besides forecasting the actual level of corruption, we also trained models to study the corruption in countries relative to the corruption in the region. Finally, it was shown how self organizing maps can be used to study the behavior of these supervised models. This allows us to open the ‘black box’ and can provide a more detailed comparison between the predictions made by the different models.

References

1. Alesina, A., Weder, B.: Do corrupt governments receive less foreign aid? National Bureau of Economic Research Working Paper 7108 (1999)
2. Azcarraga, A., Hsieh, M., Pan, S., Setiono, R.: Extracting salient dimensions for automatic SOM labeling. *Transactions on Systems, Management and Cybernetics, Part C* 35(4), 595–600 (2005)
3. Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J.: Benchmarking state of the art classification algorithms for credit scoring. *Journal of the Operational Research Society* 54(6), 627–635 (2003)
4. Bohara, A., Mitchell, N., Mittendorff, C.: Compound democracy and the control of corruption: A cross-country investigation. *The Policy Studies Journal* 32(4), 481–499 (2004)
5. Breiman, L., Friedman, J.H., Olsen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth and Brooks (1984)
6. Brockett, P.L., Xia, X., Derrig, R.: Using Kohonen's self-organizing feature map to uncover automobile bodily injury claims fraud. *International Journal of Risk and Insurance* 65, 245–274 (1998)
7. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2(2), 121–167 (1998)
8. CIA Factbook, retrieved from
<http://www.cia.gov/cia/publications/factbook/>
9. De Brabanter, J.: LS-SVM Regression Modelling and its Applications. Ph.D. thesis, Katholieke Universiteit Leuven, Faculty of Engineering (2004)
10. Deboeck, G., Kohonen, T.: Visual Explorations in Finance with Self Organizing maps. Springer, Heidelberg (1998)
11. Freedom House, Freedom in the world country ratings (2005), retrieved from
<http://www.freedomhouse.org>
12. Gerring, J., Thacker, S.: Political institutions and corruption: The role of unitarism and parliamentarism. *The British Journal of Political Science* 34, 295–330 (2004)
13. Honkela, T., Kaski, S., Lagus, K., Kohonen, T.: WEBSOM-Self-Organizing Maps of Document Collections. In: Proceedings of WSOM 1997, Workshop on Self-Organizing Maps, Espoo, Finland, Helsinki University of Technology, pp. 310–315 (1997)
14. Karalic, A.: Linear regression in regression tree leaves. In: ISSEK 1992 (International School for Synthesis of Expert Knowledge), pp. 151–163 (1992)
15. Kohavi, R., Quinlan, J.R.: Decision-tree discovery. In: Klosgen, W., Zytkow, J. (eds.) *Handbook of Data Mining and Knowledge Discovery*, pp. 267–276. Oxford University Press, Oxford (2002)
16. Kohonen, T.: Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43, 59–69 (1982)
17. Kohonen, T.: *Self-Organising Maps*. Springer, Heidelberg (1995)
18. Lagus, K., Kaski, S.: Keyword selection method for characterizing text document maps. In: Proceedings of ICANN 1999, Ninth International Conference on Artificial Neural Networks, pp. 371–376 (1999)
19. Lambsdorff, J.: Corruption in empirical research: a review. Transparency International Working paper (1999)
20. Leite, C., Weidmann, J.: Does mother nature corrupt? natural resources, corruption and economical growth. International Monetary Fund Working Paper 99/85 (1999)
21. Mauro, P.: Corruption and growth. *The Quarterly Journal of Economics* 110(3), 681–712 (1995)

22. Montinola, G., Jackman, R.: Sources of corruption: a cross-country study. *British Journal of Political Science* 32, 147–170 (2002)
23. Rauber, A., Merkl, D.: Automatic labeling of self-organizing maps: Making a treasure-map reveal its secrets. In: Zhong, N., Zhou, L. (eds.) PAKDD 1999. LNCS (LNAI), vol. 1574, pp. 228–237. Springer, Heidelberg (1999)
24. Seldadyo, H., de Haan, J.: The determinants of corruption: a literature survey and new evidence. In: EPCS 2006 Conference (2006)
25. Smola, A., Schölkopf, B.: A tutorial on support vector regression. *Statistics and Computing* 14, 199–222 (2004)
26. Suykens, J., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific, Singapore (2002)
27. Swamy, A., Knack, S., Lee, Y., Azfar, O.: Gender and corruption. *Journal of Development Economics* 64, 25–55 (2001)
28. Transparency International. Corruption Perceptions Index, retrieved from <http://www.transparency.org/>
29. Treisman, D.: The causes of corruption: a cross-national study. *Journal of Public Economics* 76(3), 339–457 (2000)
30. Vapnik, V.N.: The nature of statistical learning theory. Springer, New York (1995)
31. Vesanto, J.: Som-based data visualization methods. *Intelligent Data Analysis* 3, 111–126 (1999)
32. Vesanto, J., Alhoniemi, E.: Clustering of the self-organizing map. *IEEE Transactions on Neural Networks* 11(3), 586–600 (2000)

Online Resources

1. Matlab SOM Toolbox:
<http://www.cis.hut.fi/projects/somtoolbox/>
2. LS-SVMLab: a Matlab toolbox.
<http://www.esat.kuleuven.ac.be/sista/lssvmlab/>
3. UCI Machine Learning Repository: a collection of data sets to benchmark machine learning algorithms.
<http://www.ics.uci.edu/~mlearn/MLRepository.html>
4. WEBSOM: visualization of text collections.
<http://websom.hut.fi/websom/>
5. World Poverty Map:
<http://www.cis.hut.fi/research/som-research/worldmap.html>
6. DemoGNG: A Java applet to visualize various methods of competitive learning.
<http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/choice.html>
7. LIBSVM: A Library for Support Vector Machines.
<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

Questions for Discussions

1. One of the main problems of techniques like neural networks and support vector machines is the opaqueness of these models, i.e. it is very difficult to understand how they arrive at a certain decision. In this chapter, we used self organizing maps to make these models better comprehensible. Do you know other techniques that can provide more insight in such models?

2. One of the problems that were encountered is that the properties of the data on which the models were trained deviate from the data on which the performance was tested. How could this problem be alleviated?
3. Which other variables can you think of that might be relevant for predicting corruption. Do you expect a causal relation or a correlation?
4. How do you expect that the component planes of ‘GDP per capita’ and ‘Infant Mortality Rate’ will look like (base yourself on Figs. 13.1–3)

Protecting Private Information in Online Social Networks

Jianming He and Wesley W. Chu

Computer Science Department,
University of California, USA
`{jmhek, wwc}@cs.ucla.edu`

Abstract. Because personal information can be inferred from associations with friends, privacy becomes increasingly important as online social network services gain more popularity. Our recent study showed that the causal relations among friends in social networks can be modeled by a Bayesian network, and personal attribute values can be inferred with high accuracy from close friends in the social network. Based on these insights, we propose schemes to protect private information by selectively hiding or falsifying information based on the characteristics of the social network. Both simulation results and analytical studies reveal that selective alterations of the social network (relations and/or attribute values) according to our proposed protection rule are much more effective than random alterations.

14.1 Introduction

With the increasing popularity of Web 2.0, more and more online social networks (OSNs) such as Myspace.com, Facebook.com, and Friendster.com have emerged. People in OSNs have their own personalized space where they not only publish their biographies, hobbies, interests, blogs, etc., but also list their friends. Friends or visitors can visit these personal spaces and leave comments. OSNs provide platforms where people can place themselves on exhibit and maintain connections with friends, and that is why they are so popular with the younger generation. However, as more people use OSNs, privacy becomes an important issue. When considering the multitude of user profiles and friendships flooding the OSNs (e.g., Myspace.com claims to have about 100 million membership accounts), we realize how easily information can be divulged if people mishandle it [8]. One example is a school policy violation identified on Facebook.com. In November 2005, four students at Northern Kentucky University were fined when pictures of a drinking party were posted on Facebook.com. The pictures, taken in one of NKU's dormitories, were visual proof that the students were in violation of the university's dry campus policy. In this example, people's private activities were disclosed by themselves.

There is another type of privacy disclosure that is more difficult to identify and prevent. In this case, private data can be indirectly inferred by adversaries. Intuitively, friends tend to share common traits. For example, high school classmates have similar ages and the same hometown, and members of a dance club like dancing. Therefore, to infer someone's hometown or interest in dancing, we can check the values of these

attributes of his classmates or club mates. In another example, assume Joe does not wish to disclose his salary. However, a third party, such as an insurance company, uses OSNs to obtain a report on Joe's social network, which includes Joe's friends and office colleagues and their personal information. After looking carefully into this report, the insurance company realizes that Joe has quite a few friends who are junior web developers of a startup company in San Jose. Thus, the insurance company can deduce that most likely Joe is also a programmer (if this information is not provided by Joe himself). By using the knowledge concerning a junior programmer's salary range, the insurance company can then figure out Joe's approximate salary and advertise insurance packages accordingly. Therefore, in this example, Joe's private salary information is indirectly disclosed from Joe's social relations.

Information privacy is one of the most urgent research issues in building next-generation information systems, and a great deal of research effort has been devoted to protecting people's privacy. In addition to recent developments in cryptography and security protocols [1, 2] that provide secure data transfer capabilities, there has been work on enforcing industry standards (e.g., P3P [21]) and government policies (e.g., the HIPAA Privacy Rule [19]) to grant individuals control over their own privacy. These existing techniques and policies aim to effectively block direct disclosure of sensitive personal information. However, as we mentioned in the previous examples, private information can also be indirectly deduced by intelligently combining pieces of seemingly innocuous or unrelated information. To the best of our knowledge, none of the existing techniques are able to handle such indirect disclosures.

In this chapter we shall discuss how to protect the disclosure of private information that can be inferred from social relations. To preserve the inference properties from the social network characteristics, we encode the causality of a social network into a Bayesian network, and then use simulation and analysis to investigate the effectiveness of inference on private information in a social network. We have conducted an experiment on the Epinions.com that operates in a real environment to verify the performance improvements gained by using the Bayesian network for inferring private information. Based on the insights obtained from the experiment, a privacy protection rule has been developed. Privacy protection methods derived from the protection rule are proposed, and their performance is evaluated.

The chapter is organized as follows. After introducing the background in Sect. 14.2, we propose a Bayesian network approach in Sect. 14.3 to infer private information. Sect. 14.4 discusses simulation experiments for studying the performance of Bayesian inference. Privacy protection rules, as well as protection schemes, are proposed and evaluated in Sect. 14.5. In Sect. 14.6 we use analysis to show that based on our protection rules, selective alterations of the social network (social relations and/or attribute values) yield much more effective privacy protection than the random alterations. We present some related work on social networks in Sect. 14.7. Finally, future work and conclusions are summarized in Sect. 14.8. Sect. 14.8 is followed by several questions related to the discussion in this chapter.

14.2 Background

A Bayesian network [9, 10, 7, 22] is a graphic representation of the joint probability distribution over a set of variables. It consists of a network structure and a collection

of conditional probability tables (CPT). The network structure is represented as a directed acyclic graph (DAG) in which each node corresponds to a random variable and each edge indicates a dependent relationship between connected variables. In addition, each variable (node) in a Bayesian network is associated with a CPT, which enumerates the conditional probabilities for this variable given all the combinations of its parents' value. Thus, for a Bayesian network, the DAG captures causal relations among random variables, and CPTs quantify these relations.

Bayesian networks have been extensively applied to fields such as medicine, image processing, and decision support systems. Since Bayesian networks include the consideration of network structures, we decided to model social networks with Bayesian networks. Basically, we represent an individual in a social network as a node in a Bayesian network and a relation between individuals in a social network as an edge in a Bayesian network.

14.3 Bayesian Inference Via Social Relations

In this section we propose an approach to map social networks into Bayesian networks, and then illustrate how we use this for attribute inference. The attribute inference is used to predict the private attribute value of a particular individual, referred to as the target node Z , from his social network which consists of the values of the same attribute of his friends. Note that we do not utilize the values of other attributes of Z and Z 's friends in this study, though considering such information might improve the prediction accuracy. Instead, we only consider a single attribute so that we can focus on the role of social relations in the attribute inference. The single attribute that we study can be any attribute in general, such as gender, ethnicity, and hobbies, and we refer to this attribute as the target attribute. For simplicity, we consider the value of the target attribute as a binary variable, i.e., either true (or t for short) or false (f). For example, if Z likes reading books, then we consider Z 's book attribute value is true.

People are acquainted with each other via different types of relations, and it is not necessary for an individual to have the same attribute values as his friends. Which attributes are common between friends depends on the type of relationship. For example, diabetes could be an inherited trait in family members but this would not apply to officemates. Therefore, to perform attribute inference, we need to filter out the non-related social relations. For instance, we need to remove Z 's officemates from his social network if we want to predict his health condition. If the types of social relations that cause friends to connect with one another are specified in the social networks, then the filtering is straightforward. However, in case such information is not given, one possible solution is to classify social relations into different categories, and then filter out non-related social relations based on the type of the categories. In Sect. 14.4, we show such an example while inferring personal interests from data in Epinions.com. For simplicity, in this section we assume that we have already filtered out the non-related social relations, and the social relations we discussed here are the ones that are closely related to the target attribute.

The attribute inference involves two steps. Before we predict the target attribute value of Z , we first construct a Bayesian network from Z 's social network, and then apply a Bayesian inference and obtain the probability that Z has a certain attribute

value. In this section we shall first start with a simple case in which the target attribute values of all the direct friends are known. Then, we extend the study by considering the case where some friends hide their target attribute values.

14.3.1 Single-Hop Inference

Let us first consider the case in which we know the target attribute values of all the direct friends of Z . We define Z_{ij} as the j^{th} friend of Z at i hops away. If a friend can be reached via more than one route from Z , we use the shortest path as the value of i . Therefore, Z can also be represented as Z_{00} . Let Z_i be the set of Z_{ij} ($0 \leq j < n_i$), where n_i is the number of Z 's friends at i hops away. For instance, $Z_1 = \{Z_{10}, Z_{11}, \dots, Z_{1(n_1-1)}\}$ is the set of Z 's direct friends who are one hop away. Furthermore, we use the corresponding lowercase variable to represent the target attribute value of a particular person, e.g., z_{10} stands for the target attribute value of Z_{10} .

An example of a social network with six friends is shown in Fig. 14.1(a). In this figure, Z_{10} , Z_{11} and Z_{12} are direct friends of Z . Z_{20} and Z_{30} are the direct friends of Z_{11} and Z_{12} respectively. In this scenario, the attribute values of Z_{10} , Z_{11} , Z_{12} and Z_{30} are known (represented as shaded nodes).

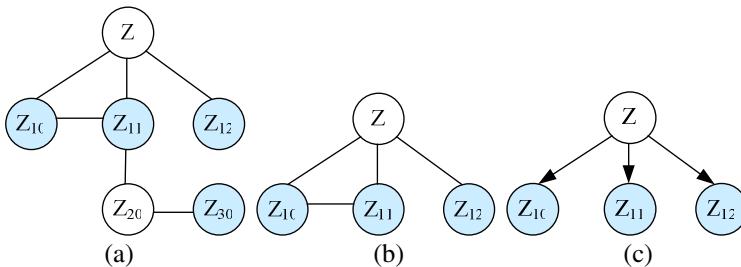


Fig. 14.1. Reduction of a social network (a) into a Bayesian network to infer Z from his friends via localization assumption (b), and via naïve Bayesian assumption (c). The shaded nodes represent friends whose attribute values are known.

Bayesian Network Construction

To construct the Bayesian network, we make the following two assumptions.

Intuitively, our direct friends have more influence on us than friends who are two or more hops away. Therefore, to infer the target attribute value of Z , it is sufficient to consider only the direct friends of Z_1 . Knowing the attribute values of friends at multiple hops away provides no additional information for predicting the target attribute value. Formally, we state this assumption as follows.

Localization Assumption

Given the attribute values of the direct friends Z_1 , friends at more than one hop away (i.e., Z_i for $i > 1$) are conditionally independent of Z .

Based on this assumption, Z_{20} and Z_{30} in Fig. 14.1(a) can be pruned, and the inference of Z only involves Z_{10} , Z_{11} and Z_{12} (Fig. 14.1(b)). Then the next question is how

to decide a DAG linking the remaining nodes. If the resulting social network does not contain cycles, a Bayesian network is formed. Otherwise, we must employ more sophisticated techniques to remove cycles, such as the use of auxiliary variables to capture non-causal constraints (exact conversion) and the deletion of edges with the weakest relations (approximation conversion). We adopt the latter approach and make a naive Bayesian assumption. That is, the attribute value of Z influences that of Z_{lj} ($0 \leq j < n_l$), and there is a direct link pointing from Z to each Z_{lj} . By making this assumption, we consider the inference paths from Z to Z_{lj} as the primary correlations, and disregard the correlations among the nodes in Z_l . Formally, we have:

Naïve Bayesian Assumption

Given the attribute value of the target node Z , the attribute values of direct friends Z_l are conditionally independent of each other.

This naïve Bayesian model has been used in many classification/prediction applications including textual-document classification. Though it simplifies the correlation among variables, this model has been shown to be quite effective [14]. Thus, we also adopted this assumption in our study. For example, a final DAG is formed as shown in Fig. 14.1(c) by removing the connection between Z_{l0} and Z_{l1} in Fig. 14.1(b).

Bayesian Inference

After modeling the specific person Z 's social network into a Bayesian network, we use the Bayes decision rule to predict the attribute value of Z . For a general Bayesian network with maximum depth i , let \bar{Z} be the maximum conditional (posterior) probability for the attribute value of Z given the attribute values of other nodes in the network, as in Eq. 14.1:

$$\bar{Z} = \arg \max_z P(Z | Z_1, Z_2, \dots, Z_i) \quad Z \in \{t, f\}. \quad (14.1)$$

Since single-hop inference involves only direct friends Z_l which are independent of each other, the posterior probability can be further reduced using the conditional independence property encoded in the Bayesian network:

$$\begin{aligned} P(Z | Z_l) &= \frac{P(Z_l | Z=z) \cdot P(Z=z)}{\sum_z [P(Z_l | Z=z) \cdot P(Z=z)]} \\ &= \frac{P(Z=z) \prod_{j=0}^{n_l-1} P(Z_{lj} = z_{lj} | Z=z)}{\sum_z [P(Z=z) \prod_{j=0}^{n_l-1} P(Z_{lj} = z_{lj} | Z=z)]}, \end{aligned} \quad (14.2)$$

where z and z_{lj} are the attribute values of Z and Z_{lj} respectively ($0 \leq j < n_l$, $z, z_{lj} \in \{t, f\}$) and the value of each z_{lj} is known.

To compute Eq. 14.2, we need to further learn the conditional probability table (CPT) for each person in the social network. In our study we apply the parameter

estimation [7] technique on the entire network. For every pair of parent X and child Y , we obtain Eq. 14.3:

$$P(Y = y | X = x) = \frac{\# \text{ of friendship links connecting people with } X = x \text{ and } Y = y}{\# \text{ of friendship links connecting a person with } X = x}, \quad (14.3)$$

where $x, y \in \{t, f\}$. $P(Y = y | X = x)$ is the CPT for every pair of friends Z_{lj} and Z in the network. Since $P(Z_{lj} | Z)$ is the same for $0 \leq j < n_l$, Z_{lj} becomes equivalent to one another, and the posterior probability now depends on N_{lt} , the number of direct friends with attribute value t . We can rewrite the posterior probability $P(Z = z | Z_l)$ as $P(Z = z | N_{lt} = n_{lt})$. Given $N_{lt} = n_{lt}$, we obtain:

$$P(Z = z | N_{lt} = n_{lt}) = \frac{P(Z = z) \cdot P(Z_{10} = t | Z = z)^{n_{lt}} \cdot P(Z_{10} = f | Z = z)^{n_l - n_{lt}}}{\sum_z [P(Z = z) \cdot P(Z_{10} = t | Z = z)^{n_{lt}} \cdot P(Z_{10} = f | Z = z)^{n_l - n_{lt}}]}. \quad (14.4)$$

where $z \in \{t, f\}$.

After obtaining $P(Z = t | N_{lt} = n_{lt})$ and $P(Z = f | N_{lt} = n_{lt})$ from Eq. 14.4, we predict Z has attribute value t if the former value is greater than the latter value, and vice versa.

14.3.2 Multi-hop Inference

In single-hop inference, we assume that we know the attribute values of all the direct friends of Z . However, in reality, not all of those attribute values may be observed since people may hide their sensitive information, and the localization assumption in the previous section is no longer valid. To incorporate more attribute information into our Bayesian network, we propose the following generalized localization assumption.

Generalized Localization Assumption

Given the attribute value of the j^{th} friend of Z at i hops away, Z_{lj} ($0 \leq j < n_l$), the attribute of Z is conditionally independent of the descendants of Z_{lj} .

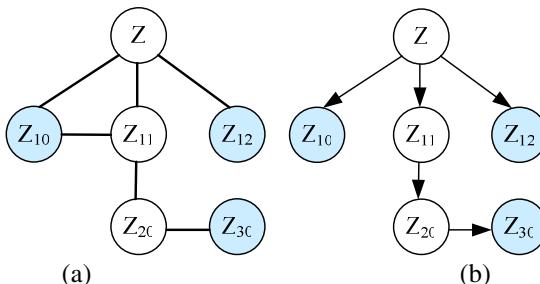


Fig. 14.2. Reduction of a social network (a) into a Bayesian network to infer Z from his friends via generalized localization assumption (b). The shaded nodes represent friends whose attribute values are known.

This assumption states that if the attribute value of Z 's direct friend Z_{lj} is unknown, then the attribute value of Z is conditionally dependent on those of the direct friends of Z_{lj} . This process continues until we reach a descendent of Z_{lj} with known attribute value. For example, the network structure in Fig. 14.2(a) is the same as in Fig. 14.1(a), but the attribute value of Z_{11} is unknown. Based on the generalized localization assumption, we extend the network by branching to Z_{11} 's direct child Z_{20} . Since Z_{20} 's attribute value is unknown, we further branch to Z_{20} 's direct friend Z_{30} . The branch terminates here because the attribute value of Z_{30} is known. Thus, the inference network for Z includes all the nodes in the graph. After applying the naive Bayesian assumption, we obtain the DAG shown in Fig. 14.2(b). Similar to single-hop inference, the resulting DAG in multi-hop inference is a tree rooted at the target node Z . One interpretation of this model is that when we predict the attribute value of Z , we always treat Z as an egocentric person who has strong influences on his/her friends. Thus, the attribute value of Z can be reflected by the attributes of friends.

For multi-hop inference, we still apply the Bayes decision rule. Due to additional unknown attribute values such as Z_{11} , the calculation of the posterior probability becomes more complicated. One common technique for solving this equation is variable elimination [19]. In this chapter, we use this technique to derive the value of \bar{Z} in Eq. 14.1.

14.4 Experimental Study of Bayesian Inference

In the previous section we discussed the method for performing the attribute inference in social networks. In this section we study several characteristics of social networks to investigate under what condition and to what extent the value of a target attribute can be inferred by Bayesian inference. Specifically, we study the influence strength between friendship, prior probability of target attributes, and society openness. We use simulations and experiments to evaluate their impact on inference accuracy, which is defined as the percentage of nodes predicted correctly by the inference.

14.4.1 Characteristics of Social Networks

Influence Strength

Analogous to the interaction between inheritance and mutation in biology, we define two types of influence in social relations. More specifically, for the relationship between every pair of parent X and child Y , we define $P(Y = t | X = t)$ (or P_{nt} for simplification) as inheritance strength. This value measures the degree to which a child inherits an attribute value from his/her parent. A higher value of P_{nt} implies that both X and Y will possess the attribute value with a higher probability. On the other hand, we define $P(Y = t | X = f)$ (or P_{tf}) as mutation strength. P_{tf} measures the potential that Y develops an attribute value by mutation rather than inheritance. An individual's attribute value is the result of both types of strength.

There are two other conditional probabilities between X and Y ; i.e., $P(Y = f | X = t)$ (or P_{ft}) and $P(Y = f | X = f)$ (or P_{ff}). These two values can be derived from P_{nt} and P_{tf} .

respectively ($P_{flt} = 1 - P_{nlr}$ and $P_{flf} = 1 - P_{nlf}$). Therefore, it is sufficient to only consider inheritance and mutation strength.

Prior Probability

Prior probability $P(Z = t)$ (or P_t for short) is the percentage of people in the social network who have the target attribute value as t . When no additional information is provided, we can use prior probability to predict attribute values for the target nodes: if $P_t \geq 0.5$, we predict that every target node has value t ; otherwise, we predict that it has value f . We call this method *naive inference*. The average naive inference accuracy that can be obtained is $\max(P_b, 1 - P_t)$. In our study, we use it as a base line for comparison with the Bayesian inference approach.

It is worth pointing out that when $P_{nlr} = P_t$, people in a society are in fact independent of each other, thus $P_{flr} = P_t$. Hence, having additional information about a friend provides no contribution to the prediction for the target node.

Society Openness

We define society openness O_A as the percentage of people in a social network who release their target attribute value A . The more people who release their values, the higher the society openness, and the more information observed about attribute A . Using society openness, we study the amount of information needed to know about other people in the social network in order to make a correct prediction.

14.4.2 Data Set

For the simulation, we collect 66,766 personal profiles from an online weblog service provider, Livejournal [12], which has 2.6 million active members all over the world. For each member, Livejournal generates a personal profile that specifies the member's biography as well as a list of his friends. Among the collected profiles, there are 4,031,348 friend relationships. The degree of the number of friends follows the power law distribution (Fig. 14.3). About half of the population has less than ten direct friends.

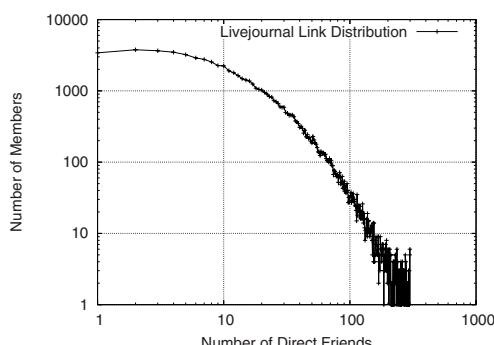


Fig. 14.3. Number of direct friends vs. number of members in Livejournal on a log-log scale

In order to evaluate the inference behaviors for a wide range of parameters, we use a hypothetical attribute and synthesize the attribute values. For each member, we assign a CPT and determine the actual attribute value based on the parent's value and the assigned CPT. The attribute assignment starts from the set of nodes whose in-degree is zero and explores the rest of the network following friendship links. We use the same CPT for each member. For all the experiments, we evaluate the inference performance by varying CPTs.

After the attribute assignment, we obtain a social network. To infer each individual, we build a corresponding Bayesian network and then conduct Bayesian inference as described in Sect. 14.3.

14.4.3 Simulation Results

Comparison of Bayesian and Naïve Inference

In this set of experiments, we compare the performance of Bayesian inference to naïve inference. We shall study whether privacy can be inferred from social relations. We fix the prior probability P_t to 0.3 and vary inheritance strength P_{tl} from 0.1 to 0.9.¹ We perform inference using both approaches on every member in the network. The inference accuracy is obtained by comparing the predicted values with the

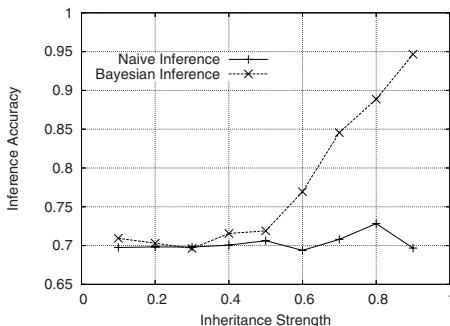


Fig. 14.4. Inference accuracy of Bayesian vs. naïve inference when $P_t = 0.3$

corresponding actual values. Fig. 14.4 shows the inference accuracy of the two methods as the inheritance strength, P_{tl} , increases. It is clear that Bayesian inference outperforms naïve inference. The curve for naïve inference fluctuates around 70%, because with $P_t = 0.3$, the average accuracy we can achieve is 70%. The performance of Bayesian inference varies with P_{tl} . We achieve a very high accuracy, especially at high inheritance strength. The accuracy reaches 95% when $P_{tl} = 0.9$, which is much higher than the 70% accuracy of the naïve inference. Similar trends are observed between these two methods for other prior probabilities as well.

¹ In an equilibrium state, the value of P_{tlf} can be derived from P_t and P_{tl} . When P_t is fixed, increasing P_{tl} results in a decrease in P_{tlf} .

Effect of Influence Strength and Prior Probability

Fig. 14.5 shows the accuracy of Bayesian inference when the prior probability P_t is 0.05, 0.1, 0.3 and 0.5, and the inheritance strength P_{tl} varies from 0.1 to 0.9. As P_t varies, the inference accuracy yields different trends with P_{tl} . The lowest inference accuracy always occurs when P_{tl} is equal to P_t . For example, the lowest inference accuracy (approximately 70%) at $P_t = 0.3$ occurs when P_{tl} is 0.3. At this point, people in the network are independent of each other. The inference accuracy increases as the difference between P_{tl} and P_t increases.

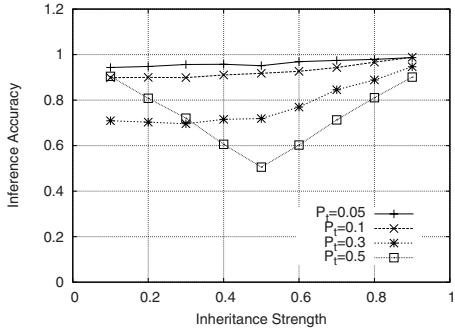


Fig. 14.5. Inference accuracy of Bayesian inference for different prior probabilities

Society Openness

In the previous experiments, we assumed that society openness is 100%. That is, the attribute values of all the friends of the target node are known. In this set of experiments, we study the inference behavior at different levels of society openness. We randomly hide the attribute values of a certain percentage of members, ranging from 10% to 90%, and then perform Bayesian inference on those members.

Fig. 14.6 shows the experimental results for the prior probability $P_t = 0.3$ and the society openness $O_A = 10\%, 50\%$ and 90% . The inference accuracy decreases as the openness decreases (i.e., the number of members hiding their attribute values increases). For instance, at inheritance strength 0.7, when the openness is decreased from 90% to 10%, the accuracy reduces from 84.6% to 81.5%. However, the reduction in inference accuracy is relatively small (on average less than 5%). We also observe similar trends for other prior probabilities. This phenomenon reveals that randomly hiding friends' attribute values only results in a small effect on the inference accuracy. Therefore, we should consider selectively altering social networks to improve privacy protection.

Robustness of Bayesian Inference on False Information

To evaluate the robustness of Bayesian inference when people provide false information, we control the percentage of members (from 0% to 100%) who can randomly set

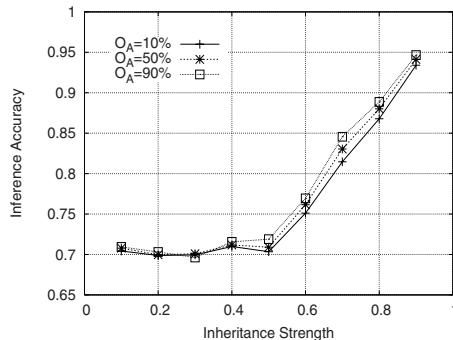


Fig. 14.6. Inference accuracy of Bayesian inference for different society openness when $P_t = 0.3$

their attribute values (referred to as randomness). Fig. 14.7 shows the impact of randomness on the inference accuracy at prior probability $P_t = 0.3$ and inheritance strength $P_{rl} = 0.7$. At low randomness, we note that the Bayesian inference clearly has a higher accuracy than the naïve inference. For example, when the randomness is 0.1, the inference accuracy of Bayesian and naïve inferences is 79.7% and 72.9% respectively. However, the advantage of Bayesian inference decreases as the randomness increases. This is especially so when the randomness reaches 1.0. At that point, there is almost no difference in the inference accuracy between Bayesian and naïve inferences. This is because their attribute values no longer follow the causal relations between friends when they randomly negate their attribute values. As a result, Bayesian inference behaves similar to naive inference. Thus, from a privacy protection point of view, falsifying personal attribute values can be an effective technique. Based on these characteristics, we will propose several schemes for privacy protection and evaluate their effectiveness in Sect. 14.5.

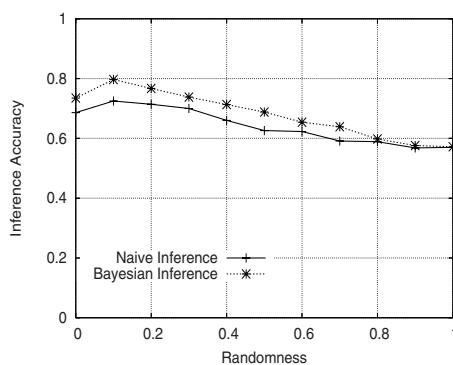


Fig. 14.7. Inference accuracy of Bayesian inference for different randomness when $P_t = 0.3$ and $P_{rl} = 0.7$

14.4.4 Experiments on Epinions.com

To evaluate the performance of Bayesian inference in a real social network, we conduct some experiments on Epinions.com [6]. Epinions is a review website for products including digital cameras, video games, hotels, restaurants, etc. Epinions divides these products into 23 categories and hundreds of subcategories. We consider that people have interests in a particular category if they write reviews on products in this category. In addition, registered members can also specify members in Epinions that they trust. Thus, a social network is formed where people are connected by trust relations. In this trust network, if person A trusts person B , it is very likely that A also likes the products that B is interested in. In this experiment, we use Bayesian inference to predict people's interests in some categories from the friends that they trust, and then compared the prediction with the actual categories of their reviews published on Epinions. The higher the percentage of the matches, the better the prediction.

We collect 66,390 personal profiles from Epinions. Each profile represents an individual with his product reviews and the people he trusts. We remove people who have no review and have no friend at all, which reduces the collection to 44,992 personal profiles. On average, each person writes 17 reviews, and has reviews in four categories. Among all categories, the most popular ones are movies, electronics and books. In terms of trust relations, each individual trusts 17 persons on average, and the distribution of the trust relations per user falls into a power law distribution again (as shown in Fig. 14.8).

Before we perform Bayesian inference on Epinions, we need to further prune the social network by filtering out social relations that are not related to the target attribute. Although people in Epinions are connected by trust relations, the persons that an individual trusts may be different from category to category. Since this information is not given in Epinions, we apply a heuristic assumption that friends with similar types of common interests have similar types of relations. We perform a K-means clustering [15] over 23 categories in Epinions. Each cluster represents a group of similar interests. Several examples of clusters are shown in Table 14.1. For example, electronic and computer hardware are clustered together, online store & services is clustered with music and books, etc. Once we have the clusters, we filter out the social relations

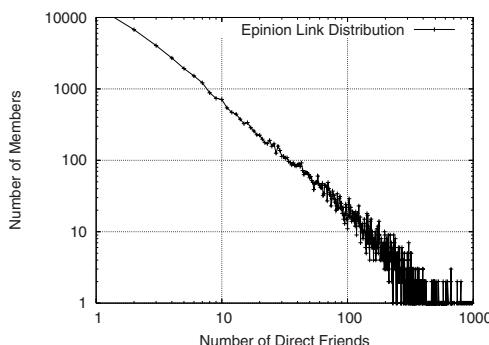


Fig. 14.8. Number of direct friends vs. number of members in Epinions on a log-log scale

Table 14.1. Examples of the clustered interests in Epinions

Cluster
Health, Personal Finance, Education
Online Store & Services, Music, Books
Restaurants & Gourmet, Movies
Electronics, Computer Hardware

Table 14.2. Inference accuracy comparison between Bayesian and naïve inferences

Target Attribute	P_t	$P_{t t}$	Accuracy	
			Naïve Inference	Bayesian Inference
Health	0.461	0.734	53.9%	63.8%
Online Store & Services	0.522	0.735	52.2%	60.6%
Restaurants & Gourmet	0.432	0.667	56.8%	64.2%
Electronics	0.766	0.833	76.6%	76.5%

if connected people have no common interests with others in the cluster. In other words, when predicting the target attribute values of health, we only consider the social relations where connected persons have a common interest in at least one category in the health cluster, i.e., personal finance, education or health categories. This filtering process reduces the original social network into a more focused social network. Once the social network is pruned, we perform Bayesian inference.

Table 14.2 compares the inference accuracy of Bayesian and naïve inferences. Note that the openness used in this experiment is 100%. As we can see from this table, Bayesian inference achieves higher predictions than the naïve inference. For the health category, the inference accuracy of naïve inference is 53.9%, and the corresponding accuracy of Bayesian inference is 63.8%. The results of other attributes show a similar trend, except for the electronics categories. This is because electronics is a very popular interest with prior probability P_t 0.766. Thus, most people will have this interest themselves, and the influence from friends is not comparatively strong enough.

14.5 Privacy Protection

We have shown that private attribute values can be inferred from social relations. One way to prevent such inference is to alter an individual's social network, which means changing his social relations or the attribute values of his friends. For social relations, we can either hide existing relations or add fraudulent ones. For friends' attributes, we can either hide or falsify their values. Our study on society openness suggests that random changes on a social network have only a small effect on the result of Bayesian inference. Therefore, an effective protection method requires choosing appropriate candidates for alteration.

In this section we shall study privacy protection schemes. We first present a theorem that captures the causal effect between friends' attribute values in a chain topology. We then apply this theorem to develop our protection schemes. We conduct

experiments on the Livejournal network structure and evaluate the performance of the proposed protection schemes.

14.5.1 Causal Effect between Friends' Attribute Values

As mentioned earlier, children's attribute values are the result of the interaction between the inheritance strength P_{tl} and the mutation strength P_{tf} of their parents. For example, in a family where the inheritance strength is stronger than the mutation strength, children tend to inherit the same attribute value from their parents; thus, the evidence of a child having the attribute value t increases our belief that his/her parent has the same attribute value t . On the contrary, when the inheritance strength is weaker than the mutation strength, parents and children are more likely to have opposite attribute values, and the evidence of a child having an attribute value t reduces our belief that his/her parent has the same attribute value t . Inspired by this observation, we derive a theorem to quantify the causal effects between friends' attribute values.

Theorem: Given a social network with a chain topology, let Z be the target node, Z_{n0} be Z 's descendant at n hops away. Assuming the attribute value of Z_{n0} is the only evidence observed in this chain, and the prior probability P_t satisfies $0 < P_t < 1$, we have $P(Z = t | Z_{n0} = t) > P(Z = t)$ iff $(P_{tl} - P_{tf})^n > 0$, and $P(Z = t | Z_{n0} = f) > P(Z = t)$ iff $(P_{tl} - P_{tf})^n < 0$, where P_{tl} and P_{tf} are the inheritance strength and mutation strength of the network respectively.

Proof: see Appendix.

This theorem states that when $P_{tl} > P_{tf}$, the posterior probability $P(Z = t | Z_{n0} = t)$ is greater than the prior probability $P(Z = t)$. Thus, the evidence of $Z_{n0} = t$ increases our prediction for $Z = t$. On the other hand, when $P_{tl} < P_{tf}$, whether $P(Z = t | Z_{n0} = t)$ is greater than $P(Z = t)$ or not also depends on the value of n , i.e., the depth of Z_{n0} . When n is even, the evidence that $Z_{n0} = t$ will increase our prediction for $Z = t$. However, when n is odd, the evidence that $Z_{n0} = t$ will decrease our prediction for $Z = t$.

14.5.2 A Privacy Protection Rule

Based on the above theorem, we propose a privacy protection rule as follows. Assume the protection goal is to reduce others' belief that the target node has the attribute value t . We alter the nodes in the social network with attribute value t when $P_{tl} > P_{tf}$. The alteration could be: 1) hide or falsify the attribute values of friends who satisfy the above conditions, or 2) hide relationships to friends who satisfy the above conditions, or add fraudulent relationships to friends who do not. On the other hand, when $P_{tl} < P_{tf}$ we alter nodes with attribute value t when that node is even hops away from the target node; otherwise, we alter nodes with attribute value f . To mislead people into believing the target node possesses an attribute value t , we can apply these techniques in the opposite way.

Based on the protection rule, we propose the following four protection schemes:

- Selectively hiding attribute value (SHA). SHA hides the attribute values of appropriate friends.

- Selectively falsifying attribute value (SFA). SFA falsifies the attribute values of appropriate friends.
- Selectively hiding relationships (SHR). SHR hides the relationship between the target node and selected direct friends. When all the friend relationships of this individual are hidden, the individual becomes a singleton, and the prediction will be made based on the prior probability.
- Selectively adding relationships (SAR). Contrary to hiding relationships in SHR, based on the protection rule, SAR selectively adds fraudulent relationships with people whose attribute values could cause incorrect inference to the target node

14.5.3 Performance of Privacy Protection

In this section we conduct a set of controlled experiments to evaluate different schemes for privacy protection. To provide privacy protection on an individual's attribute value (target node), we incrementally alter this individual's social network until the attribute value from inference predication changes its value and becomes contrary to its original value. The protection is considered a failure if it fails to change the attribute value prediction and no further alteration can be made.

We use a randomly hiding attribute value (RHA) as a baseline to evaluate the performance of the proposed protection schemes. RHA randomly selects a friend in the individual's social network and hides his/her attribute value without following the protection rule. We repeatedly perform such operations with the individual's direct friends. If protection fails after we hide all the direct friends' attribute values, we proceed to hide attribute values of indirect friends (e.g., at two hops away from this individual) and so on.

We have performed simulation experiments on 3000 individuals (nodes) in the Livejournal data set. For each node, we apply the above protection schemes and compare their performance. The two metrics used are: the percentage of individuals whose attribute values are successfully protected and the average number of alterations needed to reach such protection.

Fig. 14.9 displays the percentage of successful protection for different inheritance strengths P_{ht} at $P_t = 0.3$. We note that the effectiveness of the selected schemes follows the order: SAR > SFA > SHR > SHA > RHA. We shall now discuss the behavior of these schemes to explain and support our experimental findings. For RHA, SHA, SHR and SFA, the maximum number of alterations is the number of descendants (e.g., for RHA and SHA) and the number of direct friends (e.g., for SFA and SHR) of the target node. Since SAR can add new friend relationships and support the highest number of alterations to the social network, SAR provides more privacy protection to the target node. The performance of SFA and SHR follows SAR. We can view SFA as a combination of SHR and SAR, i.e., hiding a friend relationship followed by adding a fraudulent relationship. Therefore, the performance of SFA is better than that of SHR. SHA does not perform as well as SFA and SHR because friends at multiple hops away still leave clues for privacy inference. Finally, RHA does not follow the protection rule to take advantage of the properties of the social network, so it yields the worst performance.

Fig. 14.10 presents the performance based on the average number of alterations required to successfully protect the attribute value of a target node. We noted that RHA

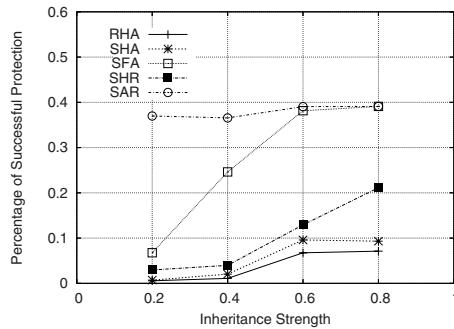


Fig. 14.9. Performance comparison of selected schemes based on the percentage of successful protection for $P_t = 0.3$

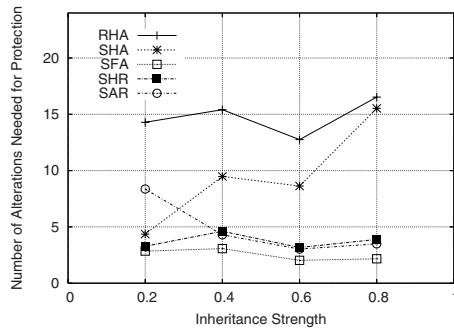


Fig. 14.10. Number of alterations required to successfully protect the attribute value of a target node at $P_t = 0.3$

has the worst and SFA has the best performance among the proposed schemes. The average number of alterations of SHR and SAR are comparable for most cases. Note that the average number of required changes for SAR is higher than that of SHR at $P_{tl} = 0.2$. This is because at the low inheritance strength region, SAR can protect many cases that other schemes cannot protect by adding a large number of fraudulent friend relationships. Finally, SHA performs better than RHA but not as good as the other schemes.

Figs. 14.9 and 14.10 reveal the effectiveness of using the protection rule for deriving privacy protection schemes. Furthermore, SFA can provide successful protection for most cases, yet does not require an excessive number of alterations to the original social network.

14.6 Analysis of RHR and SHR

In the previous section we demonstrated that selective social network alterations based on the protection rule are more effective than the method that does not follow the protection rule. We shall now use analysis to compare the difference between

randomly hiding friend relationships (RHR) and selectively hiding friend relationships (SHR). Specifically, we use the frequency of posterior probability variation after hiding friend relationships as a metric. A hiding scheme that has a high frequency of posterior probability variation is considered more effective in privacy protection than that of the low frequency ones.

14.6.1 Randomly Hiding Friend Relationships (RHR)

Hiding friend relationships means removing direct friends of the target node. The social network can be represented as a two-level tree with the target node Z as the root and n_1 direct friends $Z_{10}, \dots, Z_{l(n_1-1)}$ as leaves. We want to derive the probability distribution of the posterior probability variation due to randomly hiding friend relationships, i.e., the difference between the posterior probability after hiding their attribute values and the corresponding probability of this occurrence.

Let random variables N_{1t} and N'_{1t} be the number of friends having attribute value t before and after hiding h friend relationships, where $0 \leq h \leq n_1$ and $\max(0, N_{1t} - h) \leq N'_{1t} \leq \min(N_{1t}, n_1 - h)$. If $N_{1t} = n_{1t}$ and $N'_{1t} = n'_{1t}$, we can compute the posterior probabilities $P(Z = t | N_{1t} = n_{1t})$ and $P(Z = t | N'_{1t} = n'_{1t})$ from Eq. 14.4 respectively. Therefore, the posterior probability variation caused by hiding h friend relationships is (Eq. 14.5):

$$\begin{aligned} \Delta P(Z = t | N_{1t} = n_{1t}, N'_{1t} = n'_{1t}) &= \\ |P(Z = t | N_{1t} = n_{1t}) - P(Z = t | N'_{1t} = n'_{1t})|. \end{aligned} \quad (14.5)$$

Now we want to derive the probability that each possible value of $\Delta P(Z = t | N_{1t} = n_{1t}, N'_{1t} = n'_{1t})$ occurs. In other words, we want to compute the probability of the joint event $N_{1t} = n_{1t}$ and $N'_{1t} = n'_{1t}$ (before and after hiding friend relationships), which is equal to (Eq. 14.6):

$$P(N_{1t} = n_{1t}, N'_{1t} = n'_{1t}) = P(N_{1t} = n_{1t}) \cdot P(N'_{1t} = n'_{1t} | N_{1t} = n_{1t}). \quad (14.6)$$

Initially, if we know Z 's attribute value is Z ($z \in \{t, f\}$), the probability that $N_{1t} = n_{1t}$ follows the Binomial distribution (Eq. 14.7):

$$\begin{aligned} P(N_{1t} = n_{1t} | Z = t) &= \binom{n_1}{n_{1t}} \cdot P_{t|t}^{n_{1t}} \cdot P_{f|t}^{n_1 - n_{1t}}, \\ P(N_{1t} = n_{1t} | Z = f) &= \binom{n_1}{n_{1t}} \cdot P_{t|f}^{n_{1t}} \cdot P_{f|f}^{n_1 - n_{1t}}. \end{aligned} \quad (14.7)$$

By un-conditioning on Z , we obtain (Eq. 14.8):

$$\begin{aligned} P(N_{1t} = n_{1t}) &= P(Z = t) \cdot P(N_{1t} = n_{1t} | Z = t) + \\ P(Z = f) \cdot P(N_{1t} = n_{1t} | Z = f). \end{aligned} \quad (14.8)$$

We define h_t and h_f as the numbers of removed nodes (i.e., hidden friend relationships) with attribute value t and f , respectively ($h_t = n_{lt} - n'_{lt}$ and $h_f = h - h_t$). Then we can compute the conditional probability that $N'_{lt} = n'_{lt}$ given $N_{lt} = n_{lt}$ as (Eq. 14.9):

$$P(N'_{lt} = n'_{lt} | N_{lt} = n_{lt}) = \frac{\binom{n_{lt}}{h_t} \binom{n_l - n_{lt}}{h_f}}{\binom{n_l}{h}}. \quad (14.9)$$

In this equation, the numerator represents the number of ways to remove h_t nodes with value t and h_f nodes with value f , and the denominator represents the number of combinations when choosing any h nodes from a total of n_l nodes.

Substituting Eq. 14.8 and Eq. 14.9 into Eq. 14.6, we obtain $P(N_{lt} = n_{lt}, N'_{lt} = n'_{lt})$.

14.6.2 Selectively Hiding Friend Relationships (SHR)

We perform a similar analysis for selectively hiding friend relationships in a two-level tree. Unlike random selection which randomly selects nodes with attribute values t or f , this method follows the protection rules and selects all the nodes with the same attribute values to hide. Thus, we can compute $\Delta P(Z = t | N_{lt} = n_{lt}, N'_{lt} = n'_{lt})$ as in the previous section. However, the distribution of posterior probability variation needs to be computed differently.

Given h , the number of nodes to remove, n'_{lt} is deterministic. For example, if we remove nodes with attribute t , then $n'_{lt} = m - h$; otherwise $n'_{lt} = m$. Consequently, in the former case (Eq. 14.10),

$$P(N_{lt} = n_{lt}, N'_{lt} = n'_{lt}) = \begin{cases} P(N_{lt} = n_{lt}), & \text{if } n'_{lt} = m - h \\ 0, & \text{otherwise} \end{cases} \quad (14.10)$$

whereas in the latter case (Eq. (11)),

$$P(N_{lt} = n_{lt}, N'_{lt} = n'_{lt}) = \begin{cases} P(N_{lt} = n_{lt}), & \text{if } n'_{lt} = m \\ 0, & \text{otherwise} \end{cases} \quad (14.11)$$

where $P(N_{lt} = n_{lt})$ can be obtained from Eq. 14.8.

14.6.3 Randomly vs. Selectively Hiding Friend Relationships

We first compute the average variation in the posterior probability of both RHR and SHR, as shown in Fig. 14.11. We fix n_l to be ten and vary h from one to nine. The x-axis is the number of friends that we hide, and the y-axis is the posterior probability variation based on Eq. 14.5. Clearly, SHR has higher posterior probability variation than RHR, especially for the case of a large number of hidden friends.

We also plot the histogram of the posterior probability variation $\Delta P(Z = t | N_{lt} = n_{lt}, N'_{lt} = n'_{lt})$. We divide the range of posterior probability variation into ten equal width intervals. Then we compute the probability that the posterior probability variation falls in each interval.

Fig. 14.12 shows the histogram of the posterior probability variation for RHR and SHR, when the prior probability is 0.3 and the influence strength is 0.7. The x axis represents the intervals and the y axis represents the frequency of the posterior probability variation within the interval. The frequency is derived from Eq. 14.6 for RHR and from Eqs. 14.11 and 14.12 for SHR. For SHR, we remove friends with attribute value t . The maximum number of removed friends k cannot exceed N_{It} . As a result, we do not consider the cases when $n_{It} < k$, and we normalize the frequency results for selectively hiding friends based on the overall probability that $n_{It} \geq k$.

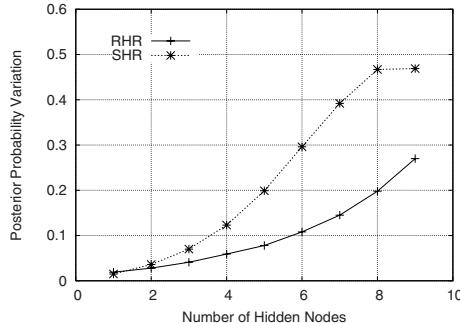


Fig. 14.11. Average posterior probability variation for selectively and randomly hiding friend relationships

For RHR, we observe that the variation is less than 0.1 for 70% to 90% of the cases in Fig. 14.12(a). Thus, the posterior probability is unlikely to be varied greatly. In contrast, the posterior probability variation in Fig. 14.12(b) is widely distributed, which means there are noticeable changes in the posterior probability after hiding nodes selectively. This trend is more pronounced when increasing the number of removed friends. For example, when $h = 8$, the frequency of the variation lying between 0.9 and 1.0 is about 28.8% as compared to 1.9% in Fig. 14.12(a). These results show the effectiveness of using the protection rule for privacy protection.

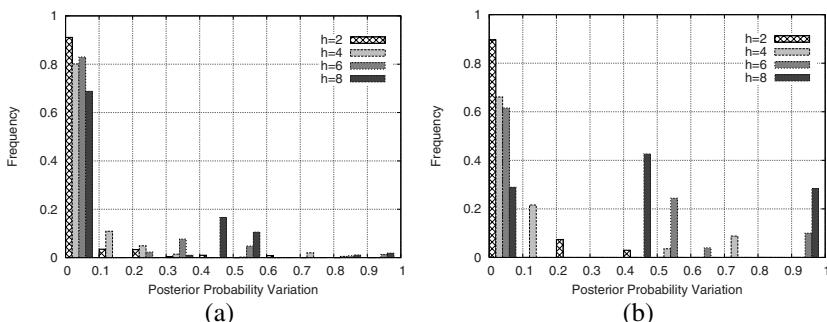


Fig. 14.12. Frequency of posterior probability variation for (a) randomly hiding friend relationships, and (b) selectively hiding friend relationships

14.7 Related Work

Social network analysis has been widely used in many areas such as sociology, geography, psychology and information science. It primarily focuses on the study of social structures and social network modeling. For instance, Milgram's classic paper [16] in 1967 estimates that every person in the world is only six hops away from one another. The recent success of the Google search engine [3] applies social network ideas to the Internet. In [17] Newman reviews the relationship between graph structures and the dynamic behavior of large networks. The Referral Web project mined social networks from a wide variety of publicly available information [11]. In sociology, social networks are often modeled as an autocorrelation model [5]. In this model, individuals' opinions or behaviors are influenced not only by those of others, but also by various other constraints in social networks. It uses a weight matrix to represent people's interactions; however, it is still not very clear how to choose the weight matrix. Leenders suggested building the weight matrix based on network structure information like node degrees [13]. Our work, on the other hand, models interpersonal relations using conditional probabilities; this depends on both structure information and personal attributes. Furthermore, Domingos and Richardson think that an individual's decision to buy a product is influenced by his friends, and they propose to model social networks as Markov random fields [4]. Because the social networks that they studied are built from a collaborative filtering database, each person is always connected to a fixed number of people who are most similar to him, which in turn forms a structure of stars with a regular degree. In contrast, we collect social networks directly from real online social network service providers. The number of friends of each individual varies. For the reasons of scalability and computational cost, we model social networks with Bayesian networks.

In terms of privacy protection, a great deal of effort has been devoted to developing cryptography and security protocols to provide security data transfer [1, 2]. Additionally, there are also models that have been developed for preserving individual anonymity in data publishing. Sweeney proposes a K-anonymity model which imposes constraints wherein the released information for each person cannot be re-identified from a group smaller than k [16]. In our study we assume that all the personal information released by the owners can be obtained by anyone in the social network. Under this assumption, we propose techniques to prevent malicious users from inferring private information from social networks.

14.8 Conclusion

We have focused this study on the impact of social relations on privacy disclosure and protection. The causal relations among friends in social networks can be effectively modeled by a Bayesian network, and personal attribute values can be inferred via their social relations. The inference accuracy increases as the influence strength increases between friends. Experimental results with real data from Epinions.com validate our findings that Bayesian inference provides higher inference accuracies than naïve inference.

Based on the interaction between inheritance strength and mutation strength, and the network structure, a protection rule is developed to provide guidance via selective network alterations (social relations and/or attribute values) to provide privacy protection. Experimental results show that alterations based on the protection rule are far more effective than random alterations. Because large variations of alterations can be provided by falsifying attribute values, this yields the most effective privacy protection among all the proposed methods.

For future study, we plan to investigate the use of multiple attributes to improve inference and protection. For example, diet and life style can reduce the risk of heart disease. Such multi-attribute semantic relationships can be obtained via domain experts or data mining. We can exploit this information to cluster target interests for inference.

References

1. Abadi, M., Needham, R.: Prudent Engineering Practice for Cryptographic Protocols. *Transactions on Software Engineering* 22, 6–15 (1995)
2. Bellovin, S.M., Merritt, M.: Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. In: IEEE Symposium on Security and Privacy, Oakland, California, May 1992, pp. 72–84 (1992)
3. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Proceedings of the Seventh International World Wide Web Conference (1998)
4. Domingos, P., Richardson, M.: Mining the Network Value of Customers. In: Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining (2001)
5. Doreian, P.: Models of Network Effects on Social Actors. In: Freeman, L.C., White, D.R., Romney, K. (eds.) *Research Methods in Social Analysis*, pp. 295–317. George Mason University Press, Fairfax (1989)
6. Epinions (1999), <http://www.epinions.com>
7. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning Probabilistic Relational Models. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden (1999)
8. He, J., Chu, W.W., Liu, Z.: Inferring Privacy Information from Social Networks. In: Mehrotra, S., Zeng, D.D., Chen, H., Thuraisingham, B., Wang, F.-Y. (eds.) *ISI 2006. LNCS*, vol. 3975, Springer, Heidelberg (2006)
9. Heckerman, D.: A Tutorial on Learning Bayesian Networks. Technical Report. MSR-TR-95-06 (1995)
10. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. In: KDD Workshop, pp. 85–96 (1994)
11. Kautz, H., Selman, B., Shah, M.: Referral Web: Combining Social Networks and Collaborative Filtering. *Communications of the ACM* 40(3), 63–65 (1997)
12. Livejournal (1997), <http://www.livejournal.com>
13. Leenders, R.T.: Modeling Social Influence Through Network Autocorrelation: Constructing the Weight Matrix. *Social Networks* 24, 21–47 (2002)
14. Lowd, D., Domingos, P.: Naive Bayes Models for Probability Estimation. In: Proceedings of the Twenty-Second International Conference on Machine Learning (ICML). ACM Press, Bonn (2005)

15. MacQueen, J.B.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)
16. Milgram, S.: The Small World Problem. Psychology Today (1967)
17. Newman, M.E.: The Structure and Function of Complex Networks. SIAM Review 45(2), 167–256 (2003)
18. Sweeney, L.: K-Anonymity: A Model for Protecting Privacy. International Journal on Uncertainty, Fuzziness, and Knowledge-Based Systems 10(5) (2002)
19. U.D. of Health and O. for Civil Rights Human Services, Standards for Privacy of Individually Identifiable Health Information (2003),
<http://www.hhs.gov/ocr/combinedregtext.pdf>
20. Watts, D.J., Strogatz, S.H.: Collective Dynamics of Small-World Networks. Nature (1998)
21. W.W.W.C. (W3C), The Platform for Privacy Preferences 1.1 (2004),
<http://www.w3.org/TR/P3P11/>
22. Zhang, N.L., Poole, D.: Exploiting Causal Independence in Bayesian Network Inference. Journal of Artificial Intelligence Research 5, 301–328 (1996)

Questions for Discussions

1. What are the reasons that the Bayesian network is suitable for modelling social networks for data inference?
2. What are the challenges in using Bayesian networks to model social networks?
3. Why can social networks improve the accuracy of information inference?
4. How does the privacy protection rule protect private attributes in social networks?
5. How can Bayesian inference accuracy be improved using multiple personal attributes?

Appendix

Theorem: Casual Effect Between Friends' Attribute Values in a Chain Network

Given a chain topology, let Z be the target node, Z_{n0} be Z 's descendant at n hops away. Assuming that the attribute value of Z_{n0} is the only evidence observed in this chain, and the prior probability P_t satisfies $0 < P_t < 1$, we have $P(Z = t | Z_{n0} = f) > P(Z = t)$ iff $(P_{nt} - P_{tf})^n > 0$, and $P(Z = t | Z_{n0} = f) > P(Z = t)$ iff $(P_{nt} - P_{tf})^n < 0$, where P_{nt} and P_{tf} are the inheritance strength and mutation strength of the network, respectively.

Proof:

Let us consider a chain topology shown in Fig. 14.13. The target node Z_{00} is the root node and each descendant (except the last one) has exactly one child. Consider the simplest example when $n=1$ (i.e., the target node Z has only one direct child Z_{10}) as shown in Fig. 14.13(a). In this example, the attribute value of Z_{10} is known.

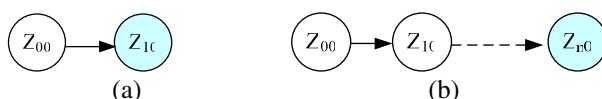


Fig. 14.13. The chain network structure: (a) the target node with one descendant; (b) the target node with n descendants

Assuming $Z_{10}=t$, from Eq. 14.1, the posterior probability $P(Z_{00} = t | Z_{10} = t)$ is:

$$\begin{aligned} P(Z_{00} = t | Z_{10} = t) &= \frac{P(Z_{00} = t) \cdot P(Z_{10} = t | Z_{00} = t)}{P(Z_{00} = t) \cdot P(Z_{10} = t | Z_{00} = t) + P(Z_{00} = f) \cdot P(Z_{10} = t | Z_{00} = f)} \\ &= \frac{P_t \cdot P_{tl|t}}{P_t \cdot P_{tl|t} + (1 - P_t) \cdot P_{tf|t}}. \end{aligned} \quad (14.12)$$

Thus,

$$\begin{aligned} P(Z_{00} = t | Z_{10} = t) > P(Z_{00} = t) &\Leftrightarrow \frac{P_t \cdot P_{tl|t}}{P_t \cdot P_{tl|t} + (1 - P_t) \cdot P_{tf|t}} > P_t \\ \Leftrightarrow P_{tl|t} - P_{tf|t} &> 0 \quad \text{for } P_t \neq 1 \end{aligned} \quad (14.13)$$

Similarly, when $Z_{10}=f$, we can prove $P(Z_{00} = t | Z_{10} = f) > P(Z_{00} = t)$ iff $P_{tl|f} - P_{tf|f} < 0$ for $P_t \neq 1$.

Now we extend this example to show how the attribute value of a node at depth n affects the prediction for Z . In Fig. 14.13(b), we show a network of $n+1$ nodes. In this figure, only Z_{n0} , Z 's descendent at depth n , has a known value. Fig. 14.14 shows the corresponding conditional probability table for these $n+1$ nodes.

Z_{00}	Z_{10}		$Z_{(n-1)0}$	Z_{n0}
t	t		t	t
t	f		t	t
t	t		f	t
t	f		f	t
f	t		t	t
f	f		t	t
f	t		f	t
f	f		f	t

Fig. 14.14. Conditional probability table for nodes in Fig. 14.13(b)

Let P_{tt}^n , P_{ft}^n , P_{tf}^n and P_{ff}^n be the joint distributions of Z and Z_{n0} :

$$\begin{aligned} P_{tt}^n &= P(Z_{00} = t, Z_{n0} = t), \\ P_{ft}^n &= P(Z_{00} = f, Z_{n0} = t), \\ P_{tf}^n &= P(Z_{00} = t, Z_{n0} = f), \\ P_{ff}^n &= P(Z_{00} = f, Z_{n0} = f). \end{aligned} \quad (14.14)$$

For example, $P_{tt}^n = P(Z_{00}=t, Z_{10}=t) = P(Z_{00}=t) P(Z_{10}=t|Z_{00}=t) = P_t P_{t|t}$ and so on.
We know,

$$\begin{aligned} P_{tt}^n + P_{tf}^n &= P(Z_{00}=t) = P_t, \\ P_{ft}^n + P_{ff}^n &= P(Z_{00}=f) = 1 - P_t. \end{aligned} \quad (14.15)$$

Further, from Fig. 14.14, we have the following relations:

$$\begin{aligned} P_{tt}^n &= P_{tt}^{n-1} \cdot P(Z_{n0}=t|Z_{n-1}=t) + P_{tf}^{n-1} \cdot P(Z_{n0}=t|Z_{n-1}=f) \\ &= P_{tt}^{n-1} \cdot P_{t|t} + P_{tf}^{n-1} \cdot P_{t|f}, \\ P_{ft}^n &= P_{ft}^{n-1} \cdot P(Z_{n0}=t|Z_{n-1}=t) + P_{ff}^{n-1} \cdot P(Z_{n0}=t|Z_{n-1}=f) \\ &= P_{ft}^{n-1} \cdot P_{t|t} + P_{ff}^{n-1} \cdot P_{t|f}. \end{aligned} \quad (14.16)$$

When $Z_{n0}=t$, the posterior probability is:

$$\begin{aligned} P(Z_{00}=t|Z_{n0}=t) &= \frac{P(Z_{00}=t, Z_{n0}=t)}{P(Z_{00}=t, Z_{n0}=t) + P(Z_{00}=f, Z_{n0}=t)} \\ &= \frac{P_{tt}^n}{P_{tt}^n + P_{ft}^n}. \end{aligned} \quad (14.17)$$

Therefore,

$$\begin{aligned} P(Z_{00}=t|Z_{n0}=t) &> P(Z_{00}=t) \Leftrightarrow \frac{P_{tt}^n}{P_{tt}^n + P_{ft}^n} > P_t \\ &\Leftrightarrow (1 - P_t) \cdot P_{tt}^n - P_t \cdot P_{ft}^n > 0. \end{aligned} \quad (14.18)$$

Based on Eq. 14.16,

$$\begin{aligned} (1 - P_t) \cdot P_{tt}^n - P_t \cdot P_{ft}^n &= \\ [(1 - P_t) \cdot P_{tt}^{n-1} - P_t \cdot P_{ft}^{n-1}] \cdot P_{t|t} + [(1 - P_t) \cdot P_{tf}^{n-1} - P_t \cdot P_{ff}^{n-1}] \cdot P_{t|f}. \end{aligned} \quad (14.19)$$

Substituting Eq. 14.15 into Eq. 14.19, we have

$$(1 - P_t) P_{tt}^n - P_t P_{ft}^n = [(1 - P_t) \cdot P_{tt}^{n-1} - P_t \cdot P_{ft}^{n-1}] \cdot (P_{t|t} - P_{t|f}). \quad (14.20)$$

Recursively, we have

$$(1 - P_t) \cdot P_{tt}^n - P_t \cdot P_{ft}^n = [(1 - P_t) \cdot P_{tt}^1 - P_t \cdot P_{ft}^1] \cdot (P_{t|t} - P_{t|f})^{n-1}. \quad (14.21)$$

Since $P_{tt}^I = P_t P_{t|t}$, and $P_{ft}^I = (1 - P_t) P_{t|f}$, we obtain

$$\begin{aligned} & (1 - P_t) \cdot P_{tt}^n - P_t \cdot P_{ft}^n \\ &= [(1 - P_t) \cdot P_t \cdot P_{t|t} - P_t \cdot (1 - P_t) P_{t|f}] \cdot (P_{t|t} - P_{t|f})^{n-1} \\ &= P_t \cdot (1 - P_t) \cdot (P_{t|t} - P_{t|f})^n. \end{aligned} \quad (14.22)$$

Combining Eq. 14.18 and Eq. 14.22, $P(Z_{00}=t | Z_{n0}=t) > P_t$ is equivalent to $(P_{t|t} - P_{t|f})^n > 0$ (when $0 < P_t < 1$). Similarly, we can show that $P(Z_{00}=t | Z_{n0}=f) > P_t$ is equivalent to $(P_{t|t} - P_{t|f})^n < 0$.

Protection of Database Security Via Collaborative Inference Detection *

Yu Chen and Wesley W. Chu

Computer Science Department,
University of California, USA
`{chenyu, wwc}@cs.ucla.edu`

Abstract. Malicious users can exploit the correlation among data to infer sensitive information from a series of seemingly innocuous data accesses. Thus, we develop an inference violation detection system to protect sensitive data content. Based on data dependency, database schema and semantic knowledge, we constructed a semantic inference model (SIM) that represents the possible inference channels from any attribute to the pre-assigned sensitive attributes. The SIM is then instantiated to a semantic inference graph (SIG) for query-time inference violation detection. For a single user case, when a user poses a query, the detection system will examine his/her past query log and calculate the probability of inferring sensitive information. The query request will be denied if the inference probability exceeds the pre-specified threshold. For multi-user cases, the users may share their query answers to increase the inference probability. Therefore, we develop a model to evaluate collaborative inference based on the query sequences of collaborators and their task-sensitive collaboration levels. Experimental studies reveal that information authoritativeness and communication fidelity are two key factors that affect the level of achievable collaboration. An example is given to illustrate the use of the proposed technique to prevent multiple collaborative users from deriving sensitive information via inference.

15.1 Introduction

Access control mechanisms are commonly used to protect users from the divulgence of sensitive information in data sources. However, such techniques are insufficient because malicious users may access a series of innocuous information and then employ inference techniques to derive sensitive data using that information.

To address this inference problem, we develop an inference detection system that resides at the central directory site. Because inference channels can be used to provide a scalable and systematic sound inference, we need to construct a semantic inference model (SIM) that represents all the possible inference channels from any attribute in the system to the set of pre-assigned sensitive attributes. The SIM can be constructed by linking all the related attributes which can be derived via attribute dependency from data dependency, database schema and semantic related knowledge. Based on the semantic inference model, the violation detection system keeps track of a user's query history.

* This research is supported by NSF grant number IIS-03113283.

When a new query is posed, all the channels where sensitive information can be inferred will be identified. If the probability to infer sensitive information exceeds a pre-specified threshold, the current query request will then be denied. Therefore, our system can prevent malicious users from obtaining sensitive information.

This inference detection approach is based on the assumption that users are isolated and do not share information with one another. This assumption, however, may not be the case in a real-life situation. Most users usually work as a team, and each member can access the information independently. Afterwards, the members may merge their knowledge together and jointly infer the sensitive information. Generalizing from a single-user to a multi-user collaborative system greatly increases the complexity of the inference detection system.

For example, one of the sensitive attributes in the system can be inferred from four different inference channels. There are two collaborators and each poses queries on two separate channels. Based on individual inference violation detection, neither of the users violates the inference threshold from their query answers. However, if the two users share information, then the aggregated knowledge from the four inference channels can cause an inference violation (see Sect. 15.7.2).

This motivates us to extend our research from a single user to the multiple user case, where users may collaborate with each other to jointly infer sensitive data. We have conducted a set of experiments, using our inference violation detector as a test bed to understand the characteristics in collaboration as well as the effect on collaborative inference. From the experiments, we learn that for a given specific task, the amount of information that flows from one user to another depends on the closeness of their relationships and the knowledge related to the task. Thus, collaborative inference for a specific task can be derived by tracking the query history of all the users together with their collaboration levels.

This chapter is organized as follows. Sect. 15.2 presents related work. Sect. 15.3 introduces a general framework for the inference detection system, which includes the knowledge acquisition module, semantic inference model and violation detection module. Sect. 15.4 discusses how to acquire and represent knowledge that could generate inference channels. Sect. 15.5 integrates all possible inference channels into a Semantic Inference Model which can be instantiated and then mapped into a Bayesian network to reduce the computation complexity for data inference. As shown in Sect. 15.6, we are able to detect inference violation at query time for both individual user and multiple collaborative users. Sect. 15.7 presents an example to illustrate the use of the proposed technique for collaboration inference detection. Sect. 15.8 presents collaboration level experiments and their estimations. Sect. 15.9 discusses the robustness of inference detection and threshold determination via sensitivity analysis. Sect. 15.10 presents the conclusion.

15.2 Related Work

Database inferences have been extensively studied. Many approaches to address the inference problem were presented in [20]. Particularly, Delugach and Hinke used database schema and human-supplied domain information to detect inference problems during database design time [18, 28, 29]. Garvey, et al. developed a tool for database

designers to detect and remove specific types of inference in a multilevel database system [22]. Both approaches use schema-level knowledge and do not infer knowledge at the data level. These techniques are also used during database design time and not at run time. However, Yip, et al. pointed out the inadequacy of schema-level inference detection, and he identifies six types of inference rules from the data level that serve as deterministic inference channels [47]. In order to provide a multilevel secure database management system, an inference controller prototype was developed to handle inferences during query processing. Rule-based inference strategies were applied in this prototype to protect the security [43]. Further, since data update can affect data inference, [21] proposed a mechanism that propagates update to the user history files to ensure no query is rejected based on the outdated information. To reduce the time in examining the entire history log in computing inference, [44] proposed to use a prior knowledge of data dependency to reduce the search space of a relation and thus reduce the processing time for inference. Open inference techniques were proposed to derive approximate query answering when network partitions occurred in distributed databases. Feasible open inference channels can be derived based on query and database schema [10].

The previous work on data inference mainly focused on deterministic inference channels such as functional dependencies. The knowledge is represented as rules and the rule body exactly determines the rule head. Although such rules are able to derive sound and complete inference, much valuable non-deterministic correlation in data is ignored. For example, salary ranges may not deterministically depend on the ranks. Further, many semantic relationships, as well as data mining rules, can not be specified deterministically. To remedy this shortcoming, we propose a probabilistic inference approach to treat the query-time inference detection problem. The contribution of our research consists of: 1) Derive probabilistic data dependency, relational database schema and domain-specific semantic knowledge and represent them as probabilistic inference channels in a Semantic Inference Model. 2) Map the instantiated Semantic Inference Model into a Bayesian network for efficient and scalable inference computation. 3) Propose an inference detection framework for multiple collaborative users.

15.3 The Inference Framework

The proposed inference detection system consists of three modules, as shown in Fig. 15.1: knowledge acquisition, semantic inference model (SIM), and security violation detection including user collaboration relation analysis.

The *Knowledge Acquisition* module extracts data dependency knowledge, data schema knowledge and domain semantic knowledge. Based on the database schema and data sources, we can extract data dependency between attributes within the same entity and among entities. Domain semantic knowledge can be derived by semantic links with specific constraints and rules. A semantic inference model can be constructed based on the acquired knowledge.

The *Semantic Inference Model (SIM)* is a data model that combines data schema, dependency and semantic knowledge. The model links related attributes and entities as well as semantic knowledge needed for data inference. Therefore SIM represents

all the possible relationships among the attributes of the data sources. A *Semantic Inference Graph (SIG)* can be constructed by instantiating the entities and attributes in the SIM. For a given query, the SIG provides inference channels for inferring sensitive information.

Based on the inference channels derived from the SIG, violation detection combines the new query request with the request log, and it checks to see if the current request exceeds the pre-specified threshold of information leakage. If there is collaboration according to collaboration analysis, the *Violation Detection* module will decide whether to answer a current query based on the acquired knowledge among the malicious group members and their collaboration level to the current user.

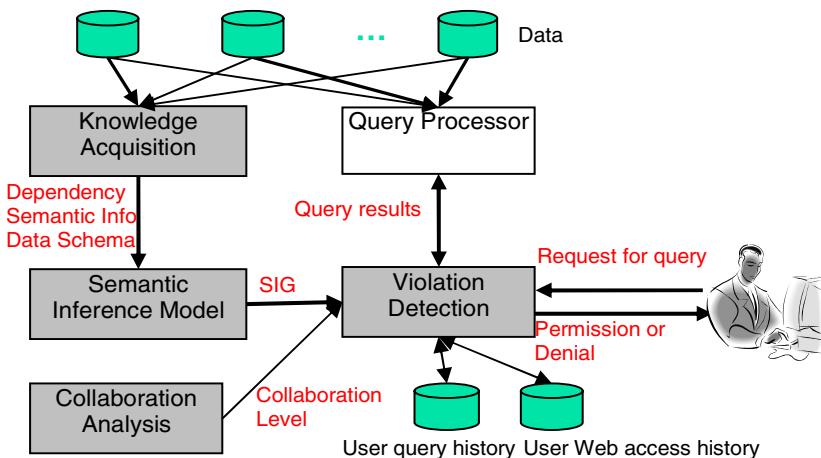


Fig. 15.1. The framework for an Inference Detection System

15.4 Knowledge Acquisition for Data Inference

Since users may pose queries and acquire knowledge from different sources, we need to construct a semantic inference model for the detection system to track user inference intention. The semantic inference model requires the system to acquire knowledge from data dependency, database schema and domain-specific semantic knowledge. This section will discuss how to acquire that knowledge.

15.4.1 Data Dependency

Data dependency represents causal relationships and non-deterministic correlations between attribute values. Because of the non-deterministic nature, the dependency between two attributes A and B is represented by conditional probabilities $p_{ij} = Pr(B=b_i | A=a_j)$. Thus, the non-deterministic data dependency is a more general representation than the relational functional dependency or other types of deterministic relationships. There are two types of non-deterministic data dependencies as defined in the Probabilistic Relational Model [19, 24]: *dependency-within-entity* and *dependency-between-related-entities*, as defined in the following.

Dependency-within-entity: Let A and B be two attributes in an entity E; if B depends on A, then for each instance of E, its value of attribute B depends on its value of attribute A with a probability value. To learn the parameter of dependency-within-entities from relational data, from a relational table that stores entity E, we can derive the conditional probabilities $p_{ij} = Pr(B=b_i | A=a_j)$ via a sequential scan of the table with a counting of the occurrences of A, B, and co-occurrences of A and B.

Dependency-between-related-entities: Let A be an attribute in entity E_1 and C be an attribute in E_2 , and E_1 and E_2 are related by R, which is a relation that can be derived from database schema. If C depends on A, then only for related instances of E_1 and E_2 , the value of attribute C in E_2 instances depends on the value of attribute A in related instances of E_1 . Such dependency-between-related-entities only exists for related instances of entities E_1 and E_2 . The parameters of dependency-between-related-entities can be derived by first joining the two entity tables based on the relation R and then scanning and counting the frequency of occurrences of the attribute pair in the joined table. If two entities have an m-to-n relationship, then the associative entity table can be used to join the related entity tables to derive dependency-between-related-entities [12].

15.4.2 Database Schema

In relational databases, database designers use data definition language to define data schema. The owners of the entities specify the primary key and foreign key pairs. Such pairing represents a relationship between two entities. If entity E_1 has primary key pk , entity E_2 has foreign key fk , and $e_1.pk = e_2.fk$, then dependency-between-related-entities from attribute A (in e_1) to attribute C (in e_2) can be derived.

15.4.3 Domain-Specific Semantic Knowledge

Other than data dependencies inside relational data sources, outside information such as domain knowledge can also be used for inferences. Specifically, domain-specific semantic relationships among attributes and/or entities can supplement the knowledge of malicious users and help their inference. For example, the semantic knowledge “can land” between Runway and Aircraft implies that the length of Runway should be greater than the minimum Aircraft landing distance, and the width of Runway should be greater than the minimum width required by Aircraft. If we know the runway requirement of aircraft C-5, and C-5 “can land” in the instance of runway r , then the values of attributes *length* and *width* of r can be inferred from the semantic knowledge. Therefore, we want to capture the domain-specific semantic knowledge as extra inference channels in the Semantic Inference Model.

Semantic knowledge among attributes is not defined in the database and may vary with context. However, from a large set of semantic queries posed by the users, we can extract the semantic constraints [50]. For example, in the WHERE clause of the following query, clauses #3 and #4 are the semantic conditions that specify the semantic relation “can land” between entity Runways and entity Aircrafts. Based on this

query, we can extract semantic knowledge “can land” and integrate it into the Semantic Inference Model shown in Fig. 15.3.¹

■ **Query: Find airports that *can land* a C-5 cargo plane.**

```
SELECT AP.APORT_NM
FROM AIRCRAFTS AC, AIRPORTS AP, RUNWAYS R
WHERE AC.AC_TYPE_NM = 'C-5' and #1
AP.APORT_NM = R.APORT_NM and #2
AC.WT_MIN_AVG_LAND_DIST_FT <= R.RUNWAY_LENGTH_FT and #3
AC.WT_MIN_RUNWAY_WIDTH_FT <= R.RUNWAY_WIDTH_FT; #4
```

15.5 Semantic Inference Model

The Semantic Inference Model (SIM) represents dependent and semantic relationships among attributes of all the entities in the information system. As shown in Fig. 15.2, the related attributes (nodes) are connected by three types of relation links: dependency link, schema link and semantic link.

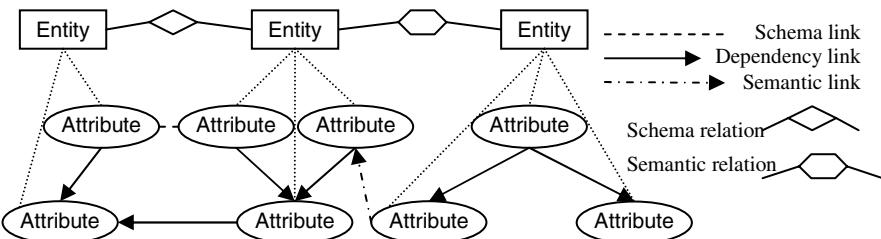


Fig. 15.2. A Semantic Inference Model. Entities are interconnected by schema relations (diamond) and semantic relations (hexagon). The related attributes (nodes) are connected by their data dependency, schema and semantic links.

Dependency link connects dependent attributes within the same entity or related entities. Consider two dependent attributes A and B. Let A be the parent node and B be the child node. The degree of dependency from B to A can be represented by the conditional probabilities $p_{ilj} = \Pr(B=b_i|A=a_j)$. The conditional probabilities of the child node given all of its parents are summarized into a conditional probability table (CPT) that is attached to the child node. For instance, Fig. 15.3(b) shows the CPT of the node “TAKEOFF_LANDING_CAPACITY” of the SIM in Fig. 15.3(a). The conditional probabilities in the CPT can be derived from the database content [19, 24]. For example, the conditional probability $\Pr(B=b_i|A=a_j)$ can be derived by counting the co-occurrence frequency of the event $B=b_i$ and $A=a_j$ and dividing it by the occurrence frequency of the event $A=a_j$.

¹ Clearly, the set of the semantic queries may be incomplete, which can result in the semantic knowledge being incomplete as well. However, additional semantic knowledge can be appended to the Semantic Inference Model as the system gains more semantic queries. The system can then reset to include the new knowledge. Otherwise, this will result in inference with knowledge update and is beyond the scope of this chapter.

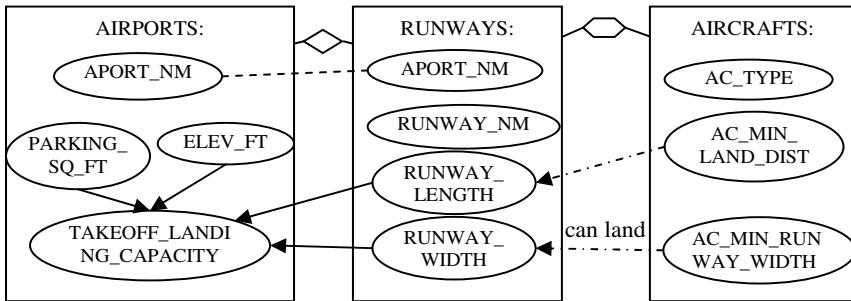


Fig. 15.3(a). A Semantic Inference Model example for Airports, Runways and Aircraft

		Conditional Probability of TAKEOFF_LANDING_CAPACITY											
Cond	parking_sq_ft	small						large					
		low			high			low			high		
		short	medium	long	short	medium	long	short	medium	long	short	medium	long
Takeoff_landing_capacity	small	0.9	0.8	0.7	0.7	0.6	0.95	0.85	0.85	0.75	0.75	0.65	0.75
	large	0.1	0.2	0.2	0.3	0.3	0.4	0.05	0.15	0.15	0.25	0.25	0.35

Fig. 15.3(b). Conditional probability table (CPT) for the attribute “TAKEOFF_LANDING_CAPACITY” summarizes its dependency on the four parent nodes. For example, $Pr(Takeoff_landing_capacity=small | Parking_sq_ft=small, Elev_ft = low, Runway_length=short, Runway_width=narrow)=0.9$. The conditional probabilities in the CPT can be derived from the database content.

node and set the value of the source node to “unknown.” In this case, the source and target node are independent, i.e., $Pr(T=t_i | P_1=v_1, \dots, P_n=v_n, P_S=unknown) = Pr(T=t_i | P_1=v_1, \dots, P_n=v_n)$. When the semantic relationship is known, the conditional probability of the target node is updated according to the semantic relationship and the value of the source node. If the value of the source node and the semantic relation are known, then $Pr(T=t_i | P_1=v_1, \dots, P_n=v_n, P_S=s_j)$ can be derived from the specific semantic relationship. For example, in Fig. 15.4(b), the semantic relationship determines that $Pr(T=t_i | P_1, \dots, P_n, P_S=s_1)=0.6$ and $Pr(T=t_i | P_1, \dots, P_n, P_S=s_2)=0.8$.

Schema link connects an attribute of the primary key to the corresponding attribute of the foreign key in the related entities. For example, in Fig. 15.3(a), APORTR_NM is the primary key in AIRPORTS and foreign key of RUNWAYS. Therefore, we connect these two attributes via schema link.

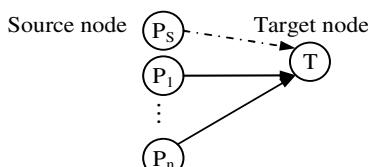


Fig. 15.4(a). Target node T with semantic link from source node P_S and dependency links from parents P_1, \dots, P_n

		Conditional Probability of T							
		unknown		s1		s2			
Cond	P _S	v ₁₁	v ₁₂	v _{n1}	v _{n2}	v ₁₁	v ₁₂	v _{n1}	v _{n2}
	P _n	v _{n1}	v _{n2}	v _{n1}	v _{n2}	v _{n1}	v _{n2}	v _{n1}	v _{n2}
T	t ₁	0.5	0.3	0.4	0.2	0.6	0.6	0.6	0.8
	t ₂	0.5	0.7	0.6	0.8	0.4	0.4	0.4	0.2

Fig. 15.4(b). The CPT of target node T summarizes the conditional probabilities of T given values of P_S and P₁, ..., P_n. For example, $Pr(T=t_1 | P_S=unknown, P_1=v_{11}, P_n=v_{n1})=0.5$.

Semantic link connects attributes with a specific semantic relation. To evaluate the inference introduced by semantic links, we need to compute the CPT for nodes connected by semantic links. Let T be the target node of the semantic link, P_S be the source node, and P₁, ..., P_n be the other parents of T, as shown in Fig. 15.4(a). The semantic inference from a source node to a target node can be evaluated as follows.

If the semantic relation between the source and the target node is unknown or if the value of the source node is unknown, then the source and target node are independent. Thus, the semantic link between them does not help inference. To represent the case of the unknown semantic relationship, we need to introduce the attribute value “unknown” to the source.

For example, the semantic relation “can land” between Runway and Aircraft (Fig. 15.5(a)) implies that the length of Runway is greater than the minimum required Aircraft landing distance. So the source node is aircraft_min_land_dist, and the target node is runway_length. Both attributes can take three values: “short,” “medium” and “long.” First, we add value “unknown” to source node aircraft_min_land_dist and set it as a default value. Then we update the conditional probabilities of the target node to reflect the semantic relationship. Here, we assume that runway_length has an equal probability of being short, medium or long. When the source node is set to “unknown,” the runway_length is independent of aircraft_min_land_dist; when the source node has a known value, the semantic relation “can land” requires runway_length is greater than or equal to aircraft_min_land_dist. Thus, the corresponding CPT for the node runway_length is shown in Fig. 15.5(b).

15.5.1 Computation Complexity of Constructing Semantic Inference Model

A SIM consists of linking related attributes (structure) and their corresponding conditional probabilities (parameters). Given a relational database, the learning of a SIM can be decomposed into two tasks: parameter learning and structure learning. In the first task, we assume that the structure of the SIM is known, i.e., the links between attributes are fixed, and our goal is to derive the conditional probability tables for each attribute. Since the parameters of semantic link are determined by its semantic constraint, let us now consider the computation complexity on learning parameters of data dependencies. Consider that given structure S has m attributes, each attribute A_i in table T_j has a set of parents $P(A_i)$. If all parents of A_i are in the same table with A_i , then the CPT of A_i can be derived by a single scan of T_j . If attribute A_i has a parent from related entity table T_k , then scanning on the joined table of T_j and T_k is needed to derive the CPT of A_i . In the worst case, the parameters can be learned in

$O(m \prod_i n_i)$ time, where m is the total number of attributes in the model and n_i is the size of the i^{th} table. When the number of dependency-between-related-entities is limited, the parameter learning can be reduced to approximately $O(\sum_i m_i n_i)$ where m_i ($< m$) is the number of attributes in the i^{th} table.

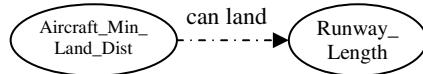


Fig. 15.5(a). The semantic link “can land” between “Aircraft_Min_Land_Dist” and “Runway_Length”

		Conditional Probability of Runway_length			
Cond	aircraft_min	unknown	short	medium	long
		short	0.33	0.33	0
Runway_Length	medium	0.33	0.33	0.5	0
	long	0.33	0.33	0.5	1

Fig. 15.5(b). Conditional Probability Table for Runway_length

If the structure of the SIM is not given by domain experts, we can generate a set of candidate structures with their corresponding parameters, and select the one that best matches the data sources. Algorithms for searching good dependency structures can be found in [19, 23].

15.5.2 Semantic Inference Graph

To perform inference at the instance level, we instantiate the SIM with specific entity instances and generate a semantic inference graph (SIG), as shown in Fig. 15.6. Each node in the SIG represents an attribute for a specific instance. To highlight the attributes of an entity instance, we group all the attributes of the instance into a rectangular box. Related attributes are then connected via instance-level dependency links, instance-level schema links and instance-level semantic links. The attribute nodes in SIG have the same CPT as in SIM because they are just instantiated versions of the attributes in entities. As a result, the SIG represents all the instance-level inference channels in the SIM.

Instance-level dependency link: When a SIM is instantiated, the dependency-within-entity is transformed into dependency-within-instance in the SIG. Similarly, the dependency-between-related-entities in the SIM is transformed into a dependency between two attributes in the related instances. This type of dependency is preserved only if two instances are related by the instantiated schema link. That is, if attribute B in instance e_2 depends on attribute A in instance e_1 , and instances e_1 and e_2 are related by R denoted as $R(e_1, e_2)$, then there is a dependency-between-related-instances from B to A.

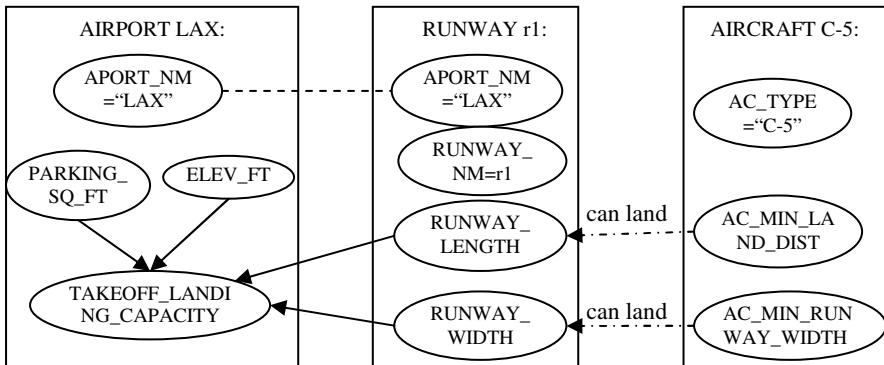


Fig. 15.6. The Semantic Inference Graph for airport instance (LAX), with runway r1 and aircraft C-5

Instance-level schema link: The schema links between entities in the SIM represent “key, foreign-key” pairs. At instance level, if the value of the primary key of an instance e_1 is equal to the value of the corresponding foreign key in the other instance e_2 which can be represented as $R(e_1, e_2)$, then connecting these two attributes will represent the schema link at the instance level. Otherwise, these two attributes are not connected.

Instance-level semantic link: At the instance level, assigning the value of the source node to “unknown” disconnects the semantic link between the attributes of two instances. On the other hand, if two instances have a specific semantic relation, then the inference probability of the target node will be computed based on its CPT and the value of the source node.

15.5.3 Evaluating Inference in Semantic Inference Graph (SIG)

For a given SIG, there are attribute dependencies within an entity, between related entities, and semantic relationships among attributes. As a result, there are many feasible inference channels that can be formed via linking the set of dependent attributes. Therefore, we propose to map the SIG to a Bayesian network to reduce the computational complexity in evaluating user inference probability for the sensitive attributes.

For any given node in a Bayesian network, if the value of its parent node(s) is known, then the node is independent of all its non-descending nodes in the network [26, 27, 30, 39, 40]. This independence condition greatly reduces the complexity in computing the joint probability of nodes in the network. More specifically, let x_i be the value of the node X_i , pa_i be the values of the parent nodes of X_i , then $P(x_i|pa_i)$ denotes the conditional probability of x_i given pa_i where $i=1,2,\dots,n$. Thus, the joint probability of the variables x_i is reduced to the product of $P(x_i|pa_i)$:

$$P(x_1, \dots, x_n) = \prod_i P(x_i | pa_i) \quad (15.1)$$

The probability for users to infer the sensitive node $S=s$ given the evidences $D_i=d_i$, $i=1, 2, \dots, n$ is:

$$P(s | d_1, d_2 \dots, d_n) = \frac{P(s, d_1, d_2 \dots, d_n)}{P(d_1, d_2 \dots, d_n)} \quad (15.2)$$

which can be further computed using Eq. 15.1. Thus, the probability of inferring a sensitive node can be computed from the conditional probabilities in the Bayesian network. Many algorithms have been developed to efficiently perform such calculations [16, 31, 35, 51, 52].

The Probabilistic Relational Model (PRM) is an extension of the Bayesian network that integrates schema knowledge from relational data sources [19, 23, 24]. Specifically, PRM utilizes a relational structure to develop *dependency-between-related-entities*. Therefore, in PRM an attribute can have two distinct types of parent-child dependencies: *dependency-within-entity* and *dependency-between-related-entities*, which match the two types of dependency links in the SIM. Since the semantic links in the SIM are similar to dependency links, we can convert each SIM to a PRM-based model. The corresponding Bayesian network can be generated after instantiating the model to instance level. Thus, for a given network, the probability of inferring a specific sensitive attribute can be evaluated via efficient Bayesian inference algorithms. In our test bed, we use SamIam [41], a comprehensive Bayesian network tool developed by the Automated Reasoning Group at UCLA, to compute the inference. The computation complexity for exact inference is mostly $O(n \cdot \exp(w))$, where n is number of nodes and w is the tree-width of the network [8, 2, 13, 17, 31, 52] and is scalable.

15.6 Inference Violation Detection for Individual User

Semantic inference graphs provide an integrated view of the relationships among data attributes, which can be used to detect inference violation for sensitive nodes. In such a graph, the values of the attributes are set according to the answers of the previous posted queries. Based on the list of queries and the user who posted those queries, the value of the inference will be modified accordingly. If the current query answer can infer the sensitive information greater than the pre-specified threshold, then the request for accessing the query answer will be denied [9].

Consider the example in Fig. 15.3. Let the TAKEOFF_LANDING_CAPACITY of any airport be the sensitive attribute, and it should not be inferred with probability greater than 70%. If the user has known that: 1) Aircraft C-5 can land in airport LAX runway r1; 2) C-5 has aircraft_min_land_dist = long and aircraft_min_runway_width = wide. Then this user is able to infer the sensitive attribute “LAX’s TAKEOFF_LANDING_CAPACITY = large” via Eqs. 15.2 and 15.1 with probability 58.30%, as shown in Fig. 15.7(a).

Now if the same user poses another query about the “Parking_sq_ft of LAX” and if this query is answered (as shown in Fig. 15.7(b), LAX_Parking_Sq_Ft=large), then the probability of inferring LAX_TAKEOFF_LANDING_CAPACITY = large by this user will increase to 71.50%, which is higher than the pre-specified threshold. Thus, this query request should be denied.

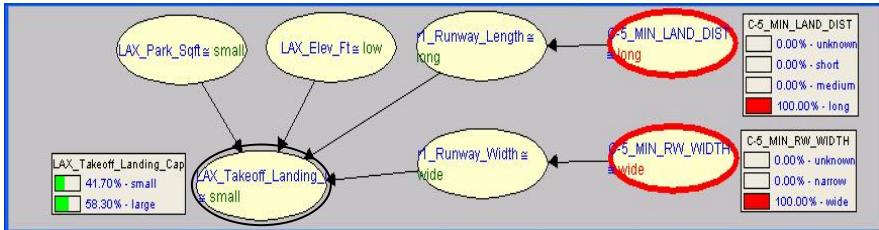


Fig. 15.7(a). Example of inference violation detection for single user. This is a portion of the Bayesian network for the transportation mission planning. The probability distribution of each node is shown in a rectangular box. The values of the bold nodes are given by previous query answers; the probability values of sensitive nodes are inferred.

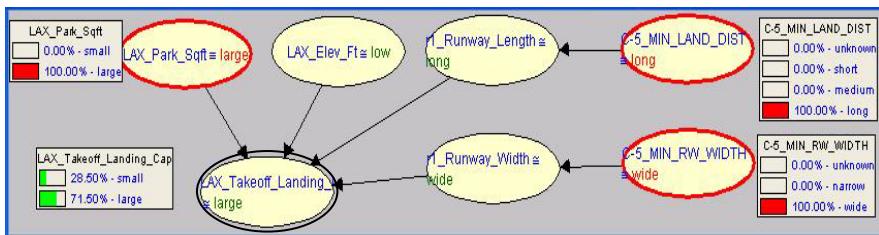


Fig. 15.7(b). Given the additional knowledge “LAX_Parking_Sq_Ft=large”, the probability for inferring the sensitive information “LAX_TAKEOFF_LANDING_CAPACITY =large” is increased to 71.50%

15.7 Inference Violation Detection for Collaborative Users

15.7.1 Collaborative Inference Violation Detection

To extend our inference violation detection module for collaborative users, we first need to define the collaboration level among users that is a metric for measuring the percentage of useful information flow from the source to the recipient. The collaboration level depends on two aspects: authoritativeness of the information source and the fidelity of the communication channel between the source and recipient. Authoritativeness can be determined by the reputability and authority of the information provider; fidelity depends on such factors as the willingness of the provider to release information, and/or the recipient’s understandability of the received information. We use collaboration levels to combine the source authoritativeness and channel fidelity. The higher the collaboration level between the pair of collaborators, the higher their collaboration effectiveness will be. More discussion of how to derive the collaboration level will be presented in Sect. 8.

Consider users A and B in Fig. 15.8. User B has a collaborative level of 85% for the information from A. Let Q_A and Q_B be the query answer set of user A and user B. User B can combine Q_A with his own previous query answer set Q_B and yield a higher inference probability for the sensitive node. For the example in Fig. 15.7(a), user B

has past query answers $Q_B = \{C-5_min_land_dist = long, C-5_min_rw_width = wide\}$ and then combines this with his acquired knowledge from user A: $Q_A = \{LAX_Park_Sqft = large\}$. Such collaboration increases the inference probability for the sensitive node from 58.30% to 66.55%, as shown in Fig. 15.8. Note that because the collaborative level of B for information from A is 85%, it yields a lower inference probability than the case where user B queries directly about LAX_Parking_Sq_Ft, as in Fig. 15.7(b).

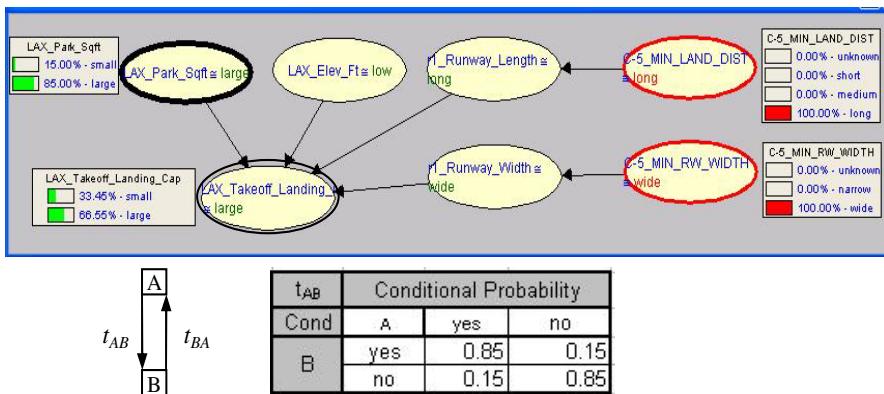


Fig. 15.8. Example of inference violation detection for multiple users. User B knows “C5_min_land_dist=long” and “C5_min_rw_width=wide” from his past query answers. User B also has the knowledge from A “LAX_Park_Sqft =large” with collaborative level 85%. Thus, the probability for user B to infer the sensitive information (shown in double ellipses) “LAX_Takeoff_Landing_Capacity=large” increases to 66.55%.

In general, according to the users' query history, there are two different types of collaborative user pairs, as shown in Fig. 15.9:

Collaboration with non-overlap inference channels: In this case, the two users pose queries on different non-overlap inference channels. The inference probability will be computed based on their combined knowledge discounted by their collaborative level.

Collaboration with overlap inference channels: In this case, the query sets posed by the two users overlap on inference channels. Such overlap may cause the users to have inconsistent belief in the same attribute on the inference channel. Thus, we need to integrate the overlapping knowledge according to the collaborative level to compute the appropriate inference probability.

Case (a) is the simple case of non-overlap inference channels. The influence from user A to user B is given by the collaboration level. Therefore, for user B, the query answers acquired by A (Q_A) can be combined with the query answers that are acquired by B (Q_B), but discounted by B's collaborative level to A. In addition, because Q_A and Q_B are from independent non-overlap inference channels, their inferences to sensitive node S are independent and can be directly combined. Thus the inference probability for the sensitive node can be computed based on the user's knowledge

from his past queries combined with his collaborator's query answers discounted by their respective collaborative level.

For Case (b), the queries posed by user A and user B overlap on their inference channels. Since Q_A and Q_B may cause inconsistent belief on some attribute nodes, these two query answer sets cannot be simply combined. For example, in Fig. 15.10(a), for attribute node X , Q_A indicates A has known $X=x$ and B can believe it with collaboration level t_{AB} ($t_{AB} \leq 1$).

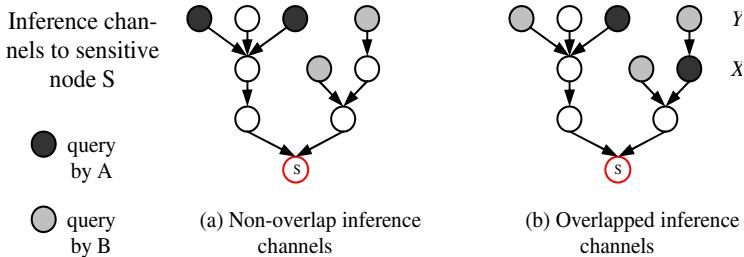


Fig. 15.9. Types of collaborative user pairs posing query sequence on the inference channels

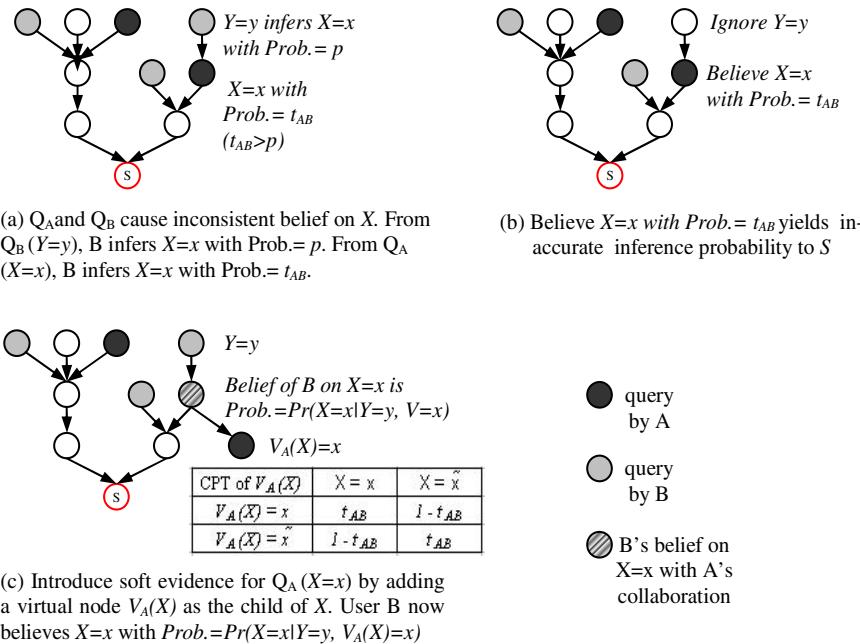


Fig. 15.10. A virtual node can be used in user B's inference network to resolve inconsistent belief when user B and A overlap on their inference channels

On the other hand, Q_B includes $Y=y$ which can infer $X=x$ with probability p . If $p \neq t_{AB}$, then Q_A and Q_B can cause B to have inconsistent belief on attribute X . Without loss of generality, we assume $p < t_{AB}$ for this example.

One approach to reconciling such inconsistent belief is to assume B will always choose to maximize his inference probability. Therefore, as shown in Fig. 15.10(b), B only follows A's advice ($X=x$ with $\text{prob.} = t_{AB}$) and ignore his own acquired knowledge ($Y=y$ infers $X=x$ with $\text{prob.} = p$). However, such a "max-inference" approach is not always correct, since people's belief is often strengthened by the confirmation and reduced by the conflicting knowledge. To represent the integration of inconclusive belief, we introduce the concept of *soft evidence* in probability calculus [14]. Soft evidence is inconclusive or unreliable information, as in the previous example, A tells B that $X=x$ and B only believes it with t_{AB} ($t_{AB} < 1$). For user B, $X=x$ is inconclusive knowledge, and therefore it needs to be set as soft evidence. To specify the soft evidence, we use the *Virtual Evidence* method developed in [14]. As shown in the Fig. 15.10(c), this method first adds a virtual attribute node $V_A(X)$ to be the child of the attribute node X to represent the event of receiving the soft evidence of X , that is, A tells B about $X=x$. Then the conditional probability of the virtual node is determined by the reliability of the soft evidence. In our example, both $\Pr(V_A(X) = x | X = x)$ and $\Pr(V_A(X) = \bar{x} | X = \bar{x})$ are determined by user B's *collaboration level* of information from A t_{AB} . Thus, the soft evidence can be integrated into the user's own knowledge. In the example, if originally B is ignorant about X , once A tells B about $X=x$, B will believe $X=x$ with probability t_{AB} . If originally B can infer X with knowledge $Y=y$, then his current belief in $X=x$ can be computed as $\Pr(X = x | Y = y, V_A(X) = x)$. Thus, we are able to integrate queries on overlapped inference channels from multiple collaborators based on their corresponding collaboration levels.

Therefore, for any type of two collaborative users, we can integrate one's knowledge to the other and detect their inference towards sensitive data. When any user poses a query, the system not only checks to see if the query requester can infer sensitive data above the threshold with a query answer, it also checks the other team members to guarantee that the query answer will not indirectly let them infer the sensitive attribute. We can iteratively generalize the above approach to an N-collaborator case. In general, when there are N collaborative users in the team, the violation detection system tracks the query posed by every team member. A query should be denied if the query answer will increase the certainty of any team member to infer the sensitive data above the pre-specified threshold.

1. Assume: current query request Q, malicious team M, sensitive data S, threshold of S is T;
2. List(M) = sort team members M in descending order of inference probability to S;
3. While(List(M) is not empty) {
 4. m = first member in List(M) with highest inference probability;
 5. max_col = the maximum collaborative level from any member in List(M) to the query requester;
 6. real_col = m's collaborative level to query requester;
 7. If (m integrate answer to Q with max_col can get inference probability < T)
 8. Then {answer query Q; goto end;}

```

9.   Else
10.    If (m integrate answer to Q with real_col can get inference probability >= T)
11.      Then {deny query Q; goto end;}
12.      Else { List(M) = List(M) - m; }
13.  }

```

An inference violation detection algorithm for N collaborative users

We can use the above greedy algorithm to efficiently decide to either answer or deny a query request from any team member. We first sort all N members by their inference probability to the sensitive attribute and start with the member having the highest inference probability. We also compute every member's collaborative level to the query requester and determine the max collaborative level. Suppose that the member with the highest inference probability integrates the current query answer adjusted by the maximum collaborative level and still cannot infer sensitive data above the threshold. Then we can stop checking the rest of the team members and answer the query. This is because no other member in the team will be able to make a higher inference. If the member with the highest inference probability integrates the query answer adjusted by his collaborative level to the requester and can infer the sensitive data above or equal to the threshold, then we can stop checking and deny this query. Otherwise, we continue on to another member with the next highest inference probability until a decision can be made.

15.7.2 An Example of Inference Violation Detection for Collaborative Users

A set of data sources for transportation and related facilities is available for mission planning. Due to the classified nature of the data sources, users can only access limited amounts of information. Malicious users want to identify whether a specific facility is capable of launching certain classified missions. However, the users are unable to access all the information that is required to derive the conclusion. Therefore, they apply inference techniques to infer the otherwise inaccessible information. In the following example, we shall demonstrate how our detection system prevents these users from accessing the relevant information.

As shown in Fig. 15.11, the transportation and facility data sources consist of four types of information: 1) takeoff and landing activities and capacity of the airport, such as parking_area, runway_length, runway_width, aircraft landing requirements etc.; 2) equipment handling capacity, such as weapons, human experts, loading facility; 3) airport cargo and warehouse capacity and activities, such as daily cargo handling capacity, warehouse space; and 4) fueling storage and consumption. Based on these entities and attributes, we can derive the dependency links between attributes, the schema links that join different aspects of information together for each airport. Furthermore, based on the following set of semantic queries:

- Query1: which airports *can land* a C-5 cargo plane?
- Query2: which airports have the loading facility that *can load* weapon type HC-1?
- Query3: which aircraft *can carry* weapon type HC-1?

We can extract the semantic knowledge for “can land,” “can load” and “can carry” for semantically linking the related attributes, as shown in Fig. 15.11.

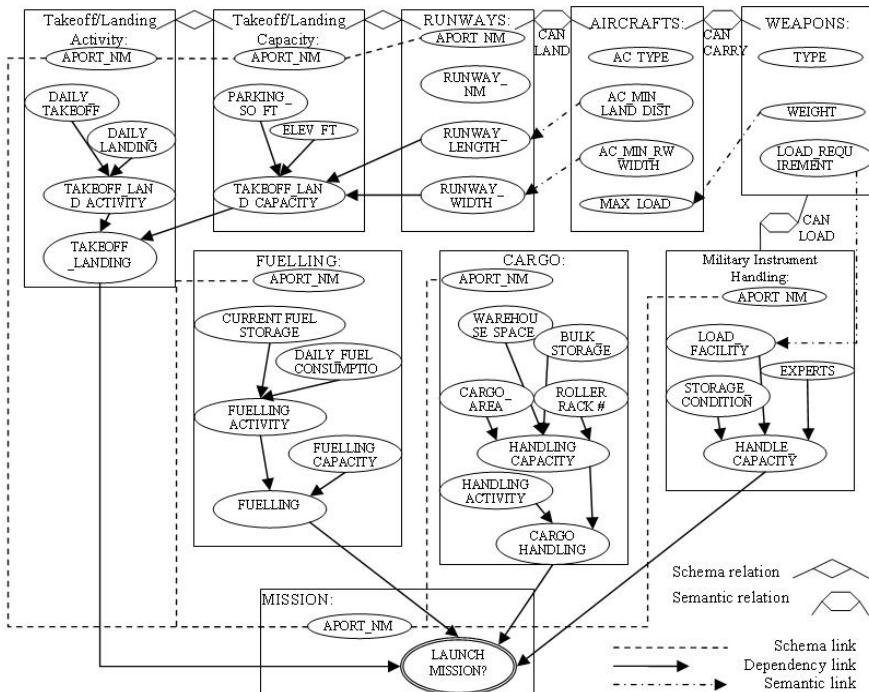


Fig. 15.11. The SIM for a transportation mission planning example

Based on these dependency links, schema links and semantic links, a reduced semantic inference model was constructed (Fig. 15.11) to represent all the possible inference channels between data attributes for all the entities. There are four data sources which yield four main inference channels to the mission entity: takeoff_landing to launch_mission; fueling to launch_mission; cargo_handling to launch_mission and handle_capacity to launch_mission. Each of the main inference channels consists of many local inference channels. To carry out the inference computation, we need to generate a semantic inference graph (SIG) by substituting the specific instance to the semantic inference model. The corresponding Bayesian network representation mapped from the SIG for airport “LAX” is shown in Fig. 15.12.

Let “Launch Mission?” be the sensitive attribute. The violation detection module examines each user’s past query log, as well as the current query request. The probability to infer “Launch Mission?” in the Bayesian network will be evaluated before answering each query. If answering the current query increases the certainty of inferring the sensitive attribute above the pre-specified threshold, then the query will be denied. Let the pre-specified threshold for launch mission be 60%, and the users have prior knowledge of: 1) Aircraft C-5 can land in airport LAX; 2) Airport LAX can load weapon HC-1. When user A poses the sequence of queries shown in Table 15.1, each query answer will update his certainty of inferring the “Launch Mission? = yes” (as shown in the table). The same is true for user B when he poses the queries in Table 15.2.

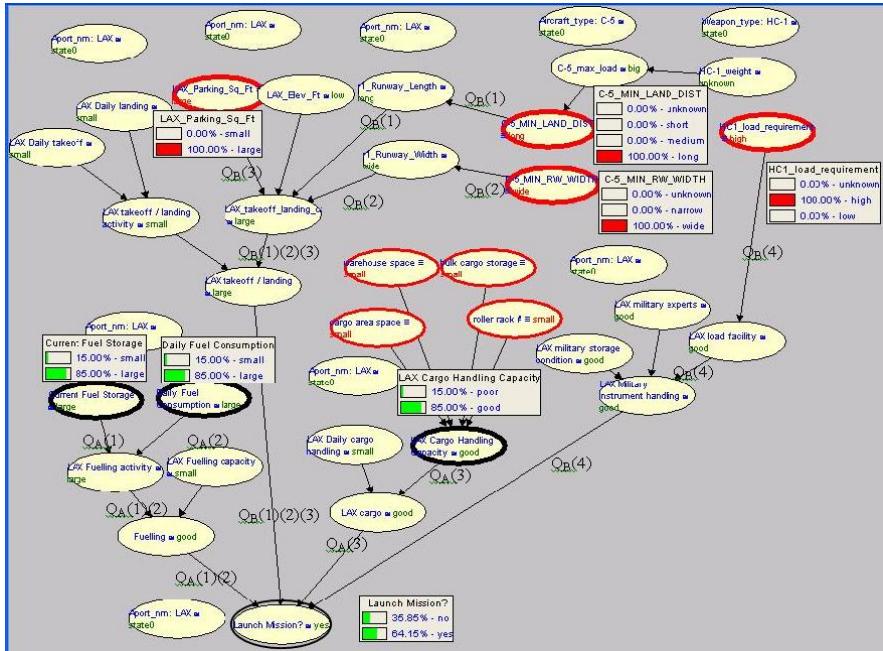


Fig. 15.12. The Bayesian network for the mission planning example. The bold nodes represent user queried attributes. Knowledge from the query answers can be accumulated along the inference channels towards the sensitive attribute. The inference channels used by each query are labeled by its query identifier. The collaborative level from user A of 85% are shown in the probability distribution boxes of $Q_A(1)$, $Q_A(2)$ and $Q_A(3)$. When all the seven queries are answered, user B can infer the sensitive attribute (shown in double ellipses) with a certainty of 64.15%.

Tables 15.1 and 15.2 are assuming that user A and user B do not collaborate. Neither A or B are getting enough information to infer the sensitive attribute above the threshold, thus all the queries are answered. However, based on the questionnaires collected from these two users, we notice that they are collaborators with an 85% collaborative level from B to A for this specific “airport mission” task. Therefore, the

Table 15.1. The inference probability of “Launch Mission? = yes” after answering user A’s queries. The probabilities are computed from the Bayesian network in Fig. 15.12.

Query Set of A $Q_A(i)$	Answer _i	$\Pr(\text{Launch_mission?} = \text{yes} \text{answer}_1, \dots, \text{answer}_i)$
What is current_fuel_storage of airport LAX?	large	52.01%
What is current_fuel_consumption of LAX?	large	56.50%
What is cargo_handling_capacity of LAX?	good	59.80%

Table 15.2. The inference probability of “Launch Mission? = yes” after answering user B’s queries. The probabilities are computed from the Bayesian network in Fig. 15.12.

Query Set of B $Q_B(i)$	Answer _i	$\Pr(\text{Launch_mission?}=\text{yes} \mid \text{answer}_1, \dots, \text{answer}_i)$
What is the min_land_dist of aircraft C-5?	long	50.31%
What is the min_rw_width of aircraft C-5?	wide	50.85%
What is the parking_area_sq_ft of airport LAX?	large	52.15%
What is the load_requirement of weapon type HC-1?	high	57.15%

knowledge from their query answers can be combined for collaborative inference. If we examine their query set Q_A and Q_B on the SIM, we notice that they do not have overlapping inference channels. This is because Q_A focused on the fueling and cargo storage of the airport while Q_B focused on the takeoff and landing activities and military instrument handling. Thus, users A and B belong to the “non-overlap inference channels” case as shown in Fig. 15.9. We can directly integrate their knowledge from query set answers based on their collaboration relation. Thus user B can integrate Q_A into Q_B and adjust the inference probability using their respective collaborative level, as shown in Table 15.3.

Table 15.3. User B integrates user A’s query set Q_A into his own query set Q_B . The Bayesian network is used to compute the inference probability in accordance with the posed query sequence and adjusted by the collaborative levels of the corresponding answers.

Integrated Query Set of B (i)	An-swer _i	Collabo-rative Level t_i (%)	$\Pr(\text{Launch_mi-ssion?}=\text{yes} \mid t_1*\text{answer}_1, \dots, t_i*\text{answer}_i)$
$Q_B(1)$ What is min_land_dist of aircraft C-5?	long	100%	50.31%
$Q_B(2)$ What is min_rw_width of aircraft C-5?	wide	100%	50.85%
$Q_A(1)$ What is current_fuel_storage of LAX?	large	85%	52.39%
$Q_A(2)$ What is current_fuel_consumption of LAX?	large	85%	55.54%
$Q_B(3)$ What is parking_area_sq_ft of LAX?	large	100%	56.84%
$Q_A(3)$ What is cargo_handling_capacity of LAX?	good	85%	59.15%
$Q_B(4)$ What is load_requirement of weapon HC-1?	high	100%	64.15%

From Table 15.3, we note that the last query posed by user B will infer sensitive information with probability higher than the pre-specified threshold of 60%. Therefore, $Q_B(4)$ should be denied by the violation detection module. In contrast, in the non-collaborative case as shown in Tables 15.1 and 15.2, all the above queries can be answered.

15.8 Collaboration Level

As defined in Sect. 15.7, a collaboration level (CL) is a metrics that can be used to estimate the collaborative inference by a group of malicious users. CL consists of two factors: information authoritativeness and communication channel fidelity. In this section we shall first conduct a set of two experiments to validate the premise of the proposed metrics and then propose a technique to estimate the parameters.

15.8.1 Experimental study of Collaboration Level

Since both information authoritativeness and fidelity are user-sensitive, we conducted an experiment using the students in one of the authors' classes as test subjects. The experiment was used as homework for the class to ensure their participation. Further, to ensure that the experiment was carried out honestly, the experiment outcome would not affect their grades. However, the winner would receive extra credit. A web interface was developed for our inference test bed so that students could pose queries directly to the test bed and receive the answers. The goal of the experiment was to study how information authoritativeness and communication fidelity affect the CL.

Before posing queries for inference, each student needed to register in the system and fill in the necessary background information, including their age, gender, major, year in school, courses taken, GPA, skills, interests, teamwork ability, social activities, friends in the class, etc. The information gave us clues about the information authoritativeness and communication fidelity of the test subjects. Based on the collected background information, we divided the class into five teams of four students to perform collaborative inference. The first team consisted of Ph.D. students with good knowledge in the database area, which should have provided good authoritativeness. The second team members were good friends, which provide good communication fidelity. The other three teams are randomly formed. In the first test, the teams were given the SIG structure based on the database, but not the SIG parameters (CPTs) nor the threshold of the security attribute. Then we allowed each team to pose a fixed number (e.g. four in this experiment) of queries to infer the security attribute. The test bed computed their inference probability after each member posed the query. The system denied the query request if the posed queries exceeded the threshold. The four members in the team could collaborate in the best way possible to increase their inference probability of the security attribute and avoid denial. In order to monitor the team communication, each team also reported its communication methods in selecting the queries, such as email, meeting, voting after debate on query selection.

Fig. 15.13 displays the maximum inference probability for the five teams. In experiment 1a, we observed that team2 reached the highest inference probability. This is because they held meetings to discuss strategies of posing queries and voted if there

was disagreement; therefore, their queries leveraged on each other to get better inference. Team1 asked a set of effective queries that spanned the inference channels based on their knowledge of the SIG structure; therefore they also performed well. This result reveals that both communication fidelity and information source authoritativeness play very important roles in determining collaboration effectiveness.

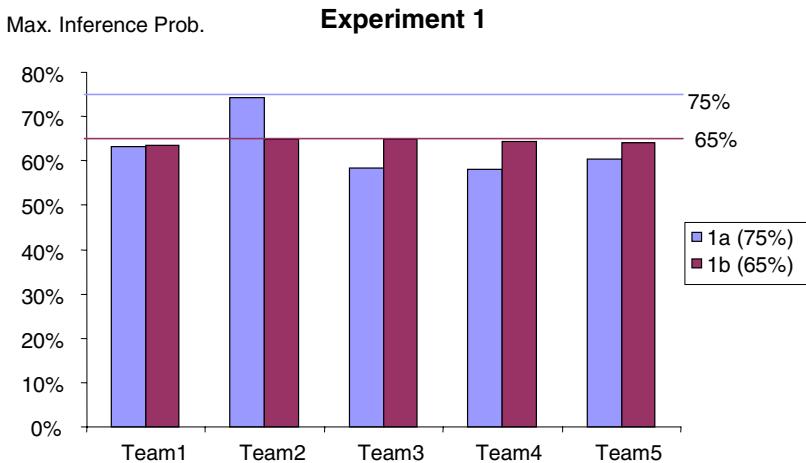


Fig. 15.13. The inference results of five collaborative teams. In experiment 1a, the teams were given the SIG structure but without the parameters (CPTs) and the threshold (75%) of the security node. In experiment 1b, the teams were given both the SIG structure and the CPTs and the inference threshold (65%) of the sensitive node.

To study how information source authoritativeness affects the CL, we repeated the same experiment in 1b. This time, we let all the teams know the SIG structure, CPTs and the threshold value of the security attribute. With the same fixed number of queries, we noticed that with the additional knowledge of the CPTs and threshold of the security attribute, all the teams improved their maximum inference probabilities. In fact, they were able to ask better queries to improve their inference probability as close to the threshold as possible. This experimental result reveals that the information source authoritativeness (in this case, the quality of queries) does affect CL in the positive way.

In the first set of experiments, we noted that both information source authoritativeness and communication fidelity played a key role in CL and therefore improved inference probability. This motivated us to study these two factors more closely in the second experiment. More specifically, we wished to investigate the collaboration effectiveness under controlled communication fidelity environment. This experiment was carried out in a manner similar to that of experiment 1, except it was conducted in another graduate class in the following quarter. Because of the small class size, we divided the students into two teams, with three members in first two teams. In order to control the communication fidelity, we assigned the communication method for each team. The first team was allowed to have “full collaboration.” Members were required

to meet and discuss query strategies and exchange their query answers in making their selection of queries. The second team was allowed “limited collaboration” in which they could only email each other with their query answers but could not discuss strategy. In terms of authoritativeness, Team2 had more task-specific knowledge (in this case, Bayesian inference) than Team1.

In this experiment, the two teams were given the SIG, its parameters (CPTs) but not the threshold of the sensitive attribute. As shown in Fig. 15.14, although Team2 was restricted in communication, they could still pose effective queries based on their task-specific knowledge to achieve a higher inference probability than Team1. On the other hand, although the first team could freely communicate and discuss, their lack of task-specific knowledge caused their failure in posing the most aggressive queries and, in turn, hurt their inference results. We notice that the second team’s knowledge of the task overcame the limitation of their collaboration, and they outperformed the first team.

The above set of experimental results validates our premise that information authoritativeness and communication fidelity are two key parameters that affect collaboration performance.

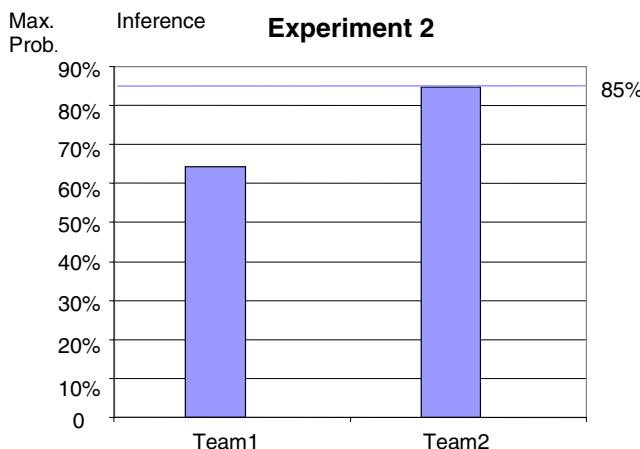


Fig. 15.14. Maximum inference probability for Experiment 2. The teams were given both the SIG structure and the CPTs, but not the inference threshold of the security node which was set at 85%.

15.8.2 Estimation of Collaboration Level

Since the collaboration level consists of two main components: information authoritativeness and communication fidelity, it can be expressed as $CL = g(A, F, e_A, e_F)$, where A is the authoritativeness of the information source, F is the communication fidelity, e_A is the error in estimation of the authoritativeness, and e_F is the error in estimating the fidelity. There are many works on trust negotiation in peer-to-peer networks that are related with e_A and e_F estimation. The interested reader should refer

to papers [1, 7, 32, 48, 49, 45, 42, 46, 15 and 37]. We shall now outline some approaches to estimate A and F under the case that $e_A = e_F = 1$, which corresponds to estimating parameter values in a trusted and honest community.

Estimation of Authoritativeness A: Information authoritativeness represents the task-specific knowledge which can be based on the provider's profile, such as reputation, education, profession, and experience that is related to the task. This authoritativeness can also be enhanced by information derived from the user social network structure . For example, if many individuals (especially highly authoritative ones) indicate user u_i as their friend, then u_i has a significant impact on others and therefore has a higher authority. The link-based similarity analysis (such as page rank) can be used to derive the authority of people [38]. Information authoritativeness can be derived from questionnaire answers and with additional correlated information from web documents or available social network information. In general, Information authoritativeness may be based on a set of multiple attributes that are related to the specific task. The estimation can become more complex and would be beyond the scope of this chapter.

Estimation of Fidelity F: For a given task, the communication fidelity of two collaborators can be based on their closeness on a set of task-sensitive attributes. Based on the registration questionnaire, we can derive their closeness by the similarity computed from the selected attribute set. Additional information from other available sources, such as their web sites and their social networks , can also be used to enhance the estimation.

After estimating A and F , we need to combine them to derive the collaboration inference. One way is to assume they have a linear relationship and thus can be combined linearly. We can then learn the coefficients (such as the weights of A and F) via a set of training data with similar tasks and users.

15.9 Robustness in Inference Detection

Usually security experts or database administrators have some idea of the required level of protection for each security attribute , but they can hardly give a quantitative measurement to describe such protection requirements. Further, in a large database system, the dependency relationship between the security attribute and other attributes is complicated. The inference towards security attribute computed from a Bayesian network depends on both the network topology (qualitative attribute dependencies) and the parameter of the network (conditional probabilities). If a small variation of a given parameter can trigger the inference probability to exceed the threshold, then the inference detection may not satisfy the robustness requirements. This motivates us to find a methodology to systematically quantify the robustness of the threshold for inference violation detection .

Sensitivity measures the impact of small changes in a network parameter on a target probability value or distribution [34]. In other words, a small change in the more sensitive attribute will cause a large impact on the inference probability . Therefore, the sensitivity values of attributes in the network provide an insight to the robustness of inference with respect to the changes in attribute parameter value. In this section, we propose to use the sensitivity analysis results to adjust the security threshold.

15.9.1 Sensitivity Definition

“Sensitivity values are partial derivatives of output probabilities with respect to parameters being varied in the sensitivity analysis . They measure the impact of small changes in a network parameter on a target probability value or distribution” [34]. More formally, for a function $f(x)$, the quantity:

$$\lim_{(x-x_0) \rightarrow 0} \frac{(f(x) - f(x_0))/f(x_0)}{(x - x_0)/x_0} \quad (15.3)$$

is typically known as the *sensitivity* of f to x at x_0 , which is the ratio of relative change in output probability over the relative change in the parameter, where x_0 is the initial value of X . If we consider the function to be the probability of security node Y given the change of attribute node X , then the sensitivity for attribute X at probability x_0 in a given network N with the initial probability of the security node y_{init} can be represented as:

$$Sen(X, Y) = \lim_{x_0, N, y_{init}} \left| \frac{(y - y_0)/y_0}{(x - x_0)/x_0} \right| = \lim_{\Delta x \rightarrow 0} \left| \frac{\Delta y / y_0}{\Delta x / x_0} \right| \quad (15.4)$$

The initial probability of the security node is the probability of Y at the state when the set of evidence was given in the network. y_{init} represents the initial probability of Y , which is different from y_0 that represents the probability of Y when $X = x_0$.

According to this definition, in a Bayesian network , if minor changes to an attribute node’s probability can result in a significant change in the output probability of the security node, then this attribute node is considered highly sensitive.

15.9.2 Adjust Security Threshold by Attribute Sensitivity Analysis

To compute the sensitivity of attributes in an inference network, we first identify all inference channels toward each security node so that the sensitivity values for the attributes along the inference channels can be computed. The inference channels include channels coming into the security node and those going out of the security node. For those out-going inference channels , we can treat them as if the channels are coming into the security node by reversing the edges along such channels and revising the corresponding conditional probabilities. This is because, in terms of inference, the security node can be thought of as the “sink” of all information. Regardless of whether the attribute is the ancestor or descendent of the security node, the inference is always from the attribute towards the security node. Thus, we can compute the attribute sensitivities on both in-coming and out-going inference channels .

In a large-scale network, because of the large number of attributes, it is time-consuming to compute the sensitivity value for each attribute on the inference channels . However, for two attribute nodes on the same inference channel , the node that is closer to the security node is more sensitive than the node that is farther from the security node at the same probability value. This difference of sensitivity value between closer and farther nodes is intuitive as closer nodes generally contain more

sensitive information and are more influential on the security node than that of farther nodes. More specifically, the farther node influences the security node through the inference channel which includes the closer node. Therefore, the amount of change at the farther node has the equivalent effect of inferring the security node as a smaller (or equal) amount of change at the closer node. For example, in the inference channel in Fig. 15.15(a), the closest attribute to security node “*Launch Mission?*” is “*Fueling*.” Based on Eq. 15.4, the sensitivity of “*Fueling*” is greater than the sensitivity of its parents “*LAX_Fueling_Activity*” for all x_0 , as shown in Fig. 15.15(b). Similarly, the sensitivity of “*LAX_Fueling_Activity*” is greater than the sensitivity of “*Daily_Fuel_Consumption*.”

By this property, we know that for each inference channel, the attribute node closer to the security node is more sensitive than the farther attribute nodes. So to measure the maximum sensitivity of each inference channel, we only need to consider the sensitivity value of the attribute node on the channel that is closest to the security node to represent the sensitivity of the entire inference channel. Thus, in the entire network, we only need to check the sensitivity of the attributes on an inference channel that is one hop away from the security node.

Each value of the security node is protected by a threshold. For example, we need threshold for “*Launch_Mission=Yes*” and another threshold for “*Launch_Mission=No*” so that the malicious user cannot infer the exact value of this attribute above the thresholds. When the data administrator proposes a threshold value based on the required protection level, he/she can check the sensitivity values of the closest attributes on each inference channel. If one of these inference channels is too sensitive which means that a small change in the attribute value can result in exceeding the threshold, then the threshold needs to be tightened to make it less sensitive.

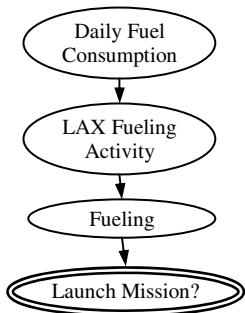


Fig. 15.15(a). A portion of the inference channel in the Bayesian network from the example

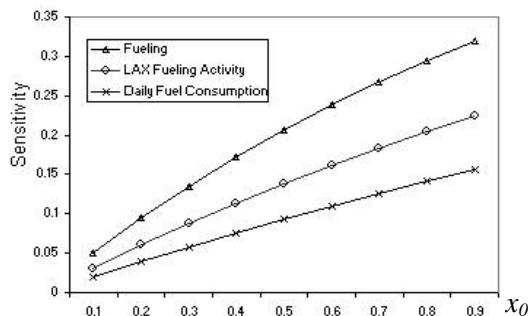


Fig. 15.15(b). The sensitivity of corresponding attribute nodes in (a) to the security node at selected initial values x_0 .

15.10 Conclusion

In this chapter we present a technique that prevents users from inferring sensitive information from a series of seemingly innocuous queries. Compared to the deterministic inference approach in previous works, we include non-deterministic relations into

inference channels for query-time inference detection . Specifically, we extract possible inference channels from probabilistic data dependency, the database schema and the semantic knowledge and construct a semantic inference model (SIM). The SIM links represent the possible inference channels from any attribute to the set of pre-assigned sensitive attributes. The parameters of attributes in SIM can be computed in polynomial time in terms of the rows and columns of the relational table. The SIM is then instantiated by specific instances and reduced to a semantic inference graph (SIG) for inference violation detection at query time. To reduce computation complexity for inference, the SIG can be mapped into a Bayesian network so that available Bayesian network tools can be used for evaluating the inference probability along the inference channels . Therefore, our proposed approach can be scalable to large systems.

When a user poses a query, the detection system will examine his/her past query log and calculate the probability of inferring sensitive information from answering the posed query. The query request will be denied if it can infer sensitive information with the probability exceeding the pre-specified threshold. We find that the Bayesian network is able to preserve the structure of the inference channels , which is very useful in providing accurate as well as scalable inference violation detection .

In the multiple-user inference environment, the users can share their query answers to collaboratively infer sensitive information . Collaborative inference is related to the collaboration level as well as the inference channels of the user-posed queries. For inference violation detection , we developed a collaborative inference model that combines the collaborators' query log sequences into inference channels to derive the collaborative inference of sensitive information .

Sensitivity analysis of attributes in the Bayesian network can be used to study the sensitivity of the inference channels . Our study reveals that the nodes closer to the security node have stronger inference effect on the security node. Thus sensitivity analysis of these close nodes can assist domain experts to specify the threshold of the security node to ensure its robustness.

User profiles and questionnaire data provide a good starting point for learning collaboration levels among collaborative users . However, gathering such information is complicated by the fact that the information may be incomplete and incorrect. In addition, the accuracy of such information is task-specific and user-community sensitive. We have constructed a test bed on the inference violation detection system to study the collaboration level for multiple collaborative users . Our preliminary study reveals that information source accuracy and communication fidelity play key roles in the collaboration level . Further research in this area is needed.

References

1. Aberer, K., Despotovic, Z.: Managing Trust in a Peer-2-Peer Information System. In: Proceedings of the tenth international conference on Information and knowledge management, Atlanta, Georgia, USA, October 05–10 (2001)
2. Chavira, M., Allen, D., Darwiche, A.: Exploiting Evidence in Probabilistic Inference. In: Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI), pp. 112–119 (2005)

3. Chan, H., Darwiche, A.: A Distance Measure for Bounding Probabilistic Belief Change. In: Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI), pp. 539–545. AAAI Press, Menlo Park (2002)
4. Chan, H., Darwiche, A.: When Do Numbers Really Matter? *Journal of Artificial Intelligence Research* 17, 265–287 (2002)
5. Chan, H., Darwiche, A.: Reasoning about bayesian network classifiers. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence, pp. 107–115 (2003)
6. Chan, H., Darwiche, A.: Sensitivity analysis in Bayesian networks: From single to multiple parameters. In: Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence (UAI), Arlington, Virginia, pp. 67–75. AUAI Press (2004)
7. Cornelli, F., Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., Samarati, P.: Choosing reputable servants in a P2P network. In: Proceedings of the 11th international conference on World Wide Web, Honolulu, Hawaii, USA, May 07–11 (2002)
8. Chavira, M., Darwiche, A.: Compiling bayesian networks with local structure. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1306–1312 (2005)
9. Chen, Y., Chu, W.W.: Database Security Protection via Inference Detection. In: Mehrotra, S., Zeng, D.D., Chen, H., Thuraisingham, B., Wang, F.-Y. (eds.) ISI 2006. LNCS, vol. 3975. Springer, Heidelberg (2006)
10. Chu, W.W., Chen, Q., Hwang, A.Y.: Query Answering via Cooperative Data Inference. *Journal of Intelligent Information Systems (JIIS)* 3(1), 57–87 (1994)
11. Chu, W.W., Yang, H., Chiang, K., Minock, M., Chow, G., Larson, C.: CoBase: A Scalable and Extensible Cooperative Information System. *Journal of Intelligence Information Systems (JIIS)* 6 (1996)
12. Date, C.J.: An Introduction to Database Systems, 6th edn. Addison-Wesley, Reading (1995)
13. Darwiche, A.: Recursive conditioning. *Artificial Intelligence* 126(1-2), 5–41 (2001)
14. Darwiche, A.: Class notes for CS262A: Reasoning with Partial Beliefs, UCLA (2003)
15. Duma, C., Shahmehri, N., Caronni, G.: Dynamic trust metrics for peer-to-peer systems. In: Proceedings of the Sixteenth International Workshop on Database and Expert Systems Applications, pp. 776–781 (2005)
16. Dechter, R.: Bucket elimination: A unifying framework for probabilistic inference. In: Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence (UAI), pp. 211–219 (1996)
17. Dechter, R.: Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence* 113, 41–85 (1999)
18. Delugach, H.S., Hinke, T.H.: Wizard: A Database Inference Analysis and Detection System. *IEEE Trans. Knowledge and Data Engineering* 8(1), 56–66 (1996)
19. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning Probabilistic Relational Models. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, August 1999, pp. 1300–1307 (1999)
20. Farkas, C., Jajodia, S.: The Inference Problem: A Survey. *SIGKDD Explorations* 4(2), 6–11 (2002)
21. Farkas, C., Toland, T.S., Eastman, C.M.: The Inference Problem and Updates in Relational Databases. In: Proceedings of the 15th IFIP WG11.3 Working Conference on Database and Application Security, pp. 181–194 (2001)
22. Garvey, T.D., Lunt, T.F., Quain, X., Stickel, M.: Toward a Tool to Detect and Eliminate Inference Problems in the Design of Multilevel Databases. In: Proceedings of the 6th Annual IFIP WG 11.3 Working Conference on Data and Applications Security (1992)
23. Getoor, L., Taskar, B., Koller, D.: Selectivity Estimation using Probabilistic Relational Models. In: Proceedings of the ACM SIGMOD (Special Interest Group on Management of Data) Conference (2001)

24. Getoor, L., Friedman, N., Koller, D., Pfeffer, A.: Learning Probabilistic Relational Models. In: Dzeroski, S., Lavrac, N. (eds.) *Relational Data Mining*. Springer, Heidelberg (2001)
25. He, J., Chu, W.W., Liu, Z.: Inferring Privacy Information From Social Networks. In: Mehrotra, S., Zeng, D.D., Chen, H., Thuraisingham, B., Wang, F.-Y. (eds.) *ISI 2006. LNCS*, vol. 3975. Springer, Heidelberg (2006)
26. Heckerman, D., Mamdani, A., Wellman, M.P.: Real-world applications of Bayesian networks. *Communications of the ACM* 38(3), 24–68 (1995)
27. Heckerman, D.: A Tutorial on Learning with Bayesian Networks. Technical Report, Microsoft Research (1996)
28. Hinke, T.H., Delugach, H.S.: Aerie: An Inference Modeling and Detection Approach for Databases. In: *Proceedings of the 6th Annual IFIP WG 11.3 Working Conference on Data and Applications Security* (1992)
29. Hinke, T.H., Delugach, H.S., Wolf, R.: A Framework for Inference-Directed Data Mining. In: *Proceedings of the 10th Annual IFIP WG 11.3 Working Conference on Data and Applications Security* (1996)
30. Jensen, F.V.: *An Introduction to Bayesian Networks*. Springer, New York (1996)
31. Jensen, F.V., Lauritzen, S.L., Olesen, K.G.: Bayesian updating in recursive graphical models by local computation. *Computational Statistics Quarterly* 4, 269–282 (1990)
32. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The Eigentrust algorithm for reputation management in P2P networks. In: *Proceedings of the 12th international conference on World Wide Web*, Budapest, Hungary, May 20–24 (2003)
33. Kautz, H., Selman, B., Shah, M.: The Hidden Web. *AI magazine* (1997)
34. Laskey, K.B.: Sensitivity Analysis for Probability Assessments in Bayesian Networks. *IEEE Transactions on Systems, Man and Cybernetics* 25, 909–909 (1995)
35. Lauritzen, S.L., Spiegelhalter, D.J.: Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems (with Discussion). *Journal of the Royal Statistical Society, Series B* 50(2), 157–224 (1988)
36. Lee, W., Stolfo, S.J., Chan, P.K., Eskin, E., Fan, W., Miller, M., Hershkop, S., Zhang, J.: Real Time Data Mining-based Intrusion Detection. In: *Proceedings of DISCEX II* (June 2001)
37. Marti, S., Garcia-Molina, H.: Taxonomy of trust: Categorizing P2P reputation systems. *Computer Networks* 50(4), 472–484 (2006)
38. Page, L., Brin, S.: The anatomy of a large-scale hypertextual web search engine. In: *Proceedings of the Seventh International World-Wide Web Conference*, Brisbane, Australia (April 1998)
39. Pearl, J.: *Probabilistic Reasoning in Intelligence Systems*. Morgan Kaufmann, San Mateo (1988)
40. Pearl, J.: Bayesian Networks, Causal Inference and Knowledge Discovery. UCLA Cognitive Systems Laboratory, Technical Report (R-281), March. Second Moment (March 1, 2001)
41. SamIam, Automated Reasoning Group, UCLA, <http://reasoning.cs.ucla.edu/samiam/>
42. Shafiq, B., Bertino, E., Ghafoor, A.: Access control management in a distributed environment supporting dynamic collaboration. In: *Workshop On Digital Identity Management, Proceedings of the 2005 workshop on Digital identity management* (2005)
43. Thuraisingham, B.M., Ford, W., Collins, M., Keeffe, J.O.: Design and Implementation of a Database Inference Controller. *Data Knowl. Eng.* 11(3), 271 (1993)
44. Toland, T.S., Farkas, C., Eastman, C.M.: Dynamic Disclosure Monitor ($D^{<\text{Superscript}>2</\text{Superscript}>}$ Mon): An Improved Query Processing Solution. In: *The Secure Data Management Workshop* (2005)
45. Winsborough, W., Li, N.: Safety in automated trust negotiation. In: *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 147–160 (2004)

46. Xiong, L., Liu, L.: Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering* 16(7), 843–857 (2004)
47. Yip, R.W., Levitt, K.N.: Data Level Inference Detection in Database Systems. In: PCSFW: Proceedings of the 11th Computer Security Foundations Workshop (1998)
48. Yu, T., Winslett, M.: A Unified Scheme for Resource Protection in Automated Trust Negotiation. In: Proceedings of the 2003 IEEE Symposium on Security and Privacy, May 11–14, 2003, p. 110 (2003)
49. Yu, T., Winslett, M.: Policy migration for sensitive credentials in trust negotiation. In: Proceedings of the 2003 ACM workshop on Privacy in the electronic society, Washington, DC, October 30 (2003)
50. Zhang, G., Chu, W.W., Meng, F., Kong, G.: Query Formulation from High-Level Concepts for Relational Databases. User Interfaces to Data Intensive Systems (UIDIS) 1999, 64–75 (1994)
51. Zhang, N.L., Poole, D.: Exploiting Causal Independence in Bayesian Network Inference. *Journal of Artificial Intelligence Research* 5, 301–328 (1996)
52. Zhang, N.L., Poole, D.: A simple approach to bayesian network computations. In: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI), pp. 171–178 (1994)

Questions for Discussions

1. Discuss what are the benefits of using probabilistic approach as compare with the deterministic approach for handling database security?
2. Discuss the types of knowledge needed to construct the semantic inference model.
3. Discuss how to acquire the conditional probability table (CPT) for each attribute from the data sources and give an example.
4. Collaboration level can be used as a metric to measure the percentage of useful information transfer from the source to the recipient in a social network. Provide a method (with an example) for determining the collaboration level.
5. Considering the introducing of virtual node in resolving inconsistent belief as shown in Fig. 15.11(c), in addition to collaborator A, suppose user B has another collaborator C who also informs B $X=x$. What will be user B's belief on X based on both A and C's input?
6. A robust threshold is defined as any small change of attribute values will not cause large impact on the security node. Discuss how can sensitivity analysis be used to improve the robust threshold of the security node.

Suspect Vehicle Identification for Border Safety

Siddharth Kaza and Hsinchun Chen

Department of Management Information Systems,
University of Arizona, USA

Abstract. Border safety is a critical part of national and international security. The U.S. Department of Homeland Security searches vehicles entering the country at land borders for drugs and other contraband. Customs and Border Protection (CBP) agents believe that such vehicles operate in groups and if the criminal links of one vehicle are known then their border crossing patterns can be used to identify other partner vehicles. We perform this association analysis by using mutual information (MI) to identify vehicles that may be involved in criminal activity. CBP agents also suggest that criminal vehicles may cross at certain times or ports to try and evade inspection. In a partnership with border-area law enforcement agencies and CBP, we include these heuristics in the MI formulation and identify suspect vehicles using large-scale, real-world data collections. Statistical tests and selected cases judged by domain experts show that the heuristic-enhanced MI performs significantly better than classical MI in identifying pairs of potentially criminal vehicles. The techniques described can be used to assist CBP agents perform their functions both efficiently and effectively.

16.1 Introduction

The commitment of the scientific community in helping the world respond to security challenges became evident after September 11, 2001. The U.S. National Research Council's report on "Making the Nation Safer: The Role of Science and Technology in Countering Terrorism" recommends that "a strategic long-term research and development agenda should be established to address counterterrorism-related areas [7]." Border safety has been identified as a critical component in the fight against transnational crime. The national strategy for homeland security [28] calls for the creation of "smart borders" that provide greater security through better intelligence, coordinated national efforts and unprecedented international cooperation. The report also emphasizes that information systems are the foundations to improve the security infrastructure.

The U.S. Department of Homeland Security (DHS) monitors vehicles entering and leaving the country, recording their license plates with a date and time of crossing using license plate readers. Customs and Border Protection (CBP) agents search vehicles for drugs and other contraband. Thorough checks are done for vehicles on watch lists (known as target/suspect vehicles) and on random vehicles as well. This process is time consuming and if the waiting times become too long, the flow of people, vehicles and commerce is impaired. So agents at the border are under pressure to balance security needs with operational efficiency.

CBP field agents and analysts believe that vehicles involved in illegal activity (especially smuggling) operate in groups. When one vehicle approaches the check-point, the others wait and join the line only if the vehicle crosses into the U.S. successfully. This ensures that the others can turn back if the vehicle before them is inspected and caught. We believe that if the criminal links of one vehicle in a group are known, then the group's crossing patterns and frequency can be used to identify other partner vehicles. So, law enforcement data can be used as a good anchor to perform such analysis and identify high-risk suspect vehicles. In a previous study [26] we found that criminal associations of vehicles crossing the border may be recorded in local law enforcement databases in border-area jurisdictions. However, CBP does not always have access to criminal records of vehicles and sometimes lacks the methods to efficiently perform such analysis on millions of crossing vehicles.

We perform this association analysis by using mutual information (MI) to identify pairs of vehicles crossing together and potentially involved in criminal activity. In initial experiments [19] we had found that the use of MI was a promising solution to this problem. In this chapter we describe a method to modify the MI formulation to incorporate domain heuristics. Domain experts (CBP agents, police detectives and analysts) and our previous study [20] suggest that groups of criminal vehicles may cross at certain times during the day or cross at different ports to try and evade inspection. It is difficult to identify these heuristics with border crossing information alone since it does not contain clear indications of criminal history or possible intent. We use law enforcement information from border-area jurisdictions to identify times and ports that criminal vehicles prefer and incorporate this knowledge in the MI formulation. The new formulations are likely to help CBP agents identify better quality target vehicles more efficiently.

The following section presents a comprehensive review of previous studies using association rule mining and mutual information. The third section presents a case study that explains the modifications needed to include domain heuristics and showcases the application of modified mutual information to real-world datasets. Finally we conclude and provide the reader with some pertinent online resources and discussion questions.

16.2 Literature Review

In this section we review previous studies that have used association rule mining to find associations between seemingly unrelated objects. We also discuss the previous uses and modifications of the concept of mutual information.

16.2.1 Association Rule Mining

Association rule mining is a popular data mining and pattern discovery technique used to find relationships between different entities in database records [15,32]. It was motivated by decision support problems faced by retail organizations where they required information on which items their customers were likely to buy together [1,36]. The direct applicability of association mining algorithms to business problems and the ease of understanding the results have made them a common pattern discovery method [15].

An association rule is a relationship of the form $A \rightarrow B$ (a person who buys A is likely to buy B), where A is the antecedent item-set and B is the consequent item-set. The antecedent and consequent item-sets can contain multiple items. $A \rightarrow B$ holds in a transaction set T with confidence ‘ c ’ if $c\%$ of transactions in T that contain A , also contain B . $A \rightarrow B$ holds with support ‘ s ’ if $s\%$ of transactions in T contain both A and B . To find associations between two item-sets, the association mining algorithms identify all relationships (rules) that have support and confidence greater than user-specified thresholds.

One of the first algorithms proposed for association mining was the AIS algorithm (named after the initials of the authors) [1] which was followed by the more commonly used Apriori algorithm [3]. In recent years, many more algorithms were developed that were focused at specialized tasks under the umbrella of association rule mining. These specialized algorithms have primarily focused on three aspects:

- *Enhancing the definition of association rules* [15]: Newer definitions of association rules have included quantitative association rules [34], generalized association rules [33], and multiple-level association rules [12].
- *Enhancing or replacing support and confidence measures*: Studies have suggested general constraints [35], maximal frequent item-sets [41], and various other flavors of the confidence measure like all-confidence and any-confidence [29].
- *Improving the performance of association mining algorithms*: Since association mining algorithms usually operate in high volume and high data-dimensionality environments, various techniques have focused on improving their performance. Many studies have replaced the support and confidence (like those in the previous category) with other more efficient measures. Other studies have used parallel processing techniques to improve the performance [2, 17] and proposed new data structures to support fast item-set generation [39].

Association rule mining has been applied to problems in many domains including ‘market basket’ data [1, 3], web log analysis (to identify online user behavior) [27], network intrusion detection [23], gene regulatory network extraction (to identify cause-effect relationships between genes) [5], recommender systems (for predicting future purchases) [24] and law enforcement [6]. Some previous studies [13, 16] have modified association rule formulations to include domain heuristics. Hilderman, et al. [13] suggested new item-set measures to replace the commonly used support measure that were more practical and useful for market basket analysis. Since classic association mining algorithms can handle only Boolean data, Huang, et al. [16] modified the technique to handle variables with three values and used it to extract large scale gene regulatory networks.

16.2.2 Mutual Information

Mutual Information is an information theoretic measure that can be used to identify interesting co-occurrences of objects. It can be considered a subset of association rule mining with 1-item antecedent and 1-item consequent item-sets. There are other more general forms of MI that include multiple item antecedent and consequent item-sets, however in this chapter we employ the more commonly used MI formulation that deals with 1-item item-sets. The earliest definitions of MI were given by Claude and

Weaver [9] and Fano [10]. It was defined as the amount of information provided by the occurrence of an event (y) about the occurrence of another event (x). They formulated it as:

$$I(x; y) = \log_2 \frac{P(x, y)}{P(x)P(y)}$$

Intuitively, this concept measures if the co-occurrence of x and y ($P(x, y)$) is more likely than their independent occurrences ($P(x).P(y)$). This formula is referred to as the classical mutual information in the rest of the chapter.

The MI measure has been used in many problem domains. It works well for phrase extraction from text documents. This is because text documents can be considered as a set of events (words), and the probability of the occurrence of a word can be calculated over the entire document. Previous studies in this area have used MI to study association between words in English texts to identify commonly occurring phrases [8, 14]. It has also been used for key phrase extraction from Chinese texts [30].

Pantel, et al. [31] used MI to match database columns containing similar information. They found that their *SIFT* model which was based on MI performed well in identifying schema correspondences across comparable datasets. In the bioinformatics domain, MI has been used to extract protein motif patterns from sequences [37] and identify building blocks of proteins from biomedical abstracts [38]. Mutual information scores have also been used for feature selection [4], extraction [22] and analyzing user preference [18].

Work on extending or modifying the classical MI measure to add domain heuristics includes studies in natural language processing. Magerman and Marcus [25] modified the MI measure (bi-gram) to include n-grams. In bioinformatics: Wren [40] extended the measure to calculate transitive MI scores for biological associations. In feature selection Fleuret [11] used conditional MI to identify important features for image classification.

Border-crossing records can be considered as a stream of text (license plates) ordered by the time of crossing. So, MI can be used to identify frequent co-occurrence between a pair of vehicle crossings. If one vehicle in the pair has a criminal record, some inferences may be made about the second vehicle if they cross together frequently. In a previous study [19] we found that the time and port of crossing may be important heuristics for improving the performance of MI in this problem domain. We propose to use conditional probability to include these domain heuristics in the MI formulation.

16.3 Case Study: Including Time and Port Heuristics in Mutual Information

In this study [21] we use MI to identify pairs of vehicles crossing together and potentially involved in criminal activity. The study also modifies the MI measure to incorporate domain heuristics like time and port of crossing.

16.3.1 Study Testbed

The testbed includes datasets obtained from various law enforcement agencies in the Tucson, AZ metropolitan region. These agencies include the Tucson Police Department (TPD), Pima County Sheriff's Department (PCSD), and many other smaller police agencies in the area. Data from these agencies is referred to as *police data*. In addition, the study also used data from the Tucson Customs and Border Protection. The TPD and PCSD datasets include information on police incidents over 16 years (1990-2005). The incidents include information on individuals and vehicles that are involved in illegal activity in southern Arizona. Some key statistics of these datasets is shown in Table 16.1.

CBP data includes information on vehicles crossing the border between Arizona and Mexico at six ports of entry. This data includes the license plate, state, date, port, and time for non-commercial vehicle crossings from March 2004 to October 2005. Details of this dataset are shown in Table 16.2.

Table 16.1. Key statistics of data from agencies in the Tucson metropolitan region

	TPD	PCSD	Others
Date Range	1990-2005	1990-2005	1990-2005
Vehicles	629,039	614,455	456,832

Table 16.2. Key statistics of CBP border crossing data

Date range	2004 – 2005
Recorded crossings	11 million
Number of vehicles	2 million

16.3.2 Study Design

Prior to presenting the research design we first define the terms *criminal vehicle* and *police contact*. A criminal vehicle is a vehicle that has been suspected, arrested, impounded, or has a warrant (with its occupant) for crimes that include narcotics (sale, possession, etc.), violence (homicide, aggravated assault, armed robbery, etc.), larceny and theft (property, vehicles, etc.), and other serious crimes in the Tucson metropolitan region. Police detectives and analysts consider these crimes and roles (suspect, arrestee) as strong indications of involvement in criminal activity. A vehicle that has had a police contact is one that is recorded in the law enforcement databases; this may be for serious crimes (as listed above) or for other activities that may include forgery and counterfeiting, suspicious activity, and others. Vehicles with police contacts are also referred to as *potentially criminal* vehicles in this study. These definitions are used in the description of the design and the evaluation process.

CBP agents check all vehicles entering the U.S. at various ports of entry. Majority of the vehicles are subjected to ‘primary checking’ which includes verifying the admissibility of the occupants and some checks for obvious presence of contraband. Primary checking usually takes between 1-2 minutes. In addition to primary checking, some vehicles with alerts on them are assigned to ‘secondary checking’ which

includes a thorough check of the vehicle and an interview of the passengers that may take 10 minutes or longer. The vehicles that are selected for secondary checking are referred to as *suspect vehicles* in this study. The target vehicles are selected based on various criteria. These include vehicle alerts from federal systems like FBI's NCIC (National Crime Information Center), random selection, and prior knowledge and intuition of CBP agents.

As mentioned before, CBP agents believe that vehicles involved in cross border crime operate in groups. These vehicles look out for each other and cross frequently within a small time frame. To identify interesting pairs of vehicles that cross the border together we use the *time of crossing* and the *port of crossing* as heuristics to enhance mutual information. The time of crossing heuristic suggests that vehicle pairs that cross during certain times of the day or night are more interesting than others. Domain experts and our previous study [19] suggest that criminal vehicles regularly cross at odd times during the night to exploit the cover of darkness. The port of crossing heuristic suggests that vehicle pairs that cross at certain ports of entry are more interesting than others. This is because criminal vehicles tend to use ports that are in areas that are desolated, near areas where contraband can be easily disposed of, or areas that are not under the radar of border patrol and CBP agents. It can be seen that both these heuristics are fluid in nature, i.e., criminal vehicles may change their time and port of crossing according to the current situation in law enforcement. The mutual information measure modified to include the time heuristic is referred to as '*MIT*', port heuristic is referred to as '*MIP*' and classical mutual information (without heuristics) is referred to as '*MIC*'.

Training Dataset and Heuristic Calculation

To evaluate the performance of MIT, MIP and MIC, the CBP border crossing records were divided into training and testing datasets. This was done using a 2/3 – 1/3 hold out procedure. The training dataset contained 7.4 million ($\approx 2/3$ of total) crossing records from March 2004 to November 2004.

To calculate the time heuristic the day was divided into six time periods corresponding to office travel (5am – 10am), travel for lunch (10am – 2am), night time (8pm – 12pm, 12pm – 5am), and others. These time periods were defined with the help of domain experts. For each of these time periods the ratio of vehicles with police contacts to the total number of crossings was calculated. This value was used to inform the mutual information score between vehicles in a given time period.

The port heuristic was calculated in a similar fashion. For each of the six ports the ratio of vehicles with police contacts to the total number of crossings was calculated. This value was used to inform the mutual information score between vehicles crossing at a given port.

Testing Dataset

The testing data contained 3.6 million ($\approx 1/3$ of total) crossings from November 2004 to Oct 2005. Police data and the border crossings in the testing dataset were used to identify two sets of vehicles:

Set A: 251 criminal vehicles that had been arrested or suspected for narcotics sale in the Tucson metropolitan region since January 2003.

Set B: All the border crossing vehicles crossing *within one hour* of each vehicle in *Set A* at the same port and in the same direction (i.e., both vehicles are either entering the U.S. or exiting it).

MIT, MIP, and MIC were calculated between paired vehicles in *Set A* and *Set B*. The vehicle pairs with high scores were considered potential suspect vehicles.

Evaluation Procedure

The potential suspect vehicles identified were evaluated by measuring their overlap with police datasets. This was done by measuring the number of vehicles with police contacts that were contained in the set of potential suspect vehicles. The overlap with the entire Tucson metropolitan region (that includes TPD and PCSD) dataset was measured. The number of potentially criminal vehicles (vehicles with police contacts) identified by MIT, MIP, and MIC were compared to each other to ascertain the performance of the modified measures. Since the aim of CBP is to target potentially criminal vehicles, a greater number of such vehicles in the target vehicle set indicate a higher quality result. In addition to statistical tests, selected cases judged by domain experts were also used in the comparison. The illustrative cases included a detailed evaluation of the criminal links of target vehicle pairs identified by the three measures.

Classical Mutual Information (MIC) Formulation

The classical mutual information score between any two vehicles is defined as:

$$MIC(A, B) = \log_2 \frac{P(A, B)}{P(A)P(B)} \quad (16.1)$$

Here *A* is a vehicle in *Set A*, and *B* is a vehicle in *Set B*. $P(A)$ and $P(B)$ are the probabilities of the vehicles *A* and *B* crossing the border, these are calculated from the border crossing datasets. $P(A, B)$ is the probability of *B* crossing within one hour of *A*, this is calculated based on the number of times *A* and *B* are seen crossing together.

Modified Mutual Information with Time Heuristics (MIT) and Port Heuristics (MIP)

In the MIT and MIP formulation, we use conditional probability to modify the definition of $P(A)$, $P(B)$ and $P(A, B)$.

$P'(A)$: Probability that vehicle *A* crosses the border and has a police contact.

$P'(B)$: Probability that vehicle *B* crosses the border and has a police contact.

$P'(A, B)$: Probability that vehicles *A* and *B* cross the border together and have police contacts.

Thus, a high $MI'(A,B)$ (based on Eq. 16.1) indicates that the vehicles are likely to cross the border and potentially commit crimes.

Given this, we can now modify the classical MI formulation to include the time heuristic: Let $P_c(a)$ be the probability that vehicle ' a ' has contact with the police, and $P_b(a)$ be the probability that ' a ' crosses the border. The probability of vehicles with police contacts crossing during the six time periods is calculated using historical information in the police databases. So, we can obtain $P_c(V|t)$, which is the probability that *any* vehicle V in time period t ($1 \leq t \leq 6$) will have a contact with the police.

Now, by definition of $P'(A)$,

$$P'(A) = \sum_{t=1}^6 P[(A_b \text{ and } A_c) | t] \quad (16.2)$$

In the above equation A_b refers to vehicle A crossing the border, and A_c refers to vehicle A having contact with the police. This is summed over all six time periods to obtain $P'(A)$. The equation reduces to

$$P'(A) = \sum_{t=1}^6 P_b(A|t)P_c(V|t) \quad (16.3)$$

since the probability of a vehicle crossing the border and having police contact are independent (so they are multiplied to obtain $P'(A)$). In addition, A is replaced by V in the second term since the probability that a vehicle in time period t has a police contact is the same for all vehicles in that time period. So basically the above process utilizes historical information (about crime) in the police datasets as a weight to modify $P'(A)$. Similar derivations can be used to obtain $P'(B)$, $P'(A,B)$ and thus $MIT(A,B)$ as shown in the following equations:

$$\begin{aligned} P'(B) &= \sum_{t=1}^6 P[(B_b \text{ and } B_c) | t] = \sum_{t=1}^6 P_b(B|t)P_c(V|t) \\ P'(A,B) &= \sum_{t=1}^6 P[((AB)_b \text{ and } (AB)_c) | t] = \sum_{t=1}^6 P_b((AB)|t)P_c(V|t)P_c(V|t) \\ MIT(A,B) &= \log_2 \frac{P'(A,B)}{P'(A)P'(B)} \end{aligned} \quad (16.4)$$

A similar derivation can be used to obtain MIP.

16.3.3 Experimental Results

To ascertain whether law enforcement information can be used to identify potential criminal vehicles, we first measured the overlap between border-crossing vehicles and police records in border-area jurisdictions. There were 66,185 border crossing vehicles that had police incident records in Tucson metropolitan area datasets. The number suggests that many vehicles crossing the border have incidents recorded in local law enforcement databases. This is a positive sign since it allows us to identify target vehicles at the border by exploring their criminal links. The existence of an overlap is also important for the calculation of heuristics based on law enforcement information.

Comparative Evaluation of MIT, MIP and MIC

Mutual information scores (MIT, MIP, and MIC) were calculated for 410,079 pairs of vehicles (the first vehicle from *Set A* and the second from *Set B*). To statistically compare the three measures, the number of police contact vehicles in Set B identified by each was counted. The results are shown in Fig. 16.1.

On the X-axis are top-n pairs (n ranging from 10-2500) of vehicles ordered by their MIT, MIP, and MIC scores. On the Y-axis is the number of vehicles with police contacts identified by the three measures. For instance, fourteen vehicles of the top-100 vehicles identified by MIP had previous police contacts. As can be seen in all three graphs, both MIT and MIP consistently identified more potentially criminal vehicles (vehicles with prior police contacts) than MIC. MIP also identified more potentially criminal vehicles than MIT. MIP identified 206 vehicles among the top 2500 pairs that had prior police contacts, i.e., 8.2% of the vehicles identified had police contacts. The average number of border crossing vehicles that have police contacts in the Tucson metropolitan region is 3.3%. Thus, the performance of MIP is better than a random selection of target vehicles from the set of border crossing vehicles. For statistical testing, thirty data points (ranging from top 5 to 3500 pairs) were taken for each of the measures and a t-test was done for the differences in the mean number of potentially criminal vehicles identified. Since all the samples had unequal variances, the Smith-Satterthwaite t-test procedure was used. It was found that MIT performed significantly better (at 95% level) than MIC. MIP also performed significantly better than MIC (at 99% level) in identifying potentially criminal vehicles. Thus, heuristic-enhanced mutual information performed better than classical mutual information.

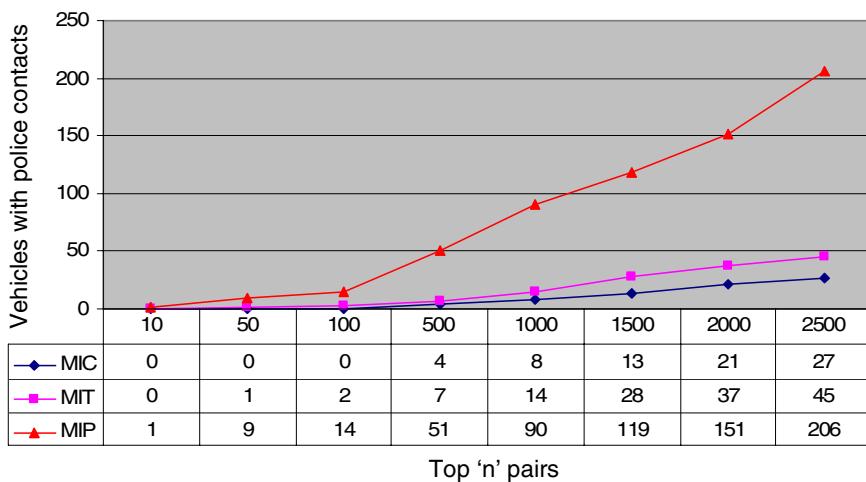


Fig. 16.1. Number of vehicles with police contacts identified by MIT, MIP and MIC

Even though the top-n pairs contained many potentially criminal vehicles (true-positives), they also included many other vehicles that had no past criminal records (false-positives). This might not look promising in other domains, but has positive connotations in this one. It suggests that many of the vehicles postulated to be

potentially criminal by the algorithms were not known to have police records before. So the new measures can be used to identify new potentially criminal vehicles that can be targeted for further inspection at the border. The low number of police contacts might also be a result of properties of the datasets. Most of the border crossing vehicles in our datasets may be headed for Phoenix, AZ and surrounding areas and thus their activity (if any) will be recorded in those police datasets. A more accurate test of the algorithms is possible if those datasets were available.

An Example Suspect Vehicle Pair

Fig. 16.2(a) shows the temporal crossing patterns of a vehicle pair (Vehicles C & D) that received a high MIT score. On the X-axis are the dates when the vehicles were

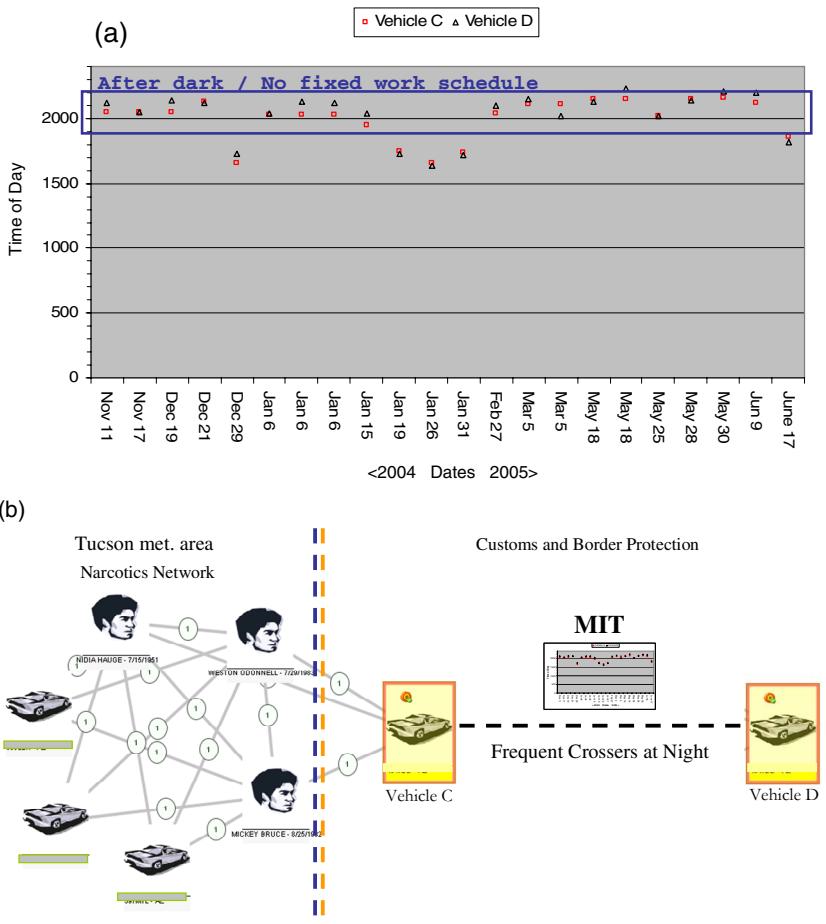


Fig. 16.2. (a) A vehicle pair identified by mutual information with time heuristics (MIT) (b) The activity of the vehicles as recorded in Tucson met. area police databases

seen crossing together. On the Y-axis are the times of crossing (0-2400). Vehicle C crossed 51 times in a 7 month period, out of which it crossed 22 times with Vehicle D. As can be seen in the figure, this vehicle pair crossed together frequently, but in addition all the crossings were after dark and did not follow a standard work schedule. This example and other like it show that MIT identifies cases that are more likely to be considered suspicious by domain experts. Since, Vehicles C & D are interesting with respect to the frequency and times of crossing together; we explored their police contacts further. Fig. 16.2(b) shows the criminal links of Vehicle C and Vehicle D. Vehicle C was found to have strong connections to a narcotics network in the Tucson metropolitan area. It had links to other people and vehicles that had been arrested / suspected for narcotics sales and possession in the region. These connections suggested that the vehicle might be an active member of a narcotics sale and smuggling ring. Domain experts also suggested that viewing the vehicles' border crossing activity in this context made them a candidate for further investigation. Both MIT and MIP identified many other such examples.

16.4 Conclusion

Exploring the criminal links of border crossing vehicles in local law enforcement databases can be used to enhance border security. In this chapter we described the use of mutual information to identify pairs of border crossing vehicles that may be involved in criminal activity. We found that mutual information can be used to identify high quality potential suspect vehicles that may warrant more inspection at the border. In addition, we concluded that the mutual information measure modified to include domain heuristics like time and port of crossing performs better than classical mutual information in the identification of potentially criminal vehicles. The method can be used to assist Customs and Border Protection agents to perform their functions both effectively and efficiently. If validated in the field, then the methodology can be used in border-states and help in enhancing national and international security. Since the methodology is based on information sharing, the results may also encourage law makers to formulate policies to increase co-operation among agencies.

Acknowledgements

This research was supported in part by the NSF Digital Government (DG) program: "COPLINK Center: Information and Knowledge Management for Law Enforcement" #9983304, NSF Knowledge Discovery and Dissemination (KDD) program: "COPLINK Border Safe Research and Testbed" #9983304, NSF Information Technology Research (ITR) program: "COPLINK Center for Intelligence and Security Informatics" #0326348, Department of Homeland Security (DHS) through the "BorderSafe" initiative #2030002 and NSF Regional Information Sharing and Collaboration #0636422.

We thank our BorderSafe project partners: Tucson Police Department, Pima County Sheriff's Department, Tucson Customs and Border Protection, ARJIS (Automated Regional Justice Information Systems), San Diego Super Computer Center (SDSC), SPAWAR, Department of Homeland Security, and Corporation for National

Research Initiatives (CNRI). We also thank Homa Atabakhsh, Hemanth Gowda and Yuan Wang of the AI Lab at the University of Arizona, Tim Petersen and Chuck Violette of the Tucson Police Department, and Ron Friend of Tucson Customs and Border Protection for their contributions to this research.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in large Databases. In: ACM SIGMOD Conference on Management of Data. ACM Press, New York (1993)
2. Agrawal, R., Shafer, J.C.: Parallel Mining of Association Rules. IEEE Transactions on Knowledge and Data Engineering 8, 962–969 (1996)
3. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: 20th Int. Conf. Very Large Data Bases, VLDB (1994)
4. Battiti, R.: Using Mutual Information for Selecting Features in Supervised Neural Net Learning. IEEE Transactions on Neural Networks 5 (1994)
5. Berrar, D., Dubitzky, W., Granzow, M., Eils, R.: Analysis of Gene Expression and Drug Activity Data by Knowledge-Based Association Mining. In: Critical Assessment of Microarray Data Analysis Techniques (CAMDA 2001). Springer, Berlin (2001)
6. Brown, D.E., Hagen, S.: Data Association Methods with Applications to Law Enforcement. Decision Support Systems 34, 369–378 (2003)
7. Chen, H.: Intelligence and Security Informatics for International Security. Springer, Berlin (2006)
8. Chruch, K.W., Hanks, P.: Word Association Norms, Mutual Information, and Lexicography. Computational Linguistics 16, 22–29 (1990)
9. Claude, S., Weaver, E.: The Mathematical Theory of Communication. University of Illinois Press, Chicago (1949)
10. Fano, R.M.: Transmission of Information. MIT Press, Cambridge (1961)
11. Fleuret, F.: Fast Binary Feature Selection with Conditional Mutual Information. Journal of Machine Learning Research 5 (2004)
12. Han, J., Fu, Y.: Mining Multiple-level Association Rules in Large Databases. IEEE Transactions on Knowledge and Data Engineering 11, 798–805 (1999)
13. Hilderman, R.J., Carter, C.L., Hamilton, H.J., Cercone, N.: Mining Association Rules from Market Basket Data using Share Measures and Characterized Itemsets. In: Wu, X., Kotagiri, R., Korb, K.B. (eds.) PAKDD 1998. LNCS, vol. 1394. Springer, Heidelberg (1998)
14. Hindle, D.: Noun Classification from Predicate-Argument Structures. In: 28th Conference on Association for Computational Linguistics (1990)
15. Hipp, J., Guntzer, U., Nakhaeizadeh, G.: Algorithms for Association Rule Mining - A General Survey and Comparison. SIGKDD Explorations 2, 58 (2000)
16. Huang, Z., Li, J., Su, H., Watts, G.S., Chen, H.: Large-scale Regulatory Network Analysis from Microarray Data: Modified Bayesian Network Learning and Association Rule Mining. Decision Support Systems (in press, 2006)
17. Javed, A., Khokhar, A.: Frequent Pattern Mining on Message Passing Multiprocessor Systems. Distrib. Parallel Databases 16, 321–334 (2004)
18. Jung, S.Y., Hong, J.H., Kim, T.S.: A statistical model for user preference. Knowledge and Data Engineering, IEEE Transactions 17, 834–843 (2005)

19. Kaza, S., Wang, T., Gowda, H., Chen, H.: Target Vehicle Identification for Border Safety using Mutual Information. In: Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems. IEEE Press, New York (2005)
20. Kaza, S., Wang, Y., Chen, H.: Target Vehicle Identification for Border Safety with Modified Mutual Information. In: Mehrotra, S., Zeng, D.D., Chen, H., Thuraisingham, B., Wang, F.-Y. (eds.) ISI 2006. LNCS, vol. 3975. Springer, Heidelberg (2006)
21. Kaza, S., Wang, Y., Chen, H.: Enhancing Border Security: Mutual Information Analysis to Identify Suspect Vehicles. Decision Support Systems (in press, 2006)
22. Kwak, N., Choi, C.H.: Feature extraction based on ICA for binary classification problems. Knowledge and Data Engineering, IEEE Transactions 15, 1374–1388 (2003)
23. Lee, W., Stolfo, S.J.: Data Mining Approaches for Intrusion Detection. In: 7th USENIX Security Symposium (1998)
24. Lin, W., Alvarez, S.A., Ruiz, C.: Efficient Adaptive-Support Association Rule Mining for Recommender Systems. Data Mining and Knowledge Discovery 6, 83–105 (2002)
25. Magerman, D.M., Marcus, M.P.: Parsing a Natural Language using Mutual Information Statistics. In: Proceedings of the Eighth National Conference on Artificial Intelligence. AAAI Press, Menlo Park (1990)
26. Marshall, B., Kaza, S., Xu, J., Atabakhsh, H., Petersen, T., Violette, C., Chen, H.: Cross-Jurisdictional Criminal Activity Networks to Support Border and Transportation Security. In: Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems. IEEE Press, New York (2004)
27. Mobasher, B., Jain, N., Han, E.H., Srivastava, J.: Web mining: Pattern discovery from world wide web transactions. Department of Computer Science, University of Minnesota, Minneapolis (1996)
28. Office of Homeland Security, National Strategy for Homeland Security (2002)
29. Omiecienski, E.R.: Alternative Interest Measures for Mining Associations in Data-bases. IEEE Transactions on Knowledge and Data Engineering 15, 57–69 (2003)
30. Ong, T., Chen, H.: Updateable PAT-Tree Approach to Chinese Key Phrase Extraction Using Mutual Information: A Linguistic Foundation for Knowledge Management. In: Second Asian Digital Library Conference, Taipei, Taiwan (1999)
31. Pantel, P., Philpot, A., Hovy, E.: Aligning Database Columns using Mutual Information. In: Proceedings of the 6th National Conference on Digital Government Research (dg.o), Los Angeles, DGRC (2005)
32. Song, M., Rajasekaran, S.: A Transaction Mapping Algorithm for Frequent Item-sets Mining. IEEE Transactions on Knowledge and Data Engineering 18, 472–481 (2006)
33. Srikant, R., Agrawal, R.: Mining Generalized Association Rules. In: 21st Conf. on Very Large Databases, Zurich, Switzerland (1995)
34. Srikant, R., Agrawal, R.: Mining Quantitative Association Rules in Large Relational Tables. In: ACM SIGMOD Conf. on Management of Data, Canada (1996)
35. Srikant, R., Vu, Q., Agrawal, R.: Mining Association Rules with Item Constraints. In: Proceedings of the 3rd International Conf. on KDD and Data Mining (KDD 1997), Newport Beach, CA (1997)
36. Stonebraker, M., Agrawal, R., Dayal, U., Neuhold, E., Reuter, A.: The DBMS Research at Crossroads. In: The VLDB Conference, Morgan Kaufmann, San Francisco (1993)
37. Tao, T., Zhai, C.X., Lu, X., Fang, H.: A study of statistical methods for function prediction of protein motifs. Applied Bioinformatics 3, 115–124 (2004)
38. Weisser, D., Klein-Seetharaman, J.: Identification of Fundamental Building Blocks in Protein Sequences Using Statistical Association Measures. In: ACM Symposium on Applied Computing, Nicosia, Cyprus (2004)

39. Woon, Y.K., Ng, W.K., Lim, E.P.: A support-ordered trie for fast frequent itemset discovery. *Knowledge and Data Engineering, IEEE Transactions* 16, 875–879 (2004)
40. Wren, J.D.: Extending the Mutual Information Measure to Rank Inferred Literature Relationships. *BMC Bioinformatics* 5 (2004)
41. Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New Algorithms for Fast Discovery of Association Rules. In: 3rd International Conf. on KDD and Data Mining (KDD 1997), Newport Beach, CA (1997)

Online Resources

1. Sasabe, Sonora, has turned into a smugglers' haven - Arizona Daily Star:
<http://www.azstarnet.com/sn/border/95936>
This is an interesting news article describing smuggling and border crossing at a port in Arizona. The article sheds light on why port heuristics may be beneficial in identifying suspect vehicles.
2. U.S. Customs and Border Protection:
<http://www.cbp.gov>
This is the official website of Customs and Border Protection. The website contains details on ports of entry including statistics on border crossing vehicles.
3. Presidential Executive Order 13356:
<http://www.fas.org/irp/offdocs/eo/eo-13356.htm>
This is the executive order of the U.S. President requiring the sharing of information among public safety agencies.
4. University of Arizona Artificial Intelligence Lab, Public safety research:
<http://ai.eller.arizona.edu>

Questions for Discussions

1. Most data mining algorithms, including the one described in this chapter, identify true positives (TP) as well as false positives (FP). How can a trade-off between TP and FP established in this problem domain? Do you think false positives are bad for the problem discussed in this chapter?
2. In recent times, there is increasing focus on developing data-centric methods to aid in national security applications. What -in your mind- are the major obstacles in the design, implementation and success of these methods?
3. Explain the intuition behind the mutual information formulation. Try and substitute numbers to explain how the classical mutual information formula works.
4. Association rule mining is a very common data mining technique. The technique grew out of problems faced by retail organizations. Why do you think association mining has become one of the most popular pattern discovery methods?

Optimization Problems for Port-of-Entry Detection Systems

Endre Boros, Elsayed Elsayed, Paul Kantor, Fred Roberts, and Minge Xie^{*}

Rutgers University, USA

Abstract. The problem of container inspection at ports-of-entry is formulated in several different ways as an optimization problem. Data generated from different analytical methods, x-ray detectors, gamma-ray detectors and other sensors used for the detection of chemical, biological, radiological, nuclear, explosive, and other illicit agents are often relied upon to make critical decisions with regard to the nature of containers presented for possible inspection and the appropriate response mechanism. Several important questions related to the utilization and co-ordination of multiple sensors for container inspection are discussed. New and efficient algorithms for finding the best inspection strategy, including the optimal sequencing of sensors and optimal assignment of thresholds for interpreting sensor readings, are described. Models and algorithms that can be used by decision makers, allowing them to minimize expected cost of inspection, minimize inspection errors (both false positives and false negatives), and/or maximize the throughput of containers, are outlined.

17.1 Introduction

Finding ways to intercept illicit materials, in particular weapons, destined for the U.S. via the maritime transportation system is an exceedingly difficult task. Practical complications of inspection approaches involve the negative economic impacts of surveillance activities, errors and inconsistencies in available data on shipping and import terminal facilities, and the tradeoffs between costs and potential risks, among others. Until recently, even with increased budget and emphasis, and rapid development of modern technology, only a very small percentage of ships entering U.S. ports have had their cargo inspected. Thus there is a great need to improve the efficiency of the current inspection processes. There has been a series of attempts to develop algorithms that will help us to inspect for and intercept chemical, biological, radiological, nuclear, and explosive agents, as well as other illicit materials. Such algorithms need to be developed with the constraint that seaports are critical gateways for the movement of international commerce. More than 95 percent of our non-North American foreign trade arrives by ship. With “just-in-time” deliveries of goods, the expeditious flow of commerce through these ports is essential. Slowing the flow long enough to

* All five authors were supported by ONR grant number N00014-05-1-0237 and NSF grant number NSFSES 05-18543 to Rutgers University. The authors also thank Sushil Mittal and Richard Hoshino for their helpful comments.

inspect either all or a statistically significant random selection of imports would be economically intolerable.

Potential goals for container inspection include:

- Minimizing inspection errors, both false positive and false negative, subject to constraints on cost of inspections
- Maximizing inspection system “throughput”
- Minimizing total expected cost of inspection.

Here, false positive means a container that has no suspicious contents is rejected or goes through extensive manual examination and false negative means that a container that has suspicious contents is accepted. These criteria are inter-related and it is unlikely that they would be optimized by a single set of parameter values. Thus, we are dealing with a typical *multi-objective programming problem*, and one seeks ways to formulate this precisely and use methods of multi-objective programming to find solutions under different formulations.

As to the data itself, at present there are only two techniques for non-invasively “seeing” into a container, both involving projection and the resultant transmission or reflection of waves: 1) Electromagnetic (EM) waves (radio waves, light, x-rays, gamma rays, etc.) and 2) Material vibration waves (ultrasound). A variety of means exist for converting these waves into images suitable for human inspectors to interpret (James, et al. 2002). In practice, there are only a small number of types of sensors used in port-of-entry inspection protocols. In part this is due to the reality that there are not many different types of inspections that can be carried out short of actually opening a container, and in part it is due to the fact that algorithmic approaches to container inspection run afoul of combinatorial explosions when more than a small number of sensors are used. We will describe new approaches to the container inspection problem that we can hope could lead to almost doubling the number of sensors whose outputs could be considered. Such methods should prepare us well for the time when additional practical detection methods are made available for port-of-entry inspections.

Although the techniques of “seeing” into containers are limited, various types of sensors, testing for the same types of things, are being used in the current practice of port-of-entry inspection. They range from hand-held or portable devices, to heavy equipment, and from counts to numerical measurements to shape (image) classifications. These sensors vary widely in cost, resolution and sensitivity, and they are evolving quickly with innovations leading to both significant decreases in costs and to many new features as well. We rely upon their measurements to make critical decisions with regard to the nature of the containers being inspected and to detect unwanted agents. Interpretation of the output from a sensor is not always straightforward and requires calibration. The interpretation may lead to accepting a container that contains undesirable material (false negative) or may lead to unnecessary inspection of a container that is acceptable (false positive), which results in delays and added cost. Clearly, interpretation of sensor readings has a direct effect on these errors and on the entire inspection process.

Typical techniques transform sensor readings into a real output, and generally thresholds are used to separate *suspicious containers* from *innocent looking* ones. There are many critical issues involved in determining what thresholds to use, such as

noise (in the detector measurements), the sequencing of different sensors, the determination of the threshold levels of the sensors that minimize the probabilities of accepting undesirable containers and rejecting acceptable containers, and the minimization of system delays, among others. These issues must be considered simultaneously in configuring inspection and detection systems and research needs to address these threshold-related issues. We will describe a polyhedral approach to this problem, building on earlier work that formulated a large-scale linear programming model yielding optimal strategies for container inspection under certain assumptions, and we will also describe a dynamic programming approach and a new algorithm that combines the well-known optimization known as the Gradient Descent Method and Newton's Method.

17.2 Model Formulation

In the port-of-entry inspection process, we consider containers (entities) being off-loaded from ships. Containers are inspected and classified according to observations we make regarding their attributes. In the simplest case, each attribute is either “present” (1) or “absent” (0). There are several categories into which we seek to classify entities. In the simplest case, these are positive and negative, 1 or 0, with “0” designating entities that are considered “acceptable” and “1” designating entities that raise suspicion and require special treatment. After each observation, we either classify the entity as 0 or 1 or subject it to another inspection process. Thus, a container can be thought of as corresponding to a string of 0's and 1's, a binary string. Note that we might make a decision about a container before knowing all the terms in this string. The classification can be thought of as a *decision function* F that assigns to each binary string of attributes a category. In this paper, we focus on the case where there are two categories. Thus, F is a *Boolean function*. For instance, consider the Boolean function defined by $F(111) = 1$ and $F(x_1x_2x_3) = 0$ otherwise, where x_1 , x_2 , and x_3 are 0 or 1. This F is the function that classifies an entity as positive if and only if it has all of the attributes. Boolean functions provide the selection logic for an inspection scheme. If F is known, we seek to determine its value by testing the attributes one by one. In a typical case, the attributes are assumed to be independent. We shall describe models where the distribution of the attribute states is known and also where it is not. In particular, in the binary case, we consider the probability p_i that the i^{th} attribute takes on the value 0, and the probability $q_i = 1 - p_i$ that it takes on the value 1. The inspection policy determines the order of testing the container's attributes. At any point, the inspection scheme might tell us to stop inspecting and decide the value of F based on the outcomes of inspections so far. We must make enough observations to allow us to classify the entity and decide whether to designate it as an entity that needs special handling or accept it. The notation we use and assumptions we make about a port-of-entry inspection system can be formalized as follows:

1. There are n inspection stations in the system; each station is used to identify one attribute of the container being inspected. Let x_i be the state of the i^{th} attribute.
2. We think of observations (measurements) as taking place sequentially. After each observation, we either classify the entity being inspected or subject it to

another inspection process. As noted above, the classification of each container is thought of as a decision function F that assigns to each string of attribute values a class C . We focus on the case where there are only two classes, $C = 0$ and $C = 1$, i.e. $F(x_1x_2\dots x_n) = 0$ means the negative class and that there is no suspicion with the container, and $F(x_1x_2\dots x_n) = 1$ means the positive class and that additional special treatment such as manual inspection of the container is required.

3. We assume that we gather information about the probability that the attribute i is present or absent for a group of containers, which are imported by the same company or from the same origin at the same time. This information can be obtained, for example, from inspection history or other sources. As noted, we use p_i to be the probability that the attribute i is absent and q_i to be the probability that the attribute i is present, i.e., $p_i = P(x_i = 0)$, $q_i = P(x_i = 1) = 1 - p_i$, for $i = 1, 2, \dots, n$.
4. The inspection stations are assumed at first to be perfect, which means that the true attributes can always be identified at the corresponding inspection stations without any error. However, later we will assume that the sensor readings are either continuous or categorical measures and consider errors in identification. In particular once “thresholds” are included in our model, we will consider the problem with the possibility that resetting thresholds will lower the probability of misclassifying a “good” container as needing further inspection or a “bad” container as being acceptable.
5. The port-of-entry inspection problem involves three kinds of costs [41] : costs of making observations, costs of false positives, and costs of false negatives. There are many possible ways to calculate the cost of obtaining a sensor reading. For instance, we can break down the cost into two components: unit cost and fixed cost. The *unit cost* is just how much it costs to use the sensor to inspect one item, and the *fixed cost* is the cost of the purchase and deployment of the sensor itself. In many cases, the primary cost is the unit cost since many inspections are very labor intensive. The fixed cost is usually a constant and often does not contribute in optimization equations, so for simplicity we will disregard it. The *inspection cost* c_i is basically the expected cost of making observations for an average container. Stroud and Saeger (personal communication) also assign costs to false positives and false negatives, though especially the latter is very difficult to do. The cost of the former is essentially the cost of labor to manually inspect the container, though more sophisticated analysis would take into consideration the economic cost of delays in shipping. Stroud and Saeger then seek to minimize the sum of inspection costs and false positive and negative costs. Some other researchers also consider such a sum of costs, whereas others concentrate on inspection costs only.

The whole inspection process can be represented as a binary decision tree (BDT), the nodes of which correspond to sensors or decisions (0 or 1), and branches of which correspond to the decision we make after learning the sensors’ readings. For concreteness, we think of a BDT as having a left branch from a sensor node meaning that the sensor gives an outcome 0, and a right branch meaning that it gives an outcome 1.

Decision nodes are all 0 or 1 and are leaves of the tree, i.e., have no outgoing branches. The container inspection problem was considered by Stroud and Saeger [41], who provided a complete enumeration of all possible binary decision trees built from no more than 4 sensors and corresponding to Boolean functions satisfying two assumptions: completeness (all variables are needed) and monotonicity (finding a more suspicious reading on any one sensor must not decrease the probability that the container itself should be inspected). A Boolean function F is *monotone* if given two strings $x_1x_2\dots x_n, y_1y_2\dots y_n$ with $x_i \geq y_i$ for all i , then $F(x_1x_2\dots x_n) \geq F(y_1y_2\dots y_n)$. F is *incomplete* if it can be calculated by finding at most $n-1$ attributes and knowing the value of the input string on those attributes. Stroud and Saeger then modeled sensor outcomes as following independent Gaussian distributions, one for “good” containers and one for “bad” ones, and studied thresholds for container readings above which an inspection outcome of 1 would be reported. They computed approximately optimal thresholds for each sensor (one threshold per sensor) by a non-linear grid-search approach. Their method cannot be extended to more than 4 sensors, due to combinatorial explosion of the number of binary decision trees. For example, for $n = 4$ different sensors as nodes, there are 114 complete, monotone Boolean functions and 11,808 distinct corresponding BDTs. Compare this with 1,079,779,602 BDTs for all Boolean functions, which explains the need for special assumptions such as completeness and monotonicity. For $n = 5$, there are 6,894 complete, monotone Boolean functions and 263,515,920 corresponding BDTs. Even worse: compare 5×10^{18} BDTs corresponding to all Boolean functions. (Counts are from Stroud-Saeger.)

Among important extensions of this work are: doing sensitivity analysis on their results [2]; making more restrictive assumptions about the nature of the binary decision trees [47]; introducing a new and promising polyhedral approach [4]; and broadening the class of binary decision trees considered in order to introduce more computationally-efficient search procedures for optimal inspection strategies [28].

17.3 The Polyhedral Formulation

Boros, et al. [4] extended the work of Stroud and Saeger [41] and formulated a large-scale linear programming model yielding optimal strategies for container inspection. This model is based on a polyhedral description of *all decision trees* in the space of possible *container inspection histories*. The dimension of this space, while quite large, is an order of magnitude smaller than the number of decision trees. This formulation allowed them to incorporate both the problem of finding optimal decision trees and optimal threshold selection for each sensor into a single linear programming problem. The model can also accommodate budget limits, capacities, etc., and one can solve it to maximize the achievable detection rate. Boros, et al. have been able to solve this model for 4 sensors, and branching that allows up to 7 possibly different routing decisions at each sensor (in contrast to the binary routing solved by Stroud and Saeger [41], and implicit in Boolean models) in a few minutes of CPU time, on a standard desktop PC. They are also able to run the model for as many as 7 sensors, when they allow only binary decisions, as in Stroud and Saeger [41]. A challenge is to extend and improve this polyhedral formulation.

To describe the polyhedral approach in more detail, note that to containers passing through the inspection station, one can associate a “history”, which is the sequence of pairs naming a sensor and its reading. Readings at a given sensor fall within a range of values, and we can partition this range into several classes. For instance, if we have sensors a, b, c and we present a grid of partitions for these sensors, consisting of, say, 2 possible readings for sensor a , 3 for sensor b , and 5 for sensor c , then sequences like $(c,4;a,1;b,3)$, $(b,1;a,2)$, etc. are all possible histories. (In the former, we mean that sensor c has a reading in its fourth region, a in its first region, and b in its third region). Let us note that the number of possible histories is smaller by a full exponential order (that is, it is only singly exponential instead of being doubly exponential) than the number of decision trees (utilizing the same branching parameters and same set of sensors). Denoting our terminal decisions by 0 and 1, we can define decision variables $y(H,0)$ and $y(H,1)$ representing, respectively, the (unknown) fraction of containers having history H and final decisions 0 (ok) or 1 (check further with special handling).

Clearly, the equality $\sum_{H,D} y(H,D) = 1$ must hold, where the summation is taken over all histories H and final decisions D . Let us further call an initial segment of a history a pre-history, and denote by $P < H$ the statement that P is an initial segment in H . With this notation we can describe a set of consistency equalities corresponding to the knowledge that at each sensor s and for each reading range r at this sensor the fraction of containers receiving reading r is $g(s,r)$ out of all containers that arrive at s with the same pre-history (where $g(s,r)$ are considered as input parameters, derivable from a physical sensor model, or from past data, etc.). In other words, we must have:

$$g(s,r) * \sum_{\{H,D | (K,s) < H\}} y(H,D) = \sum_{\{H,D | (K,s,r) < H\}} y(H,D) \quad (17.1)$$

for all sensors s , for all pre-histories K not involving sensor s , and for all reading ranges r possible at sensor s .

Boros, et al. [4] prove that the above set of equalities, together with the nonnegativity of the decision variables, describes a polytope, the vertices of which correspond to decision trees. Since many other characteristics of an inspection policy, such as unit inspection cost, detection rate, sensor loads, etc., can all be expressed as linear functions of the decision variables, they are able to formulate various problems related to container inspection as linear programs (see more in [4]).

The preliminary results of Boros, et al. demonstrate that (1) allowing multi-fold discretization of sensor readings provides substantially better results (higher detection rate at a lower cost) with the same set of sensors; (2) optimal strategies in fact involve a mixture of several decision trees, rather than just a *single best* decision tree. There are various ways to extend this work.

17.3.1 Extension to More Sensors and Variables

The current formulation involves a quite large system of linear inequalities, which leads to numerical instability, and with this there are challenges in expanding to a larger number of sensors and larger branching factors at each of these sensors. One could exploit the algebraic structure of this system, and reformulate it to yield a numerically more stable formulation. Developing a column generation technique for this type of problem might also help. Extending the method to systems involving up to 7

to 8 sensors, and as many as 5 to 10 different possible decisions at each of these sensors, would allow us to evaluate the effects of new technologies, budget changes, etc., in realistic sizes.

17.3.2 Selection of Threshold Grids or Partitions

As noted in Sect. 17.2, another important area for further research is the selection of threshold grids. There are many important issues involved in determining what thresholds to use, including sequencing of sensors, the desire to minimize false positive or false negative results, the need to minimize system delays for commerce, and problems caused by noise and errors in measurement. Stroud and Saeger [41] developed a grid-optimization technique for selecting the thresholds. This is incorporated into the large-scale linear programming formulation of Boros, et al. [4] by choosing a grid of possible threshold values, and letting the optimization model select which ones of them are actually utilized in an optimal inspection strategy. The large-scale linear programming formulation can be solved efficiently by standard LP packages for up to 5 sensors and 5 to 7 thresholds per sensor, even if the decision trees involved are not binary. As noted above, the results of Boros, et al. [4] show that multiple thresholds provide substantial improvement at no added cost. They also show that the optimal solution is not a single decision tree, but a convex mixture of several decision trees. Of course, this approach is sensitive to the selection of the initial grid. In particular, computational complexity forces us to start with a relatively coarse grid initially. A challenge is to develop a better method for selecting initial grids. One could use the fact that decisions based on sensor readings do not depend on the actual reading, but rather on the odds ratio of dangerous to innocent container contents corresponding to the actual reading [26]. One could also develop an iterative approach for refining the partitioning of sensor readings, so that a k -fold partition at a given sensor is focused on the most critical sensor readings, as revealed by the model with $k-1$ partitions.

17.4 Different Allowable Topologies for the Decision Trees

The port-of-entry inspection problem can be decomposed into two sub-problems. The first problem deals with the determination of the optimum sequence of inspection or the structure of the inspection decision tree in order to achieve the minimum expected inspection cost; the second problem with the determination of the optimum thresholds of the sensors at inspection stations so as to minimize the cost associated with false positive (false alarm, which results in additional manual inspection) and false negative (failure to identify illicit materials or weapons). The first problem can be formulated and investigated using approaches parallel to those used in the optimal sequential inspection procedure for reliability systems as described by Butterworth [7], Ben-Dov [3], and Cox, et al. [14]. After the sequences of inspection and the structure of the inspection decision tree are determined, one determines the optimum thresholds of the sensors at inspection stations.

The Stroud-Saeger [41] approach to port-of-entry inspection concentrated on limiting the number of decision trees we must consider by making special assumptions about the underlying Boolean function, i.e., that it is complete and monotone. An alternative approach is to make special assumptions about the topology of the decision tree.

17.4.1 Series, Parallel, and Other Topologies

A topology-limiting approach can take advantage of the literature of systems reliability. Here, one considers a multi-component system, where the state of all the components is described by a vector (x_1, x_2, \dots, x_n) , and the operational state of the system is a function $F(x_1, x_2, \dots, x_n)$ of the states of its components. (In our earlier discussion, we spoke of strings $x_1 x_2 \dots x_n$ rather than vectors, but the distinction is not important.) In the so-called testing or diagnosis problem, we want to learn the state of the system (the value of F) by testing some of the components (learning the value of some of the variables x_i). It is assumed that we know the cost c_i of testing component x_i , and we also know the distribution of the values of x_i , for all components $i=1, \dots, n$. The problem is to determine that sequence of tests that minimizes the total expected cost. A simplified version considers only expected cost of inspections, disregarding costs of false positives and negatives. The word “sequence” used here is in fact misleading, since, as we have noted, the real testing strategy is in fact a decision tree, in which the next component to be tested depends on the values we learned for the components tested earlier. Such problems arise in fault testing in systems engineering, medical diagnosis, search problems in data bases, and even in quiz shows (see e.g., Butterworth [7], Chang and Slagle [10], Duffuaa and Raouf [15], Greiner [19], Joyce [21], Kadane [22], Kowalski [24, 25], Nilsson [34], Pohl [35]). Numerous papers in the extensive literature have considered this problem in one or another special case.

For instance, consider series or parallel systems, i.e., those in which components have only working or failing states, and the function F is a simple conjunction or disjunction of its components. More precisely, a *series Boolean function* is a decision function F that assigns the container class “1” if any of the attributes is present, i.e., if $x_i = 1$ for any $i \in \{1, 2, \dots, n\}$, and a *parallel Boolean function* is a decision function F that assigns the container the class “1” if all of the attributes are present, i.e., if $x_i = 1$ for all $i \in \{1, 2, \dots, n\}$.

Consider a container with n independent attributes and described by a series Boolean function. Let the inspection procedure be such that attribute $i+1$ is inspected only if attribute i is found absent, for all attributes $i = 1, 2, \dots, n-1$. If attribute i is present, the inspection of further attributes is halted and the container is subject to special handling/manual inspection. In this situation, the following result holds: For a series Boolean decision function, inspecting attributes $i = 1, 2, \dots, n$ in sequential order is optimum, i.e., minimizes total expected inspection cost, if and only if:

$$c_1 / q_1 \leq c_2 / q_2 \leq \dots c_n / q_n \quad (17.2)$$

In this case, the expected inspection cost is given by

$$C = c_1 + \sum_{i=2}^n \left[\prod_{j=1}^{i-1} p_j \right] c_i \quad (17.3)$$

In other words, the optimal strategy is to test the components in the order of the ratio $c_i/\text{Prob}(x_i=1)$. This strategy is dependent on the sensor configuration. (See e.g., Mitten [32], Butterworth [7], Natarajan [33], Alidaee [1] and many others for proofs.)

It is natural to ask: How can this be generalized for different Boolean decision functions other than series? For more complex k -out-of- n systems, the optimal strategy was determined by Chang, Shi and Fuchs [11]. Series-parallel systems (i.e., systems in which components can hierarchically be grouped into series or parallel subsystems) are also frequently considered in the literature. One natural idea is to test subsystems one-by one, without interruption, in the order of their cost/probability ratios. This was proposed by Joyce [21], and was claimed (mistakenly) to be optimal for series-parallel systems. Ben-Dov [3] showed that this idea is indeed optimal for simple, depth-2 series-parallel systems, while Natarajan [33] proved that it is optimal within those strategies that test subsystems without any interruption. Boros and Ünlüyurt [5] showed that this algorithm can unfortunately fall short of the optimal value, and by an arbitrary factor, for series-parallel systems of depth-3. A generalization was also proposed, and was shown to be optimal for the very special case of systems composed of identical components (see Ünlüyurt [44] for a survey on this topic). Zhang, Schroepfer, and Elsayed [47] have developed general total cost of inspection equations for n sensors in series and parallel configurations. It is natural to seek to develop similar equations for other configurations such as series-parallel, parallel-series, k -out-of- n systems and consecutive k -out-of- n systems. The approach for other complex configurations such as networks of sensors can be based on either a path-tracing approach or cut sets (minimum cut sets) depending on the characteristics of the sensor network. It is likely going to be difficult to obtain closed form expressions but the derived equations can be numerically evaluated through efficient algorithms.

17.4.2 Optimal Threshold Setting

In Sect. 17.3.2, we discussed a polyhedral approach to optimal threshold setting. The problems of optimal threshold setting become much more complex and difficult to solve for a larger number of sensors. An alternative approach to determining threshold levels involves a simplifying assumption about the tree topology. Assuming a “series” topology (looking at one sensor at a time in a fixed order), one can first determine an optimal sequence of sensors. Once an optimum sequencing of sensors is obtained, the threshold level problem is then formulated. Zhang, Schroepfer and Elsayed [47] have used a complete enumeration approach to determine the optimum sequence of inspection stations and the corresponding sensors’ threshold levels to solve problems with up to three sensors in series and parallel systems. The computational time increases exponentially as the number of sensors increases. They are developing efficient algorithms for solving different sensor configurations and hope to be able to utilize the topology to obtain tractable solutions for systems having 6 to 7 sensors without resorting to complete enumeration, through a combinatorial optimization algorithm based on a dynamic programming approach.

17.4.3 Unknown Boolean Function F

A similar, though very different problem, arises in the study of systems reliability when we do not explicitly know the system function F , but we know a way to determine for any proposed function $G(x_1, \dots, x_n)$ the probability that it describes the true state of our system (we call this probability the detection rate). This variant arises in several applications, including medical diagnosis [19], organizing call centers [14], and container inspections at ports [41].

There are several possible ways to formulate problems in this setting (still assuming that we know the cost of testing and probability distributions of the variables). One problem is to find that function G for which the detection rate is the highest, and then to determine the optimal decision tree for “testing its components” (that is, determining the actual value of each of the components) to minimize expected expenses. A more general problem is to consider all of the functions G for which the detection rate is above a given threshold, and minimize the expected testing cost among all these. This particular formulation was considered by Stroud and Saeger [41].

Stroud and Saeger [41] solved the above problems at once, by enumerating all the possible decision trees representing all possible complete, monotone Boolean functions. The difficulty with this approach is that the number of decision trees grows doubly-exponentially with the number of sensors. In particular, they demonstrated that these problems can be handled for up to 4 sensors, but that this approach is not feasible for 5 or more sensors, with current computing technology, due to the intensity of the combinatorial explosion.

Both of the above problem formulations are further complicated in practice by the fact that testing the components typically yields a result on a continuous scale, not a discrete or binary scale. The solution must discretize this (that is, assign some meaning to the readings), changing thus implicitly the functions (or sets of functions) considered by the model. For instance, if we decide to binarize all readings, then one possibility is to use thresholds that divide the range of readings into two, and this leads to the problem of finding the thresholds for which the optimum in the previously considered cases (whichever model we use) is the best. One approach to this problem is to bisect the range of the threshold into two equal regions and then select two threshold levels, the first in the middle of the first region and the second in the middle of the second region. One can then estimate the minimum cost for each and select the level corresponding to the lower cost. A promising approach is to repeat the process by bisecting the range between the selected level and the original level and continue the process until the global minimum cost is obtained. The computational time required for this approach will of course be dependent on the initial range and the “fine” sectioning of the region.

The effectiveness of a partition having a given number of subsets K will be increased if the partition is not defined by thresholds placed on the natural sensor reading scale, but are defined to mark off connected portions of the ROC curve. This curve plots the conditional probability of sending a dangerous container for inspection as a function of the probability that a harmless container will be unnecessarily inspected. This curve rises smoothly as the required posterior odds that a container is “bad” are lowered. With complex sensors, the natural sensor readings corresponding to a particular portion of the ROC curve may be quite widely separated. On the other

hand, statistical theory tells us that such a portion should be treated in the same or very similar ways. Therefore one can anticipate that defining partitions in terms of the ROC curve, rather than in terms of the natural sensor reading, will increase the power of an inspection scheme for any given number of partitions K .

17.4.4 Complete, Monotone Binary Decision Trees

Stroud and Saeger [41] formulate the port-of-entry inspection problem as a sequential decision making problem that involves finding an optimal (least cost) binary decision function. They reformulate this by considering the possible binary decision trees that correspond to that decision function, and seek through the space of possible binary decision trees to find least cost trees realizing each function. The problem becomes rapidly intractable unless special assumptions are made about the binary decision function. As noted earlier, Stroud and Saeger limit their analysis to complete, monotone binary decision functions. They enumerate all complete, monotone Boolean functions and then calculate the least expensive corresponding BDTs under assumptions about various costs associated with the trees. Their method is practical for n up to 4, but not $n = 5$. The problem is exacerbated by the number of BDTs (see counts in Sect. 17.2).

Madigan, et al. [28] generalized the notion of complete, monotonic Boolean functions and defined notions of complete and monotonic binary decision trees. They developed a search algorithm that identifies (locally) optimum (least cost) complete, monotonic binary decision trees that is more efficient than the method developed by Stroud and Saeger, and makes it possible to analyze trees of at least 5 sensors (types of tests). The search method is based on a notion of neighborhood in the space of complete, monotonic binary decision trees, built on work of Chipman, George, and McCullough [12, 13] and Miglio and Soffritti [31].

17.5 Multi-objective Programming Approaches

As mentioned earlier, in algorithmic approaches to inspection at ports of entry, we hope to find the optimal design of (sensor) system configuration and the best sets of threshold levels that can achieve a variety of objectives, such as maximizing inspection system throughput, minimizing the expected cost of inspection per container, and minimizing inspection errors, including both false positive and false negative. These objectives or criteria are interrelated. It is unlikely that they would be optimized by the same alternative parameter values, and there exists some trade-off between the criteria. It is a typical *multi-objective optimization* problem (see, for instance, Eschenauer, et al. [16], Statnikov and Matuso [40], Fonseca and Fleming [17, 18], and Leung and Wang [27], among others). Depending on the application, one can formulate different multi-objective algorithms for the problem. In general, there may be a large number or infinite number of optimal solutions (e.g., optimal sets of threshold values, etc.), in the sense of *Pareto-optimality*. It is desirable to find as many (optimal) solutions as possible in order to provide more choices to decision makers.

The multi-objective problem is almost always solved by combining the multiple objectives into one scalar objective whose solution is a Pareto optimal point for the

original problem. Most algorithms for the problem have been developed in combination with techniques, such as minimizing/maximizing weighted sum of the objective functions, the goal programming method, a normal-boundary intersection method, and multilevel programming, among others. A particularly promising idea is to employ a combination of a goal programming method and a modified method of using weighted sums of the objective functions.

The *goal programming* method [39, 20] is a branch of multiple objective programming. In the goal programming approach, we optimize one objective while constraining the remaining objectives to be less than given target values. One can, for example, set constraints on the inspection cost, false positive rate and false negative rate, and maximize the objective function of throughput. An alternative approach is to use a modified weighted sum approach (a hybrid version of goal programming and the weighted sum approach), where we optimize, for example, a *fitness function* (i.e., weighted sum) of the objective functions under a constraint that both the false negative rate and false positive rate are controlled within their respective tolerance levels. In both the goal programming method and the modified weighted sum approach, by using the constraints, one can avoid the problem of how to subjectively choose appropriate weights for the false positives and false negatives, as discussed in Stroud and Saeger [41]. In a small system with only 2 to 4 sensors, this multi-objective programming problem can be solved by a grid search method similar to that discussed in Zhang, Schroepfer, and Elsayed [47] and Stroud and Saeger [41]; see, also Sects. 17.4.2 and 17.3.2. However, for a large system with more sensors, it is computationally challenging to use this simple enumeration method, since the computational time increases exponentially as the number of sensors increases. Hence, one needs to investigate alternative methods to simple enumeration.

To effectively solve the problem for a system with more sensors and a more complex configuration, one needs to develop more efficient algorithms as well as more objective ways of choosing the weights. This could be tackled by using a *genetic algorithm* in combination with studying statistical designs (such as uniform designs) on the weights and by systematically searching through the domain of parameters (i.e., threshold values and configurations) for optimal values. A similar algorithm was developed by Leung and Wang [27], who used a uniform design to study the weights and search for optimal parameter values.

17.6 Generalizations and Complications

17.6.1 Allowing for Stochastic Dependence of the Sensor Readings

The research described above has all been done for the case of stochastically independent sensors. The model can be extended to consider stochastically dependent sensors. In the current models, sensor readings are taken to be independent of one another, although, of course, the branching actions at a later sensor can be quite different depending on the readings provided by an earlier sensor. The polyhedral method described in Sect. 17.3 can in principle be extended to deal with the case of stochastically *dependent* sensors, although the constraints required to represent the inspection effort needed to identify a particular region of the “sensor reading space” become rather more complicated.

17.6.2 Measurement Error and Optimal Threshold Setting

One important issue in modeling and computing sensor reading data and the objective functions is *measurement error*. The measurement error is also known as ‘error-in-variables’ [9, 8, 23]. Some work on port-of-entry inspection optimization algorithms has simplified the analysis by assuming precise and accurate measurements of sensors at inspection stations. This is a reasonable first approach. However, most measurements are subject to errors due to external sources, which include environmental conditions such as temperature fluctuations, humidity, dust particles, vibration, natural radiation; and internal sources that are dependent on the sensing component material and its ability to provide accurate measurements over a wide range of measurements as well as its manufacturing and assembly issues.

As noted earlier, Stroud and Saeger [41] assume that both “good” containers and “bad” containers follow Gaussian distributions of readings. When readings exceed a threshold value, the outcome is viewed as suspicious. Based on whether or not an outcome is suspicious, the next test to be applied is determined or the container is finally categorized as “ok” or not. Of course, depending on the distribution of readings and the threshold values, errors can occur with certain probabilities. Anand, et al [2], also using Gaussian distributions, experiment with models for setting the thresholds so as to minimize total cost of the corresponding binary decision tree. Their approach involves incrementing individual sensor thresholds in fixed-size steps in an exhaustive search for threshold values that will minimize the expected cost of a binary decision tree. More efficient algorithms, based on combinations of the Gradient Descent Method and Newton’s Method in optimization, are reported on in Madigan, et al. [28].

Other approaches are also of interest. For example, building on work of Yi and Lawless [46], one can aim to “extract” the “true” values of the measurements in order to determine accurate threshold levels and minimize the probability of misclassification of inspected items (containers) investigated using sensor observations. One can also apply analysis of the “meaningfulness” of conclusions from combinatorial optimization, developed in the measurement theory literature. (See Mahadev, Pekec, and Roberts [29, 30], Roberts [36, 37, 38].) This approach analyzes the sensitivity of conclusions about optimality if parameters are measured on different kinds of scales and the scales change in permissible ways. Conditions are given for conclusions of optimality to be invariant under permissible changes of scale. A major application of this line of work has been to scheduling problems and one can seek to modify the methods of Mahadev, Pekec, and Roberts [29, 30] to apply to the port-of-entry inspection problem, which involves complications in the scheduling problems considered previously in the literature.

17.7 Case Study: Container Risk Scoring

The first step in the container inspection process actually starts outside the United States. To determine which containers are to be inspected, the United States Customs and Border Protection (CBP) uses a layered security strategy. One key element of this

strategy is the Automated Targeting System (ATS). CBP uses ATS to review documentation, including electronic manifest information submitted by the ocean carriers on all arriving shipments, to help identify containers for additional inspection. CBP requires the carriers to submit manifest information 24 hours prior to a United States-bound sea container being loaded onto a vessel in a foreign port.

ATS is a complex mathematical model that uses weighted rules that assign a risk score to each arriving shipment in a container based on manifest information. The CBP officers then use these scores to help them make decisions on the extent of documentary review or physical inspection to be conducted [45]. This can be thought of as the first inspection test and the “sensor” is the risk scoring algorithm. Thus, in some sense, all trees start with the first sensor and this sensor is then not used again. It is not unreasonable to think of more sophisticated risk scoring algorithms that also involve sequential decision making, going to more detailed analysis of risk on the basis of initial risk scoring results. The Canadian government uses similar methods. The Canadian Border Services Agency (CBSA) uses an automatic electronic targeting system to risk-score each marine container arriving in Canada. As with ATS, this Canadian system has several dozen risk indicators, and a score/weight for each indicator.

As an application of how these two risk-assessment systems can be enhanced using the methods discussed in this paper, one could think of the following approach. Each risk indicator could be categorized into a specific “risk category”, based on general themes such as an unreliable trade chain partner or suspicious commodity information. Based on various algorithms, one could create a real-valued function or “super rule” to assess the potential risk in each of these risk categories, returning a value between 0 and 1. Each of these functions could be thought of as a “sensor”. The methods described in this paper, in particular the polyhedral methods described in Sect. 17.3, could provide both the American and Canadian container targeting systems with tools to determine an optimal multi-level decision tree to determine whether a container should be targeted or authorized to clear based on the results of these super rules. These and other ideas are under consideration.

17.8 Closing Comments

It is quite impressive how much work has already been done to develop formal methods for improving container inspection procedures using optimization techniques. Future work would benefit greatly from agreement upon a method for representing the cost-effectiveness of various sensor strategies. The problem always involves representation of risks, with assumptions about probabilities and costs (or utilities). It would be very useful to find ways to make precise the range of possible parameter values for things like inspection costs, loss of trade costs, etc. The problem is particularly complex since we are dealing with low probability, high consequence events. Measurement of both the probability and the cost of such events is very difficult and of course risk assessment for such events is a central challenge in homeland security.

The many parameters involved and many criteria for a “good” inspection strategy suggest that there will be more than one way to formulate the inspection problem. The uncertainties involved about parameter values suggest that even for each formulation, a significant amount of sensitivity analysis should be carried out. Because the problem of finding solutions to the formalizations of the inspection problem becomes

dramatically more difficult as the number of tests available increases by even a small number, there is great need to develop methods for finding such solutions in increasingly efficient ways.

The chances of extending present methodology to the number of potential tests we might have in the future will be small without some dramatically new approaches.

Because our ports handle billions of dollars of goods each year, even small improvements in efficiency of handling these goods, or of inspecting them while not disrupting trade, can be very important. Because the consequences of mistakes in inspection can be catastrophic, even small improvements in the likelihood of successfully preventing weapons of mass destruction from passing through our borders can also be important.

References

1. Alidaee, B.: Optimal ordering policy of a sequential model. *Journal of Optimization Theory and Applications* 83, 199–205 (1994)
2. Anand, S., Madigan, D., Mammone, R., Pathak, S., Roberts, F.S.: Experimental analysis of sequential decision making algorithms for port of entry inspection procedures. In: Mehrotra, S., Zeng, D.D., Chen, H., Thuraisingham, B., Wang, F.-Y. (eds.) ISI 2006. LNCS, vol. 3975. Springer, Heidelberg (2006)
3. Ben-Dov, Y.: Optimal testing procedures for special structures of coherent systems. *Management Science* 27(12), 1410–1420 (1981)
4. Boros, E., Fedzhora, L., Kantor, P.B., Saeger, K., Stroud, P.: Large scale LP model for finding optimal container inspection strategies. *Naval Research Logistics Quarterly* (submitted, 2006), Preprint at, http://rutcor.rutgers.edu/pub/rrr/reports2006/26_2006.pdf
5. Boros, E., Ünlüyurt, T.: Diagnosing Double Regular Systems. *Annals of Mathematics and Artificial Intelligence* 26(1-4), 171–191 (1999)
6. Boros, E., Ünlüyurt, T.: Sequential testing of series-parallel systems of small depth. In: Laguna, M., Velarde, J.L.G. (eds.) OR Computing Tools for the New Millennium, January 5–7, 2000, pp. 39–74. INFORMS Computing Society, Cancun, Mexico (2000)
7. Butterworth, R.: Some reliability fault testing models. *Operations Research* 20, 335–343 (1972)
8. Carroll, R.J., Ruppert, D., Stefanski, L.A.: Measurement error in nonlinear models. Chapman & Hall, London (1995)
9. Carroll, R.J., Spiegelman, C.H., Lan, K.K., Bailey, K.T., Abbott, R.D.: On errors-in-variables for binary regression models. *Biometrika* 71, 19–25 (1984)
10. Chang, C.L., Slagle, J.R.: An admissible and optimal algorithm for searching and-or graphs. *Artif. Intell.* 2, 117–128 (1971)
11. Chang, M., Shi, W., Fuchs, W.K.: Optimal diagnosis procedures for k-out-of-n structures. *IEEE Trans. Comput.* 39(4), 559–564 (1990)
12. Chipman, H.A., George, E.I., McCulloch, R.E.: Bayesian CART model search. *Journal of the American Statistical Association* 93, 935–960 (1998a)
13. Chipman, H.A., George, E.I., McCulloch, R.E.: Extracting representative tree models from a forest. Working Paper 98-07, Department of Statistics and Actuarial Science, pp. 98–97, University of Waterloo (1998b)
14. Cox Jr., L.A., Qiu, Y., Kuehner, W.: Heuristic least-cost computation of discrete classification functions with uncertain argument values. *Ann. Oper. Res.* 21, 1–21 (1989)

15. Duffuaa, S.O., Raouf, A.: An optimal sequence in multicharacteristics inspection. *J. Optim. Theory Appl.* 67(1), 79–87 (1990)
16. Eschenauer, H., Koski, J., Osyczka, A.: *Multicriteria Design Optimization*. Springer, Berlin (1990)
17. Fonseca, M., Fleming, P.J.: Multiobjective optimization and multiple constraint handling with evolutionary algorithms – Part I: Unified formulation. *IEEE Trans. Syst., Man. Cybern. A* 28, 26–37 (1998a)
18. Fonseca, M., Fleming, P.J.: Multiobjective optimization and multiple constraint handling with evolutionary algorithms – Part II: Application example. *IEEE Trans. Syst., Man. Cybern. A* 28, 38–47 (1998b)
19. Greiner, R.: Finding optimal derivation strategies in redundant knowledge bases. *Artif. Intell.* 50, 95–115 (1990)
20. Jones, D.F., Tamiz, M.: Goal programming in the period 1990–2000. In: Ehrgott, M., Gandibleux, X. (eds.) *Multiple Criteria Optimization: State of the art annotated bibliographic surveys*, pp. 129–170. Kluwer, Dordrecht (2002)
21. Joyce, W.B.: Organizations of unsuccessful R&D projects. *IEEE Transactions on Engineering Management* 18(2), 57–65 (1971)
22. Kadane, J.B.: Quiz show problems. *J. Math. Anal. Appl.* 27, 609–623 (1969)
23. Kalbfleisch, J.D., Prentice, R.L.: *The statistical analysis of failure time data*, 2nd edn. Wiley, New York (2002)
24. Kowalski, R.: Search strategies for theorem proving. In: Meltzer, B., Mitchie, D. (eds.) *Machine Intelligence*, vol. 5, pp. 181–201. Edinburgh University Press, Edinburgh (1969)
25. Kowalski, R.: And-or graphs, theorem proving graphs and bi-directional search. In: Meltzer, B., Mitchie, D. (eds.) *Machine Intelligence*, vol. 7, pp. 167–194. Edinburgh University Press, Edinburgh (1972)
26. Kushner, H., Pakut, A.: A Simulation Study of Decentralized Detection Problem. *IEEE Trans. On Automatic Control* 27(5), 1116–1119 (1982)
27. Leung, Y.W., Wang, Y.: Multiobjective programming using uniform design and genetic algorithm. *IEEE Trans. Syst. Man Cyber. C* 30(3), 293–304 (2000)
28. Madigan, D., Mittal, S., Roberts, F.S.: Sequential decision making algorithms for port of entry inspection: Overcoming computational challenges. DIMACS Center, Rutgers University (January 2007) (preprint) (submitted for publication)
29. Mahadev, N.V.R., Pekec, A., Roberts, F.S.: Effect of change of scale on optimality in a scheduling model with priorities and earliness/tardiness penalties. *Mathematical and Computer Modelling* 25, 9–22 (1997)
30. Mahadev, N.V.R., Pekec, A., Roberts, F.S.: On the meaningfulness of optimal solutions to scheduling problems: Can an optimal solution be non-optimal? *Operations Research* 46(suppl.), 120–134 (1998)
31. Miglio, R., Soffritti, G.: The comparison between classification trees through proximity measures. *Computational Statistics and Data Analysis* 45, 577–593 (2004)
32. Mitten, L.G.: An analytic solution to the least cost testing sequence problem. *The journal of Industrial Engineering*, 17 (January–February 1960)
33. Natarajan, K.S.: Optimizing depth-first search of AND-OR trees. Technical Report, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 (1986)
34. Nilsson, N.J.: *Problem-Solving Methods in Artificial Intelligence*. McGraw-Hill, New York (1971)
35. Pohl, I.: Bi-directional search. In: Meltzer, B., Mitchie, D. (eds.) *Machine Intelligence*, vol. 6, pp. 127–140. Edinburgh University Press, Edinburgh (1971)

36. Roberts, F.S.: Meaningfulness of conclusions from combinatorial optimization. *Discrete Applied Math.* 29, 221–241 (1990)
37. Roberts, F.S.: Limitations on conclusions using scales of measurement. In: Barnett, A., Pollock, S.M., Rothkopf, M.H. (eds.) *Operations Research and the Public Sector*, pp. 621–671. Elsevier, Amsterdam (1994)
38. Roberts, F.S.: Meaningless statements *Contemporary Trends in Discrete Mathematics*. DIMACS Series, vol. 49, pp. 257–274. American Mathematical Society, Providence (1999)
39. Scniederjans, M.J.: Goal programming methodology and applications. Kluwer Publishers, Boston (1995)
40. Statnikov, R.S., Matusov, J.B.: *Multicriteria Optimization and Engineering*. Chapman and Hall, New York (1995)
41. Stroud, P., Saeger, K.: Enumeration of Increasing Boolean Expressions and Alternative Digraph Implementations for Diagnostic Applications. In: Chu, H., Ferrer, J., Nguyen, T., Yu, Y. (eds.) *Proceedings, Computer, Communication and Control Technologies: I*, vol. IV, pp. 328–333 (2003)
42. Sultan, A.M., Templeman, A.B.: Generation of Pareto solutions by entropy-based methods. In: Tamiz, M. (ed.) *Multiobjective Programming and Goal Programming: Theories and Applications*, pp. 164–195. Springer, Berlin (1996)
43. Ünlüyurt, T.: Sequential testing of complex systems: A review. *Discrete Applied Mathematics* 142(1-3), 189–205 (2004)
44. Ünlüyurt, T.: Testing systems of identical components. *Journal of Combinatorial Optimization* 10(3), 261–282 (2005)
45. United States Government Accountability Office, Cargo container inspection. GAO-06-591T (March 30, 2006)
46. Yi, G.Y., Lawless, J.F.: A corrected likelihood method for the proportional hazards model with covariates subject to measurement error. *Journal of Statistical Planning and Inference* (in press, 2006)
47. Zhang, H., Schroepfer, C., Elsayed, E.A.: Sensor Thresholds in Port-of-Entry Inspection Systems. In: *Proceedings of the 12th ISSAT International Conference on Reliability and Quality in Design*, Chicago, Illinois, USA, August 3–5, pp. 172–176 (2006)

Modeling and Validation of Aviation Security

Uwe Glässer, Sarah Rastkar, and Mona Vajihollahi

Software Technology Lab, School of Computing Science,
Simon Fraser University, Canada
`{glaesser, srastkar, mvajihol}@cs.sfu.ca`

Abstract. Security of civil aviation has become a major concern in recent years, especially due to the increasing number of potential and real threats imposing dynamically changing risks on airport and aircraft security. We propose here a novel computational approach to checking consistency, coherence and completeness of aviation security requirements and provide a framework for systematic analysis of the efficiency and effectiveness of procedural security measures. Our approach deals with the inherent uncertainty of security systems by utilizing advanced computational and probabilistic modeling techniques (namely, Abstract State Machines and Probabilistic Timed Automata) in combination with Probabilistic Model Checking tools.

18.1 Introduction

Civil aviation security encompasses a variety of protective measures related to airport and aircraft security that collectively aim at safeguarding airports, aircrafts and air traffic control against any unlawful interference. Vital aviation security mechanisms routinely deal with a wide range of procedural aspects of security controls and screening procedures, for instance, as practiced for airport passenger screening, checked baggage and cabin baggage screening, and likewise measures for air cargo security. With an increasing number of possible threats, the task of operationalizing security goals into functional requirements and constraints on procedures and routines poses a challenging problem. Aviation security standards, guidelines and recommended practices [17, 31] are constantly being revised and updated to improve and intensify security measures, trying to catch up with dynamically changing risks caused by a multitude of threats—a moving target that is difficult to predict. Practicality considerations also demand for cost sensitive solutions. Ultimately, one wants acceptable safety at reasonable cost, where cost factors in particular also include timing aspects such as delays.

Security engineering can no longer rely on empirical deduction and statistical methods alone to cope with the notorious problem of establishing the soundness and completeness of aviation security measures. Striving to eliminate any conceivable vulnerabilities as far as possible, critical inspections of both the effectiveness and the efficiency of security controls are crucial to provide feedback for enhancing and improving protective measures. Arguably, there has been a lack of analytical means for systematic reasoning about procedural aspects of complex security models. Practical

tests can provide valuable insights and reveal deficiencies that may not be noticed otherwise [11]. However, real-world experiments are often problematic and have their limitations, making any exhaustive testing virtually impossible to do. So what are more feasible alternatives then? We try to answer this question by introducing a novel computational approach to systematically check procedural security models, performing a thorough analysis and validation of their abstract operational requirements.

Our approach is inspired by the observation that functional requirements on procedural aspects of aviation security can abstractly be described in terms of *process flow models* and, consequently, can be analyzed, validated and verified by innovative software technology methods much like abstract system requirements. Formalizing abstract requirements properly within a well defined computational framework has a number of advantages. Informal requirements make it difficult to identify and eliminate hidden deficiencies and weaknesses¹, potentially causing severe misinterpretations that can result in security holes with fatal consequences. Formalization helps uncovering such problems in early design phases. It is also a prerequisite for practically any form of machine assisted inspection of models and their characteristic properties through computer simulation (testing) and/or symbolic execution (verification). Finally, an abstract formal model can serve as an accurate and reliable documentation, a ‘blueprint’ of the key requirements, preserving important knowledge for reuse and future development.

Security mechanisms as considered here rely to some non-negligible extend on assumptions that involve uncertainty. For instance, screening measures for baggage, like x-ray or hand search, can identify an unlawful object in a certain baggage item with a relatively high probability, especially when the two measures are combined. However, there is no absolute guarantee that every unlawful object in the baggage being screened will indeed be identified as such. Depending on a number of human and technical factors, an object may indeed escape the attention of the operator of the x-ray machine and also pass a hand search performed afterwards. In the presence of such real-world phenomena, probabilistic modeling techniques provide a rational choice. For the construction of formal models derived from aviation security guidelines, we formalize procedural requirements in abstract computational terms as state transition systems, combining *Abstract State Machines* (ASM) [5] and *Probabilistic Timed Automata* (PTA) [20]. In combination with these two modeling techniques, we use symbolic *Model Checking* (MC) [7], specifically probabilistic MC techniques [28], as a structured computational framework for establishing the consistency, coherence, and completeness of procedural security requirements.

The main purpose of this work, carried out under a project called Safeguard², is to provide a practical instrument and a tool for quantitative analysis and experimental studies of security procedures and routines with the intention to improve existing standards and practices by identifying potential deficiencies and weaknesses. To this end, the goal is to deduce critical facts as meaningful feedback for quality assurance rather than proving correctness.

The chapter is structured as follows. Sect. 18.2 discusses related work and presents a literature review. A case study on airport screening procedures, presented

¹ Frequent problems are accidental ambiguities, loose ends, and logical inconsistencies.

² Safeguard Project, <http://www.cs.sfu.ca/~se/projects/safeguard>

in Sect. 18.3, illustrates the modeling approach at an intuitive level of understanding using most common practices as an example. Experimental results from model checking the screening model are presented in Sect. 18.4 followed by a discussion of the results and the conclusions in Sect. 18.5. Finally, Sect. 18.6 lists related on-line resources and Sect. 18.7 a number of open questions for further discussion.

18.2 Related Work

Conceptually, the scope of the problem tackled here spans aviation security, decision support, requirements engineering, system modeling, design validation, and formal verification. Devising reliable and robust aviation security plans is a challenging task, one that naturally demands for systematic methods for operationalizing complex security goals into functional requirements on procedures and routines, making sure that the resulting security plan is coherent, consistent and complete. While the computational framework proposed here concentrates on decision support for aviation security measures, the same approach will likely be meaningful in other application contexts as well, e.g., critical infrastructure protection and emergency response.

The following subsections review literature and work related to our approach, starting with a brief outline of aviation security guidelines and standards.

18.2.1 Aviation Security Guidelines

International, regional and national authorities have devised a series of standards that specify various procedures and security measures to be implemented to ensure the security of civil aviation. The International Civil Aviation Organization (ICAO) provides guidelines at the international level defined in Annex 17 to the Convention on International Civil Aviation [17]. In order to synchronize the operations among the countries in a region, regional authorities also introduce security standards and regulations. For instance, recognizing the fact that Annex 17 provides minimum standards to ensure the security of civil aviation, the European Parliament and the Council of the European Union have approved a set of regulations establishing common rules in the field of aviation security [31]. The international and regional standards must be implemented by all contracting states involved in civil aviation, and are usually refined to capture the national laws of each country, forming national standards for civil aviation.

Principal rules and guidelines for ‘Preventive Security Measures’ in civil aviation security are outlined in [17, Sect. 4]:

The objective is “*to prevent weapons, explosives, or any other dangerous devices which may be used to commit an act of unlawful interference, the carriage or bearing of which is not authorized, from being introduced, by any means whatsoever, on board an aircraft engaged in international civil aviation.*” [17, Sect.4.1]³

For example, each ‘Contracting State’ shall establish measures to ensure that [17, Sect. 4.1]:

³ Hold baggage refers to baggage intended for carriage in the hold of an aircraft [31].

- “*Operators when providing service from that State do not transport the baggage of passengers who are not on board the aircraft unless that baggage is subjected to appropriate security control which may include screening.*” (4.4.3)

18.2.2 Modeling Airport Security

In response to the 9/11 incidents, several programs for enhancing airport security have been launched. Specifically, there have been initiatives suggesting the use of modeling and simulation to support airport security [32]. One of the main works is a European project called EDEMOI [23] with the final report published in October 2006 [24]. This project aimed at applying computational modeling techniques to address two different aspects of airport security: i) assessing the quality of international standards and recommended practices (by checking for logical deficiencies), and ii) evaluating the conformance of a given airport to these standards.

The proposed approach combines two different kinds of modeling techniques: UML graphical models of aviation security standards facilitate validation by domain experts; transformation of graphical models into formal models using B [1] supports in-depth analysis and the generation of test scenarios for conformance tests. In comparison to our approach, there are two major differences. First, they do not consider probabilistic aspects caused by the inherent uncertainty of security procedures. Second, the optimization problem which arises from the trade-off between cost and safety is also not addressed.

A number of other papers focus on applying simulation techniques in the context of airport modeling [12, 30]. Most of them concentrate on improving performance measures within an airport without addressing the security problem. Related to our work, is the work presented in [26], which utilizes simulation-based methodologies with the goal of redesigning the passenger screening process to adhere to security standards while improving the performance and minimizing the cost (i.e. inconvenience to the public and time delays). They use a discrete event simulation model to represent the passenger and luggage screening systems. Starting from the current (“As Is”) simulation model of the airport, a series of “To Be” design alternatives are built. Each “To Be” model is analyzed to verify whether it meets the agreed to security requirements while improving the overall performance. This process is repeated until the final design emerges.

18.2.3 Abstract State Machines

Nondeterminism and probabilistic choice are two key concepts in modeling the discrete event systems addressed by airport security measures. Nondeterminism greatly simplifies modeling by reducing complexity, while probabilistic choice deals with the phenomenon of uncertainty. Additionally, concurrency plays a major role as we consider distributed systems characterized by decentralized control structures and asynchronous interaction patterns. To this end, Abstract State Machines [5] mean a sensible choice for modeling system requirements abstractly in terms of process flow models. Building on a simple but universal mathematical framework, this modeling

paradigm is known for its versatility in specification and design, experimental validation and formal verification of algorithms, architectures, languages, protocols and virtually all kinds of sequential, parallel and distributed systems.⁴

The ASM formalism and abstraction principles have been studied extensively by researchers in academia and industry both in Europe and North America. Widely recognized applications include semantic models of popular programming languages [29], international industry standards [13, 4], Web service architectures [10, 2], crime patterns in computational criminology [6], and the development of supporting tool environments [25, 9]. Together, the many applications have led to a solid methodological foundation for building *ASM ground models* [5, 3]. Intuitively, a ground model of the system under study serves as a ‘blueprint’ that establishes the key functional requirements in a reliable form with a degree of detail and precision as needed, not restricting any conceivable refinements. Emphasizing semantic rather than syntactic aspects, a ground model provides a proper basis for critically checking its consistency, completeness and validity.

18.2.4 Probabilistic Model Checking

We adopt here the technique of Probabilistic Model Checking (PMC) [28] to quantitatively analyze and validate our airport security model. Generally, model checking [7] uses symbolic computation methods to automatically verify formal models defined as state transition systems. That is, the state space of a model is inspected by performing a reachability analysis to check if a specified property holds on every reachable state. Due to the probabilistic nature of our airport security model, we use a probabilistic variant of classical model checking to analyze properties of interest. Specifically, we use PRISM [27, 16], a tool developed at the University of Birmingham. PRISM is an internationally leading and widely used probabilistic model checker, which has been applied to a wide range of real-life systems [18].

Probabilistic Timed Automata (PTA) [20], a variant of timed automata with discrete probability distributions, is a formal framework for modeling state transition systems. This formalism has shown to be suitable for the description of timed systems exhibiting both stochastic and nondeterministic characteristics. In a PTA, real-valued clocks measure the passage of time, and transitions can be probabilistic, that is, be expressed as a discrete probability distribution on the set of target states. PTA models can be used as input for the PRISM model checker as will be explained in Sect. 18.4. Further examples of analyzing PTA models with PRISM are presented in [21, 22].

18.3 Airport Security Model

For the purpose of illustrating our approach, we concentrate on airport screening procedures for a hypothetical airport. We build our airport security model incrementally upon the guidelines and standards introduced by ICAO [17] and ECAC [31], starting with an ASM ground model of the principal requirements defined by ICAO. This model is then refined in several consecutive refinement steps. First, the *screening*

⁴ See also the ASM Web site at www.eecs.umich.edu/gasm/ for numerous case studies.

procedure is refined by using the more specific European guidelines. Second, the actual *screening operation* is described in detail assuming a hypothetical airport security plan. In the last step, this model is then transformed into a Probabilistic Timed Automata (PTA) model in order to facilitate the use of the probabilistic model checker PRISM. Fig. 18.1 illustrates the overall architectural model which will be explained in the following sections.

It is important to note that, since the standards and recommendations specified by ICAO have to be followed at the international level by all the member countries, building a ground model based on [17] provides the general framework for further refining the model according to ‘any’ more specific guidelines. We have chosen to use the European standard for further refinements, simply because it is publicly available.

The guidelines and rules in [17] thus form the general framework for our ground model of airport security procedures.

18.3.1 Security Flow Model

The ASM ground model presented here captures the *process flow* for ensuring civil aviation security as outlined in [17, Sect. 4]. We define a domain ENTITY that contains all the entities in the system security of which must be ensured. This includes passengers, cabin baggage, and hold baggage.⁵ We also define a domain ACTOR that represents all the active players controlling the security procedures in the system and includes security personnel and the screening equipment they operate. The behavior of both actors and entities is modeled by identifying them with autonomously operating ASM agents of our distributed ASM.

$$\text{AGENT} \equiv \text{ACTOR} \cup \text{ENTITY}$$

$$\text{ACTOR} \equiv \text{OPERATOR} \cup \text{MACHINE}$$

$$\text{ENTITY} \equiv \text{PASSENGER} \cup \text{PACKAGE}$$

$$\text{BAGGAGE} \equiv \text{CABIN_BAGGAGE} \cup \text{HOLD_BAGGAGE}$$

Each entity has to follow the required security procedure before boarding the aircraft. As discussed in Sect. 18.2.1, Annex 17 [17] only provides a set of rules describing the required security measures for each type of entities; no visible architecture or further description of the security procedures is provided by this document. However, after close examination of the standard, we have devised an architectural description in the form of an abstract flow model. This model applies to all the entities that are subject to security controls. We define the flow model in terms of a *control-state ASM*⁶ as shown in Fig. 18.2.

From the time an entity enters the airport, to the time it leaves the airport by plane, it goes through different *modes* as represented by control states. Initially, when

⁵ Annex 17 also provides measures relating to cargo and special categories of passengers. The process flows for these entities are similar to those of passenger and baggage, however, they are not considered as part of the model presented here.

⁶ Control state ASMs represent “a normal form for UML activity diagrams and allow the designer to define machines which below the main control structure of finite state machines provide synchronous parallelism and the possibility to manipulate data structures”[5].

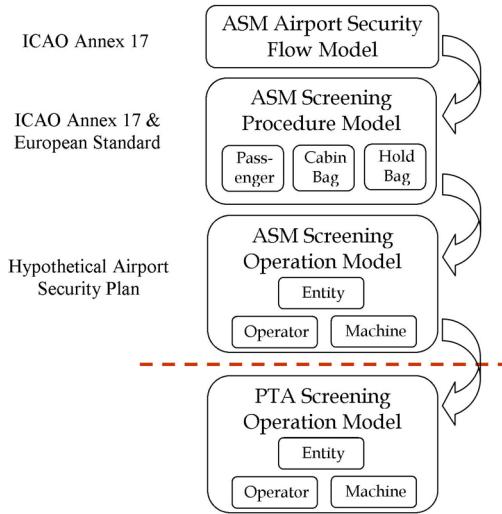


Fig. 18.1. The Architecture of our Airport Security Model

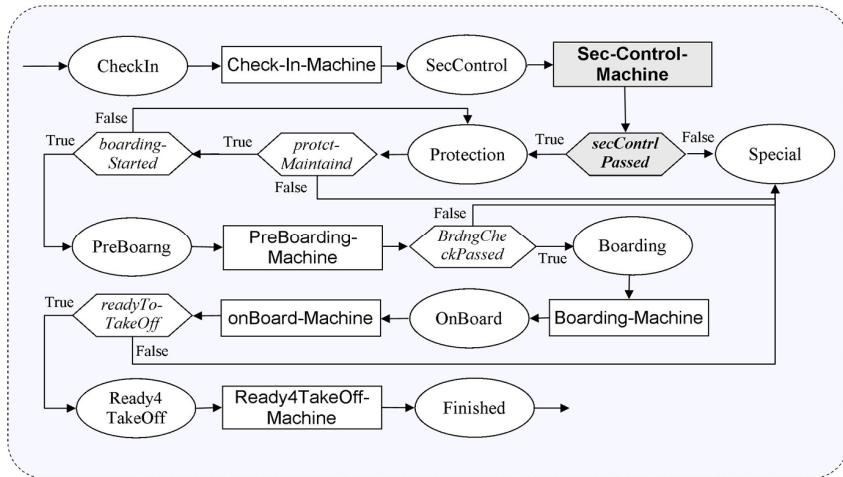


Fig. 18.2. Control-State ASM of the Airport Security Flow Model

entering the airport, an entity is in the mode ‘CheckIn’, meaning that it has to go through the check-in procedure as modeled by the **Check_In_Machine**. When completed, its mode switches to ‘SecControl’, where the required security measures are invoked by the **Sec_Control_Machine**. If the security control procedure is completed successfully, the mode switches to ‘Protection’, where it needs to be protected from any “unauthorized interference”. On the other hand, if the entity does not pass the security control, its mode switches to ‘Special’. This mode is designed to capture any exceptions and special cases where the routine security flow is interrupted. An entity

stays in the mode ‘Protection’, until boarding starts or the protection condition is violated, i.e. an unauthorized contact occurs. A violation results in the mode ‘Special’, whereas if the boarding is started without any violation, the mode is changed to ‘Pre-Boarding’. After additional security checks are performed by the PreBoarding_Machine, the mode is changed to ‘Boarding’. Once the entity is ‘OnBoard’, final security checks are performed by the OnBoard_Machine. For instance, before the plane can take off, Rule 4.4.3 (see Sect. 18.2.1) requires that if the passenger of a piece of baggage is not on board of the aircraft, the baggage must not be transported unless it is subjected to appropriate security controls. Once the plane is ready to take off, the mode is changed to ‘Ready4TakeOff’. The mode then changes to ‘Finished’ as soon as the plane takes off. At this point, the entity is removed from the system.

18.3.2 Incremental Refinements

The abstract security flow model of Sect. 18.3.1 can now further be refined to capture the specific security control requirements for each type of entity. Regarding the screening of baggage, the security control machine (highlighted in Fig. 18.2) can be refined into $\text{Sec_Control_Machine}_{\text{passCabin}}$ and $\text{Sec_Control_Machine}_{\text{holdBag}}$. We focus here on the security procedure for accompanied hold baggage and describe how this modeled based on the European regulations [31].

Screening Procedure Model

We regard the European standard as a source for clarifying and refining the guidelines outlined in [17], providing a more detailed and practical view of the security measures. The European standard clearly defines “hold baggage” as “baggage intended for carriage in the hold of an aircraft” and specifies two different procedures for screening of hold baggage depending on whether it is accompanied and unaccompanied baggage [31, Sect. 5]. We restrict our attention to the screening procedure for accompanied hold baggage⁷ and refer to [14] for a more detailed discussion on the difference between screening of accompanied and unaccompanied hold baggage.

According to the standard, “*All items of accompanied hold baggage [...] shall be screened by one of the following methods before being loaded onto an aircraft: i) hand search; or ii) conventional x-ray equipment with at least 10% of screened baggage also being subjected to either: hand search; or EDS or EDDS or PEDS⁸ or ...*”

Screening of accompanied hold baggage is formally described in terms of another control-state ASM as shown in Fig. 18.3. To determine the faith of a given piece of hold baggage, the baggage goes through a number of internal modes, starting with the mode ‘Idle’. Before the screening operation is finished (and the machine enters the mode ‘Done’), the value of the *secContrlPassed* function (highlighted in Fig. 18.2) is set. This function is then used in the airport security flow model to decide whether the entity can enter the mode ‘Protection’ or it should switch to the mode ‘Special’, indicating the need for special attention.

⁷ Accompanied hold baggage is defined as “baggage accepted for carriage in the hold of an aircraft, on which the passenger who checked it in is on-board.”[31, Sect. 1].

⁸ EDS: Explosive Detection System, EDDS: Explosive Device Detection System, PEDS: Primary Explosive Detection System [31, Sect. 1].

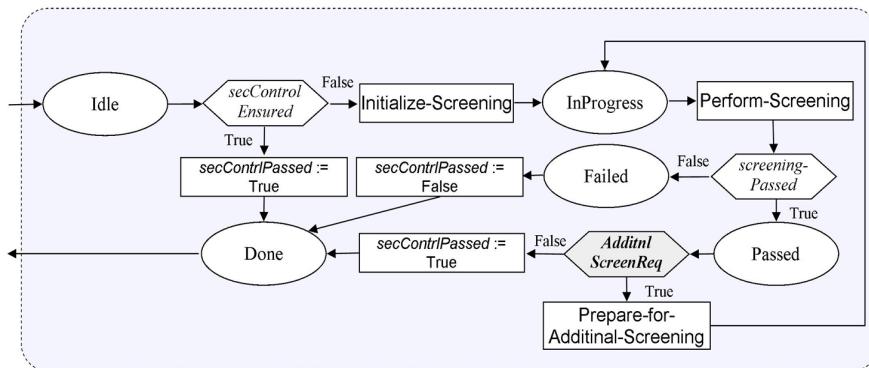


Fig. 18.3. Control-State ASM for Screening of Accompanied Hold Baggage

The *additnlScreenReq* function (highlighted in Fig. 18.3) allows for the hold baggage to be randomly re-screened according to the requirements of the European standard.⁹

Screening Operation Model

Screening a piece of baggage typically involves additional actors, in particular, an x-ray machine and its operator. In order to model the screening operation as required for refining the *Perform_Screening* rule (cf. Fig. 18.3), we have to capture the behavior of each of those actors and also model the interactions between them.¹⁰ We do this here for the x-ray machine and the operator.

A piece of baggage is first screened by the x-ray machine generating an image of its content. At the same time, with the use of modern image detection techniques [8], the machine signals to the operator if it detects/suspects an unlawful object contained in the screened baggage item. The operator then analyzes the image to make a final decision. This decision depends on the image being produced, as well as time constraints and also various human factors such as the operator's experience, distractions affecting attention, fatigue reducing concentration, et cetera. Finally, the operator classifies a piece of baggage as *passed* or *failed*, or sends it back to be *re-screened* by the machine. We specify the behavior of the operator in terms of the control-state ASM illustrated in Fig. 18.4, abstractly modeling what can be considered the operator's 'internal state' by means of a function *mode*.

The operator remains in the 'Idle' mode until the x-ray machine has generated either a positive signal (Pos) or a negative signal (Neg). In case of a positive signal (i.e. the machine has detected an unlawful object), the operator will also classify the baggage as positive. The mode switches to 'Pos' and the status of the inspected item is updated to *failed*. In case of a negative signal (i.e. nothing suspicious has been

⁹ For a more detailed discussion of the probabilistic nature of this function, we refer the interested reader to [14].

¹⁰ Although the focus in this paper is on hold baggage screening, the same principles apply to screening operations of other entities as well.

detected), the baggage item is not automatically classified as passed; rather, the operator needs to make a decision on the resulting screening status. This decision needs to be made in a timely fashion¹¹. The transition phase is modeled by changing the mode to ‘Decide’. With the probability of $1 - pDoubt$, the operator does not doubt the conformity of the screened item with the regulations, and the status of this item is set to *Passed*. On the other hand, the operator may suspect a problem with the probability of $pDoubt$, being unable to classify the baggage right away. In this case, the mode switches to ‘Doubt’, where a decision is made non-deterministically between classifying the item as positive or sending it back for re-screening.¹² Meanwhile, another timing constraint is also considered in this state. The operator can not let any baggage go through re-screening indefinitely; thus, the total time spent on screening of a piece of hold baggage must be restricted. As such, if the total time spent on screening the same piece of baggage exceeds the limit, and the operator is still doubtful, the mode is changed to ‘Pos’, implying that alternate means of screening will be applied. In any case, when the inspection of one piece of baggage is completed (the status of the baggage is updated), the mode changes back to ‘Idle’, and the operator waits for the next baggage item on the x-ray belt.

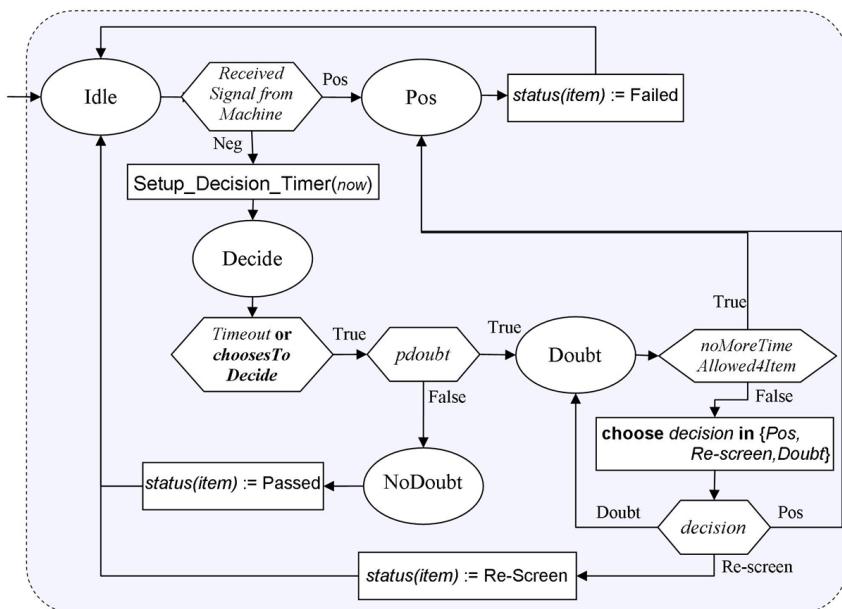


Fig. 18.4. Control-State ASM for the Operator of an X-Ray Machine

¹¹ Timing constraints have a major impact on the performance and accuracy of security procedures and, hence, must be captured in the model appropriately.

¹² It is important to note that nondeterminism serves here as an abstraction of the various factors that, possibly in combination with one another, may influence this decision.

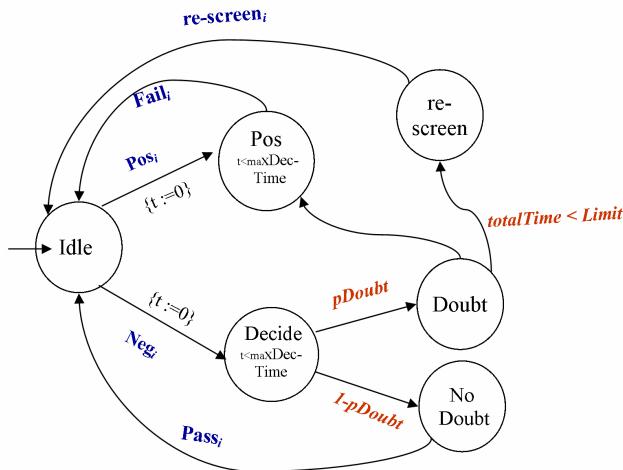


Fig. 18.5. PTA of the Operator of an X-Ray Machine

18.3.3 Probabilistic Timed Automata (PTA) Model

The refined ASM ground model of the screening operation (see the operator control-state ASM of Fig. 18.4 in Sect. 18.3.2) provides a detailed description of the ASM agent abstractly representing the observable behavior of an operator. We now transform this model into a PTA. The precise semantics of the ASM and the PTA formalisms facilitates a direct transformation between the two models.

The behavior of the operator is captured by the PTA of Fig. 18.5. In order to model the synchronized interaction between the operator and the baggage (or the x-ray machine), we utilize the labels assigned to PTA transitions.¹³ In the state ‘Idle’, when a *Pos* signal is received from the machine (i.e. the transition with the same label is fired by the machine PTA), a *Failed* signal is sent to the baggage. On receiving a *Neg* signal from the machine, the operator makes a decision (mode ‘Decide’) within the time period specified by *maxDecTime*. If there is no doubt (with probability $1-pDoubt$), a *Passed* signal is sent to the machine. Otherwise, a non-deterministic choice is made between *re-screening* and classifying the baggage item as ‘Pos’. However, rescreening is only possible, if the total time spent on screening this baggage item is below the overall time limit.

The complete PTA model of the whole screening procedure consists of the operator PTA, the machine PTA, and the baggage PTA. This model finally serves as the input to the PRISM model checker and will be discussed in more detail in Sect. 18.4. Due to space limitations, here we only exemplify the operator PTA, and refer the interested reader to [15] for a comprehensive description of the model.

¹³ PTA models are synchronized over common events (transition labels); i.e. two transitions with the same label always occur simultaneously.

18.4 Model Analysis and Validation

This section explains in detail how PRISM can be used to analyze the airport model presented in Sect. 18.3.1.

18.4.1 Analysis of PTA Models with PRISM

The current implementation of PRISM [27] only supports the analysis of *finite-state* probabilistic models of the following three types: Discrete Time Markov Chain (DTMC), Markov Decision Process (MDP) and Continuous-time Markov Chains (CTMC). MDP extends DTMC by supporting also nondeterministic in addition to probabilistic behavior. CTMC allows transitions to occur in real-time as opposed to discrete steps, but differ from PTA in that delays are represented by exponential distributions. Furthermore, there is no nondeterminism in continuous-time Markov chains.

Since a PTA has *infinitely* many states as a result of the dense nature of time it assumes (allowing positive real numbers as time values), one can not directly use PRISM to analyze systems modeled as PTA. There are several ways to tackle this problem. One is based on region equivalence [20], which results in prohibitively large state spaces for realistic systems and leads to approximate results. Another way is to digitize time by adopting an integral time model [19], which leads to finite-state systems and allows the use of efficient symbolic methods developed for untimed systems. Here, we use the integral time semantics to turn the PTA model of the screening operation into an MDP model that can be directly checked by PRISM.

Using a probabilistic model checker one can analyze a set of quantitative properties expressed in Probabilistic Computational Tree Logic (PCTL). One advantage of such a quantitative analysis is that the results can be plotted as graphs that can be inspected for trends and anomalies related to reliability, consistency, completeness, and the performance of the model. This approach is discussed in more detail in the following subsections.

18.4.2 Experimental Results

In [14], we showed how DTMC models can be used in combination with PCTL to identify probabilistic trends of a system. It was argued that probabilistic model checking can be used to investigate and explore the effect of different parameters on selected security properties and, hence, on the overall security of the system. Here, we apply PRISM to the complete model consisting of the three MDP models generated from the respective PTA models of Sect. 18.3.3 using integral time semantics (cf. Sect. 18.4.1). This model is referred to as *mdp1*.

The basic components of PRISM's input language are *modules* and *variables*. A system is described as a parallel composition of interacting modules, each of which contains a number of local variables. The values of these variables at any given time constitute the state of the module. The behavior of each module is described by a set of commands. Each actor in our PTA model of the screening operation (Sect. 18.3.3)

is mapped to a module in the PRISM input language. Hence we have three PRISM modules, namely `baggage`, `screeningMachine`, and `screeningOperator`.¹⁴

A global clock is used for measuring the total time spent on the screening operation for each single baggage item. The `screeningMachine` and `screeningOperator` modules also have their own local clocks. The local clock of the `screeningMachine` module measures the time spent while screening, and the local clock of the `screeningOperator` module measures the time spent by the operator in the mode ‘Decide’. All three clocks are synchronized and modeled as integer-typed variables that can be mapped to real-world units of time as appropriate.

Given the nondeterministic nature¹⁵ of MDP models, PRISM only computes either the *minimum* or the *maximum* probability of a property of an MDP model. Still, it is possible to determine if the minimum or maximum probability satisfies a given bound or to obtain the actual probability value. Although, the use of nondeterminism normally imposes restrictions on the extent to which one can analyze behavior (discussed further below), we content that these models can still be used as a reliable source for analyzing properties and identifying trends.

The total allowable time for the screening operation is represented in our PRISM model by a constant named `limit`. We inspect the effect of different values of `limit` on the properties related to safety and cost.

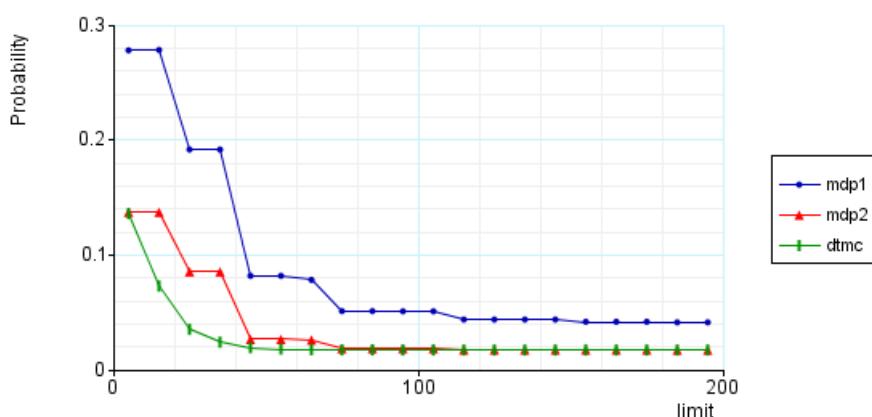


Fig. 18.6. Probability of reaching false-negative depending on time limit

Fig. 18.6 shows the *maximum* probability of reaching the *false-negative* state under different time limits. The maximum probability corresponds to the worst-case scenario, meaning that the screening operation fails to identify a baggage item containing

¹⁴ The model presented here only uses one instance of each module. In principle, it is possible to add more baggage items to the system or use multiple screening machines by utilizing the *module renaming* feature of PRISM, which allows duplication of modules.

¹⁵ The scale of nondeterministic behavior exhibited by a model basically corresponds to the chosen level of abstraction.

one or more unlawful objects. This safety-related property is formally defined in PCTL as follows.

```
Pmax =? [true U baggage.mode = False Neg] where 0 < limit ≤ 200
```

As one would expect, the probability of reaching the false-negative state decreases as one allows more time for the screening operation. In other words, with more time spent on the screening operation, this failure is less likely to occur. In addition to the original model *mdp1*, this property has also been checked for two different variants of *mdp1*, referred to as *mdp2* and *dtmc* (cf. Fig. 18.8). The two variants were derived from *mdp1* by gradually reducing nondeterminism. Our goal is to investigate the impact of nondeterminism on the obtained results in order to calibrate the level of abstraction, being able to control complexity without loosing relevant information. In the DTMC variant, all the nondeterministic transitions were replaced with probabilistic ones, corresponding to the case where detailed information on all the transitions is available. The results confirm the reliability of the original MDP model for analysis purposes. For a more detailed description of the different variants we refer to [15].

Regarding the cost of the screening operation, there is a simple equation: the longer the time one allows for an individual baggage item the less throughput one can expect and the higher are the cost. In order to investigate the effective cost factors more differentiated, we use the cost/reward structure provided by PRISM. Numerical cost values can be assigned to different transitions and/or states in the system under study.

For our model, there are two critical factors contributing to the total cost of the screening operation: (a) baggage items that are sent for re-screening, and (b) baggage items not containing any unlawful objects that are falsely classified as dangerous baggage (false-positive). We model the accumulation of costs caused by these two factors by assigning cost values to the respective transition (re-screening) and the respective state (false-positive).

Fig. 18.7 shows the combined accumulated cost for different values of *limit*. The saturation effect with respect to the cost values for longer time limits is due to the

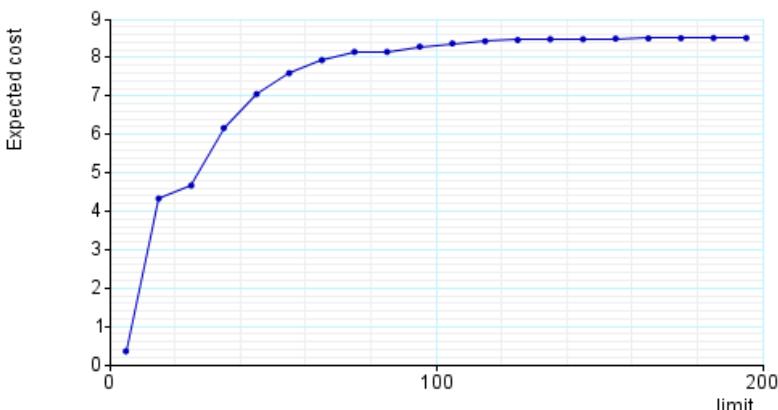


Fig. 18.7. Overall cost of the screening operation depending on time limit

fact that longer sequences of iterations of re-screening of the same baggage item are less likely than shorter sequences, whereas the probability of false positives relatively soon reaches a stable value virtually not decreasing further with increasing time.

The diagrams in Figs. 18.6 and 18.7 show a trade-off between safety and cost. We argue that these quantitative results can be analyzed to explore the impact of the value of limit on the performance with respect to safety and cost. For instance, both diagrams show significant fluctuation for time values between 10 and 80 time units. This interval is considered the ‘critical’ segment of the time line, where the diagrams serve as a reliable source for investigating the impact of choosing a value within this segment on both safety and cost. For a more detailed discussion, we refer to [15].

18.5 Discussions and Conclusions

Formalization of informal requirements is a common approach to deal with the inherent ambiguity and impreciseness of natural language and also a prerequisite for validation and verification of abstract system models by computational means. Inspired by the observation that functional requirements on procedural security models can be naturally expressed in terms of process flow models formally described as state transition systems and, thus, can be analyzed, validated and verified using innovative software technology methods, we propose here a novel approach to systematically check airport security procedures and routines. We illustrate the basic concepts by means of simple yet meaningful examples.

In the presence of real-world phenomena affecting security mechanisms in various ways, uncertainty is a non-negligible aspect that needs to be addressed by the applied formal approach. We use probabilistic modeling techniques in combination with probabilistic model checking to capture this aspect properly. Building on existing security regulations and standards introduced by aviation security authorities, we model abstract operational requirements on baggage screening at airports in terms of Abstract State Machine models and Probabilistic Timed Automata.

The concept of ASM ground models greatly simplifies the task of turning informal requirements into precise specifications, bridging the gap between empirical and formal approaches. To this end, a ground model provides an accurate and complete description of the system under study, ensuring that all the crucial security requirements are captured, while less important details are left abstract; as such, it serves as a reliable foundation for incremental refinements. On the other hand, Probabilistic Timed Automata can easily be transformed into Markov Decision Processes, a probabilistic model representation that is accepted as input by the PRISM model checker. Properties of interest to be checked are represented in Probabilistic Computational Tree Logic. In addition to analyzing probability values, PRISM also supports the computation of relevant cost factors.

In principle, the proposed computational framework can be used for modeling procedural aspects of security at any desired level of abstraction. Formalization facilitates the task of checking the logical consistency, coherence and completeness of airport security plans. Experimental studies help to better understand how the various system parameters affect the resulting behavior and performance. Based on extensive experience with modeling a variety of real-life distributed systems, we can say that

the proposed modeling approach based on ASM abstraction principles is scalable and suitable to capture much more complex procedural requirements. To what extend the resulting models can also be analyzed using machine-assisted verification techniques is not sufficiently clear without further investigation. However, the proper use of refinement methods in combination with the firm semantic foundation allows for systematically decomposing a system into sub-parts where each part can be analyzed separately. The overall behavior of the system then can be defined and analyzed as the composition of the sub-systems.

In summary, the proposed combination of computational techniques seems a promising approach to overcome existing limitations of traditional security engineering solely based on empirical deduction and statistical methods. In the struggle for continuous improvements and enhancements to ensure acceptable safety at reasonable cost, the use of computational methods and tools allows to systematically reason about the effectiveness and efficiency of security measures. The approach suggested here likely has similar benefits in any security context in which the problem of operationalizing complex security goals into functional requirements on procedures and routines poses a challenge.

References

1. Abrial, J.R.: *The B-Book: Assigning Programs to Meanings*. Cambridge University Press, Cambridge (1996)
2. Altenhofen, M., Börger, E., Lemcke, J.: A High-Level Specification for Mediators (Virtual Providers). In: Bussler, C., Haller, A. (eds.) *Proceedings of Business Process Management Workshops*, pp. 116–129 (2005)
3. Börger, E.: The ASM ground model method as a foundation for requirements engineering. *J. Verification: Theory and Practice*, 145–160 (2003)
4. Börger, E., Glässer, U., Müller, W.: Formal Definition of an Abstract VHDL'93 Simulator by EA-Machines. In: Kloos, C.D., Breuer, P.T. (eds.) *Formal Semantics for VHDL*, pp. 107–139. Kluwer Academic Publishers, Dordrecht (1995)
5. Börger, E., Stärk, R.: *Abstract State Machines: A Method for High-Level System Design and Analysis*. Springer, Heidelberg (2003)
6. Brantingham, P.L., Kinney, B., Glässer, U., Singh, K., Vajihollahi, M.: A Computational Model for Simulating Spatial Aspects of Crime in Urban Environments. In: Jamshidi, M. (ed.) *Proceedings of 2005 IEEE International Conference on Systems, Man and Cybernetics*, pp. 3667–3674. IEEE, Los Alamitos (2005)
7. Clarke, E.M., Grumberg, O., Peled, D.A.: *Model Checking*. MIT Press, Cambridge (2000)
8. The Economist Airport Screening Technology: Full Exposure. *The Economist* 389(8491), 21–21 (2006)
9. Farahbod, R., et al.: The CoreASM Project, <http://www.coreasm.org>
10. Farahbod, R., Glässer, U., Vajihollahi, M.: An Abstract Machine Architecture for Web Service Based Business Process Management. The special issue on Business Processes and Services of the International Journal of Business Process Integration and Management (IJBPIM) (to appear, 2006)
11. The fifth estate, Fasten Your Seatbelts. First aired Nov 9, 2005 on CBC-TV (2006) Last visited November 2006,
<http://www.cbc.ca/fifth/fastenseatbelts/index.html>

12. Gatersleben, M.R., van der Weij, S.W.: Analysis and Simulation of Passenger Flows in an Airport Terminal. In: Farrington, P.A., Nembhard, H.B., Evans, G.W. (eds.) Proceedings of the 1999 Winter Simulation Conference, pp. 1226–1231 (1999)
13. Glässer, U., Gotzhein, R., Prinz, A.: The formal semantics of SDL-2000: status and perspectives. *J. Comput. Networks* 42(3), 343–358 (2003)
14. Glässer, U., Rastkar, S., Vajihollahi, M.: Computational Modeling and Experimental Validation of Aviation Security Procedures. In: Mehrotra, S., Zeng, D.D., Chen, H., Thuraisingham, B., Wang, F.-Y. (eds.) ISI 2006. LNCS, vol. 3975, pp. 420–432. Springer, Heidelberg (2006)
15. Glässer, U., Rastkar, S., Vajihollahi, M.: Computational Modeling and Experimental Validation of Aviation Security Procedures. Technical Report SFU-CMPT-TR-2006-23. Simon Fraser University (2006)
16. Hinton, A., Kwiatkowska, M., Norman, G., Parker, D.: Prism: A tool for automatic verification of probabilistic systems. In: Hermanns, H., Palsberg, J. (eds.) TACAS 2006 and ETAPS 2006. LNCS, vol. 3920, pp. 441–444. Springer, Heidelberg (2006)
17. International Civil Aviation Organization (ICAO), Annex 17 to the Convention on International Civil Aviation: Standards and Recommended Practices - Security - Safeguarding International Civil Aviation against Acts of Unlawful Interference (2002)
18. Kwiatkowska, M., Norman, G., Parker, D.: Probabilistic model checking in practice: Case studies with PRISM. *J. ACM SIGMETRICS Performance Evaluation Review* 32(4), 16–21 (2005)
19. Kwiatkowska, M., Norman, G., Parker, D., Sproston, J.: Performance analysis of probabilistic timed automata using digital clocks. *J. Formal Methods in System Design* 29, 33–78 (2006)
20. Kwiatkowska, M., Norman, G., Segala, R., Sproston, J.: Automatic verification of real-time systems with discrete probability distributions. *J. Theoretical Computer Science* 282, 101–150 (2002)
21. Kwiatkowska, M., Norman, G., Sproston, J.: Probabilistic model checking of the IEEE 802.11 wireless local area network protocol. In: Hermanns, H., Segala, R. (eds.) PROBMIV 2002, PAPM-PROBMIV 2002, and PAPM 2002. LNCS, vol. 2399, pp. 169–187. Springer, Heidelberg (2002)
22. Kwiatkowska, M., Norman, G., Sproston, J.: Probabilistic model checking of deadline properties in the IEEE 1394 FireWire root contention protocol. *J. Formal Aspects of Computing* 14(3), 295–318 (2003)
23. Laleau, R., Vignes, S., Ledru, Y., Lemoine, M., Bert, D., Donzeau-Gouge, V., Dubois, C., Peureux, F.: Application of Requirements Analysis Techniques to the Analysis of Civil Aviation Security Standards. In: Ralyté, J., Ågerfalk, P.J., Kraiem, N. (eds.) Proceedings of the First International Workshop on Situational Requirements Engineering Processes (SREP 2005), pp. 91–107 (2006)
24. Ledru, Y.: Project EDEMOI - An Approach to Model and Validate Airport Security. (Final report) (2006)
25. Microsoft FSE Group, The Abstract State Machine Language (2003),
<http://research.microsoft.com/fse/asml/>
26. Pendergraft, D.R., Robertson, C.V., Shrader, S.: Simulation of an Airport Passenger Security System. In: Ingalls, R.G., Rossetti, M.D., Smith, J.S., Peters, B.A. (eds.) Proceedings of the 2004 Winter Simulation Conference, pp. 874–878 (2004)
27. PRISM Web Site, <http://www.cs.bham.ac.uk/~dxp/prism>

28. Rutten, J., Kwiatkowska, M., Norman, G., Parker, D.: Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems. In: Panangaden, P., van Breugel, F. (eds.). CRM Monograph Series, vol. 23. American Mathematical Society (2004)
29. Stärk, R., Schmid, J., Börger, E.: Java and Java Virtual Machine: Definition, Verification, Validation. Springer, Heidelberg (2001)
30. Takakuwa, S., Oyama, T.: Simulation Analysis of International-Departure Passenger Flows in an Airport Terminal. In: Chick, S., Sanchez, P.J., Ferrin, D., Morrice, D.J. (eds.) Proceedings of the 2003 Winter Simulation Conference (2003)
31. The European Parliament and the Council of the European Union, Regulation (EC) No 2320/2002 of the European Parliament and of the Council - Establishing Common Rules in the Field of Civil Aviation Security (2002)
32. Wilson, D.L.: Use of Modeling and Simulation to Support Airport Security. IEEE Aerospace and Electronic Systems Magazine 20(8), 3–6 (2005)

Online Resources

- Abstract State Machine Research Center: <http://www.asmcenter.org> The main goal of the ASM Research Center is to support the ASM method for rigorous modeling and analysis of complex computational systems through its applications in a variety of domains and its theoretical and practical enhancements.
- CoreASM: an extensible ASM execution engine.<http://www.coreasm.org/> The CoreASM project focuses on the design of a lean executable ASM (Abstract State Machines) language, in combination with a supporting tool environment for high-level design, experimental validation and formal verification (where appropriate) of abstract system models. The tool is open-source and freely available from the website.
- EDEMOI: modeling airport security.<http://www-lsr.imag.fr/EDEMOI/> The EDEMOI project aims to provide an industrial methodology and tools to build models from these natural language documents, and use specialized tools to study their consistency.
- PRISM: probabilistic symbolic model checker. <http://www.cs.bham.ac.uk/~dpx/prism/> PRISM is open source software and is freely from the website. The website also provides a number of case-studies and a comprehensive list of related projects.
- Safeguard: A novel computational approach to aviation security. <http://www.cs.sfu.ca/~se/projects/safeguard> Technical reports on the project as well as detailed description of the case study presented here are available on the website.

Questions for Discussions

1. What are the main challenges that regulators and designers of aviation security systems are facing? How can systematic analytical approaches help in dealing with these challenges?
2. What exactly is a process flow model? How can this modeling paradigm be utilized in the context of security procedures and routines?
3. What is an ASM ground model? What are the advantages (or possible disadvantages) of using the ASM ground model method in analyzing abstract functional requirements on security systems?

4. Why is the use of probabilistic modeling techniques important for capturing characteristic properties of aviation security systems?
5. What is the role of nondeterminism vs. probabilistic choice in modeling complex behavior? What is the trade-off between these two choices with respect to systematic analysis of the resulting behavior?
6. What are the main elements in the cost-effect analysis of a security system? How can one more precisely characterize the notion of acceptable security at reasonable cost' and how can this goal be achieved?
7. How can the approach discussed in this chapter be utilized to effectively improve existing aviation security standards and practices? In your experience, what are advantages and disadvantages of this approach?
8. Provide examples of other application domains where the proposed approach may be applied.
9. In this chapter, we have investigated the effect of time on safety and cost. What other parameters contribute to cost and/or safety of screening operations within an airport?

Anomaly Detection in Moving Object

Xiaolei Li, Jiawei Han, Sangkyum Kim, and Hector Gonzalez

University of Illinois at Urbana-Champaign, USA

Abstract. With recent advances in sensory and mobile computing technology, many interesting applications involving moving objects have emerged. One of them is *identification of suspicious movements*: an important problem in homeland security. The objects in question can be vehicles, airplanes, or ships; however, due to the sheer volume of data and the complexities within, manual inspection of the moving objects would require too much manpower. Thus, an automated or semi-automated solution to this problem would be very helpful. That said, it is challenging to develop a method that can efficiently and effectively detect anomalies. The problem is exacerbated by the fact that anomalies may occur at arbitrary levels of abstraction and be associated with multiple granularity of spatiotemporal features.

In this study, we propose a new framework named ROAM (Rule- and Motif-based Anomaly Detection in Moving Objects). In ROAM, object trajectories are expressed using discrete pattern fragments called *motifs*. Associated features are extracted to form a hierarchical feature space, which facilitates a multi-resolution view of the data. We also develop a general-purpose, rule-based classifier which explores the structured feature space and learns effective rules at multiple levels of granularity. Such rules are easily readable and can be easily provided to humans to aid better inspection of moving objects.

19.1 Introduction

Moving object research has been gathering much momentum in recent years. With more and more historical or real-time data on moving objects being collected, more and more applications are becoming viable. Technologies such as GPS devices, RFID sensors, RADAR, and satellites allow trajectory information to be recorded for objects of all sizes, whether it be a tiny cellphone or a giant ocean liner.

One particular problem that is of interest in homeland security and surveillance is automated detection of suspicious or anomalous moving objects. For instance, at any one time, there are over 160,000 vessels traveling in the United States' waters. They include military cruisers, private yachts, commercial liners, and so on. In past years, they have been examined manually; but the cost of doing so is staggering. Thus, it has become highly desirable to create automated tools that can evaluate the behavior of all maritime vessels and develop situational awareness on the abnormal ones.

In addition to the maritime application, many other homeland security tasks can benefit from an automated solution in detecting anomalies in moving objects. More and more cities are installing closed-circuit cameras to prevent terrorism. Through algorithms in vision and pattern recognition, moving objects such as cars or humans can be extracted from these videos and provided to an automated algorithm. GPS devices

embedded in cellphones or vehicles can also provide similar data; as would automated toll-booths. Another direct application is the monitoring of airplanes. In addition to the big commercial jets, there are many small private planes that roam the skies. A radar operator would have a hard time in tracking all of them manually. An automated algorithm that points out suspicious behavior to the operator would be very useful in better directing the operator's attention.

An outlier is, in general, viewed as “*an observation (or a set of observations) which appears to be inconsistent with the remainder of that set of data* [2].” However, the term “inconsistent” has many far-reaching implications. The decision is often subjective and depends heavily on the context. Outliers are also rare in the population, which makes search harder.

Though outlier detection has been studied in many contexts [2, 14], the moving objects domain [9] poses unique challenges. Additionally, problems such as indexing [19, 21], clustering [13, 8, 12], anomaly detection [17, 16] have been studied in moving objects domain as well. However, they focus almost exclusively on the trajectories. In practice, trajectories are associated with non-spatiotemporal features and such associations are often more valuable for analysis. In this paper, we take a different approach by constructing a multi-dimensional feature space oriented on segmented trajectories. This allows us to analyze complex relationships between multiple features associated with different granularities and dimensions in each trajectory.

There are in general two mechanisms for anomaly detection: *classification*, which relies on labeled training data sets, and *clustering*, which performs automated grouping without the aid of training data. Although both are interesting methods for mining moving object outliers, classification often leads to stronger mining results with the help of training data. Therefore, our focus will be on constructing a classification model.

19.2 Trajectory Representation

The problem of anomaly detection in moving object data is defined as follows. The input data is a set of **labeled trajectories**: $D = \{(t_1, c_1), (t_2, c_2), \dots\}$, where t_i is a trajectory and c_i is the associated class label. A *trajectory*¹ is a sequence of spatiotemporal records of a moving object, e.g., GPS records. Each record has the geographic location as well as a timestamp, and records can be made at arbitrary time intervals. The set of possible class labels is $C = \{c_1, c_2, \dots\}$. In simple anomaly detection, there could just be two classes: c^{normal} and c^{abnormal} .

The goal of the problem is to learn a function f which maps trajectories to class labels: $f(t) \rightarrow c \in C$. f should be consistent with D as well as future trajectories not in D . In other words, we want to learn a model which can classify trajectories as being normal or abnormal.

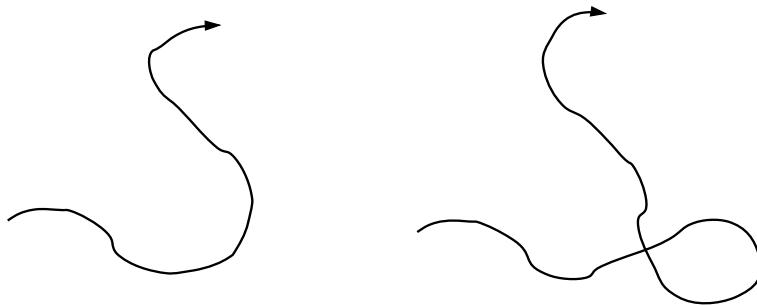
19.2.1 Representation with Motifs

There have been some prior work in the area of trajectory prediction [16, 17]. Markov models or other sequential models can model a single trajectory and predict its future

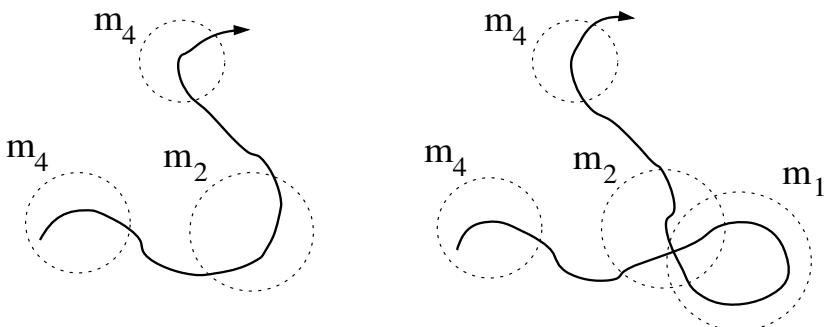
¹ Trajectory in this paper is just data and does not imply path prediction.

behavior. However, when used in the context of a large population with many different distributions, such approaches may not be effective.

Consider the two trajectories in Fig. 19.1. They have similar shapes except the one on the right has an extra loop. The impact of this additional loop depends on the task, but one would remark that the other portions are remarkably similar.



(a) Two similar trajectories. The loop in the right trajectory is difficult to handle in holistic approaches



(b) Same two trajectories after motif extraction. The right trajectory has an extra m_1 .

Fig. 19.1. Motif representation

This example presents some problems for holistic models. It is difficult to represent the semantics of “mostly the same with the exception of an extra loop” using distance metrics between models. Local differences could either dominate the metric or be drowned out by the rest of trajectory. Furthermore, it is difficult to capture thousands or tens of thousands of trajectories in a single model. While a single object or a small set may have clear patterns, a large population (such as in real-world anomaly detection) presents a wide range of patterns across all granularities of time and space signals.

19.2.2 Motif-Based Feature Space

In this paper, we propose that semantic analysis should be based on a rich feature space constructed using trajectory fragments. In ROAM, raw trajectories are partitioned into fragments. These fragments are overlaid on top of each other and the common patterns become what we call **motifs**. Motifs are represented by a tuple (motif expression) which includes additional spatiotemporal attributes that may be helpful in analysis. The set of motif expressions observed then forms a feature space in which the original trajectories are placed. Using such a feature space, we can leverage algorithms in machine learning and data mining to learn complex associations between trajectory fragments and also other important information.

A *motif* is a prototypical movement pattern. Examples include *right turn*, *u-turn*, and *loop*. One could view them as parallels to gene expressions in DNA sequences or entity mentions in text. Fig. 19.1(b) shows the motifs in Fig. 19.1(a) as drawn by the dotted circles. In this form, the two trajectories now have much in common: They share one m_2 and two m_4 's, and differ in one m_1 (i.e. *loop* motif).

19.2.3 Multi-resolution Feature Hierarchies

Another observation we make is raw recordings and semantic analysis often occur at different spatiotemporal granularities. While time recordings may be made at the minute or second level, analysis is usually more sensible on the hour level or even higher. The same scenario applies to the location measure. One might record at the meter level but analyze at the city block or district level.

By using a more general representation, fewer distinct measure values are used and the analysis task could become easier. In addition, it would improve human readability. If the final classification model produces human readable results (as ROAM does), having high level features not only reduces the size of the results but also increases their understandability.

Sometimes, these hierarchies are readily available, as in the case of time. With other features, however, it may not be obvious. In ROAM, we use a clustering-based technique to automatically extract hierarchies based on the behavior of the trajectories. Given such hierarchies, it is still hard to know *a priori* which levels will work the best for classification so we let ROAM adjust dynamically.

19.3 Framework

Fig. 19.2 shows the ROAM (Rule- and Motif-based Anomaly Detection of Moving Objects) framework. Square boxes are computation modules, round boxes are data sources, and arrows show the flow of data. There are three computation modules in ROAM: Motif Extractor, Feature Generator, and Hierarchical Rule-based Classifier. Data flows through them in that sequence.

19.3.1 Motif Extractor

The first computation module in ROAM is the Motif Extractor. A motif is a prototypical movement pattern. ROAM use a sliding window technique to process the

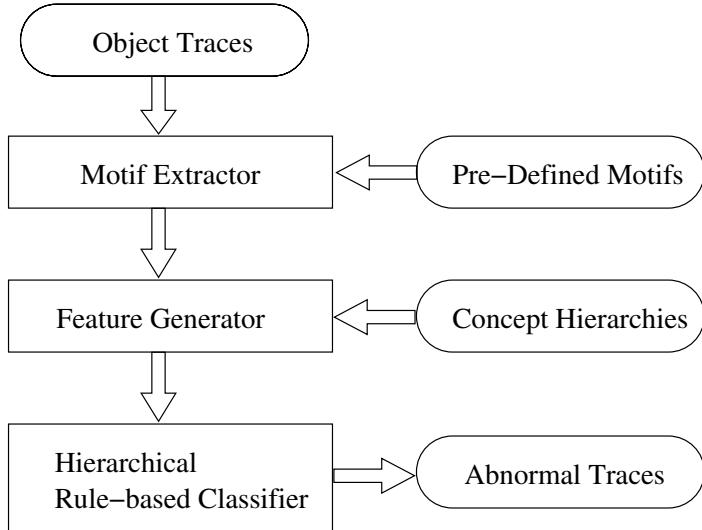


Fig. 19.2. ROAM Framework

trajectories. All windows are overlaid on top of each other and clustering is used to group them into representative sets. These representative ones then form the set of interesting *motifs* in D .

Given a trajectory, we slide a window of length ω across it. ω could be defined with respect to time or distance. If it is time, then different speeds (and thus distance traveled) would result in different motifs. If it is distance, then speed variances would be normalized. In our experiments, we used time since speed was relatively stable though more complex data might require more complex normalization methods. For each resultant window w , we compute the vector from the first point in w to the last point in w . The width of the vector is then expanded to accommodate all other points within w ; this bounding box allow us to smooth over noises in the trajectory.

All bounding boxes are overlaid on top of each other. And using the Euclidean distance, we cluster them to find the representative patterns. The resultant cluster centers then define the set of motifs. In ROAM, a motif is represented just like a window: a vector with a bounding box around it. Depending on the task, other variables may be kept as well. Once the motifs are set, we then go through D again using the same sliding windows. This time, a window w in a trajectory is *similar* to a motif m if $\|w - m\| \leq \varepsilon$. And if a particular window is similar to a motif, we say that motif is “expressed” in the trajectory.

A natural question to raise is how to set ω . A too small of a value could miss motifs by dividing them into indistinguishable pieces and a too large of a value could bundle motifs together and lose discriminative power. Fortunately, it turns out that most reasonable values will perform just fine. As we will show empirically in Sect. 19.4, classification accuracy is fairly robust with regards to different ω values.

Given a trajectory, the motif extractor returns the sequence of **motif expressions** found in the trajectory. Each motif expression has the form

$$(m_i, t_{start}, t_{end}, l_{start}, l_{end}) \quad (19.1)$$

where m_i is the motif, t_{start} and t_{end} are the starting and ending times, and l_{start} and l_{end} are the starting and ending locations. The complete sequence is known as the **motif trajectory** of the original trajectory.

Motif Expression Attributes

The form of motif expression shown in Eq. 19.1 is only the first step in full motif expression extraction. Additional information on when, where, and how the motif was expressed is needed. Take Fig. 19.3 as an example. There are two objects moving in an area with the same trajectories; however, the left one is near an important landmark. This extra piece of information (i.e., proximity to landmark) can be crucial in decision making. If we also knew that the left object was moving slowly during the middle of the night, the combination of all such information is telling in anomaly detection.

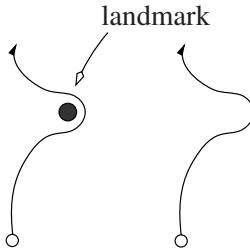


Fig. 19.3. Two objects moving with the same trajectory

For each motif expression, we introduce a set of **attributes** in addition to the simple time and location ones in Eq. 19.1. Some examples include duration, top_speed, avg_speed, radius, and general_location. Some of these attributes can be derived easily from the time and location attributes, e.g., $\text{avg_speed} = \text{path-distance}(l_{start}, l_{end}) \div (t_{start}, t_{end})$. Others may require a more sophisticated motif extractor. Let there be A such attributes: $\{a_1, a_2, \dots, a_A\}$. We now represent each motif expression as follows,

$$(m_i, v_1, v_2, \dots, v_A) \quad (19.2)$$

where m_i is the motif and v_i is the value of attribute a_i . Note that a_i may be continuous or even multi-dimensional.

19.3.2 Feature Generator

Once the motif expressions have been extracted, semantic analysis can begin. One could try the following naïve classification scheme. For each distinct (*motif, attribute, attribute value*) combination we see in the trajectory data, we map it to a feature. For example, (right-turn, speed, 11mph) would map to a feature and (right-turn, speed,

$12mph$) would map to another feature. Formally, $\forall i, j, k; (m_i, a_j, v_k) \leftrightarrow f_x \in F$ where F is the resulting feature space. We then use the following classifier.

Algorithm 1 (Flat-Classifier)

1. Transform the motif-trajectories into vectors in the F feature space. Suppose $f_x \leftrightarrow (m_i, a_j, v_k)$. Then the x_{th} component of the vector has the value of the number of times motif i 's attribute j expressed value k in the trajectory.
2. Feed the feature space and the data points as input into a learning machine.

This particular transformation from trajectories to a feature space is complete. Every motif attribute value is preserved as a feature and the frequencies of their expressions are preserved as the feature values. However, it is ineffective by the following observations. First, a large number of distinct motif attribute values leads directly to a high dimensional feature space. Specifically, suppose there are M motifs, A attributes, and each attribute has V distinct possible values. The instance space is then N^{MAV} . Second, the high granularity or continuous motif attribute values make generalization difficult. Because these distinct values are transformed to distinct features, generalization becomes essentially impossible. Learning on a feature that is at 10:31am will have no bearing on a feature that is at 10:32am.

Feature Generalization

In order to overcome the difficulties in the FLAT-CLASSIFIER, generalization in the feature space is needed. For example, (*right-turn*, *time*, 2am), (*right-turn*, *time*, 3am), and (*right-turn*, *time* }, 4am) features could be generalized into one feature: (*right-turn*, *time*, *early_morning*). This not only reduces the dimensionality of the feature space but also helps the learning machine through feature extraction.

Recall that each feature has the form (m_i, a_j, v_k) , where each attribute a_j is either numerical (1D) or spatiotemporal (2D or 3D). We assume that each a_j has a distance metric defined on its values. Thus, features having the same m_i and a_j values (*i.e.*, $(m_i, a_j, *)$) can be compared with a formal distance metric. For example, (*right-turn*, *time*, 2am) is more similar to (*right-turn*, *time*, 2:02am) than (*right-turn*, *text*, 6pm). But, it does not make sense to compare features with different m_i or a_j values (*e.g.*, (*right-turn*, *time*, 2am) is not comparable to (*u-turn*, *speed*, 10mph)).

We partition the features in F into sets with distinct (m_i, a_j) values. If there are M motifs and A attributes, there are $M \times A$ disjoint sets. We propose to generalize the features in each (m_i, a_j) set into a smaller set. Further, this new set will be hierarchical where appropriate. This will be the task of the **Feature Generator** in the ROAM framework. Specifically, it will

1. discretize or cluster continuous or high granularity motif attribute values.
2. form a hierarchy over the attribute values, which in turn offers a multi-resolution view of the data.

We will treat each (m_i, a_j) space independently. Since the attribute values can have different forms (e.g., numerical values, 2D spatial locations), we will use different methods where appropriate. We explain them in detail in the following two sections.

Spatial Attributes

Attributes such as `location` are spatial points in a 2D or 3D space. In such scenarios, we use a hierarchical “micro-clustering” technique similar to BIRCH [28] to discover prototypical patterns. Features are inserted into a tree-based data structure where nodes represent micro-clusters. A micro-cluster is a small, tightly grouped neighborhood, and features belong to the same micro-cluster only when they are closely related. A tree of these micro-clusters represents a concept hierarchy of the attribute.

Take the `location` attribute as an example. A micro-cluster may only include features which are within a few meters of each other. During insertion of features into the tree, each micro-cluster has a maximum radius parameter. If a feature cannot fit inside a micro-cluster, a new micro-cluster is created. The tree also had a maximum branching factor so insertions and rotations occur like a typical B-tree. After all features have been inserted into the tree, the leaf nodes form the set of micro-clusters. Each micro-cluster can be viewed as a meta data point that represents similar features. We then feed the set of micro-clusters into a hierarchical agglomerative clustering algorithm to construct the final hierarchy.

The final clustering tree is hierarchical in the following sense: any node in the tree contains summarized information for all data points in that node’s subtree. For example, the root contains a summary of the entire tree. The summary information is sufficient for the calculation of the centroid and the radius of the points. The reason we choose a BIRCH-like algorithm in our system is two-fold. First, it performs micro-clustering, which fits our needs better. Second, building the CF tree is time and space efficient ($O(n)$). More properties are described in [28].

Numerical Attributes

Attributes such as `time` and `avg_speed` are numerical. Usually, in the presence of continuous attributes, *discretization* is performed. Doing so has many advantages. First, it makes the learning problem easier. A decision tree, for example, would have fewer splits to consider. A discrete feature allows better generalization. Second, it makes human readability easier. For instance, it is much easier to understand the feature value of 1pm–2pm as opposed to reading all the distinct values between 1pm and 2pm.

Discretization techniques [18] can be split into two main groups: unsupervised and supervised. Since we have labeled data, supervised algorithms are more appropriate. There is an abundant number of methods available for this; most of them would function just fine here. An additional requirement we have is a hierarchy over the resultant discrete values. While most discretization methods do not have this property, we can easily add it by performing hierarchical agglomerative clustering as a post-processing step.

Since spatial attributes are a generalization of numerical attributes, we use the same clustering methods in our implementation of ROAM for both types of attributes. Clustering in one-dimensional data still provides meaningful groupings based on behavior.

Multi-resolution View

After building hierarchies in each of the (m_i, a_j) spaces, the overall feature space is now structured as a set of hierarchies. Fig. 19.4 shows a partial illustration. In it, there are two **motif-attribute hierarchies**: $(right-turn, \text{time})$ and $(right-turn, \text{speed})$. Each node corresponds to a micro-cluster feature discovered in F . For example, the black node in Fig. 19.4 represents all right-turns taken between 2 and 8am. High level nodes in the hierarchies correspond to high level features and low level nodes correspond to low level features. By choosing different subsets of the nodes, a user can create distinctly different views of the data. For example, suppose one only used level one features in Fig. 19.4 (*i.e.*, “morning”, “slow”, etc). This generates a very rough view of the data and with only four features. On the other hand, choosing the leaf nodes in Fig. 19.4 generates a detailed view but with many more features.

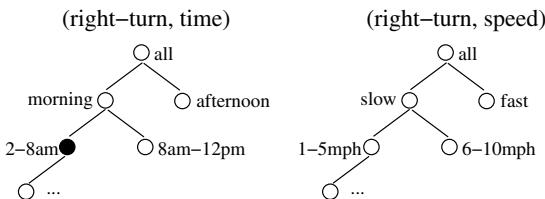


Fig. 19.4. Two sample motif-attribute hierarchies

As mentioned previously, concept hierarchies may already exist for some attributes (*e.g.*, time). In such cases, one may just choose to use them to construct the motif-attribute hierarchy. However, in other cases or sometimes in place of the existing hierarchies, one could use automated techniques in ROAM to construct the hierarchies. This has the distinct advantage that the hierarchies are built based on the behavior of the data. As a result, more features could be dedicated to the dense regions and fewer features to the sparse regions. Clustering and discretization techniques can adjust dynamically based on the data and could facilitate more effective analysis.

19.3.3 Classification

With the features extracted, the last step in ROAM is classification. The problem is relatively classical so previous machine learning algorithms may be leveraged. We first explain how a Support Vector Machine (SVM) can be applied and then introduce our own classifier to take advantage of certain properties within the ROAM feature space.

Support Vector Machine

In machine learning theory, the class boundary function is judged on its generalization performance. That is, how does it perform on unseen data as opposed to the training data? For instance, a rote learner (*i.e.*, one which simply remembers data to class label pairs) can produce 100% accuracy on the training data but would fail miserably at unseen data. Often, classifiers such as the rote learner fall into the trap of overfitting to

the training data. SVM is a classification method that tries to maximize the generalization performance by maximizing the *margin*.

The margin in SVM denotes the distance from the class boundary to the closest data points in the feature space. Fig. 19.5 shows an example; in it, the class boundary is positioned in such a way that the margin is maximized. In addition to the margin maximization, SVM also have several other nice properties. First, it may “project” the data points to a higher dimensional or possibly infinitely dimensional space. In doing so, it may become easier to find a good class boundary. Second, a shortcut is available in such cases that avoids the actual creation of the high dimensional feature space. More details of SVM can be found in [3, 24].

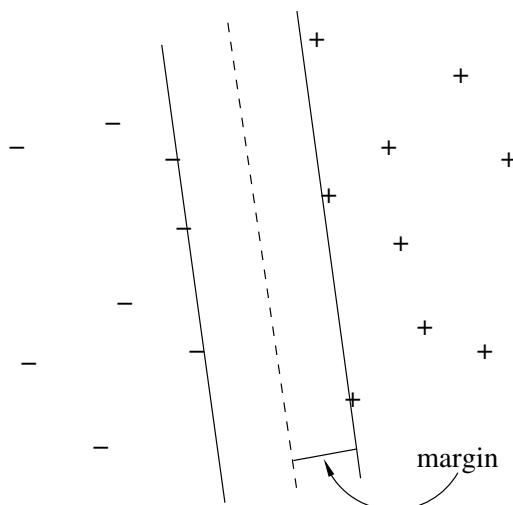


Fig. 19.5. Margins in SVM

CHIP Intuitions

Let F' be the set of all nodes in the motif-attribute hierarchies. F' is the largest feature space where all resolutions are included. Though this feature space is “complete”, it is unlikely to be the best one for classification. It creates a high dimensional feature space; this makes learning slow and possibly ineffective. On the other hand, suppose we choose only the root level nodes in all the motif-attribute hierarchies. That is, only the “all” nodes in Fig. 19.4. The problem with this feature space is that the features are too general to be useful. What we seek is something in between those two extremes.

To this end, we propose a rule-based classification method: **CHIP** (Classification Hierarchical Interpretable Prediction Rules). We chose a rule-based learning algorithm for several reasons. First and foremost, it produces human-readable results. This is useful in practice. Second, it is efficient. **CHIP** is $O(N)$ with respect to either the number of examples or the number of features. Other classifiers such as Naïve Bayes or SVM are $O(N^2)$ with respect to one. The problems we are dealing could be large and high dimensional. Lastly, the classification problem is unbalanced: the abnormal

class has few training examples. In such contexts, rule-based learner have been shown to be effective [6].

Before formally describing **CHIP**, we give some intuitions. **CHIP** iteratively and greedily searches for the best available rule until all positive examples are covered. In addition, **CHIP** tries to use high-level features whenever possible. For example, suppose all ships that move at location X between the hours of *12pm* and *5pm* are normal. Then a single rule using the *afternoon* feature will suffice. Using this principle has several benefits. First, the feature space is kept to a small size. This speeds up learning and keeps the problem tractable. Second, features are kept high level whenever possible. This produces rules which are general and easily understood by a human. Third, it avoids the problem of over-fitting.

In machine learning research, the study of feature space simplification or generalization is known as *feature selection* [11]. Given a set features, choose a subset which will perform better (in terms of efficiency and/or accuracy) in the learning task. A typical approach scores each feature and iteratively inserts or removes them. In our setting, however, we have something that is different than the standard setting: there are hierarchical structures over the features. Thus, selection should be a little smarter.

With this in mind, we propose a top-down search in the feature hierarchies. We start with an initial high level feature space and try to describe the data (in the rules sense). If these features produce accurate rules, we are satisfied. But if at some point we find that a more specific feature will produce a better rule, we *expand* the existing feature space to include that specific feature. This process repeats until all the training data is sufficiently covered.

CHIP Algorithm

We introduce some definitions first. **CHIP** learns a set of rules, much like **FOIL** [20] and **CPAR** [27]. A single rule r has the conjunctive form of:

$$l_1 \wedge l_2 \wedge \dots \wedge l_n \rightarrow c$$

where each l_i is a literal (or predicate) of the form (*feature* = *value*) and c is the class label. An example is “covered” by r if all the literals in r are satisfied in the example. Next, recall that F' is the complete set of features. For any feature f in F' , let $Exp(f)$ return the set of f 's children in F' 's hierarchy. For example, $Exp(\text{morning}) = \{2 - 8\text{am}, 8\text{am}-12\text{pm}\}$. At any time, **CHIP** uses a subset of the features in F' . Let F_C be this set.

A rule is learned one literal at a time. Literals are selected according to a weighted version of *Foil_Gain* [20], which is based on the positive and negative coverage of the rule before and after adding the literal. Let p_0 and n_0 be the number of positive and negative examples covered by rule r without literal l . Let p_1 and n_1 be the number of positive and negative examples covered by rule $r \wedge l$. *Foil_Gain*(l, r) is then defined as

$$p_1 \left(\log_2 \frac{p_1}{p_1 + n_1} - \log_2 \frac{p_0}{p_0 + n_0} \right)$$

The weighted version of *Foil_Gain* [27] allows previously covered positive examples to be used again but just weighs them down. This adjusts the p and n values appropriately in the above equation.

In the previous section, we gave the intuitive notion of discovering that a more specific feature will perform better than a current feature. Here, we formalize this notion in the function $Exp_Gain(f, r)$ where f is a feature and r is a rule. It is defined as

$$Exp_Gain(f, r) = \max_{(l, f_i) \forall l, f_i \in Exp(f)} Foil_Gain(l, r)$$

The *Exp_Gain* (expansion gain) of a feature is the maximum *Foil_Gain* achieved by any literal in any of its child features. It is defined with respect to a non-empty rule similar to *Foil_Gain*. We chose this function because it allows sensible direct numerical comparisons between *Foil_Gain* and *Exp_Gain*.

Algorithm 2 (CHIP)

Input: (1) Training set $D = P \cup N$, where P and N are the positive and negative examples. (2) Initial feature set $F_C \in F'$

Output: Set of classification rules R

Method:

1. **while** not all of P is covered
2. initialize new rule r
3. **while true**
4. find literal l with highest $Foil_Gain(l, r)$
5. find feature f with highest $Exp_Gain(f, r) \setminus\setminus$
6. if both gains < min_gain **then break**
7. if $Foil_Gain(l, r) > \beta Exp_Gain(f, r)$ **then**
8. add l to r
9. **else**
10. add feature f to F_C
11. add r to R
12. **return** R

Discussion. CHIP starts with all examples uncovered and iteratively searches for the best rule to cover the positive examples. The search is greedy and halts when enough positive examples are covered. Rules are learned one literal at a time, choosing them based on *Foil_Gain* (line 4). In line 5, the *Exp_Gain* of each feature is calculated. If the better gain is *Foil_Gain*, the literal is added to the current rule (line 8). Otherwise, the feature space is expanded (line 10) and the process repeats.

Complexity. CHIP has running time of $O(nSR)$ where n is the number of examples, S is the size of the used feature space, and R is the number of learned rules. In our implementation, we collapsed examples (trajectories) which appear the same into meta-examples. Thus, with a high initial feature space, n can be quite small if the data is skewed. S can also be small initially since there are only a few high level features. As the algorithm executes, both n and S will increase with feature expansion. This is another reason to avoid careless expansion.

19.4 Experiments

In this section, we show our framework's performance in a variety of settings. We conduct our experiments using both real and generated data to show efficiency and effectiveness. For data generation, we used GSTD [22] (which generates raw trajectories) and also our own data generator (which generates motif-trajectories). The latter allows us to test some parts of the framework independently of others. Efficiency experiments were run on an Intel Pentium 4 2.6GHz machine with 1.5GB of memory. The Motif Extractor was written in Python and the rest was written in C++ and compiled with GCC.

Each dataset consists of two classes: normal and abnormal. In GSTD, we achieve this by generating two datasets with slightly different parameters. In our own generator, the base data is motif-trajectories. A set of motif-expression seeds are initialized in the model and a Gaussian mixture model is used to create randomness. We generate the abnormal class by mixing “abnormal” motifs with a background model that is shared between both the normal and abnormal classes.

There are two parameters which controls **CHIP**. One is the starting level in the motif-attribute hierarchy. Level 0 denotes the root level. The other is β , the feature expansion weight. These two parameters are indicated as **ROAM** (*starting_level*, β). Finally, since the number of abnormal examples is small, we used the standard F1 metric instead of accuracy. $F1^2$ is a harmonic mean of recall and precision and better reflects the effectiveness of the classifier. An F1 score of 100 indicates 100% recall and precision. F1 scores were the result of 10-fold cross validation. Experiments were run 5 times to get an average.

19.4.1 Real Data

We obtained real ship navigational data from the Monterey Bay Aquarium Research Institute (MBARI³). Under the MUSE project, several ships traveled in ocean waters near Northern California to conduct various aquatic experiments. The ships' navigational data, which includes time, longitude, and latitude, were recorded. Depending on the ship, the sampling rate varied from 10 seconds to a few minutes; the end result are fairly continuous paths. Fig. 19.6 shows a typical path of a vessel named Point Sur.

We collected data from two different ships (namely Point Sur and Los Lobos) and assigned different class labels to them. The two ships carried out different tasks and thus naturally had different movement patterns. There was a total of 23 paths (11 of one, 12 of another), each with 1500 to 4000 points. Using **ROAM**, we extracted 40 motifs, constructed features, and tried to recover the class labels using **CHIP**. Fig. 19.7 shows two sets of trajectory segments that were marked as motifs 10 and 14. Motif 10 is a simple straight trajectory towards the northeastern corner. Motif 14 is a 3-part move of going north, northwest, and then north again.

Motifs were extracted from a window of approximately 3 minutes, and had two additional attributes. One is the distance traveled, which indicates speed, and the other

² $F1 = (2 \times \text{recall} \times \text{precision}) \div (\text{recall} + \text{precision})$.

³ <http://www.mbari.org/MUSE/platforms/ships.htm>

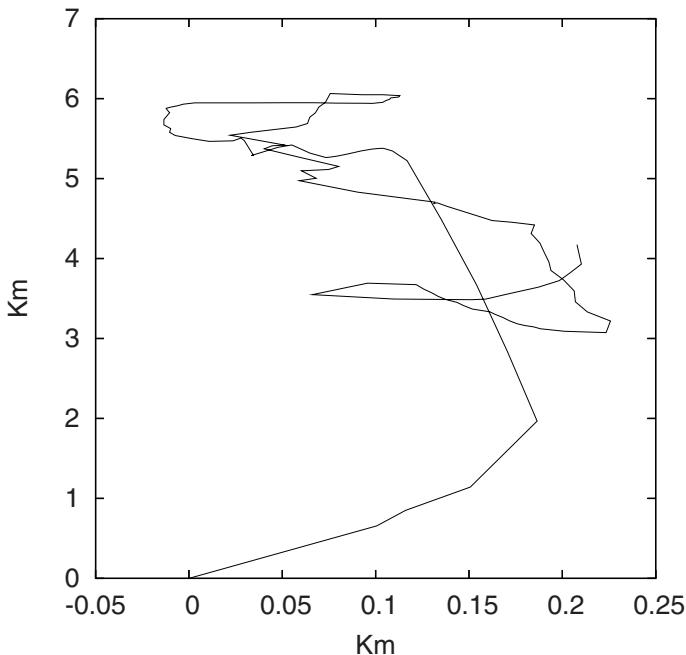


Fig. 19.6. Sample path of ship Point Sur from 16:00 to 24:00 on 8/23/00 starting at point (0,0)

is the general Euclidean distance to the stored motif. We did not include the time-of-day attribute since the two ships had regular but different schedules and including them would make the problem too easy. Motif-attribute hierarchies (branching factor of 4) were also generated, which ranged from 2 levels deep to 7 levels deep.

An issue raised before was the setting of ω , the width of the window to extract motifs. Fig. 19.8 shows the effect on classification as ω increases from 4 minutes to 60 minutes on the MBARI data. As shown, accuracy with too small or too large of a window is poor, but in the intermediate, it is relatively stable. Thus, we believe that as long as the window is reasonable, performance should not be affected too much. Another issue is how *many* motifs to extract. This was set to 40 in Fig. 19.8, and Fig. 19.9 shows the effect as that number changes from 5 to 60. The curve shows that we were able to achieve 100% classification accuracy with 10 and 15 motifs. And as the number increases, accuracy decreases but not too drastically. In general, a reasonable number should not be too difficult to find.

19.4.2 Synthetic Data

While the real data experiments provided some validation of our methods, we were unable to thoroughly test other aspects due to the small dataset size. To combat this, we experimented with synthetic data from GSTD and also our own data generator.

Notation

For our own data generator, we use the following notation to denote the parameters used in generation. Each data set's name is in the form of “N#B#M#A#S#L#”, where

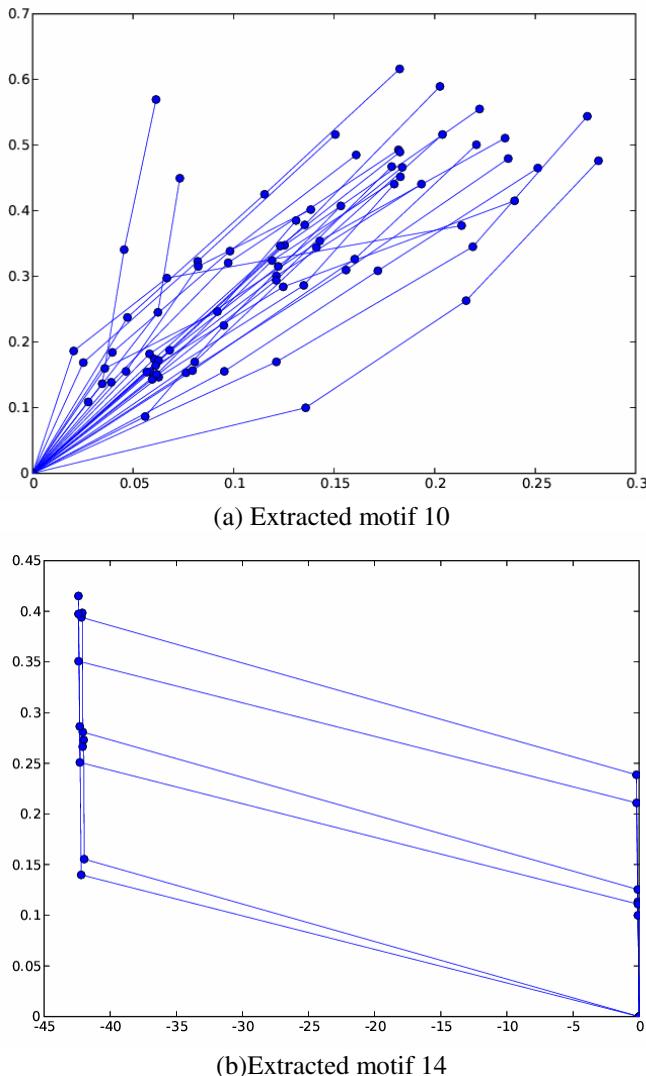


Fig. 19.7. Extracted motifs from MBARI data

N is the number of normal trajectories, B is the number of abnormal ones, M is the number of motifs, A is the number of attributes, S is the standard deviation in the Gaussian mixture distributions, and L is the average length of the trajectory.

Classification Accuracy

First, we tested accuracy using GSTD. In GSTD, we generate two different classes of data using Gaussian distributed movement centers. The two models shared the same

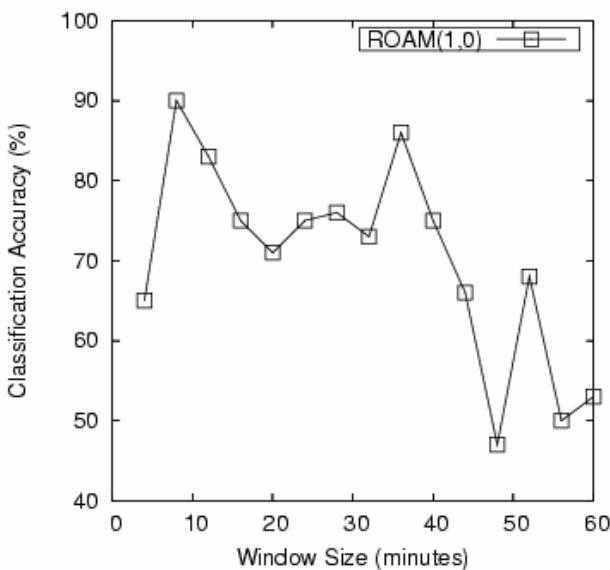


Fig. 19.8. Effect of ω on classification accuracy

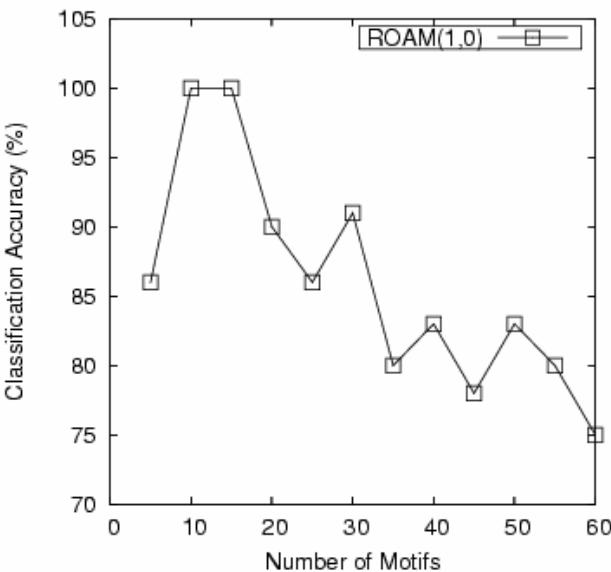


Fig. 19.9. Effect of number of motifs on classification accuracy

parameters except the mean of the centers differed by 0.01 (0.50 vs. 0.51). Fig. 19.10 shows the results as we also varied the variance in the distributions. As expected, accuracy improves as the trajectories differ more from each other. But even at small differences, ROAM was able to distinguish the two classes.

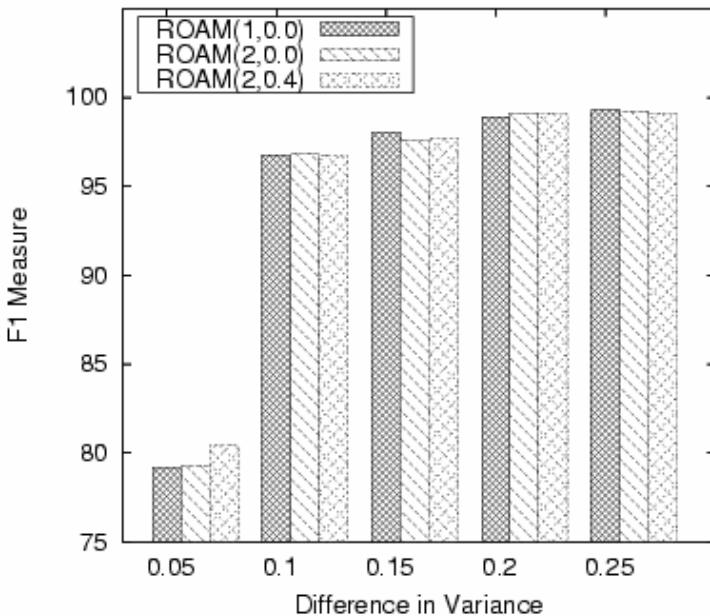


Fig. 19.10. GSTD \$N2000B200M30\$: Accuracy with respect to difference in variance

Next, we tested the accuracy using our own data generator. Fig. 19.11 shows F1 results as the number of motifs in the data increased from 10 to 100 on the y-axis. For comparison, we used SVM⁴ (nu-SVC with radial kernel) with the Flat-Classifier as described before and also SVM with level 2 features. The first thing we notice is that SVM with Flat-Classifier is hopeless, as expected. We also observe that ROAM with level 1 features and a little bit expansion is almost as good as SVM with level 2 features. ROAM with level 2 and a little bit expansion is equal to or better than SVM.

We note that the size of the classification feature space is much larger than the number of motifs. For example, when the number of motifs equals 100, the number of level 2 features equals nearly 1200.

Fig. 19.12 shows F1 as the motif-trajectory length varies. As the length increases, the data gets denser and we observe that SVM's performance deteriorates. However, ROAM with its various configurations were fairly stable. Fig. 19.13 shows the effect as standard deviation is increases from 5 to 40. As expected, F1 decreases as the values get more spread out. One might have noticed that we have rather large standard deviation values. This is because the range of values is large (~1000).

Recall that a larger value of β , the expansion factor, increases the chances that CHIP will expand the feature space during learning. The effect of different β values vary from one dataset to another. Fig. 19.14 shows a typical result. In the ROAM(1, *) curve, ROAM starts with level 1 features and improves significantly with expansion.

⁴ <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

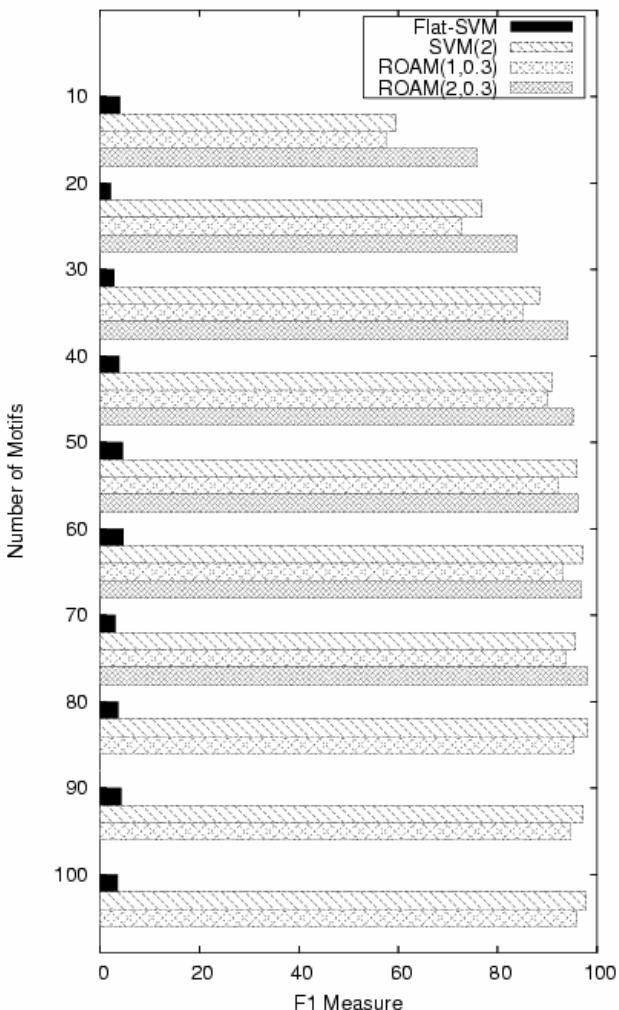


Fig. 19.11. *N4kB200A3S5.OL20*: Accuracy with respect to number of motifs

In the ROAM(2, *) curve, F1 is high initially. It improves slightly with some expansion but eventually drops down. This is the effect of over-fitting. In other words, CHIP has expanded too greedily and the feature space has become too specific.

Finally, Fig. 19.15 compares a general feature space vs. a specific one. One is ROAM(2,,0), which is level 2 features with no expansion. The other is ROAM(MAX), which is only the leaf features. We see that ROAM(2,0) is significantly better in accuracy. Furthermore, it is also faster. With 60 motifs, ROAM(2,0) took an average of 84 seconds with 705 features while ROAM(MAX) took 850 seconds with approximately 4350 features.

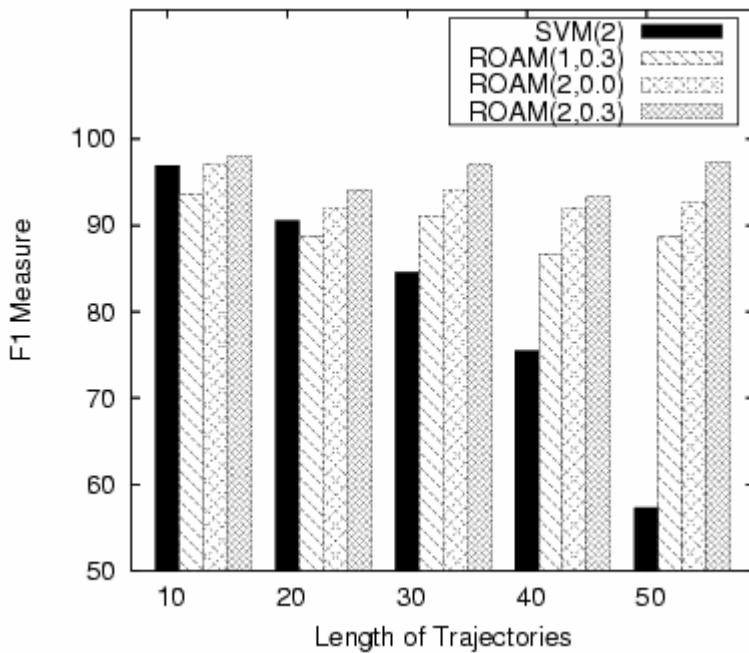


Fig. 19.12. *N4kB200A3S5.0L20*: Accuracy with respect to length of motif-trajectories

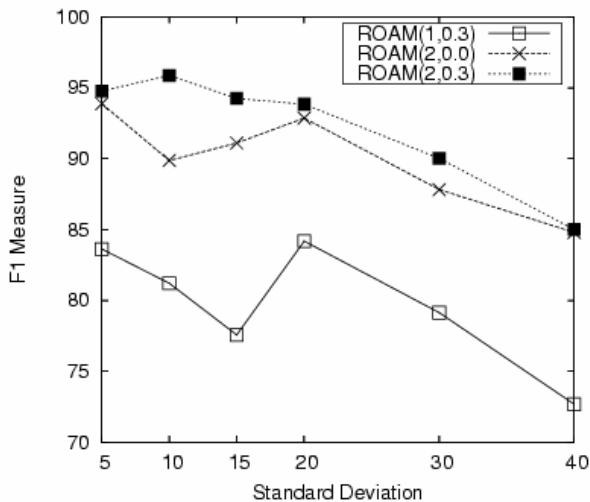


Fig. 19.13. *N5kB100M20A3L20*: Accuracy with respect to standard deviation

Efficiency

With regards to efficiency, we first check sensitivity to the number of trajectories. Fig 19.16 shows a plot broken down into ROAM's components, note the log scale. As we

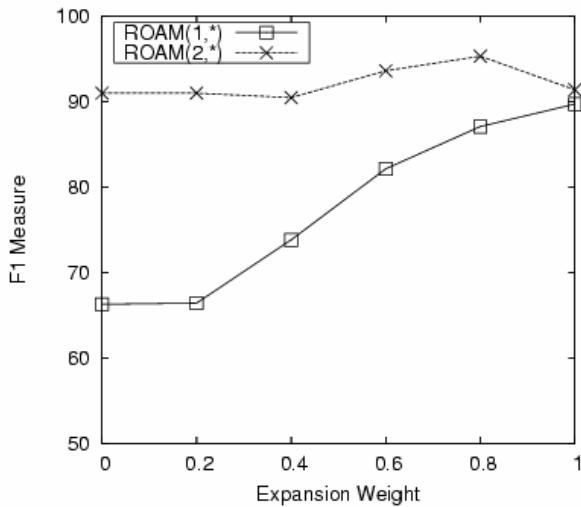


Fig. 19.14. *N500B100M20A3S25L20*: Accuracy with respect to β

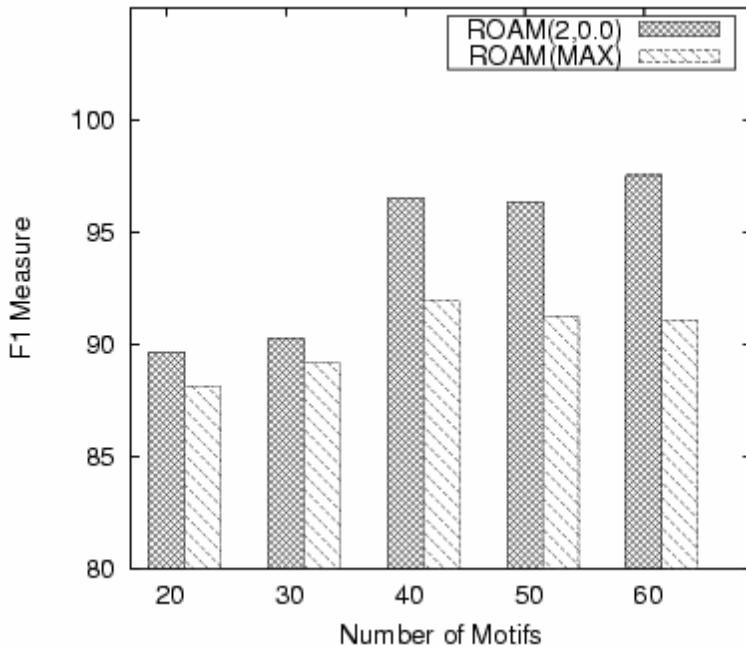


Fig. 19.15. *N15kB500A3S25L20*: Accuracy with respect to number of motifs

can see, all components scale nicely with respect to the number of trajectories. The Motif Extractor is the slowest, but it was implemented in Python (10–30 slower than C++) while the other components were in C++.

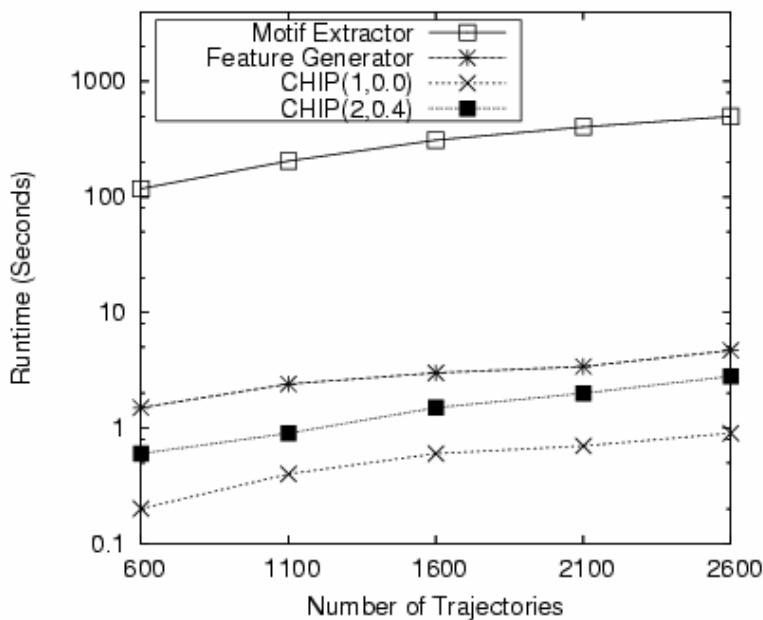


Fig. 19.16. GSTD: \$B200M20\$: Efficiency with respect to number of trajectories

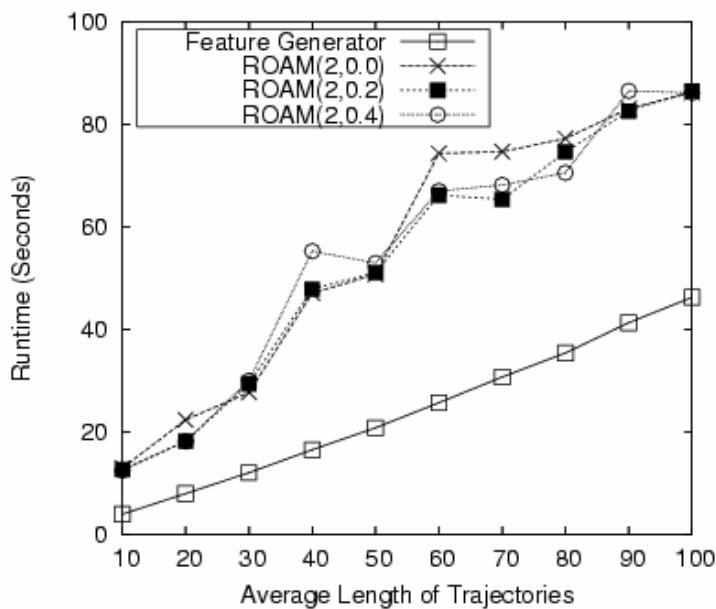


Fig. 19.17. N20000B1000M20A3S10.0: Efficiency with respect to length of motif-trajectories

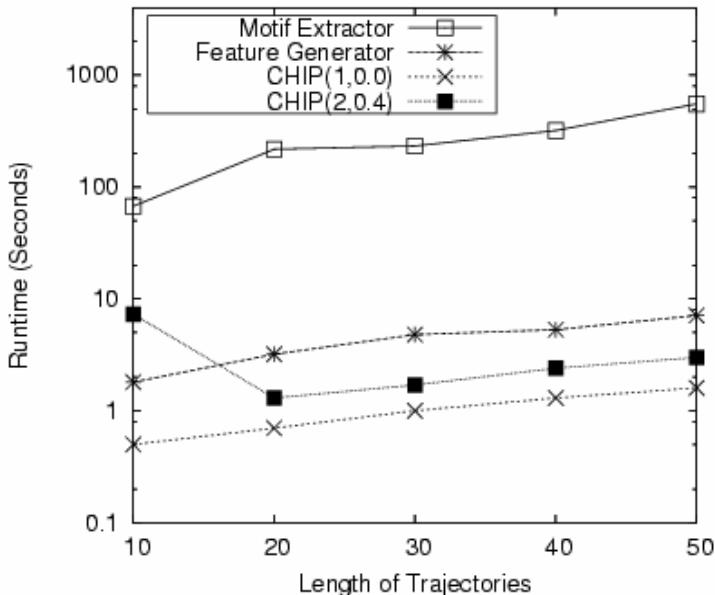


Fig. 19.18. GSTD: *N2000B100M10*: Efficiency with respect to length of trajectories

Another aspect of efficiency is sensitivity to the length of the trajectories. Fig. 19.17 shows the running time as the length was increased from 10 to 100 in our own data generator. Fig 19.18 shows a similar experiment using GSTD data. Again, we see a linear increase in running time as trajectory length increased. The reason is that with longer trajectories, there is a linear increase in the number of motif expressions ROAM has to process.

19.5 Related Work

Prior work in moving object databases (MOD) have addressed problems similar to ours. Discrete data model of moving objects was introduced in [7] and used subsequently in [17, 4, 16]. However, to our knowledge, ROAM is the first work to represent trajectories in a feature space oriented on the discrete units (fragments). Prior work used fragments as a dimensionality-reduction technique and still retained a trajectory model.

In [17, 16], the authors construct models to predict trajectories as well as recognize anomalies. However, both works assume the existence of a single trajectory model. This works well when there is only a few objects ([17] experimented with one object and [16] experimented with three). In such cases, the object(s) have clear individual patterns and could be captured in a compact model. However with anomaly detection in a large population, it is unclear whether such approaches will work. Within the population, there is a very large variation of trajectories and anomalies could occur in any form. In ROAM, we do not assume the existence of a single or a few global trajectory models. The classifier can learn rules specific to any scenario.

Clustering moving objects [8, 12] and time series [13] as well as classification of time series [26, 25] are also related to this work. However, most of them focus analysis on the raw trajectories using techniques such as dynamic time warping. In ROAM, the interactions between the trajectories and non-spatiotemporal attributes play a crucial role that is usually ignored in previous work. In real world cases, such interactions often convey the most useful information.

In addition to specific problems, representation in MOD [9] is a core issue. In [10], abstract data types are added to a DBMS to model the geometries. In [7], a discrete data model is proposed. Trajectories are decomposed into “slices” where each slice is represented by a simple function. In comparison to our work, these slices are simpler motifs and do not translate to features. Related to representation, indexing and query processing [19, 21] are also key MOD issues. However, they focus analysis on the raw spatial and trajectory data and query processing. For example, discovering which moving objects are physically located within a certain query window. In ROAM, we focus on the semantic problem of anomaly detection which requires higher level analysis.

Work in time series and traditional anomaly detection also touch on our problem. [1] presents techniques which can query for shapes in time series, and [5] automatically discovers motifs in time series. Algorithms in this area are helpful and could be applied in ROAM, but our framework does more than just motif discovery. It builds a hierarchical feature space using the motifs and performs high level feature analysis. Traditional outlier detection [14] is closely tied to our problem. However, they are only concerned with fixed data points in 2D space. We focus on moving object data.

In data mining, there are work which focus on finding frequent patterns related to moving objects. [23] mines sequential patterns in spatial locations, and [15] mines co-location association rules. However, such studies are often at a higher semantic level than ROAM. That is, they capture associations between locations but ignore patterns in raw trajectories (as well as their associations).

19.6 Discussion and Conclusion

In this paper, we have proposed the ROAM framework for the problem of anomaly detection in massive moving object data sets. With advances in tracking technology and increases in the need for better security, automated solutions for detecting abnormal behavior in moving objects such as ships, planes, vehicles, *etc.* are needed more than ever. However, this is a difficult problem since patterns of movement linked with the environment are complex. In ROAM, we use a novel motif-based feature space representation with automatically derived hierarchies. Combined with a rules-based classification model that explores the hierarchies, ROAM is shown to be both effective and efficient in our testing.

References

1. Agrawal, R., Psaila, G., Wimmers, E.L., Zait, M.: Querying shapes of histories. In: Proceedings of 1995 Int. Conf. Very Large Data Bases (VLDB 1995), Zurich, Switzerland, September 1995, pp. 502–514 (1995)

2. Barnett, V., Lewis, T.: *Outliers in Statistical Data*. John Wiley & Sons, Chichester (1994)
3. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2(2), 121–167 (1998)
4. Cao, H., Wolfson, O.: Nonmaterialized motion information in transport networks. In: Eiter, T., Libkin, L. (eds.) *ICDT 2005*. LNCS, vol. 3363, pp. 173–188. Springer, Heidelberg (2005)
5. Chiu, B., Keogh, E., Lonardi, S.: Probabilistic discovery of time series motifs. In: *Proceedings of 2003 ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2003)*, Washington, DC (August 2003)
6. Denis, F.: Pac learning from positive statistical queries. In: Richter, M.M., Smith, C.H., Wiegagen, R., Zeugmann, T. (eds.) *ALT 1998*. LNCS (LNAI), vol. 1501. Springer, Heidelberg (1998)
7. Forlizzi, L., Güting, R.H., Nardelli, E., Schneider, M.: A data model and data structures for moving objects databases. In: *Proceedings of 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD 2000)*, Dallas, TX, May 2000, pp. 319–330 (2000)
8. Gaffney, S., Smyth, P.: Trajectory clustering with mixtures of regression models. In: *Proceedings of 1999 Int. Conf. Knowledge Discovery and Data Mining (KDD 1999)*, pp. 63–72 (1999)
9. Güting, R.H., Schneider, M.: *Moving Objects Databases*. Morgan Kaufmann, San Francisco (2005)
10. Güting, R.H., Bohlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M.: A foundation for representing and querying moving objects. *ACM Trans. Database Systems (TODS)* (March 2000)
11. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, 1157–1182 (2003)
12. Kalnis, P., Mamoulis, N., Bakiras, S.: On discovering moving clusters in spatio-temporal data. In: Bauzer Medeiros, C., Egenhofer, M.J., Bertino, E. (eds.) *SSTD 2005*. LNCS, vol. 3633, pp. 364–381. Springer, Heidelberg (2005)
13. Keogh, E., Pazzani, M.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In: *Proceedings of 1998 Int. Conf. Knowledge Discovery and Data Mining (KDD 1998)*, New York, NY, August 1998, pp. 239–243 (1998)
14. Knorr, E., Ng, R.: Algorithms for mining distance-based outliers in large datasets. In: *Proceedings of 1998 Int. Conf. Very Large Data Bases (VLDB 1998)*, New York, NY, August 1998, pp. 392–403 (1998)
15. Koperski, K., Han, J.: Discovery of spatial association rules in geographic information databases. In: Egenhofer, M.J., Herring, J.R. (eds.) *SSD 1995*. LNCS, vol. 951, pp. 47–66. Springer, Heidelberg (1995)
16. Kostov, V., Ozawa, J., Yoshioka, M., Kudoh, T.: Travel destination prediction using frequent crossing pattern from driving history. In: *Proceedings of 8th Int. IEEE Conf. Intelligent Transportation Systems*, Vienna, Austria, September 2005, pp. 970–977 (2005)
17. Liao, L., Fox, D., Kautz, H.: Learning and inferring transportation routines. In: *Proceedings of 2004 Nat. Conf. Artificial Intelligence (AAAI 2004)* (2004)
18. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: An enabling technique. *Data Mining and Knowledge Discover* 6, 393–423 (2002)
19. Pfoser, D., Jensen, C.S., Theodoridis, Y.: Novel approaches to the indexing of moving object trajectories. In: *Proceedings of 2000 Int. Conf. Very Large Data Bases (VLDB 2000)*, Cairo, Egypt, September 2000, pp. 395–406 (2000)

20. Quinlan, J.R., Cameron-Jones, R.M.: FOIL: A midterm report. In: Proceedings of 1993 European Conf. Machine Learning, Vienna, Austria, pp. 3–20 (1993)
21. Saltenis, S., Jensen, C.S., Leutenegger, S.T., Lopez, M.A.: Indexing the positions of continuously moving objects. In: Proceedings of 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD 2000), Dallas, TX, May 2000, pp. 331–342 (2000)
22. Theodoridis, Y., Silva, J.R.O., Nascimento, M.A.: On the generation of spatiotemporal datasets. In: Güting, R.H., Papadias, D., Lochovsky, F.H. (eds.) SSD 1999. LNCS, vol. 1651. Springer, Heidelberg (1999)
23. Tsoukatos, I., Gunopulos, D.: Efficient mining of spatiotemporal patterns. In: Jensen, C.S., Schneider, M., Seeger, B., Tsotras, V.J. (eds.) SSTD 2001. LNCS, vol. 2121, pp. 425–442. Springer, Heidelberg (2001)
24. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
25. Wei, L., Keogh, E.: Semi-supervised time series classification. In: Proceedings of 2006 Int. Conf. Knowledge Discovery and Data Mining (KDD 2006), Philadelphia, PA (August 2006)
26. Xi, X., Keogh, E., Shelton, C., Wei, L.: Fast time series classification using numerosity reduction. In: Proceedings of 2006 Int. Conf. Machine Learning (ICML 2006) (2006)
27. Yin, X., Han, J.: CPAR: Classification based on predictive association rules. In: Proceedings of 2003 SIAM Int. Conf. Data Mining (SDM 2003), San Fransisco, CA, May 2003, pp. 331–335 (2003)
28. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an efficient data clustering method for very large databases. In: Proceedings of 1996 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD 1996), Montreal, Canada, June 1996, pp. 103–114 (1996)

Intelligent Face Recognition

Adnan Khashman

Electrical & Electronic Engineering Department,
Near East University, Northern Cyprus
khashman@ieee.org

Abstract. A human face is a complex object with features that can vary over time. However, we humans have a natural ability to recognize faces and identify persons in a glance. Of course, our natural recognition ability extends beyond face recognition, where we are equally able to quickly recognize patterns, sounds or smells. Unfortunately, this natural ability does not exist in machines, thus the need to simulate recognition artificially in our attempts to create intelligent autonomous machines.

Face recognition by machines can be invaluable and has various important applications in real life, such as, electronic and physical access control, national defense and international security. While the world is in war against terrorism, the list of wanted persons is getting larger, however, in most cases there is a database containing their face images with various different features such as: with and without eyeglasses or bearded and clean shaven...etc. These different face images of persons (wanted or not) can be used as database in the development of face recognition systems.

Current face recognition methods rely on either: detecting local facial features and using them for face recognition or on globally analyzing a face as a whole. This chapter reviews known existing face recognition methods and presents two case studies of recently developed intelligent face recognition systems that use global and local pattern averaging for facial data encoding prior to training a neural network using the averaged patterns.

20.1 Introduction

Research into applications of neural networks is fascinating and has lately attracted more scientists and indeed “science-fictionists”. The idea of simulating the human perceptions and modeling our senses using machines is great and may help humankind in medical advancement, space exploration, finding alternative energy resources or providing national and international security and peace. Commonly, we refer to our *five* senses; although lately some research suggested there are 21 senses [3]. One sense that is related to this work is sight. We see and perceive objects in different ways depending on our individuality. However, we share the ability to recognize objects or patterns quickly even though our experience of these objects is minimal. A quick “glance” onto a “familiar” face and recognition occurs.

Intelligent systems are being increasingly developed aiming to simulate our perception of various inputs (patterns) such as images, sounds...etc. Biometrics is an example of popular applications for artificial intelligent systems. The development of an

intelligent face recognition system requires providing sufficient information and meaningful data during machine learning of a face.

This chapter presents a brief review of known face recognition methods such as Principal Component Analysis (PCA) [22], Linear Discriminant Analysis (LDA) [1] and Locality Preserving Projections (LPP) [7], in addition to intelligent face recognition systems that use neural networks [9, 10]. There are many works emerging every year suggesting different methods for face recognition, however, these methods are appearance-based or feature-based methods that search for certain global or local representation of a face.

The chapter will also provide two detailed case studies on intelligent face recognition systems. In both cases neural networks are used to identify a person upon presenting his/her face image. Pattern averaging is used for face image preprocessing prior to training or testing the neural network. Averaging is a simple but efficient method that creates “fuzzy” patterns as compared to multiple “crisp” patterns, which provides the neural network with meaningful learning while reducing computational expense.

The two systems differ in their methods of preparing facial data. The first case (Intelligent Global Face Recognition) considers a person’s face and its background and suggests that a quick human “glance” can be simulated in machines using image preprocessing and global pattern averaging, whereas, the perception of a “familiar” face can also be achieved by exposing a neural network to the face via training [9].

The second case (Intelligent Local Face Recognition) considers a person’s essential face features (eyes, nose and mouth) and suggests that a persons face can be recognized regardless of his/her facial expression whether being smiley, sad, surprised...etc. [10].

Finally, analysis of the results presented in these cases will be provided and conclusions of this chapter will be drawn.

The chapter is organized as follows: Sect. 20.2 presents a review on face recognition methods and briefly describes known methods, intelligent methods and difficulties in face detection. Sect. 20.3 presents the first case study on intelligent global face recognition. Sect. 20.4 presents the second case study on intelligent local face recognition. Sect. 20.5 concludes this chapter and provides a discussion on the efficiency of intelligent face recognition by machines.

20.2 A Review on Face Recognition

This section provides a brief review of face recognition in general. Commonly used face databases will be listed, difficulties with face detection will be discussed and examples of successful face recognition methods will be briefly described.

20.2.1 Face Image Databases

A simple definition of “Face Recognition” is the visual perception of familiar faces or the biometric identification by scanning a person’s face and matching it against a library of known faces. In both definitions the faces to be identified are assumed to be familiar or known. Luckily, for researchers we have rich libraries of face images that

are usually freely available for developers. Additionally, “own” face image databases can be built and used together with known databases. The commonly used known libraries include [4]:

- The Color FERET Database, USA
- The Yale Face Database
- The Yale Face Database B
- PIE Database, CMU
- Project - Face In Action (FIA) Face Video Database, AMP, CMU
- AT&T “The Database of Faces” (formerly “The ORL Database of Faces”)
- Cohn-Kanade AU Coded Facial Expression Database
- MIT-CBCL Face Recognition Database
- Image Database of Facial Actions and Expressions - Expression Image Database
- Face Recognition Data, University of Essex, UK
- NIST Mugshot Identification Database
- NLPR Face Database
- M2VTS Multimodal Face Database (Release 1.00)
- The Extended M2VTS Database, University of Surrey, UK
- The AR Face Database, Purdue University, USA
- The University of Oulu Physics-Based Face Database
- CAS-PEAL Face Database
- Japanese Female Facial Expression (JAFFE) Database
- BioID Face DB - HumanScan AG, Switzerland
- Psychological Image Collection at Stirling (PICS)
- The UMIST Face Database
- Caltech Faces
- EQUINOX HID Face Database
- VALID Database
- The UCD Colour Face Image Database for Face Detection
- Georgia Tech Face Database
- Indian Face Database

Web links to the above databases are included in Sect. 20.7: online resources. The following section discusses some of the problems that should be accounted for when selecting a certain database or when making one’s own database.

20.2.2 Difficulties in Face Detection

The databases used in developing face recognition systems rely on images of human faces captured and processed in preparation for implementing the recognition system. The variety of information in these face images makes face detection difficult. For example, some of the conditions that should be accounted for, when detecting faces are [23]:

- Pose (Out-of Plane Rotation): frontal, 45 degree, profile, upside down
- Presence or absence of structural components: beards, mustaches and glasses

- Facial expression: face appearance is directly affected by a person's facial expression
- Occlusion: faces may be partially occluded by other objects
- Orientation (In Plane Rotation)::face appearance directly varies for different rotations about the camera's optical axis
- Imaging conditions: lighting (spectra, source distribution and intensity) and camera characteristics (sensor response, gain control, lenses), resolution

Face Recognition follows detecting a face. Face recognition related problems include [12]:

- Face localization
 - Aim to determine the image position of a single face
 - A simplified detection problem with the assumption that an input image contains only one face
- Facial feature extraction
 - To detect the presence and location of features such as eyes, nose, nostrils, eye-brow, mouth, lips, ears, etc
 - Usually assume that there is only one face in an image
- Face recognition (identification)
- Facial expression recognition
- Human pose estimation and tracking

The above obstacles to face recognition have to be considered when developing face recognition systems. The following section reviews briefly the known face recognition methods.

20.2.3 Recognition Methods

Many face recognition techniques have been developed over the past few decades. These techniques use different methods such as the appearance-based method [18]; where an image of a certain size is represented by a vector in a dimensional space of size similar to the image. However, these dimensional spaces are too large to allow fast and robust face recognition. To encounter this problem other methods were developed that use dimensionality reduction techniques [1, 11, 13, 17]. Examples of these techniques are the Principal Component Analysis (PCA) [22] and the Linear Discriminant Analysis (LDA) [1].

PCA is an eigenvector method designed to model linear variation in high-dimensional data. PCA performs dimensionality reduction by projecting an original n -dimensional data onto a k ($\ll n$)-dimensional linear subspace spanned by the leading eigenvectors of the data's covariance matrix. Its aim is to find a set of mutually orthogonal basis functions that capture the directions of maximum variance in the data and for which the coefficients are pairwise decorrelated. For linearly embedded manifolds, PCA is guaranteed to discover the dimensionality of the manifold and produces a compact representation. PCA was used to describe face images in terms of a set of basis functions, or "eigenfaces".

LDA is a supervised learning algorithm. LDA searches for the projection axes on which the data points of different classes are far from each other while requiring data

points of the same class to be close to each other. Unlike PCA which encodes information in an orthogonal linear space, LDA encodes discriminating information in a linearly separable space using bases that are not necessarily orthogonal. It is generally believed that algorithms based on LDA are superior to those based on PCA. However, some recent work [17] shows that, when the training data set is small, PCA can outperform LDA, and also that PCA is less sensitive to different training data sets.

Another linear method for face analysis is Locality Preserving Projections (LPP) [6] where a face subspace is obtained and the local structure of the manifold is found. LPP is a general method for manifold learning. It is obtained by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the manifold. Therefore, though it is still a linear technique, it seems to recover important aspects of the intrinsic nonlinear manifold structure by preserving local structure. This led to a recently developed method for face recognition; namely the Laplacianface approach, which is an appearance-based face recognition method [7].

The main difference between PCA, LDA, and LPP is that PCA and LDA focus on the global structure of the Euclidean space, while LPP focuses on local structure of the manifold, but they are all considered as linear subspace learning algorithms. Some nonlinear techniques have also been suggested to find the nonlinear structure of the manifold, such as Locally Linear Embedding (LLE) [21]. LLE is a method of nonlinear dimensionality reduction that recovers global nonlinear structure from locally linear fits. LLE shares some similar properties to LPP, such as a locality preserving character. However, their objective functions are totally different. LPP is obtained by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the manifold. LPP is linear, while LLE is nonlinear. LLE has also been implemented with a Support Vector Machine (SVM) classifier for face authentication [19].

Approaches that use the Eigenfaces method [22], the Fisherfaces method [1] and the Laplacianfaces method [7] have shown successful results in face recognition. However, these methods are appearance-based or feature-based methods that search for certain global or local representation of a face. None so far has considered modeling the way we humans recognize faces.

The following section reviews some face recognition methods that incorporate artificial intelligence in order to provide an intelligent system for face recognition.

20.2.4 Artificial Intelligence and Face Recognition

Intelligent systems are being increasingly developed aiming to simulate our perception of various inputs (patterns) such as images, sounds...etc. Biometrics is an example of popular applications for artificial intelligent systems. Face recognition by machines can be invaluable and has various important applications in real life. The development of an intelligent face recognition system requires providing sufficient information and meaningful data during machine learning of a face.

The use of neural networks for face recognition has also been addressed in [19, 24, 5, 16]. More recently, Li, et al. [14] suggested the use of a non-convergent chaotic neural network to recognize human faces. Lu, et al. [15] suggested a semi-supervised learning method that uses support vector machines for face recognition. Zhou, et al.

[25] suggested using a radial basis function neural network that is integrated with a non-negative matrix factorization to recognize faces. Huang and Shimizu [8] proposed using two neural networks whose outputs are combined to make a final decision on classifying a face. Park, et al. [20] used a momentum back propagation neural network for face and speech verification.

Many more face recognition methods that use artificial intelligence are emerging continually; however, two intelligent face recognition methods will be studied in this chapter. These are described in the following sections.

20.3 Case Study 1: Intelligent Global Face Recognition

This case study presents an intelligent face recognition system that uses *global* pattern averaging of a face and its background and aims at simulating the way we see and recognize faces. This is based on the suggestion that a human “glance” of a face can be approximated in machines using pattern averaging, whereas, the “familiarity” of a face can be simulated by a trained neural network [9]. A real-life application will be presented using global averaging and a trained back propagation neural network to recognize the faces of 30 persons.

20.3.1 Image Database for Global Face Recognition

One common problem with processing images is the large amount of data that is needed for meaningful results. Although neural networks have the advantage of parallel processing, there is still a need to pre-process images to reduce the amount of data while retaining meaningful information on the images. This is an important requirement for an efficient system that has low time and computational expense.

There are 30 persons whose faces were to be recognized and thus their face images would be used as the database for the work presented within this case study. Each face has three different projections, which were captured while looking: Left (*LL*), Straight (*LS*) and Right (*LR*) as shown in Fig. 20.1 resulting in 90 images that are used for implementing the intelligent system. Fig. 20.2 shows these 90 images representing 30 persons of various gender, ethnicity and age.

All original images are gray and of size (512x512) pixels. The images were compressed and their size reduced to 128x128 pixels. A window of size 100x100 pixels;



Fig. 20.1. Person 30 looking: a- left (LL) b- straight (LS) c- right (LR)

that contains the face and its background, is then extracted and the data within this relatively smaller size image is used for training and eventually testing the neural network.

20.3.2 Image Pre-processing (Global Averaging)

The method used for presenting the images to the neural network is based on global pattern averaging, which provides the glance approximation. A face image of size 100x100 pixels is segmented and the values of the pixels within each segment are averaged. The result average values are then used as input data for the neural network.

The averaging of the segments within an image reduces the amount of data required for neural network implementation thus providing a faster recognition system. This also provides flexible mathematical inputs for neural networks that simulate the quick glance of a human which is sufficient for pattern recognition. Global pattern averaging can be defined as follows:

$$PatAv_i = \frac{1}{s_k s_l} \sum_{l=1}^{s_l} \sum_{k=1}^{s_k} p_i(k, l), \quad (20.1)$$

where k and l are segment coordinates in the x and y directions respectively, i is the segment number, S_k and S_l are segment width and height respectively, $P_i(k, l)$ is pixel value at coordinates k and l in segment i , $PatAv_i$ is the average value of pattern in segment I , that is presented to neural network input layer neuron i . The number of segments in each window (of size XY pixels) containing a face, as well as the number of neurons in the input layer is i where

$$i = \{-1, 0, 1, 2, \dots, n\}, \quad (20.2)$$

and

$$n = \left(\frac{X}{s_k} \right) \left(\frac{Y}{s_l} \right). \quad (20.3)$$

Segment size of 10x10 pixels ($S_k = S_l = 10$) has been used and average values representing the image were obtained, thus resulting in 100 average values in total ($n = 100$) that were used as the input to the neural network for both training and testing.

Fig. 20.3 shows an example of this pre-processing phase. The original 512x512 pixel image is reduced to 256x256 pixels and then to 128x128 pixels. This is followed by extracting a region of size 100x100 pixels that contains the face. The extracted region is then segmented (tiled) and averaged yielding a 10x10 pixel pattern that represents the original image.

20.3.3 Neural Network Implementation

The multilayer perceptron neural network, which was developed as part of this global face recognition system, is based on the back propagation learning algorithm, with a total number of three layers, comprising, input layer, hidden layer and output layer.



Fig. 20.2. Own database of 30 persons

The input layer has 100 neurons, each receiving an averaged value of the face image segments. The hidden layer consists of 99 neurons, whereas the output layer has 30 neurons according to the number of persons. Fig. 20.4 shows the topology of this neural network and data presentation to the input layer.

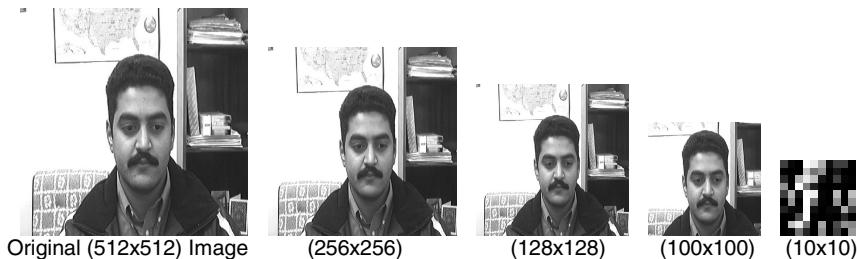


Fig. 20.3. Image pre-processing before neural network training or testing

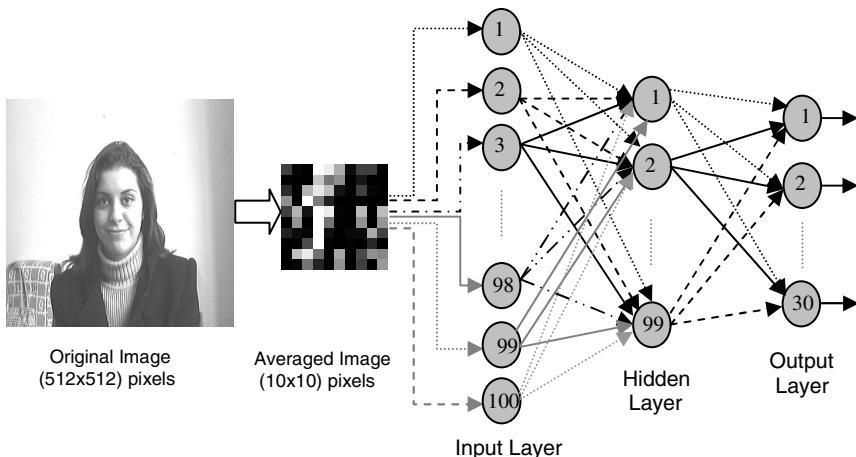


Fig. 20.4. Global pattern averaging and neural network design

The approach within this case study is based on simulating the “glance” and “familiarity” of faces in humans. The glance effect is approximated via image pre-processing and global pattern averaging as described in (Sect. 20.3.2), whereas, familiarity of a face is simulated by training the neural network using face images with different orientations.

The implementation of a neural network consists of training and testing. In this work a total of 90 face images (corresponding to 30 persons) were used. For training the neural network 60 face images (looking left *LL* and looking right *LR*) were used. The 30 remaining face images (looking straight *LS*) were used for testing purposes where the system is expected to recognize the person looking straight at the camera by training it on face images looking left and right. This simulates the familiarity of a face in machines, even though the test images (looking straight) present a neural network with different pixel values as a result of the difference in the orientation of the face.

A recognition system “sensitivity” feature was also developed as part of the neural network classification of input face images. Three levels of tolerance, namely Low

(minimum 80% face resemblance), Medium (minimum 65% face resemblance) or High (minimum 50% face resemblance) can be used depending on the required level of accuracy. The results that are presented in the next section were obtained using *Low* tolerance.

20.3.4 Results and Discussion (Case 1)

The back propagation neural network learnt and converged after 4314 iterations and within 390 seconds, whereas the running time for the trained neural network after training and using one forward pass was 0.21 seconds. These time cost results were obtained using the following system configuration: 2.4 GHz PC with 256 MB of RAM using Windows XP operating system, C-language source code and Borland C⁺⁺ compiler. Table 20.1 lists the final parameters of the successfully trained neural network.

All training images (60 face images – looking left and right) were recognized when used for testing the trained neural network yielding 100% recognition rate. The recognition of the testing face images (30 face images – looking straight) indicates the success and robustness of this intelligent system, as these image faces had not been presented to the neural network before. Additionally, the look straight face images have different orientation and, thus, different pixel values in comparison to the

Table 20.1. Trained neural network final parameters using global face data

Input Layer Nodes	100
Hidden Layer Nodes	99
Output Layer Nodes	30
Learning Rate	0.008
Momentum Rate	0.32
Minimum Error	0.002
Iterations	4314
Training Time	390 seconds
Generalization (run) Time	0.21 seconds



Person 23



Person 21

Fig. 20.5. Person 23 incorrectly identified as person 21

training face images look left and right at similar coordinates. Testing the neural network using these test images yielded a successful 96.67% recognition rate where 29 out of 30 face images were correctly recognized.

The only incorrect result, out of the testing image set, was person 23 identified as person 21. Both persons have close face resemblance, where a quick “glance” may not be able to distinguish. This incorrect identification occurred only when presenting the neural network with the face image looking straight (LS). Fig. 20.5 shows both persons. Table 20.2 shows the recognition rates where a total recognition rate of 98.89% has been achieved.

Table 20.2. Intelligent Global Face recognition results for 30 persons

Image Set	Recognition Rate
Training Set (60 images)	(60/60) %100
Testing Set (30 images)	(29/30) %96.67
Total (90 images)	(89/90) % 98.89

In summary, the recognition process has two phases. First, simulating the quick look (glance) via image pre-processing which involves face image compression, cropping, segmentation and global pattern averaging. This phase yields segment pattern average values that are global representations of the face and consequently form the input to a neural network. The second phase (simulating familiarity) is training the neural network using the output of the first phase. Once the network converges or learns, classification and face recognition is achieved.

20.3.5 Conclusions

This case study, which is based on using global (complete face and background) data averaging, introduced a novel approach to face recognition, based on simulating the human “glance” and face “familiarity”. The glance effect is approximated via image pre-processing and pattern averaging. When we humans have a quick look (glance) at faces, we do not observe the detailed features but rather a general global impression of a face. This can be approximated by averaging the face image instead of searching for features within the face. The averaged patterns are representation of a face regardless of its expression or orientation. The quick glance is followed by familiarity with a face, which is simulated by training a neural network using face images with different orientations.

The presence or absence of structural components such as beards, mustaches or glasses could of course affect the recognition results. This depends on the differences in pixel values due to the structural component. A large difference in pixel values would marginally change the averaged pattern value, whereas a small difference would cause a minimal change in averaged pattern values. This problem can be solved by updating the recognition system with any changes to a face due to a structural component; in other words familiarizing the intelligent system with any changes to a face. This problem was investigated by testing the trained neural network using face images of “person 3” wearing a dark hat, thus resulting in minimal changes to the

averaged pattern value. The system was able to correctly recognize “person 3” with and without the hat. On the other hand, “person 2” face image without beard and moustache was used for testing and the system yielded “unknown person” result, thus requiring updating the recognition system with the new look of the person, after which “person 2” was correctly recognized. Another interesting result was the correct recognition of person 20 and person 28, who happen to be identical twins. This demonstrates the flexibility of the developed system where face image database can be updated as required. Fig. 20.6 shows the images used for these additional tests.

A real life application, using 90 face images of 30 persons of different gender, age and ethnicity, was implemented using this global intelligent recognition system. A total recognition rate of 98.89% was obtained using 90 face images of the 30 persons in different orientations. Only one person’s face image (looking straight) was mistaken for another person (looking straight too) as shown in Fig. 20.5. The robustness and success of this face recognition system was further demonstrated by its quick run time

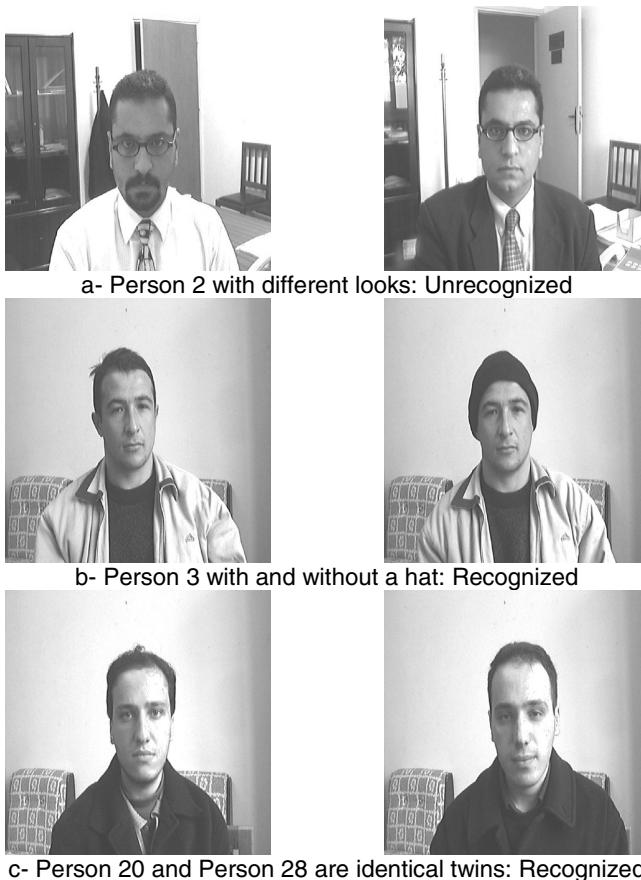


Fig. 20.6. Examples of further recognition system tests

(one neural network forward pass) of 0.21 seconds. Time cost was kept minimal through image-preprocessing and reduction of input/hidden layer neurons in the topology of the neural network.

Finally, three levels of tolerance can be used in this system depending on the required level of accuracy: low tolerance (80% face resemblance), medium tolerance (65% face resemblance) or high tolerance (50% face resemblance). All results shown in this case study were obtained using the low tolerance classification, where a minimum of 80% face resemblance is required. This is believed to be a good resemblance ratio considering the neural network is trained using globally averaged patterns of faces.

20.4 Case Study 2: Intelligent Local Face Recognition

This case study presents an intelligent face recognition system that uses *local* pattern averaging of essential facial features (eyes, nose and mouth). Here, multiple face images of a person with different facial expressions are used, where only eyes, nose and mouth patterns are considered. These essential features from different facial expressions are averaged and then used to train a supervised neural network [10]. A real-life application will be presented using local averaging and a trained neural network to recognize the faces of 30 persons.

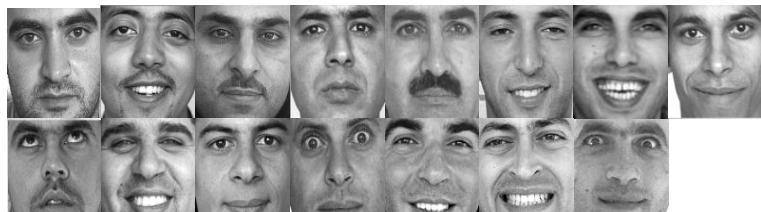
20.4.1 Image Database for Local Face Recognition

The face images, which are used for training and testing the neural network within the intelligent local face recognition system, represent persons of various ethnicities, age and gender. A total of 180 face images of 30 persons with different facial expressions are used, where 90 images are from the ORL face database [2], and 90 images are from our own face database. Our face database was built using face images captured under the following conditions:

- Similar lighting condition
- No physical obstruction
- Head pose is straight without rotation or tilting
- Camera at the same distance from the face

Each person has six different face expressions captured and the image is resized to (100x100) pixels, thus resulting in 90 face images from each face database. Fig. 20.7 shows the faces of the 30 persons from our face database and the ORL face database, whereas Fig. 20.8 shows examples of the six facial expressions.

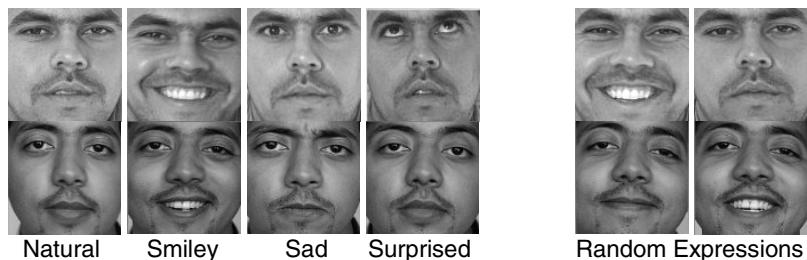
The 180 face images of the 30 persons with different expressions were used for the development and implementation of the intelligent local face recognition system. Approximation or local averaging of four multi-expression faces is applied only during the neural network training phase where the four facial expressions (natural, smiley, sad and surprised) images are reduced to one face image per person by separately averaging the essential features (eyes, nose, and mouth), thus providing 30 averaged face images for training the neural network. Testing the neural network is implemented using the six facial expressions *without* the averaging process, thus providing 180 face images for testing the trained neural network.



a- Faces of 15 persons (Own Database)



b- Faces of 15 persons (ORL Database [Error! Reference source not found.])

Fig. 20.7. Face Databases for Local Face Recognition

a- Examples of Multi-expression faces from our database



b- Examples of Multi-expression faces from ORL database [25]

Fig. 20.8. Examples of the different facial expressions

20.4.2 Image Pre-processing (Local Averaging)

The implementation of recognition system comprises the image preprocessing phase and the neural network arbitration phase.

Image preprocessing is required prior to presenting the training or testing images to the neural network. This aims at reducing the computational cost and providing a faster recognition system while presenting the neural network with sufficient data representation of each face to achieve meaningful learning.

The back propagation neural network is trained using approximations of four specific facial expressions for each person, which is achieved by averaging the essential features, and once trained; the neural network is tested using the six different expressions without approximation.

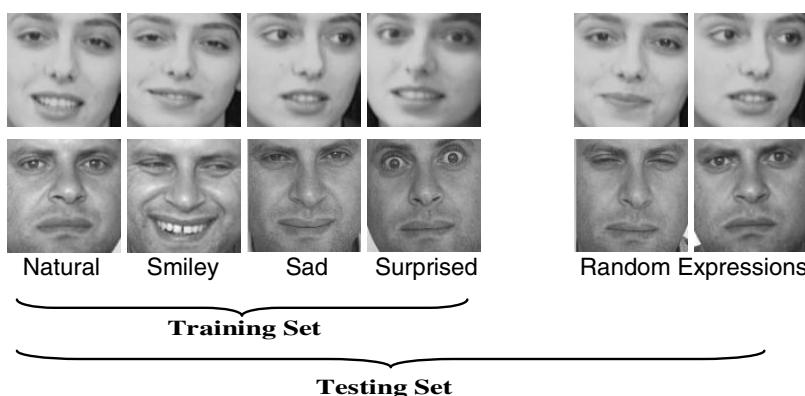


Fig. 20.9. Examples of training and testing face images

There are 180 face images of 30 persons with six expressions for each. Training the neural network uses 120 images (which will be averaged to 30 images) representing the 30 persons with four specific expressions. The remaining 60 images of the 30 persons with random different expressions are used together with the 120 training images (prior to averaging) for testing the trained neural network, as can be seen in Fig. 20.9, thus resulting in 180 face images for testing.

The four essential features (eyes, nose and mouth) from four expressions (natural, smiley, sad and surprised) are approximated via local averaging into one single vector that represents the person. Fig. 20.10 shows the scheme for the intelligent local face recognition system.

The features are, firstly extracted for each facial expression of each subject as shown in Fig. 20.11. Feature extraction is manually performed using Photoshop. Secondly, the dimensions of each feature are reduced by interpolation. The right eye, left eye, nose and mouth dimensions are reduced to (5 x 10) pixels, (5 x 10) pixels, (7 x 10) pixels and (6 x 17) pixels respectively.

Thus, the output matrices dimension after interpolation process will be 1/3 of the input matrices; for example, the 15x30 pixels input matrix will be after interpolation

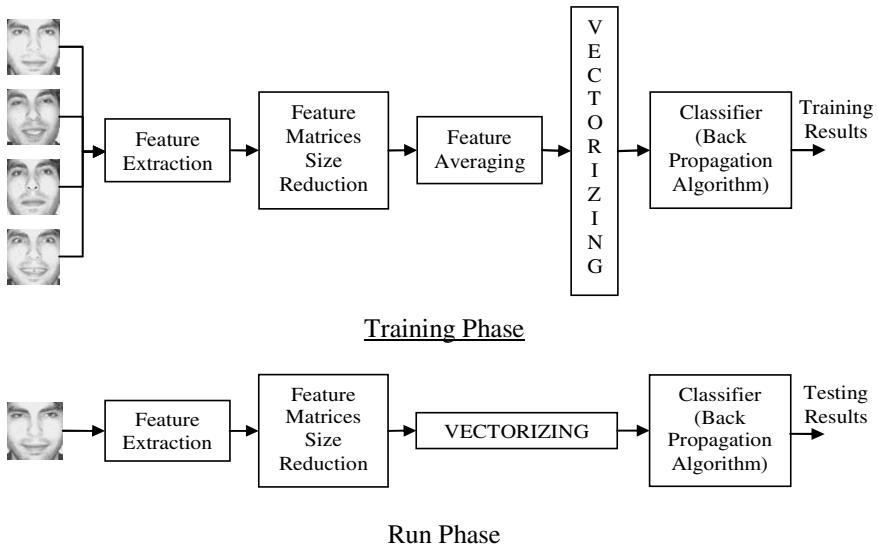


Fig. 20.10. General architecture of the intelligent local face recognition system



Fig. 20.11. Extracted local features from four different expressions

5x10 pixels. Local averaging is then applied where the 120 training images are reduced to 30 averaged images by taking the average for each feature in the four specific expressions for each subject.

The local feature averaging process for each feature can be implemented using the following equation:

$$f_{avg} = \frac{1}{4} \sum_{i=1}^4 f_i , \quad (20.4)$$

where f_{avg} is the feature average vector and f_i is feature in expression i of one person. Finally, the averaged features are represented as (272x1) pixel vectors, which will be presented to the input layer of the back propagation neural network.

20.4.3 Neural Network Implementation

The back propagation algorithm is used for the implementation of the proposed intelligent face recognition system, due to its simplicity and efficiency in solving pattern recognition problems. The neural network comprises an input layer with 272 neurons

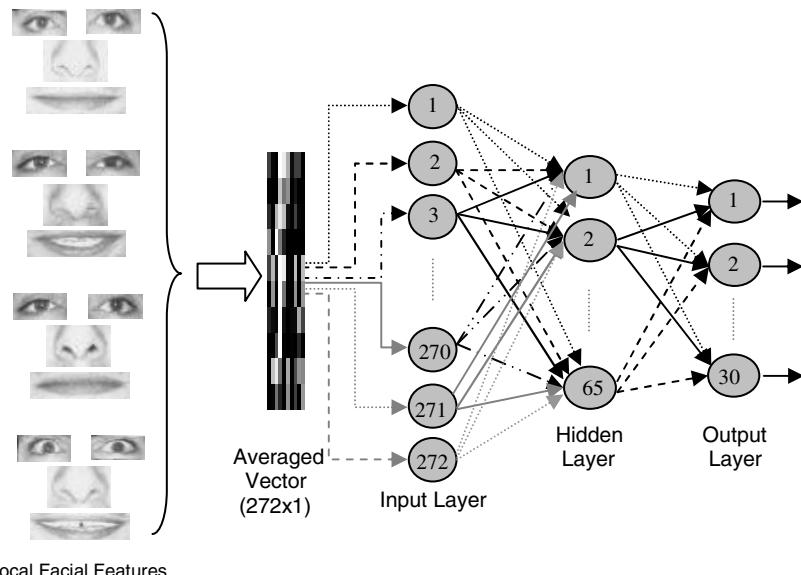


Fig. 20.12. Local pattern averaging and neural network design

that carry the values of the averaged features, a hidden layer with 65 neurons and an output layer with 30 neurons which is the number of persons. Fig. 20.12 shows the topology of this neural network and data presentation to the input layer.

20.4.4 Results and Discussion (Case 2)

The neural network learnt the approximated faces after 3188 iterations and within 265 seconds, whereas the running time for the trained neural network using one forward pass was 0.032 seconds. These results were obtained using a 1.6 GHz PC with 256 MB of RAM, Windows XP OS and Matlab 6.5 software. Table 20.3 shows the final parameters of the successfully trained neural network. The reduction in training and testing time was achieved by the novel method of reducing the face data via averaging selected essential face features for training, while maintaining meaningful learning of the neural network. The face recognition system correctly recognized all averaged face images in the training set as would be expected.

The intelligent system was tested using 180 face images which contain different face expressions that were not exposed to the neural network before; these comprised 90 images from our face database and 90 images from the ORL database. All 90 face images in our database were correctly identified yielding 100% recognition rate with 91.8% recognition accuracy, whereas, 84 out of the 90 images from the ORL database were correctly identified yielding 93.3% recognition rate with 86.8% recognition accuracy.

Table 20.3. Neural network final parameters

Number of Input Neurons	272
Number of Hidden Neurons	65
Number of Output Neurons	30
Learning Coefficient	0.0495
Momentum Coefficient	0.41
Minimum Error	0.001
Training Iterations	3188
Training Time	265 Seconds
Generalization (run) Time	0.032 seconds

Table 20.4. Recognition Rates, Accuracy and Run Time

Database	Own	ORL	Total
Recognition Rate	100 %	93.3 %	96.7 %
Recognition Accuracy	91.8 %	86.8 %	89.3 %

The overall recognition rate for the system was 96.7% where 174 out of the available 180 faces were correctly recognized with an accuracy rate of 89.3%. The recognition rate refers to the percent of correctly recognized faces, whereas the recognition accuracy refers to the classification real output value in comparison to the desired output value of “1”, using binary output coding.

The processing time for face image preprocessing and feature averaging was 7.5 seconds, whereas running the trained neural network took 0.032 seconds. The recognition rates and recognition accuracy the trained system are shown in Table 20.4.

Further investigations of the capability of the developed face recognition system were also carried out by testing the trained neural network ability to recognize two subjects with eyeglasses. Fig. 20.13 shows the two persons with and without glasses; person 1 wears clear eyeglasses whereas, person 2 wears darker eyeglasses.

The effect of the presence of facial detail such as glasses on recognition performance was investigated. The neural network had not been exposed to the face images with glasses prior to testing. Correct recognition of both persons, with and without their glasses on, was achieved. However, the recognition accuracy was reduced due to the presence of the glasses. The ability of the trained neural network to recognize these faces despite the presence of eyeglasses is due to training the network using feature approximations or “fuzzy” feature vectors rather than using “crisp” feature vectors. Table 20.5 shows the accuracy rates for both persons with and without glasses.

20.4.5 Conclusions

This case study, which is based on using local (facial features) data averaging, introduced a novel method to intelligent face recognition. The method approximates four essential face features (eyes, nose and mouth) from four different facial expressions



Fig. 20.13. (a) Clear eyeglasses (b) Darker eyeglasses

Table 20.5. Recognition Accuracy With and Without Eyeglasses

Person 1		Person 2	
No Eyeglasses	Clear Eyeglasses	No Eyeglasses	Dark Eyeglasses
96 %	86 %	90 %	73 %

(natural, smiley, sad and surprised), and trains a neural network using the approximated features to learn the face. Once trained, the neural network could recognize the faces with different facial expressions. Although the feature pattern values (pixel values) may change with the variations in facial expression, the use of averaged-features of a face provides the neural network with an approximated understanding of the identity and is found to be sufficient for training a neural network to recognize that face with any expression, and with the presence of minor obstructions such as eyeglasses.

The successful implementation of the proposed method was shown throughout a real-life implementation using 30 face images showing six different expressions for each. An overall recognition rate of 96.7% with recognition accuracy of 89.3% was achieved.

The use of feature approximation helped reducing the amount of training image data prior to neural network implementation, and provided reduction in computational cost while maintaining sufficient data for meaningful neural network learning. The overall processing times that include image preprocessing and neural network implementation were 272.5 seconds for training and 0.032 seconds for face recognition.

20.5 Discussions and Conclusions

This chapter presented a review of related works on face recognition in general and on intelligent face recognition in particular. Research work on the later has been increasing lately due to the advancement in Artificial Intelligence and the availability of fast computing power.

The recognition of a face that has been seen before is a natural and easy task that we humans perform everyday. What information we pick from a face during a glance may be mysterious but the result is usually correct recognition. Do we only look at features such as eyes or nose (local recognition) or do we ignore these features and look at a face as a whole (global recognition)? How about the other “input” information in

addition to our visual information such as sounds or smell? The brain is an efficient parallel processor that receives enormous amount of data and processes it at incredibly high speeds. Therefore, in real life face recognition, the brain would be processing not only the image of a face, but also gestures, sounds, odor and any other information that might help achieving a quick recognition. Of course, to simulate such a parallel perceiving machine that would use multi-senses is yet to be achieved. Meanwhile, we focus on the visual input that is represented as facial images.

Many research works on face recognition attempt to answer the above questions, while scientist differ in their approaches or methods to how face recognition can be simulated in machines. One common concept that is shared by most methods is that the detection of a face requires facial information, which can be obtained locally (using local facial features such as eyes) or globally (using a whole face).

The diversity of the different methods and approaches is more evident when investigating the development of artificial intelligent face recognition systems. These intelligent systems aim to simulate the way we humans recognize faces. Here one can pause for a while and think “What do I really look at when I look at a face?”. The answer could be that we all have our own ways which might differ, thus the diversity in simulating intelligent face recognition in machines.

This chapter described two examples of intelligent face recognition methods that have been recently suggested. The first method simulates the way some of us might be using when recognizing a face by taking a quick global look at the whole face. This was referred to as “Intelligent Global Face Recognition” and was described in Sect. 20.3. The second method uses essential local face features of a person with different facial expression. This was referred to as “Intelligent Local Face Recognition” and was described in Sect. 20.4. In both cases the artificial intelligent system was implemented using supervised neural networks whose tasks were to simulate the function and structure of a brain that receives visual information.

The Global averaging neural network learnt to classify the faces within 390 seconds, whereas the running time for the trained neural network was 0.21 seconds. The Local averaging neural network learnt to classify the faces within 265 seconds, whereas the running time for the trained neural network was 0.032 seconds. These time costs can be further reduced by using faster machines, which will inevitably occur in the near future.

The Global average neural network implementation yielded 100% recognition rate of all training images as would be expected, while testing this neural network using the test images yielded a successful 96.67% recognition rate. Thus, the overall recognition rate for the Global average method was determined as 98.89%. The Local average neural network implementation also yielded 100% recognition rate when using the 30 locally averaged face images in the training set. Testing was carried out using 180 face images which contain different face expressions that were not exposed to the neural network before. Here, 174 out of the 180 test images were correctly identified yielding 96.7% recognition rate. Thus, the overall recognition rate for the local average method was determined as 97.14%.

In conclusion, both methods have shown successful results which suggests they can be used as part of a robust intelligent face recognition system. Global averaging can be successfully applied in real life where the faces are at different orientations and the image contains non-facial features such as hats, hair, eyeglasses and

background. Local averaging can also be applied successfully to identify faces with different expressions. Here, the image databases contain only the faces as the method uses essential facial features such as eyes, nose and mouth; therefore the existence of background and occlusions is irrelevant. Although the local feature pattern values (pixel values) may change with the change of facial expression, the use of local averaging of a face provides the neural network with an approximated understanding of the identity and is found to be sufficient for training a neural network to recognize that face with any expression.

Despite successful implementations of artificial intelligent face recognition systems such as those shown in the above case studies, there are questions that are yet to be answered before we can completely trust a machine whose intelligence “evolves” in minutes in comparison with our natural intelligence that took thousands of years to evolve. There is no doubt that the advancement in technology provides us with the means to develop artificially intelligent systems, but how intelligent are they really are?

Consider a real life scenario in an airport where *trained* security officers monitor people coming *in* and *out* of a terminal.

“Trained” means the officer has spent time on training and learning before he/she is considered qualified and is assigned the important task of quickly identifying a suspicious person or a potential terrorist. The officer would have also be shown face images of wanted persons, probably with possible different looks. Such training is possible to simulate in intelligent machines albeit with many limitations. These include the scope of the area to be monitored and the flexibility of visual input device (i.e. camera in a system or the eyes in a human). Another problem for an artificially intelligent system is the detection of suspicious persons by observing their gestures and body language, while detecting the face at the same time. Then there is detection and identifying a person when going “in” and “out” of the building, where at some point the face can not be seen. Such tasks, which can be simultaneously and quickly performed by one trained security officer, require different artificial intelligent systems. So why do we need the intelligent machines when their capabilities are marginally less than our own?

Most of the currently developed intelligent recognition systems are aimed to be used as an aid to human operators. A completely, autonomous system would be our eventual target. The development of more powerful and faster computing systems is continuing, and with this increase in computational power we can design intelligent recognition systems that could perform many recognition tasks at once. So the simulation of our parallel information processing is getting closer albeit slowly and gradually.

References

1. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *J. IEEE Trans (PAMI)* 19(7), 711–720 (1997)
2. Cambridge University, Olivetti Research Laboratory face database (2002),
<http://www.cl.cam.ac.uk/Research/DTG/attarchive/facedatabase.html>
3. Durie, B.: Senses Special: Doors of Perception. *New Scientist Magazine* 2484, 34 (2005)
4. Face Recognition Homepage–Databases (March 26, 2007),
<http://www.face-rec.org/databases/>

5. Fan, X., Verma, B.: A Comparative Experimental Analysis of Separate and Combined Facial Features for GA-ANN based Technique. In: Proceedings of Conference on Computational Intelligence and Multimedia Applications, pp. 279–284 (2005)
6. He, X., Niyogi, P.: Locality Preserving Projections. In: Proceedings of Conference on Advances in Neural Information Processing Systems (2003)
7. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.J.: Face Recognition Using Laplacianfaces. J. IEEE Trans (PAMI) 27(3), 328–340 (2005)
8. Huang, L.L., Shimizu, A.: Combining Classifiers for Robust Face Detection. In: Wang, J., Yi, Z., Žurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3972, pp. 116–121. Springer, Heidelberg (2006)
9. Khashman, A.: Face Recognition Using Neural Networks and Pattern Averaging. In: Wang, J., Yi, Z., Žurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3972, pp. 98–103. Springer, Heidelberg (2006)
10. Khashman, A., Garad, A.: Intelligent Face Recognition Using Feature Averaging. In: Mehta, S., Zeng, D.D., Chen, H., Thuraisingham, B., Wang, F.-Y. (eds.) ISI 2006. LNCS, vol. 3975, pp. 432–439. Springer, Heidelberg (2006)
11. Levin, A., Shashua, A.: Principal Component Analysis over Continuous Subspaces and Intersection of Half-Spaces. In: Proceedings of the European Conference on Computer Vision, vol. 3, pp. 635–650 (2002)
12. Li, S.Z., Jain, A.K.: Handbook Of Face Recognition. Springer, Heidelberg (2005)
13. Li, S.Z., Hou, X.W., Zhang, H.J., Cheng, Q.S.: Learning Spatially Localized, Parts-Based Representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 207–212 (2001)
14. Li, G., Zhang, J., Wang, Y., Freeman, W.J.: Face Recognition Using a Neural Network Simulating Olfactory Systems. In: Wang, J., Yi, Z., Žurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3972, pp. 93–97. Springer, Heidelberg (2006)
15. Lu, K., He, X., Zhao, J.: Semi-supervised Support Vector Learning for Face Recognition. In: Wang, J., Yi, Z., Žurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3972, pp. 104–109. Springer, Heidelberg (2006)
16. Lu, X., Wang, Y., Jain, A.K.: Combining Classifiers for Face Recognition. IEEE Conference on Multimedia & Expo 3, 13–16 (2003)
17. Martinez, A.M., Kak, A.C.: PCA versus LDA. J. IEEE Trans (PAMI) 23(2), 228–233 (2001)
18. Murase, H., Nayar, S.K.: Visual Learning and Recognition of 3-D Objects from Appearance. J. Computer Vision 14, 5–24 (1995)
19. Pang, S., Kim, D., Bang, S.Y.: Face Membership Authentication Using SVM Classification Tree Generated by Membership-Based LLE Data Partition. J. IEEE Trans. Neural Networks 16(2), 436–446 (2005)
20. Park, C., Ki, M., Namkung, J., Paik, J.K.: Multimodal Priority Verification of Face and Speech Using Momentum Back-Propagation Neural Network. In: Wang, J., Yi, Z., Žurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3972, pp. 140–149. Springer, Heidelberg (2006)
21. Roweis, S.T., Saul, L.K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science 290, 2323–2326 (2000)
22. Turk, M., Pentland, A.P.: Face Recognition Using Eigenfaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591 (1991)
23. Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting Faces in Images: A Survey. J. IEEE Trans (PAMI) 24(1), 34–58 (2002)

24. Zhang, B., Zhang, H., Ge, S.: Face recognition by applying wavelet subband representation and kernel associative memory. *J. IEEE Trans. Neural Networks* 15, 166–177 (2004)
25. Zhou, W., Pu, X., Zheng, Z.: Parts-Based Holistic Face Recognition with RBF Neural Networks. In: Wang, J., Yi, Z., Zurada, J.M., Lu, B.-L., Yin, H. (eds.) *ISNN 2006. LNCS*, vol. 3972, pp. 110–115. Springer, Heidelberg (2006)

Online Resources

Comprehensive Face Recognition Resources

<http://www.cbsr.ia.ac.cn/users/szli/FR-Handbook/>

<http://www.face-rec.org/general-info/>

<http://www.epic.org/privacy/facerecognition/>

http://www.findbiometrics.com/Pages/face_articles/face_2.html

Fun with Face Recognition

<http://www.myheritage.com/FP/Company/tryFaceRecognition.php>

<http://faculty.washington.edu/chudler/java/faces.html>

<http://faculty.washington.edu/chudler/java/facemem.html>

Face Databases

- The Color FERET Database, USA:
<http://www.itl.nist.gov/iad/humanid/colorferet/home.html>
- The Yale Face Database:
<http://cvc.yale.edu/projects/yalefaces/yalefaces.html>
- The Yale Face Database B:
<http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html>
- PIE Database, CMU:
http://www.ri.cmu.edu/projects/project_418.html
- Project - Face In Action (FIA) Face Video Database, AMP, CMU:
<http://amp.ece.cmu.edu/projects/FIADataCollection/>
- AT&T “The Database of Faces” (formerly “The ORL Database of Faces”):
<http://www.cl.cam.ac.uk/Research/DTG/attarchive/facedatabase.html>
- Cohn-Kanade AU Coded Facial Expression Database:
http://vasc.ri.cmu.edu/idb/html/face/facial_expression/index.html
- MIT-CBCL Face Recognition Database:
<http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html>
- Image Database of Facial Actions and Expressions - Expression Image Database:
<http://mambo.ucsc.edu/psl/joehager/images.html>
- Face Recognition Data, University of Essex, UK:
<http://cswww.essex.ac.uk/mv/allfaces/index.html>
- NIST Mugshot Identification Database:
<http://www.nist.gov/srd/nistsd18.htm>
- NLPR Face Database:
<http://nlpr-web.ia.ac.cn/english/irds/facedatabase.htm>
- M2VTS Multimodal Face Database (Release 1.00):
<http://www.tele.ucl.ac.be/PROJECTS/M2VTS/m2fdb.html>

- The Extended M2VTS Database, University of Surrey, UK:
<http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb/>
- The AR Face Database, Purdue University, USA:
http://rvl1.ecn.purdue.edu/~aleix/aleix_face_DB.html
- The University of Oulu Physics-Based Face Database:
<http://www.ee.oulu.fi/research/imag/color/pbfd.html>
- CAS-PEAL Face Database:
<http://www.jdl.ac.cn/peal/index.html>
- Japanese Female Facial Expression (JAFFE) Database:
<http://www.ircatr.jp/~mlyons/jaffe.html>
- BioID Face DB - HumanScan AG, Switzerland:
<http://www.humanscan.de/support/downloads/facedb.php>
- Psychological Image Collection at Stirling (PICS):
<http://pics.psych.stir.ac.uk/>
- The UMIST Face Database:
<http://images.ee.umist.ac.uk/danny/database.html>
- Caltech Faces:
<http://www.vision.caltech.edu/html-files/archive.html>
- EQUINOX HID Face Database:
<http://www.equinoxsensors.com/products/HID.html>
- VALID Database:
<http://ee.ucd.ie/validdb/>
- The UCD Colour Face Image Database for Face Detection:
<http://ee.ucd.ie/~prag/>
- Georgia Tech Face Database:
http://www.anefian.com/face_reco.htm
- Indian Face Database:
<http://vis-www.cs.umass.edu/~vidit/IndianFaceDatabase/>

Questions for Discussions

1. Do you think that one day artificially intelligent machines could become more “intelligent” than human? Justify your answer.
2. Where do you draw the line regarding the tasks you can trust artificial intelligent systems with?
3. What will be the most likely application fields for artificial intelligence within the next 10 years?
4. How efficient are current intelligent face recognition systems?
5. Do you think the employment of biometrics and artificial intelligent systems in real life applications would make the world safer?
6. (Related to Question 5) Would there be an increase in crime in society? How?
7. What are the possibilities of turning artificial intelligence against us? think of an artificial intelligent terrorist robot!

Video Analysis of Vehicles and Persons for Surveillance

Sangho Park and Mohan M. Trivedi

Computer Vision and Robotics Research Laboratory
University of California at San Diego, USA
{parks, mtrivedi}@ucsd.edu

Abstract. This chapter presents a multi-perspective vision-based analysis of the activities of vehicles and persons for the enhancement of situational awareness in surveillance. Multiple perspectives provide a useful invariant feature of the object in the image, i.e., the footage area on the ground. Moving objects are detected in the image domain, and the tracking results of the objects are represented in the projection domain using planar homography. Spatio-temporal relationships between human and vehicle tracks are categorized as safe or unsafe situation depending on the site context such as walkway and driveway locations. Semantic-level information of the situation is achieved with the anticipation of possible directions of near-future tracks using piecewise velocity history. Crowd density is estimated from the footage on the homography plane. Experimental data show promising results. Our framework can be applied to broad range of situational awareness for emergency response, disaster prevention, human interactions in structured environments, and crowd movement analysis in a wide field of view.

21.1 Introduction

There has been a growing interest in the society and industry for making sensor-based systems that enhance the safety and efficiency of human inhabited environments. Enhanced *situational awareness* is one of the key issues in developing intelligent infrastructures for safer environments. The situational awareness discussed in this chapter means representation, modeling, and recognition of semantic context of events occurring in the monitored environment with respect to the activities of persons and vehicles in terms of their sizes, spatial distributions, velocities, relative configurations, traffic flow, recent history, near future anticipation, etc. Fig. 21.1 shows an example of surveillance video frames. A specific region of interest (ROI) denoted by the rectangle on the satellite image is monitored with two network cameras installed in the outdoor environment. In order to develop automatic situational awareness system, it is important to understand how people interact with each other and with the environment. It will be useful to detect, represent and estimate what kinds of events are occurring or about to occur in the monitored site. Pedestrian safety and crowd behavior analysis are good examples.

In this chapter, we present a methodology for multi-perspective vision-based analysis of human interactivity with other persons and vehicles for enhanced situational awareness. This chapter addresses issues and challenges in video surveillance and presents

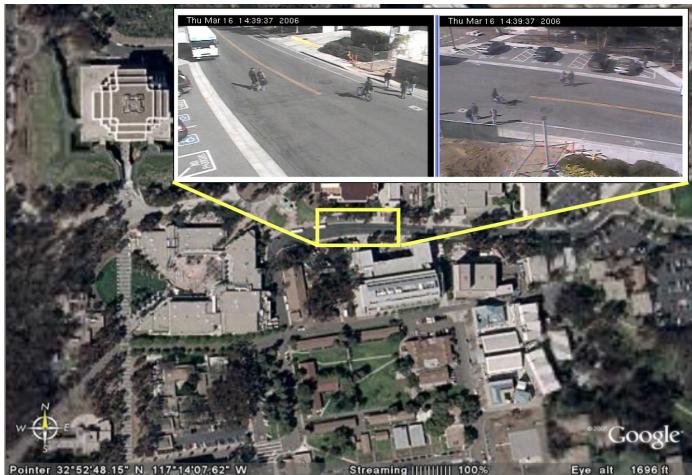


Fig. 21.1. Synchronized video frames (inlet) of a specific region of interest in a university campus. (Satellite image by Google.)

a case study that achieves enhanced situational awareness by incorporating view-invariant features of moving objects and spatio-temporal context of their motions.

The topic of this chapter belongs to more general research problems of analyzing and recognizing human behavior in active environments. We present our methodology in the context of pedestrian safety and crowd monitoring domain.

21.2 Literature Review

Several research issues have been addressed in the context of behavior analysis when visual modality is used as the main source of information. First of all, the vision-based system is required to distinguish pedestrians from vehicles and their typical movement patterns, respectively. Extraction of view-invariant features from raw data is critical for this purpose. It is also desirable to locate all moving objects (i.e., persons or vehicles) and to effectively map them on the world coordinate system of the site of interest. Extraction and formation of semantic information from raw video signal is at the heart of the *situational awareness* of the system.

Analysis of the movements of vehicles has been mainly done in the research domain of intelligent transportation systems (ITS). Early research on highway traffic monitoring was based on inductive loop technology [4]. Recent research uses vision sensors (i.e., cameras) [13] or multimodal sensors including vision sensors, audio sensors, seismic sensors [18]. Classification of vehicles and persons is an important issue in ITS [9, 7].

There has been active research effort for vision-based analysis of human activity in computer vision including video surveillance, human-computer interaction, virtual reality, choreography, and medicine. Reviews of general research on vision-based understanding of human motion can be found in [1, 6, 12]. Most of outdoor human monitoring systems have been developed under certain specific environmental contexts and assumptions: i.e., specific time, place, and activity scenarios involved in the

situation [8, 11, 20]. Exemplar surveillance systems have been either based on track analysis [14, 15, 19, 22] or body analysis [8]. Track analysis represents individual moving object as a moving point that corresponds to the center of gravity of the object, and aims at understanding such trajectory patterns and their meanings for the recognition of event (i.e., what is happening in the scene.) Track-level analysis is usually applied to wide-area surveillance of multiple moving vehicles or pedestrians in open space such as a parking lot or a pedestrian plaza in which individual object occupies tiny portion of the pixels in the image frame. In some wide-area surveillance situations, coarse representation of human body in terms of a moving bounding box or an ellipse may be enough for tracking [14].

Other researchers have applied more detailed representation of a human body such as a moving region or a blob [19, 22] to higher-resolution image data captured by Pan-Tilt-Zoom cameras in distributed surveillance systems. One of the recent developments in video surveillance is the usage of distributed system to cover multiple monitored scenes with various FOV's. Review of recent developments in intelligent distributed surveillance systems can be found in [21]. Body-level analysis usually focuses on more detailed activity analysis of individual persons. Velastin, et al. [22] estimated optical flow to compute the motion direction of pedestrians in subway environments. Makris, et al. [10] presented a method to learn scene semantics from multiple views. Park, et al. [16, 17] presented a synergistic framework that combines track-level and body-level analysis of multi-person interaction and activity.

Another important categorization of exemplar systems is related to indoor vs. outdoor setup. Comparing to indoor environments, outdoor environments have a lot of variations such as weather change, time shift from morning to evening, and moving backgrounds. Outdoor surveillance systems have to deal with those variations, and robustness is still an issue in outdoor surveillance. Most of the outdoor surveillance systems apply track analysis due to the limited image resolution, because the wide field of view (FOV) for outdoor surveillance usually limits the resolution of person appearance to relatively low-resolution images. Most of the research mentioned above mainly focuses on recognition of human activity, i.e., human-human interactions.

Recognition of human-machine interaction in outdoor environments such as human-vehicle interaction has not been actively addressed. This chapter presents a new framework to analyze human activity and interaction with vehicles as well as other humans for the enhancement of automatic situational awareness.

21.3 A Case Study

In this section, we present a multi-view based video surveillance system (MuVis) for analyzing the activity and interaction of persons and vehicles. Our system uses multiple cameras with different perspectives and analyzes the visual information at multiple levels. At gross level, we represent each moving object as a trajectory point of the center of gravity of the object. Track of the moving object is formed along the video sequence. At detailed level, we represent the object in terms of its footage area on the ground in order to estimate the view-invariant size of the object. We observe that the approximate size of the object's footage area is invariant to translation and rotation, unless the object falls or flips over. Planar homography is used to locate the object's footage position on the world coordinate system. At semantic level, the interaction

among persons and vehicles is analyzed. Contextual information including site model and activity scenario is integrated at the semantic level. The concepts of *spatio-temporal interaction boundary* and *time to collide* are introduced to represent and predict various interaction patterns among moving objects.

21.3.1 Planar Homography Mapping

Foreground moving objects are detected and segmented by background subtraction. Tracking of each object is performed by data association of foreground object blobs on the homography projection plane. However, image appearance of the same object varies significantly according to camera perspectives as shown in Fig. 21.2. Therefore, even though the tracker keeps following the same object, it does not classify the object category into vehicle or person. For the reliable classification of object types, we need to estimate the invariant size of the object. We rectify the images and map the objects to the world coordinate system using planar homography. We estimate the footage area and location of an object in the world coordinate system by using multiple-view geometry.

The geometric registration of a camera viewpoint is performed using a planar homography. Planar homography is a linear projective transformation H that relates two points P and P_v on a plane Π from two different views [5].

Fig. 21.3 depicts the process of estimating the footage area using homography. Multiple views of the same object are transformed by planar homography and the intersection of the projected images are used as the footage region of the object on the ground.

In a homogeneous coordinate system, let $p = (x, y, 1)$ denote the image location of a 3D scene point in one view and let $p' = (x', y', 1)$ be its coordinates in another view.



Fig. 21.2. Perspective effect on image appearance. The same object appears very different at different time frames as well as from different perspectives.

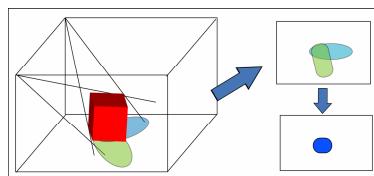


Fig. 21.3. Schematic diagrams for footage area estimation using multiple planar homography

Then, the homography H between the two views can be determined by the 4-point algorithm [5] as follows:

Given a set of 4 matching points p_i and p'_i , $i \in \{1, 4\}$, between two views, the perspective parameters h_{ij} (i.e., the elements of matrix H) correspond to a null space of the matrix A defined in Eq. 21.1 and are estimated using singular value decomposition (SVD) of A .

$$AH = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_1 & -x_1y_1 & -x_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y_1x_1 & -y_1y_1 & -y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x_2 & -x_2y_2 & -x_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y_2x_2 & -y_2y_2 & -x_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x_3 & -x_3y_3 & -x_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y_3x_3 & -y_3y_3 & -y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4x_4 & -x_4y_4 & -x_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y_4x_4 & -y_4y_4 & -y_4 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (21.1)$$

If we denote H_m^n as the homography from view m to n , we can register multiple cameras by series of concatenated homographies given in Eq. 21.2.

$$P_m^n = H_{n+1}^n H_{n+2}^{n+1} \cdots H_{m-1}^{m-2} H_m^{m-1} \quad (21.2)$$

In the current system, we map a point P_1 in view-1 and a point P_2 in view-2 to the points P_1^v and P_2^v , respectively, on a common virtual-view plane by homography matrices H_1^v and H_2^v , respectively. P_1^v and P_2^v are then averaged on the virtual-view plane.

$$\begin{aligned} P_1^v &= H_1^v P_1 \\ P_2^v &= H_2^v P_2 \end{aligned} \quad (21.3)$$

The coordinate system of the virtual view is specified by the 3D CAD model in Fig. 21.5.

Planar homography assumes all the pixels lie on the same plane (i.e., the ground plane in 3D world.) Pixels that violate this assumption result in mapping to a skewed location on the projection plane. By intersecting multiple projection maps of the same object, we can estimate the object's common footage region that observes the assumption.

Foreground moving objects on the homography plane are detected and segmented by frame differencing over multiple frames with a moving window. The foreground map F_k at frame k is obtained by thresholding the joint-likelihood map $F_k^0 = \sum_j F_k^j$ from j -th cameras which is computed by:

$$F_k^j = \left| \frac{1}{M} \sum_m I_m - \frac{1}{N} \sum_n I_n \right| \quad (21.4)$$

$$\begin{aligned} m &\in \{k, \dots, k - M + 1\} \\ n &\in \{k - M, \dots, k - M - N + 1\} \end{aligned} \quad (21.5)$$

The most recent M frames ($M = 3$) of input image I_m at frame m are averaged as the input frame and the next recent N frames ($N = 60$) are averaged as the moving window of background for the j -th camera at frame k . In the current paper, two camera inputs are summed for the joint-likelihood map F_k^0 . The motivation of using multi-frame differencing is that the mis-detection rate with the planar homography constraint is quite high. Therefore we want to reduce mis-detection and raising false alarm first. Then the raised false-alarm rate is effectively reduced by combining multiple homography constraints from multiple views.

Fig. 21.4 shows the results of homography mapping from two synchronized input video frames (Fig. 21.4(a),(b)) to a common virtual projection plane (Fig. 21.4(c).) Note that the raw image points lying on the ground plane (e.g., foot regions) coincide on the common virtual projection plane, while the raw image points that do not lie on the ground (e.g., torsos and heads of pedestrians) get skewed to form warped images on the projection plane.

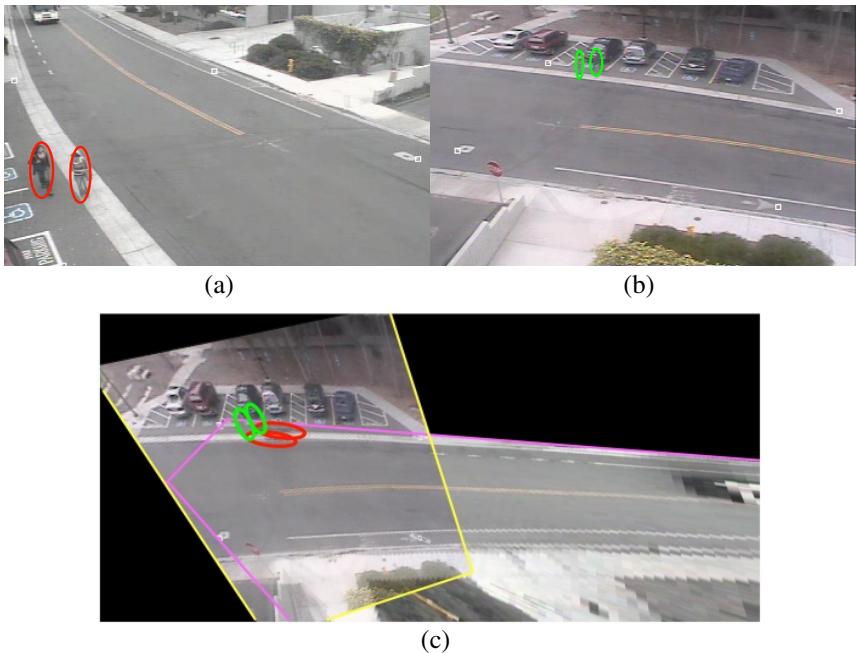


Fig. 21.4. Multi-view images registration into a virtual view using homography

21.3.2 Track Analysis in Spatio-temporal Domain

Moving object's true velocity (i.e., speed and direction) in 3D world coordinate system is estimated on the projection plane (e.g., Fig. 21.4 (c)). The velocity of a moving

object determines reaching boundary in a given time. This reaching boundary defines the *spatio-temporal interaction boundary* of the object. If there exists a foreign object at the vicinity of a moving object, the estimation of *time to collide* becomes important; the time of arrival or the time to collide has significant implication regarding safety in transportation systems. In the next section, we discuss the spatio-temporal analysis of tracks.

The spatio-temporal characteristics of the interaction boundary provides a useful tool to analyze human-human interactions as well as human-vehicle interactions in terms of time, velocity, and distance as described below.

A physical law of dynamics tells that the distance x that can be reached within time t is proportional to velocity v according to dynamics as formulated in Eq. 21.1. This implies that, with higher velocity, the range of impact of interaction can reach farther within a given time period.

$$x = vt \quad (21.6)$$

where v has directional component as well as magnitude.

Humans are subconsciously aware of this fact, and anticipate the consequence of speed with respect to safety. In the case of human movement, the direction of motion is ambiguous due to the possibility of agile body motion. Therefore, we make the directionality broader, resulting in a circular interaction boundary. It means that we model the interaction boundary as a circular shape with radius proportional to track velocity. In circular interaction boundary, the velocity v is replaced by speed $|v|$ and the reaching distance is represented in terms of distance $|x|$. In the case of vehicle movement, the direction of motion would be more deterministic depending on the driver's intent. Therefore, it would be more realistic to shape the interaction boundary of a vehicle more directional depending on velocity. However, for the sake of simplicity, we assume circular boundary for humans as well as vehicles.

The spatio-temporal interaction boundary can be categorized into *interaction potential* from *interaction region*. Both concepts are expressed in terms of spatial boundary that surrounds a moving object, but the former is related to anticipatory interaction, while the latter indicates actual interaction. We derive the effective radius of *interaction potential*, r_p , of a moving object:

$$r_p \approx |v|t \quad (21.7)$$

We model the radial shape of the interaction potential as a probability distribution function (PDF) in terms of a 2D Gaussian distribution, $R = N(\mu, \Sigma)$, truncated by the circle of radius r_p . The actual parameters of the PDF can be learned with training data. A similar formulation of pedestrian's moving directionality was proposed by Antonini and Bierlairein [2]. However, their method using manual tracking is computed on image plane from a single perspective and is perspective-dependent, whereas our approach is computed on projection plane using planar homography and is view-independent.

As seen above, the spatial and temporal analysis of tracks are highly correlated. We will present more details about the spatio-temporal analysis at track-level

modeling of human/vehicle activity in later sections in this chapter. The significance and connotation of a specific human track pattern depends on the site context: driveway, walkway, crowded area, etc. The relation between human track patterns and site context is mediated by policy, by which we mean which activity needs to be regulated/monitored and which activity is allowed. In this chapter, we are interested in the combination of spatial and temporal relations in the site context as summarized in Tables 21.1, 21.2. Table 21.1 shows the *spatial* site context of human activity, while Table 21.2 shows the *temporal* site context of interactivity between two objects. A person may stay, walk, or run at different sites such as walkway, driveway, or specific region of interest (ROI) at a bus-stop area or a building entrance zone. |, /, and × in the tables denote normal, cautious, and abnormal track patterns, respectively. Cautious or abnormal pattern at a specific ROI depends on the duration of stay and the site context. *Interaction region* is the actual boundary in which interaction between two objects occurs. We define the interaction region between two objects (i.e., person or vehicle) to be the intersection of the two interaction potentials.

Diagrams for the track-level analysis of human activity and interactivity are shown in Fig. 21.5. The figures from the left to the right show a track in 3D spatiotemporal space in *xyt* dimensions, the track's interaction potential boundary in speed (*v*) vs. spatial (*y*) dimension, planar view of the track and interaction potential in space (*x* vs. *y* axis), and the interaction duration between two tracks depicted by the rectangle along a time line, respectively.

The main focus on moving-person interactions in this chapter is regarding the macro-level concepts such as *approach*, *pass-by*, *depart*, etc. This kind of interactions is characterized by short duration of the interaction period.

Table 21.1. Spatial context dependency of human activity

Person \ Site	Stay	Walk	Run
Walkway		/	×
Driveway	×		
ROI	, ×		×

|, /, and × denote normal, cautious, and abnormal track patterns, respectively.

Table 21.2. Temporal context dependency of interactivity between two objects

Object-2 \ Object-1	Stay	Slow	Fast
Stay		/	/
Slow		/	×
Fast	/	×	×

Legends are the same as in Table 21.1.

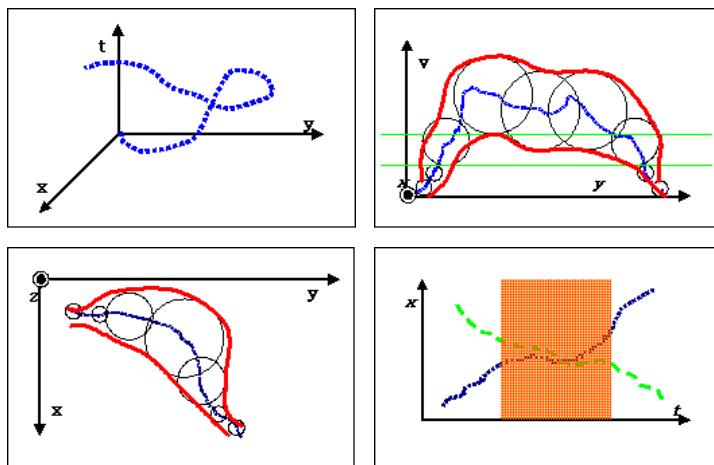


Fig. 21.5. Schematic diagrams for trajectory analysis in spatio-temporal space. Circles represent interaction potential boundaries at a given space/time. Red curves represent the envelopes of the interaction boundary along tracks.

21.3.3 Distributed Sensor Placements and Site Model

Environmental context, especially spatial environment, can be represented by site modeling. Various approaches are possible depending on the available site information. If the 3D structural information is available, we can build a 3D CAD model of the site, which is useful for representing important structures such as buildings and roads. The merit of 3D CAD model is that it provides actual 3D world coordinate systems for the site. But this modeling usually requires multiple cameras and accurate camera calibration.

If the site is mainly composed of a flat ground plane, then we can build a planar homography. The advantage of homography-based modeling is that it provides perspective-compensated plan view of the site. It may require multiple cameras with overlapped field of view (FOV).

If the site is arbitrarily complex or spatial configuration is ambiguous from camera view, we can still manually assign region of interest (ROI) for specific interest regions. Most of the single camera-based 2D site modeling falls in this category. The advantage of 2D site modeling is that it is flexible and simple. Some ambiguity is inevitable due to occlusion, perspective distortion of the view, etc.

We have a real-world test bed for an intelligent infrastructure (called ‘smart space’) with the combination of the above modeling options to generate a heterogeneous site model. Fig. 21.6(a) shows an image of the actual building which is located in the satellite image in Fig. 21.6 (d). A 3D CAD model is made and texture-mapped based on architectural data about the building structure and floor plans (Fig. 21.6(a), (b).) Four cameras are mounted on specific locations of the building to cover surrounding roads (Fig. 21.6 (c)). Camera placements are indicated by (C1–C4) with viewing directions and the corresponding view areas (A1–A4). Cameras 1 and 2 view Area-1, Camera-3

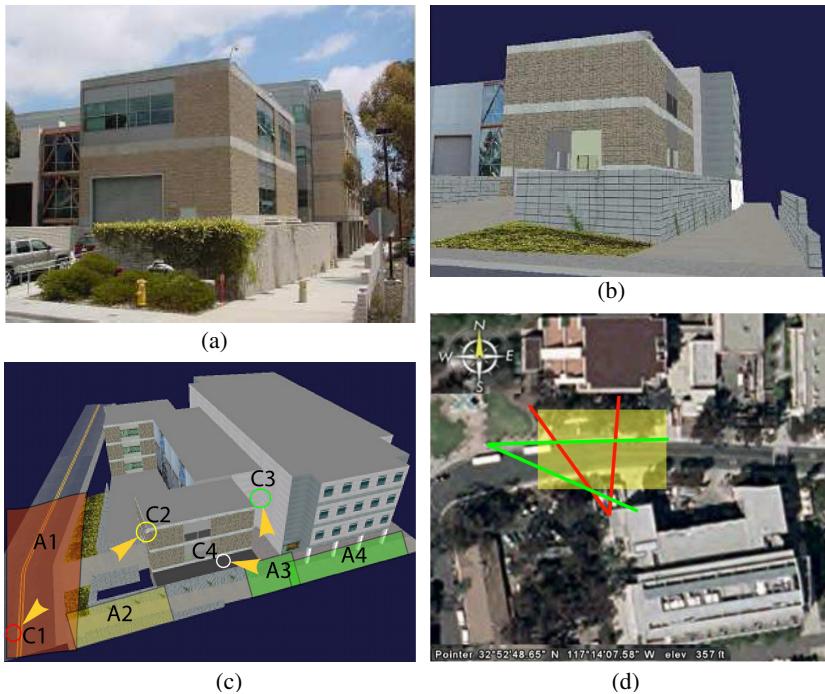


Fig. 21.6. The real test bed for the current system: (from top-left to bottom-right): (a) actual building, (b) its 3D site model, (c) camera placements (C1-C4) with viewing directions to areas (A1-A4), and (d) Area-1 (A1) in yellow in the satellite image, respectively.

views Area-3, and Camera-4 views Area-4, respectively. This chapter focuses on Area-1 viewed from Cameras 1 and 2. Area-1 viewed from C1 and C2 is shown in yellow in Fig. 21.6 (d); straight lines depict the camera fields of view, and the yellow rectangular region corresponds to the planar homography result in Fig. 21.5 (c).

21.3.4 Tracking Multiple Objects

Vision-based tracking of multiple objects starts from the processing of foreground segmentation. We use the frame differencing technique with posterior morphological operation for the segmented foreground.

Tracking of the detected object is performed by data association between consecutive frames in the homography domain. Fig. 21.7 shows the multi-view based video surveillance system (MuViS) for analyzing the activities of persons and vehicles. Upper panel shows examples of synchronized video input frames. Lower panel shows the front end of the graphical user interface of the MuViS; detected objects are mapped and tracked on the virtual top-down view projection plane by homography.

The projection plane is divided by the virtual thick horizontal bars that segment the plane into three ROIs called ‘Walkway-North’, ‘Driveway-Middle’, and ‘Walkway-South’, from top to bottom, respectively. Note that a pedestrian (of object ID: 2) is

walking along the ‘Walkway-North’, while two small vehicles (of object ID’s 0 and 1) are passing by in the ‘Driveway-Middle’.

The multi-object tracking uses 2D Gaussian ellipse representations of foreground regions. Tracking is performed by using a variant of the modified probabilistic data association filter (PDAF) [3] combined with a Kalman filter. Fig. 21.7 shows an example of object tracking on the homography plane. Each detected object is represented on the virtual grid of 10×10 pixels by a tightly surrounding ellipse, and the Kalman-filter based tracking parameters provide the velocity estimation. The three rectangles depict the detected objects. Moving object’s true velocity (i.e., speed and direction) in 3D world coordinate system is estimated at the projection (i.e., homography) plane. The velocity of a moving object determines the object’s reaching boundary in a given time. If there exists a foreign object at the vicinity of a moving object, the estimation of *time to collide* becomes important; the time of arrival or the time to collide has significant implication regarding safety in transportation systems.

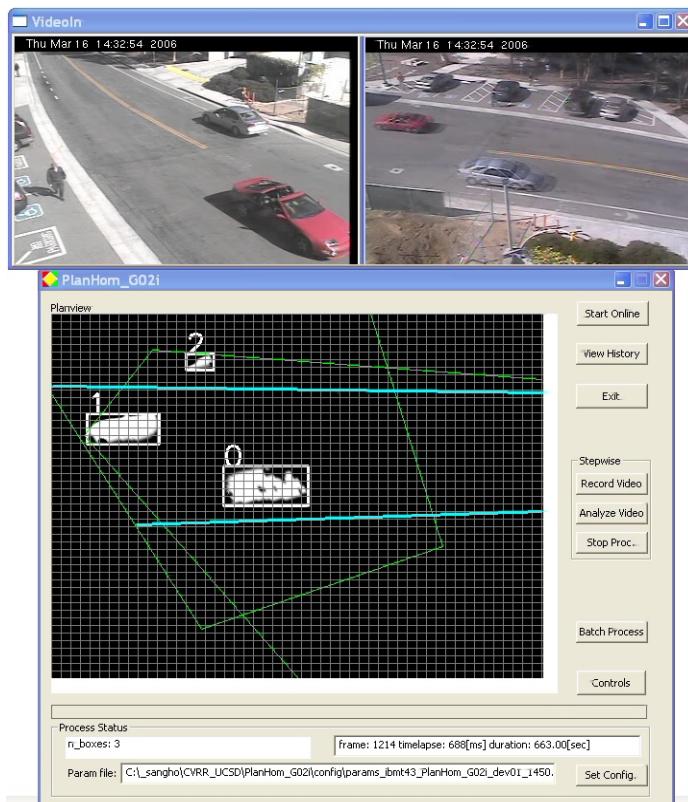


Fig. 21.7. Multi-view based video surveillance system (MuViS) for analyzing the activities of persons and vehicles. Upper panel: synchronized video input frames. Lower panel: the front end of the graphical user interface of the MuViS.

The virtual fine grid overlaid on the projection plane effectively estimates the footage areas of individual moving object blobs, which forms the dynamic density patterns of object occupancy on the ROIs.

We have tested our system with video data captured at area A1 in Fig. 21.6 during different day times for several days. Two cameras C1 and C2 were used to capture the views. Images in Fig. 21.4(a)(b) are example views from camera C1 and C2 with detected persons, respectively, and the homography-based registration result is shown in Fig. 21.4(c). The ground truth for the image registration is obtained from satellite imagery in Fig. 21.6(d).

The site context information of the system also includes various statistics computed on the fly for crowd density plot, pedestrian flow directions, vehicle traffic histogram, etc.

21.3.5 Dynamic Density Estimation

In wide-view open area, counting individuals may not be possible or robust especially when the site is crowded. Therefore, it would be more useful to estimate the detected objects' density, range, and moving velocity in the world coordinate system using the footage areas for each group of objects. Fig. 21.8 shows our estimation of dynamic density patterns of crowds and moving vehicles observed on a sunny day. Each row in Fig. 21.8 shows multi-perspective image frames, and the moving objects' detected footage regions mapped on homography plane. The upper four rows in Fig. 21.8 show the scene change in terms of spatial distribution of moving objects. The last row shows the probability density functions (PDFs) of crowdedness of the upper four rows.

The PDFs were estimated by dividing the homography plane inherently into grid regions and computing the density of the footage pixels in each grid cell.

From the tested experimental site, it is observed that the driveway is sporadically occupied by fast moving high-density large blobs classified to vehicles, whereas the pedestrian walkways are frequently occupied by slow-moving sparse blobs classified as moving crowds. This empirical observation supports our framework for the spatio-temporal analysis of site context in Tables 21.1, 21.2.

The density patterns and their dynamic changes provide the information about how each region of the monitored site is occupied by people or vehicles for how many frames and how they interact. Fig. 21.8 shows example plots of dynamic density patterns acquired on different times on different days. The information is effective to provide enhanced situational awareness. The dynamic density plots can be utilized for video query in order to perform post-mortem analysis.

Fig. 21.9 shows the results of experiments on dynamic density estimation in natural settings where unobtrusive busy traffic flows of multiple pedestrians and vehicles were involved (as in Fig. 21.8.) High peaks with the number of cells greater than 50 were classified as large vehicles such as buses and trucks, mid-size peaks between 10 and 50 cells were classified as small vehicles such as sedan and minivan, and small peaks lower than 10 were classified as pedestrians in isolation or in a crowd.

Table 21.3 shows the confusion matrix of the object classification using the dynamic density plots including those in Fig. 21.9. A single sample in the table represents the whole span of a peak from any region of the ROIs. In the confusion matrix,

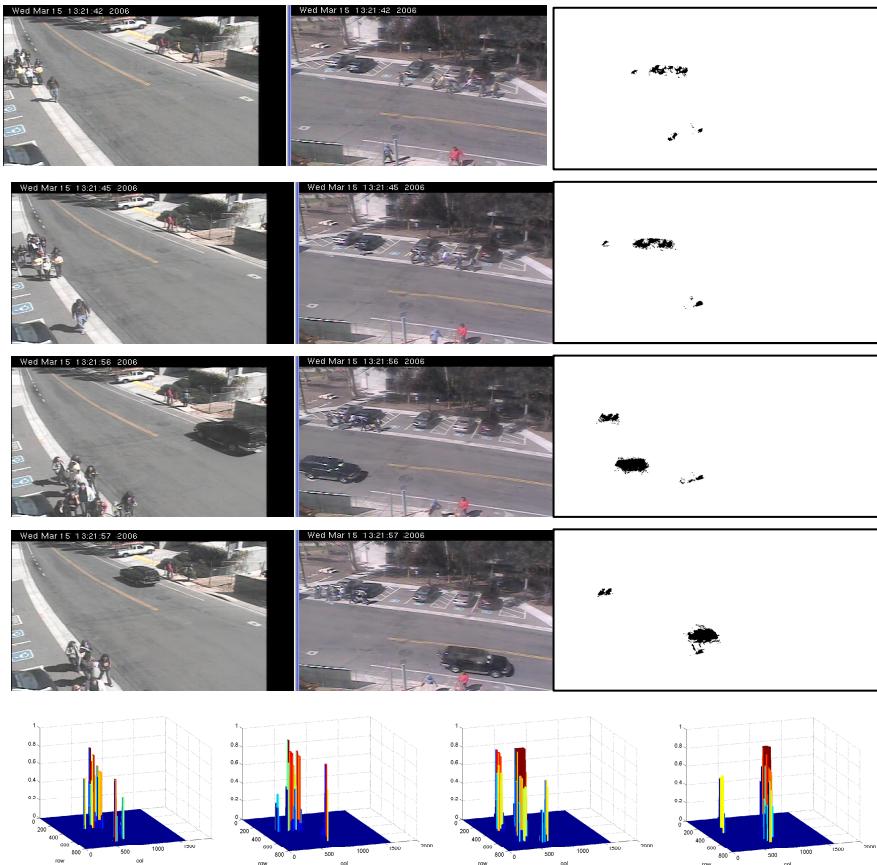


Fig. 21.8. Dynamic density estimation of crowds and moving vehicles captured at 0, 3, 14, and 15 seconds. PDFs correspond to the homography maps.

Table 21.3. Confusion matrix of object classification results using the dynamic density plots

	LV	SV	Ps	recall
LV	0.95	0.05	0	0.95
SV	0.05	0.95	0	0.95
Ps	0	0.06	0.94	0.94
precision	0.93	0.90	1.0	

LV, SV, Ps denote large vehicle, small vehicle, and pedestrian(s), respectively.

recall is defined as the fraction of the total number of objects in a particular class that are classified correctly by the system for that class. *Precision* is defined as the fraction of objects recognized for a particular class that are actually correct.

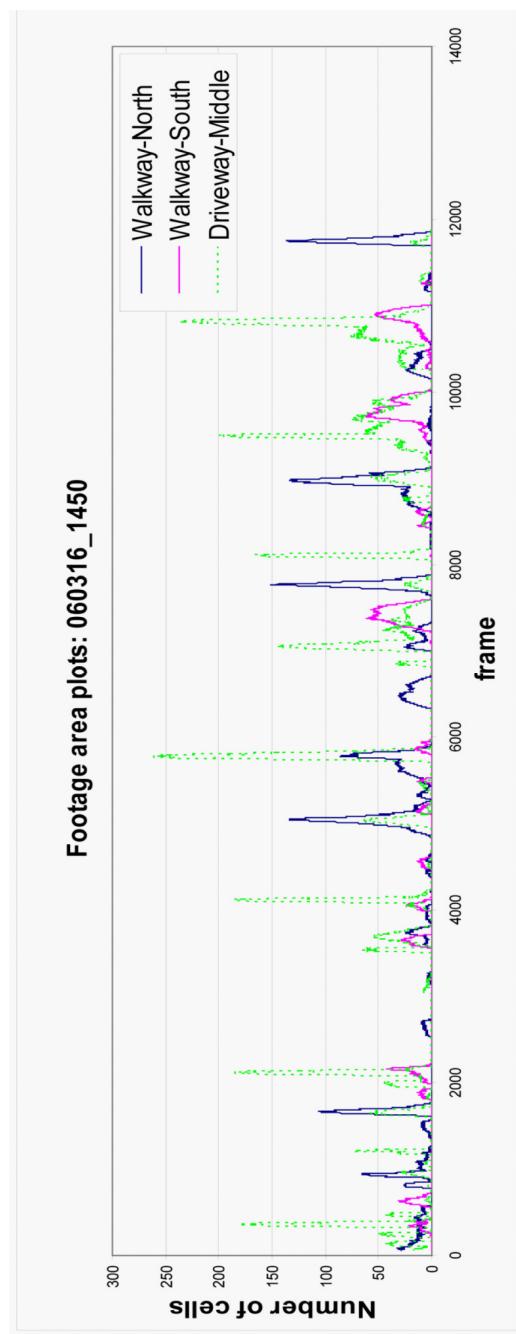


Fig. 21.9. Dynamic Density estimation of moving objects at different ROIs captured on different days

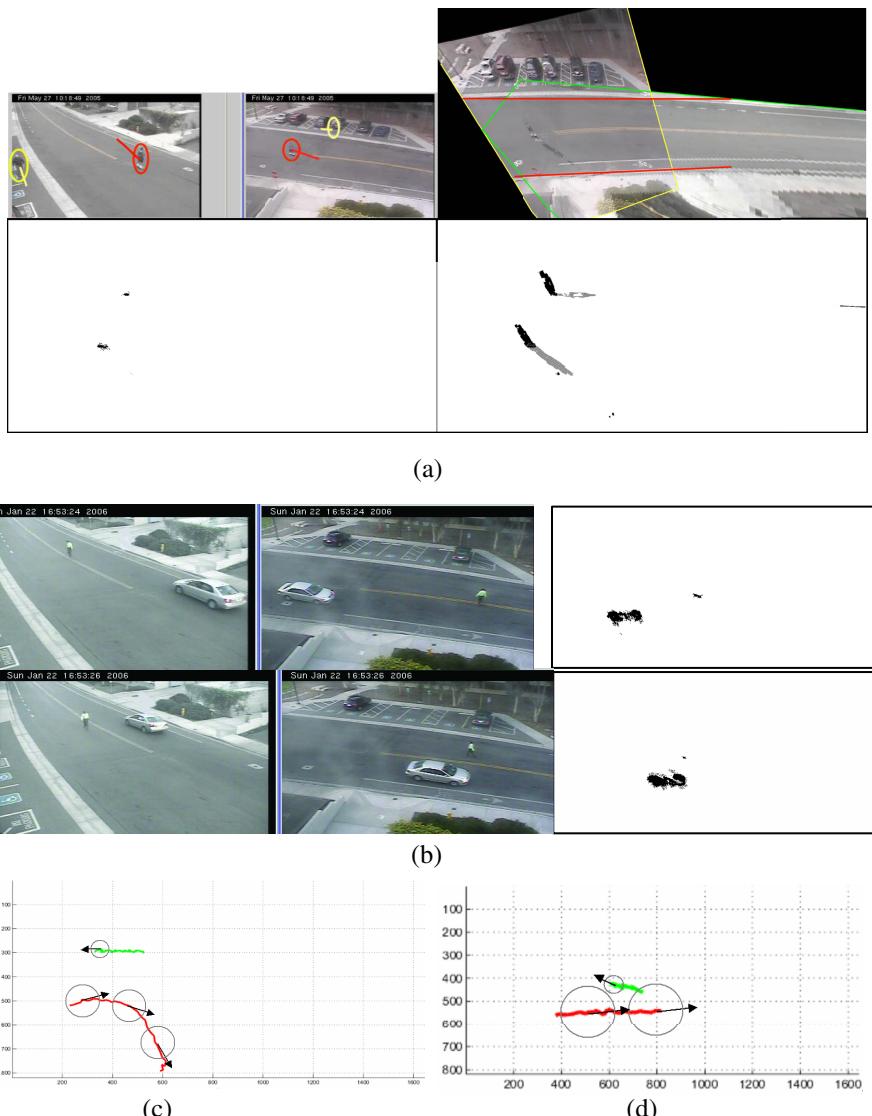


Fig. 21.10. Estimation of interaction patterns of moving objects with different velocities using the homography-mapped footage regions in the projection planes. Walking person in green plot vs. skateboarding person in red plot (a)(c). Walking person in green plot vs. driving car in red plot (b)(d).

Fig. 21.10 shows more detailed analysis: the estimation of interaction patterns between different moving objects.. The interaction patterns are analyzed using the tracks of the homography-mapped footage regions in the projection plane corresponding to the region of interest in Fig. 21.5(d). In Fig. 21.10(a), the images starting clockwise from upper left panel show that multiple views of the site with two detected persons

in circle, homography projection plane map of the two views, overlay of the two projections of foreground regions (not shadow!), and the footage areas of each person obtained from the overlay, respectively. Fig. 21.10(c) shows two simultaneous tracks of (a): a walking person's track in green and a skateboarding person's track in red in different speeds. Piecewise velocities are represented by arrows, and spatio-temporal interaction potentials are denoted by gray circles at each time instance. The absence of overlap between the interaction potentials of the two tracks successfully indicates that the monitored scene is in safe situation. In Fig. 21.10(b), the images on the upper and lower rows show the raw input views and the detected footage regions in projection plane at different moments. Fig. 21.10 (d) shows two simultaneous tracks of (b): a walking person's track in green and a moving vehicle's track in red. The person's proximity to the big interaction potentials (denoted by gray circles) of the vehicle properly indicates the danger of a possible hit.

21.4 Discussions and Conclusions

In this chapter we have presented a multi-perspective vision-based analysis framework to analyze human and vehicle activities for enhanced situational awareness in video surveillance. Planar homography using multiple perspectives provides view-invariant estimation of footage area of objects for object classification. Tracks of the moving objects are robustly estimated with the footage regions in the world coordinate system. Spatio-temporal interrelationship between human and vehicle tracks is capitalized in terms of different combinations of track vs. site context such as walkway and driveway. The concepts of *interaction boundary* and *time to collide* of each moving objects are introduced in order to build semantically meaningful situational awareness. A case study demonstrated experimental evaluation of the proposed method applied to pedestrian safety and disaster prevention domain. The proposed multi-perspective vision system and the multi-level analysis framework can be applied to broader domains including emergency response, human interactions in structured environments, and crowd movement analysis in wide-view sites.

References

1. Aggarwal, J.K., Cai, Q.: Human motion analysis: a review. *Computer Vision and Image Understanding* 73(3), 295–304 (1999)
2. Antonini, G., Bierlaire, M.: Capturing interactions in pedestrian walking behavior in a discrete choice framework. *Transportation Research Part B* (2005)
3. Bar-Shalom, Y., Blair, W.: Multitarget-multisensor tracking: applications and advances, Norwood, MA, vol. 3, pp. 199–231 (2000)
4. Coifman, B.: A new algorithm for vehicle reidentification and travel time measurement on freeways. *ASCE Applications of Advanced Technology in Transportation* (1998)
5. Criminisi, A., Reid, I., Zisserman, A.: A plane measuring device. *Image and Vision Computing* 17(8), 625–634 (1999)
6. Gavrila, D.: The visual analysis of human movement: a survey. *Computer Vision and Image Understanding* 73(1), 82–98 (1999)
7. Gupte, S., Masoud, O., Martin, R.F.K., Papanikolopoulos, N.P.: Detection and classification of vehicles. *IEEE Trans. Intell. Transport. Syst.* 3(1), 37–47 (2002)

8. Haritaoglu, I., Harwood, D., Davis, L.S.: W4: Real-time surveillance of people and their activities. *IEEE transactions on Pattern Analysis and Machine Intelligence* 22(8), 797–808 (2000)
9. Lipton, A.J., Fujiyoshi, H., Patil, R.S.: Moving target classification and tracking from real-time video. In: *Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision*, Princeton, New Jersey, pp. 8–14 (1998)
10. Makris, D., Ellis, T., Black, J.: Learning scene semantics. In: *ECOVISION 2004 Early Cognitive Vision Workshop*, Isle of Skye, Scotland, UK (2004)
11. McKenna, S.J., Jabri, S., Duric, Z., Wechsler, H.: Tracking interacting people. In: *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2000)*, pp. 348–353 (2000)
12. Moeslund, T.B., Granum, E.: A Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding* 81(3), 231–268 (2001)
13. Morris, B., Trivedi, M.M.: Improved Vehicle Classification in Long Traffic Video by Co-operating Tracker and Classifier Modules. In: *IEEE Conference on Advanced Video and Signal based Surveillance* (2006)
14. Oliver, N.M., Rosario, B., Pentland, A.P.: A Bayesian Computer Vision System for Modeling Human Interactions. *IEEE Trans. Pattern Analysis and Machine Intelligence* 22(8), 831–843 (2000)
15. Park, S., Trivedi, M.M.: A track-based human movement analysis and privacy protection system adaptive to environmental contexts. In: *IEEE International Conference on Advanced Video and Signal based Surveillance*, Como, Italy (2005)
16. Park, S., Trivedi, M.M.: Analysis and Query of Person-Vehicle Interactions in Homography Domain. In: *IEEE Conference on Video Surveillance and Sensor Networks*, Santa Barbara, USA (2006)
17. Park, S., Trivedi, M.M.: Multi-person Interaction and Activity Analysis: A Synergistic Track- and Body- Level Analysis Framework. *Machine Vision and Applications* (to appear, 2007)
18. Ploetner, J., Trivedi, M.M.: A Multimodal Approach for Dynamic Event Capture of Vehicles and Pedestrians. In: *Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks*, pp. 203–209 (2006)
19. Remagnino, P., Shihab, A.I., Jones, G.A.: Distributed intelligence for multicamera visual surveillance. *Pattern Recognition: Special Issue on Agent-based Computer Vision* 37(4), 675–689 (2004)
20. Trivedi, M.M., Gandhi, T., Huang, K.: Distributed interactive video arrays for event capture and enhanced situational awareness. In: *IEEE Intelligent Systems, Special Issue on Artificial Intelligence for Homeland Security* (2005)
21. Valera, M., Velastin, S.A.: Intelligent distributed surveillance systems: a review. *IEEE Proceedings Vision, Image and Signal Processing* 152(2), 192–204 (2005)
22. Velastin, S.A., Boghossian, B.A., Lo, B., Sun, J., Vicencio-Silva, M.A.: Prismatic: Toward ambient intelligence in public transport environments. *IEEE Transactions on Systems, Man, and Cybernetics -Part A* 35(1), 182–214 (2005)

Online Resources

1. **Intel Open Source Computer Vision Library:** Open source and library for high level programming in C++ for computer vision.
<http://www.intel.com/research/mrl/research/opencv/index.htm>

2. **MatLab help:** Online help page for programming in MatLab.
<http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml>
3. **CVonline:** A useful website for the evolving, distributed, non-proprietary, on-line compendium of computer vision.
<http://homepages.inf.ed.ac.uk/rbf/CVonline/CVentry.htm>
4. **Computer Vision Homepage:** A useful website at Carnegie Melon University for links to computer vision related resources.
<http://www.cs.cmu.edu/~cil/vision.html>
5. **Computer Vision and Robotics Research Laboratory, UCSD:** The authors' laboratory website that contains publication links for studies on visual surveillance, intelligent and safe automobiles, etc.
<http://cvrr.ucsd.edu/>
6. **DoT:** US Department of Transportation official website.
<http://www.dot.gov/>
7. **NHTSA:** U.S. National Highway Traffic Safety Administration.
<http://www.nhtsa.dot.gov/>
8. **IEEE-ITS:** IEEE Transactions on Intelligent Transportation Systems.
<http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=6979>

Questions for Discussions

1. How can your system distinguish moving crowds from moving vehicles? What kinds of features will be discriminative?
2. How can you extend the homography-based surveillance system in the current chapter to non-coplanar sites?
3. The current implementation of the contextual dependency of connotation in Table 21.1 is static in that it does not change over time. Is it sufficient? If not, how can you improve it to be dynamic and adaptive over time?
4. The current system in this chapter is a system-level or environment-level solution for safety and security in transportation systems. From the viewpoint of an individual, how can the system communicate with an individual pedestrian or driver for his/her own safety in the environment? How can you utilize the information from the system-level solution?
5. The current configuration of the two cameras in Fig. 21.6(d) is quasi-orthogonal (i.e., almost perpendicular in their viewing directions.) Do you think if there is any specific reason for this? If this configuration is not available, what alternative configuration of two cameras would you choose for what reasons?
6. If you have more than two cameras available, what kind of configurations would you prefer for what reasons?

Video-Based Deception Detection

Matthew L. Jensen, Thomas O. Meservy, Judee K. Burgoon,
and Jay F. Nunamaker Jr.

Center for the Management of Information,
University of Arizona, USA
`{mjensen, tmeservy, jburgoon, jnunamaker}@cmi.arizona.edu`

Abstract. This chapter outlines an approach for automatically extracting behavioral indicators from video and explores the possibility of using those indicators to predict human-interpretable judgments of involvement, dominance, tenseness, and arousal. The team utilized two-dimensional spatial inputs extracted from video to construct a set of discrete and inter-relational features. Then three predictive models were created using the extracted features as predictors and human-coded perceptions of involvement, tenseness, and arousal as the criterion. Through this research, the team explores the feasibility and validity of the approach and identifies how such an approach could contribute to the broader community.

22.1 Introduction

Deception is a common occurrence and though researchers have enhanced our understanding of this phenomenon over the last several decades, there is still much to learn about deception and how to detect it. Numerous deception detection techniques have been tested; however, many of these techniques are invasive and impractical to use outside of the laboratory. Often, these deception detection techniques utilize low-level cues which may be difficult for a human to interpret.

Humans are not particularly good at detecting deception. Even when trained, human accuracy at detecting deception is just better than chance [20]. All too often, people rely on detrimental biases and misconceptions of which cues are the best indicators of deception. Most deception detection techniques rely on a combination of cues to make a prediction of deception rather than a single cue. Humans often have difficulty in tracking and managing the numerous cues that might be the best predictors of deception. Moreover, because deception indicators are subtle, dynamic, and transitory, they often elude humans' conscious awareness. If computer-assisted detection tools could be developed to provide additional, more meaningful interpretation of an individual's behavior—such as, how involved they are, their level of tension, dominance in a conversation, or how aroused they are—then humans could incorporate more meaningful perceptions into their final attribution of whether or not an individual is being deceptive.

This chapter outlines an approach for automatically extracting behavioral indicators from video and explores the possibility of using those indicators to predict human-interpretable judgments of involvement, dominance, tenseness, and arousal.

22.2 Literature Review

The extant knowledge that relates to deception is vast and detailed. The following section provides an overview of related works and presents our model for deception detection.

22.2.1 Humans and Deception Detection

In the search for observable cues of deception, researchers have examined countless nonverbal behaviors. Among the cues most closely correlated with deception are receiver perceptions about deceptive communication [15, 27]. Interestingly, this is in direct contrast with the notion that humans are poor at detecting deception. One research study found that receiver judgment concerning communication quality was actually more diagnostic in identifying deception than direct estimates of deception [28]. Consistent with this claim, other researchers have underscored the importance of human perception in deception detection [15]. Researchers have highlighted the importance of logical structure, sender immediacy (psychological closeness conveyed by language), and overall sender-receiver involvement. Behaviors associated with uncertainty and avoidance have also been linked to deception. Such behaviors include admitted lack of memory, lack of details, and uncooperativeness. Further, elevated levels of tenseness have also been associated with deception. Tension may be manifested in the voice, but can also be overtly displayed by gesture and other body movement [15]. These cues are more easily identified by a human observer than deception itself.

Training has had limited success in alleviating the issue of poor discriminatory ability in detecting deception. While there is currently some debate concerning the effectiveness of training in deception detection, there appears to be growing support for training having a positive, significant effect on deception detection ability [17, 20]. Indeed, some researchers believe that current research underestimates the ability of humans as lie-detectors [17].

22.2.2 Tools to Detect Deception

In addition to training, researchers have courted the idea of developing tools meant to assist those entrusted with assessing credibility. Such tools take advantage of behavioral indicators that co-occur or that are non-existent with deception. Perhaps the most well-known method of deception detection is the polygraph. The polygraph assists in detecting deception by sensing physiological changes in respiration, heart rate, and palmar sweat due to arousal or anxiousness which may accompany deception. Other deception detection tools and methods also take advantage of physiological changes that may result from deception. These include voice stress analyzers, brain activity analysis, and thermal scanning [13].

In a different approach, other researchers have explored deception detection via overt behavioral displays. Research in this vein focuses on language use and body movement. For example, micro-momentary facial expressions revealing fear or surprise are proposed to be indicative of deceit. Behavioral methods have the advantage

of not requiring sensors to be directly attached to the body, though their reliability is still being assessed [13].

By building on the notion that human perception is critical in deception detection, we have pursued a video-based method of deception detection that attempts to *automatically* capture many of the important perceptual cues. This method is a behavioral method which analyzes indicators in the kinesics channel.

22.2.3 Brunswikian Lens Model for Deception Detection

Cataloguing the nearly limitless combinations of behavior might seem an insurmountable task, were it not for pioneering efforts by Birdwhistell [2, 3] to reduce over 150,000 possible body, facial, and gestural movements to 50-60 basic building blocks called kinemes. Kinemes, which are analogous to phonemes in the field of linguistics, do not have meaning in themselves. They must be combined into kinemorphs and kinemorphic classes—specific patterns of behavior that are analogous to morphemes, or words, and clauses. This structural linguistics approach launched the study of kinesics, one of the primary nonverbal communication codes. In like vein, Trager [23] analyzed the voice, providing foundation for another nonverbal code variously labeled as vocalics, paralanguage, or prosody.

Though the structural approach is useful in identifying specific “atoms” of nonverbal communication, mapping specific behavioral components to meanings is best accomplished by taking a functional approach in which movements are categorized according to the communicative functions they serve and to which humans attach interpretations [4]. For example, gestures that accompany speech and illustrate its content are called illustrators; gestures that alleviate psychological or physical discomfort are called adaptors; gestures that regulate the flow of conversation are called regulators or interactional gestures [16, 1]. At a molar level, constellations of behavior work in concert to express meta-communicative messages such as involvement, dominance, or hostility. For example, frequent and expansive illustrator gesturing, absence of adaptors, rapid speaking tempo, interruptions, and intense language together convey dominance. Compelling evidence that many nonverbal behaviors have clearly recognized social meanings and are used with regularity within a given speech community [4, 7] implies that the possible meanings assigned to sequences and combinations of multimodal cues can be identified.

One way to model the relationship between structural indicators and functional meanings is to apply a modified Brunswikian lens model to map behavioral indicators to meta-communicative meanings [18, 24], an approach we have used successfully to identify clusters of nonverbal features that predict relational messages and credibility [6, 8]. In a lens model, objective indicators called distal cues (D) are mapped to intermediate perceptual judgments called proximal percepts (P), which in turn combine to yield subjective attributions or meanings (A). Fig. 22.1 illustrates the model. An internal state (C) such as heightened activation generates several distal indicators (e.g., frequent gesticulation, forward lean, elevated fundamental frequency, rapid speaking tempo), which are perceived as potency and intensity and which become the subjective judgments having the most proximal influence on a meta-communicative attribution such as relational dominance or hostility.

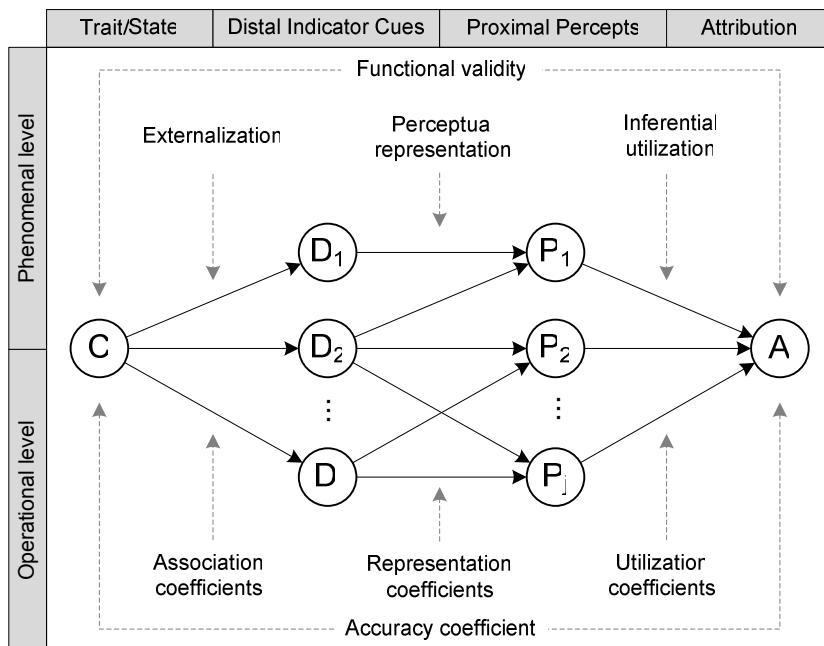


Fig. 22.1. Brunswikian lens model

We are developing a taxonomy of the fundamental meta-messages of interpersonal communication [5], which constitute the proximal percepts, and the distal indicators that should be associated with each. This taxonomy guides the selection of distal indicators to be automated, and the proximal percepts or meta-messages, measured by human judges, with which they are associated.

We have applied this model to identify configurations of micro-level deception cues that predict mid-level percepts which in turn predict attributions. Fig. 22.2 displays an operationalized view of the model using communication dimensions as proximal percepts that can be combined to arrive at an attribution of an individual's level of honesty (on a scale of 0 to 10; 0 being completely deceptive and 10 being completely honest).

In our data sets, humans participate in deception which is represented by the state characteristic (C) in the lens model. The distal indicators, D₁–D_j, are automated features extracted through kinesics analysis (described below). The proximal percepts, P₁–P_j, are communication dimensions (e.g., involvement, dominance, tenseness, arousal) derived from judgments made by third-party observers. The final attribution, A, is a prediction of a self-reported honesty score using proximal percepts as predictors. This attribution is validated through comparison with the original characteristic (C). In the case of deception detection, both the characteristic and attribution can be viewed as an inverse relationship with the level of honesty. Sample indicators are shown for each of the major components of the model in Fig. 22.2.

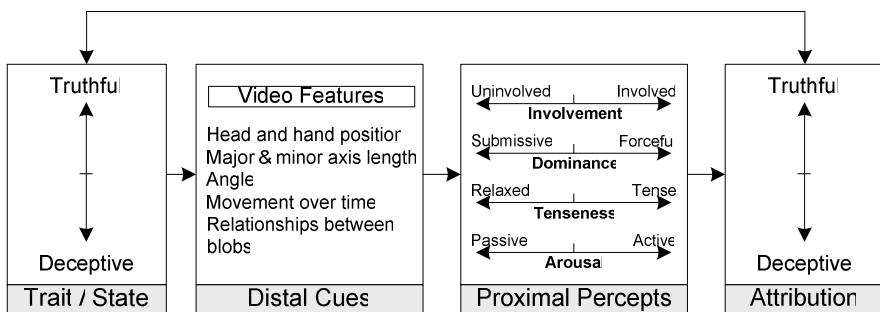


Fig. 22.2. Brunswikian lens model applied to deception detection

22.2.4 Distal Cues, Proximal Percepts, and Deception

When applying the Brunswikian lens model to the problem of deception detection, expected relationships between the components of the lens model need to be specified. Fortunately, past research has provided, to a large extent, empirically-based links between proximal percepts and attributions. In our model, proximal percepts are operationalized as human perceptions of dominance, tensionness, arousal, and involvement. These broad perceptions of human communication encompass a great deal under a single assessment. For example, involvement can be divided into subcomponents of immediacy, altercentrism, expressiveness, conversational management, and social anxiety [14]. Likewise, dominance can be divided into influence, conversational control, focus and poise, panache, and self-assurance [9]. While more granular measures of the subcomponents can be productively and selectively studied via a Brunswikian lens model, they are conglomerated in the current study.

With the selection of the proximal percepts established, associated distal cues can be determined. There are numerous possible cues which may account for perceived levels of the proximal percepts. The search for relevant distal cues will be bounded by existing research in deception detection. Cues with a demonstrated or hypothesized correlation to deception will be reviewed for ties to proximal percepts and ultimately an attribution of the level of deception or honesty.

Table 22.1. Sample proximal percepts and distal cues associated with deceptiveness

Proximal Percepts	Observed Levels	Distal Cues
Dominance	Lowered	Limited hand movement over time
Tensioness	Elevated	Minor hand movements which are close together Rigid head movement
Arousal	Mixed	Frequent hand-to-face gesturing and hand-to-hand movements
Involvement	Lowered	Limited gestures away from the body

Proximal percepts, which were predicted using automatically-extracted distal cues, are used to predict an honesty score which can also be thought of as an estimated level of deception. A sample of proximal percept levels and distal cues that may be associated with deception is shown in Table 22.1. We have adopted a continuous measure of deception as the final attribution because it does not force a dichotomous classification of truth or deception. Dichotomous classifications may be derived from the continuous measure through setting an appropriate threshold (see [26]).

22.3 Research Approach

Our general automated approach for extracting distal cues is based on a typical pattern classification approach. The input stream is segmented into meaningful units, low- and higher-level features are extracted from these units, and then well-known regression techniques predict a meaningful level of a communication perception.

22.3.1 Automated Kinesics Feature Extraction

Nonverbal feature extraction occurs via a prototypical system that has evolved over the last several years. The preferred input for this system is high quality digital video of a single subject who is in a sitting position away from any objects, such as a table, that might hide the hands or head. Higher quality video allows for more reliable position estimation of the head and hands. However, the system has also been used successfully with converted analog video. The first step in the process is to segment the video into meaningful units, such as an answer to an interview question. Next, low-level features are identified by estimating hand and face positions. Although many strategies exist for hand and face recognition, our system uses algorithms developed by the Computational Biomedicine Imaging and Modeling Center (CBIM) at Rutgers University [21] that use color analysis, eigenspace-based shape segmentation and Kalman filters to track the head and hands through each frame of a video segment.

Ellipses approximate the location and size of the head and hands. The center point, axes' lengths and angle of major axis are then calculated for each ellipse. From these basic features (shown in Fig. 22.3(a)), approximately 150 additional single-frame and multiple-frame features are calculated. The distance between two blobs (e.g., right hand to head distance) is an example of a single frame feature. How far a blob has moved between frames is an example of a multiple frame feature. The distance feature in Fig. 22.3(b), hints at gestures that may indicate nervousness (such as preening, scratching, rubbing, etc). Fig. 22.3(c) illustrates another single-frame feature, the quadrant feature, which allows us to understand how often each blob is in each spatial region surrounding the body and may help to discriminate between an open and closed posture [22].

Multi-frame features require information from two or more frames. An example of a multi-frame feature is the distance and direction a blob has traveled between frames. Distance can be calculated using the Euclidean distance formula on the center points of a blob in two successive frames. Fig. 22.4 illustrates the distance feature. Features are computed for each frame in the video clip and then summarized for every meaningful segment. Means and variances of these measurements are calculated based on total number of frames for the video segment.

Numerous features can be inferred from the general metrics that are extracted as a result of blob analysis. Many of these features attempt to capture key elements (e.g., kinemorphs) that a human coder would look for when trying to identify more complex distal cues (e.g., gestures) or proximal percepts. However, a number of additional features that typically aren't measured by human coders (e.g., distance of movement per frame) are extracted because they are potentially useful in an automated environment

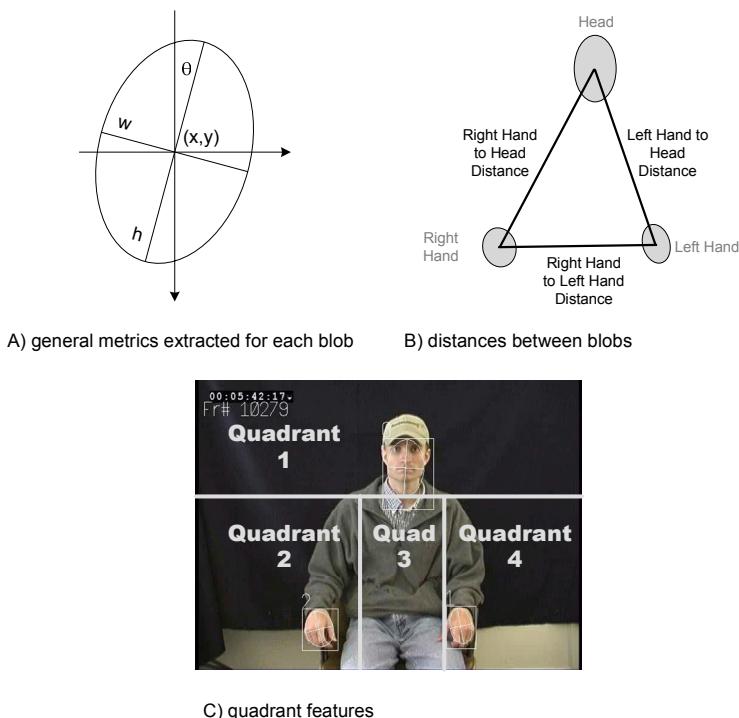


Fig. 22.3. Sample single frame features

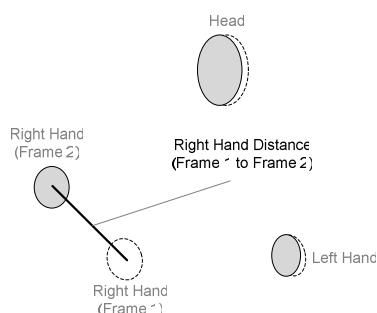


Fig. 22.4. Sample multiple frame feature

and yet difficult for a human coder to recognize. The extracted features represent kinemes that must be fused with other kinemes to form kinemorphs and higher order kinemorphic constructions that are the basis for attributions of meta-messages.

22.3.2 Research Questions

Four main research questions guide this stream of research: (A) How are characteristics (meta-messages) displayed as distal cues in the kinesics modality? (B) How can combinations of kinesics features be combined into clusters of mid-level judgments (proximal percepts)? (C) How can proximal percepts be combined to predict attributions? (D) Is predictive power improved by directly predicting meta-communication attributions from distal indicators? These questions can be seen to represent different linkages in a Brunswikian lens model (see Fig. 22.5). Descriptions of the key relationships are provided below.

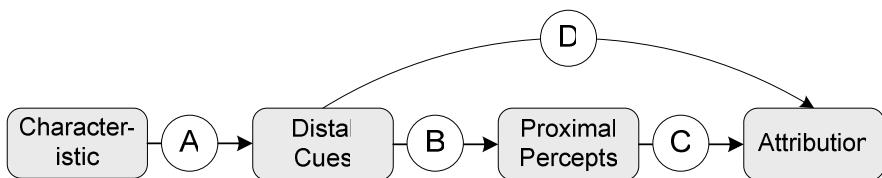


Fig. 22.5. Methods for data prediction in context of Brunswikian lens model

Association coefficients (A) may be productively studied by comparing self reports of sender communication characteristics with observable behaviors. The behaviors may be either manually coded distal cues (e.g. gesture, speech rate, vocal stress, etc.) or automatically detected ones (e.g., average position of the left hand, total distance the head moves, etc.). Results regarding this link in the Brunswikian model are not reported in this chapter; we are currently examining this relationship.

Representation coefficients (B) are calculated by using multiple regression to predict proximal percepts and identify the cues accounting for the most variance in percepts. For example, average hand movement and average head angle can be combined to predict a continuous value on a relaxed-tense perceptual continuum. If the proximal percept is categorical, the representation coefficients could be calculated using other methods of classification such as neural networks and support vector machines.

To identify the utilization coefficients and test their efficacy in predicting meta-messages (C), we employ and compare logistic regression, discriminant analysis, neural networks, and support vector machines to link proximal percepts to attributions. For example, an interviewee who displays proximal percepts of relaxation, moderate dominance, and high involvement might be considered trustworthy.

For (D) we can also utilize linear regression for continuous measures or logistic regression, discriminant analysis, neural networks, and support vector machines for categorical measures to determine truthful or deceptive attributions using the distal cues (video features) as the inputs, rather than the proximal percepts. It is important to note that (D) is not part of the original Brunswikian model. However, the comparison

between (C) and (D) provides insight into the predictive capabilities of automated observation. As with the association coefficients, the relationship between distal cues and attributions is not directly addressed in this chapter but we have addressed this link in other work [22, 19].

In the case of interval-level measures where we use multiple regression and neural networks, we calculate the amount of variance accounted for, precision, and recall. In the case of categorical criterion measures (interviewees' judgment of which meta-communicative style was presented), we use logistic regression, discriminant analysis, and support vector machines to calculate percentages of correct classification in the test data set. We also compare them on their classification performance.

We recognize that we have chosen only a small subset of the methods that could possibly be used. In the past we have successfully used time delay neural networks (TDNN) and recurrent neural networks (RNN) to classify individual gestures and address the hierarchical nature of those gestures within the context of the deceptive discourse [11]. We plan to investigate additional methods (e.g., Bayesian networks, Hidden Markov Models, and Fuzzy Logic models) to determine if they produce superior performance in predicting proximal percepts (e.g., involvement, tenseness) and interpersonal displays (e.g., suspicious, trustworthy).

22.3.3 Description of Mock Theft Dataset

The data used in validating the Brunswikian model came from a mock theft experiment. The purpose of the Mock Theft experiment [10, 11] was to identify deceptive cues that accompany deception. Participants in the experiment were undergraduate students in an introductory communication course. A wallet was placed on a chalkboard in a classroom and some participants "stole" the wallet. Other participants were present during the theft, though they may or may not have been aware when it took place. Both "thieves" and "innocents" were questioned about the theft by trained interviewers. Interviewers were conducted by text chat, audio communication, or face-to-face. The interviews were captured as digitized files.

Only the videotaped face-to-face interactions from the Mock Theft experiment were used in this study. There were a total of 42 possible face-to-face interactions that could be included in the study. Two were not used because the interviews were not manually coded for proximal percepts by any of the trained human coders.

Each interaction was composed of a number of question-answer exchanges. In this study only the theft narrative was included in the analysis. Of the 40 responses, 17 were truthful and 23 were deceptive.

Table 22.2. Reliability levels of the proximal percepts (* $p < 0.05$; ** $p < 0.01$)

	Involvement	Dominance	Tensioness	Arousal
Involvement	.720			
Dominance	.139	.445		
Tensioness	.075	-.325*	.670	
Arousal	.676**	.449**	.015	.656

22.3.4 Measurement

Each theft narrative was coded by human coders for levels of dominance, involvement, arousal, and tenseness. These human coders received extensive training in identifying and interpreting human nonverbal communication. In 10 cases, only one human coder was able to review the theft narrative. In the remaining 30 cases, at least two independent human coders reviewed and coded the theft narrative. The level of reliability (Cronbach's alpha) between the two coders is shown along the diagonal in Table 22.2. The correlations between the human coded percepts are shown in the cells below the diagonal in Table 22.2.

Although only involvement met the conventionally acceptable level of .70 for sufficient reliability, the measurements for arousal and tenseness were also retained because they are near the threshold and Scmitt has argued that even extremely low levels of reliability (e.g., .50) do not justify discarding the measurement [25]. However, the measurement of dominance was exceptionally low. Further, it was highly correlated with tenseness and arousal, indicating that its measurement could be captured in those measures. Therefore, the dominance measure was excluded from the models predicting deception.

22.4 Results

The first analysis involved using automatically generated features to infer human-coded proximal percepts. This analysis evaluated the (B) link in Fig. 22.5 and tested the plausibility of automatically replicating human perceptions of involvement, tenseness, and arousal. Following our research methodology, multiple regression was used to develop models that accurately predicted levels of individual percepts. The mean scores of the human-coded percepts constituted the dependent variables. Single-coder measurements of the percepts were taken when only one coder reviewed an interview. The independent variables used in each regression were a subset of 22 features taken from the 150 automatically generated features described in Sect. 22.3.1. These 22 features track the movements of the head and both hands and serve as candidate distal cues in the Brunswikian model. Summarized descriptions of these features are shown in Table 22.3.

The 22 features were used in three separate multiple regressions, each predicting involvement, tenseness, or arousal. In each multiple regression, a stepwise method of entering variables was used. Although this method may inflate the significance of the overall model, we desired the best predictive model for each percept. We used strict criteria for entering variables into the models for each percept ($p\text{-in} = 0.05$; $p\text{-out} = 0.10$). All models did not violate assumptions of normality and multicollinearity. The variables included in each model, R^2 , and model significance are reported in Table 22.4.

The second analysis involved using predicted proximal percepts and human judgments of proximal percepts to estimate a level of deception (the (C) link in Fig. 22.5). As part of the Mock Theft experiment, the participants were required to indicate their level of honesty during the theft narrative. This self-report was an integer on a scale from 0–10 (0 – completely deceptive; 10 – completely truthful). Although the self-report was a subjective judgment that may have been influenced by lack of memory,

Table 22.3. Automatically generated features used to infer percepts

Body Part Measured	Summarization	Variable Name	Description
Head	Avg	angle	Angle of the major axis
Head, RH, LH	Avg	angle_diff	Difference in angles between previous frame and current frame
Head, RH, LH	Avg, Var	diff	Euclidean distance between x, y position between previous and current frame
Head, RH, LH	Avg	tri_area	Area of triangle formed by connecting right, left hands and head
Head, RH, LH	Avg, Var	distance	Euclidean distance between head, and one hand; Euclidean distance between hands
RH, LH	Avg	Q1	Dichotomous flag when the hand blob is in quadrant 1 in the current frame
RH, LH	Avg	Q2	Dichotomous flag when the hand blob is in quadrant 2 in the current frame
RH, LH	Avg	Q3	Dichotomous flag when the hand blob is in quadrant 3 in the current frame
RH, LH	Avg	Q4	Dichotomous flag when the hand blob is in quadrant 4 in the current frame

Table 22.4. Results from multiple regression models predicting percepts from automatically generated distal cues

	Variables Included	R ²	Standard Error	Model Significance
Involvement	Avg_RH_LH_distance Avg_Q4_LH	.276	1.11	p = 0.003
Tensionness	Var_LH_Head_distance	.121	1.03	p = 0.028
Arousal	Avg_RH_LH_distance Avg_Head_angle_diff Avg_Q1_LH Avg_Q4_RH	.447	1.13	p < 0.001

decreased motivation, and rosy hindsight, it provided a mechanism to balance the amount of truth a deceiver blended in his or her tale along with the deception. While the majority of truth-tellers and deceivers marked either 10 or 0 on the self report, seven participants reported an honesty score between the two extremes. In addition, there were 11 cases where participants who were guilty of the theft reported being

truthful in their narrative of the events surrounding the theft. This interesting contradiction between condition and self-report may be explained by participants providing an accurate description during the theft narrative while omitting important, self-incriminating details.

With all innocent participants and a large percentage of guilty participants reporting high truthfulness in the theft narrative, the honesty self-report was not normally distributed. Therefore, numerous transformations of the honesty score were performed. None of the transformations completely remedied the violation. However, a natural log transformation decreased the violation and was therefore adopted.

First, the human-coded proximal percepts were used to predict levels of honesty. Human-coded values for involvement, tenseness, and arousal were included as independent variables and the transformed, self-report of honesty was the dependent variable.

Table 22.5. Results from a multiple regression model predicting honesty score from human coded percepts

	Variables Included	R ²	Standard Error	Model Significance
Honesty Self-report	Human_coded_involvement Human_coded_arousal	.128	0.94	P = 0.079

Table 22.6. Results from a multiple regression model predicting honesty score from predicted percepts

	Variables Included	R ²	Standard Error	Model Significance
Honesty Self-report	Predicted_tenseness	.096	0.93	p = 0.046

Review of the initial model indicated that involvement and arousal significant predictors of honesty ($p = .033$ and $p = .086$, two-tailed, respectively); tenseness was not ($p = .422$). Therefore, the model was reparameterized to exclude tenseness. The results of the model are shown in Table 22.5. In the reparameterized model, involvement was negatively associated with deception ($p = .026$) and arousal was positively associated with deception ($p = .079$). That is, lower involvement and higher arousal were associated with being deceptive.

The values for involvement, tenseness, and arousal were then used to predict an honesty score. This was accomplished via another multiple regression with predicted values for involvement, tenseness, and arousal as independent variables, and the self-reported honesty level as the dependent variable.

Review of the initial model indicated that tenseness was individually significant in predicting honesty ($p = .045$); however, involvement and arousal were not significant ($p = .882$ and $p = .524$, respectively). Therefore, the model was reparameterized to exclude involvement and arousal. The results of the model are shown in Table 22.6. In the reparameterized model, tenseness was negatively associated with deception ($p = .046$).

22.5 Discussion

We have made initial steps in validating our approach of deception detection via a Brunswikian lens model. Our automatic kinesics analysis is capable of extracting relevant distal cues that can be used to predict perceptual judgments such as involvement ($R^2=.276$), tenseness ($R^2=.121$), and arousal ($R^2=.447$). These findings validate the (B) link in our model (Fig. 22.5).

Additionally, it was shown that the manually coded and also the predicted proximal percepts could be used to determine an attribution; in this case, an individual's level of honesty (manually coded percepts: $R^2=.128$; predicted percepts: $R^2=.096$). These findings validate the (C) link in our model (Fig. 22.5).

Both the human-coded and predicted percepts significantly predict self-reported honesty. Interestingly, the models produced by human-coding and automatic analysis perform at comparable levels; however, they utilize completely different independent variables in the model. This observation invites thoughts of humans and automated tools fulfilling complementary roles in deception detection.

By building tools that can better approximate human perceptions of involvement, tenseness, and arousal (and other perceptions), this research lays a foundation to provide answers to such real-world questions as: what is needed for a machine to interact sensibly with a human? What indicators are the prototypical features necessary to simulate real communication? Are there telltale—and automatically detectable—signals from which a machine can infer a human's current internal state? To the extent that scenes have behavioral routines associated with them (e.g., leaders showing dominance in a meeting), such behavioral recognition can ultimately aid scene recognition. More generally, the multifunctional, multimodal, and molar approach represented here offers a more ecologically valid model of how micro-level behaviors create more general perceptions that drive attributions of meaning.

Care should be taken to consider the limitations of these studies. First, readers should recognize that despite transformations of the honesty score, the variable remained non-normal. Before firm conclusions can be drawn, these results need to be replicated. We are currently addressing this issue.

Second, the amount of variance accounted for (R^2) in the models, though reflecting medium effect sizes or better, is rather small. This suggests that there are other factors that are important to consider when evaluating an honesty level. Additional relevant factors may include motivation to succeed in deception, interviewing style of interviewer, individual characteristics, and cultural factors. These factors were not considered in the analysis.

Third, self-reports of honesty are not indications of guilt or an objective measure of actual deception. Studying relationships between guilt or objectively measured level of deception and the distal cues and proximal percepts would be an interesting extension of this work.

Finally, there is much progress to be made in refining the precision of our automated feature extraction methods. These methods currently suffer from issues such as misclassification of the head and hands resulting from occlusion, hands leaving the

frame, and so forth. Researchers are currently extending our capability in this arena and improved feature extraction should improve our abilities to replicate human perceptions and improve judgments concerning deception.

22.6 Conclusion

The intricate interplay and variety of nonverbal behaviors in conveying meaning has eluded scientists attempting to simulate human communication. Progress has been made at the more rudimentary level of identifying denotative meanings and discrete gestures but truly robust intelligence awaits the ability of systems to identify the complexities and dynamics of actual ongoing interactions among real people and to decipher those multi-layered meta-messages that are essential context for understanding what is said. Our research advances this goal. In the fields of communication and psychology, the research contributes to theories of relational communication (i.e., how people define their interpersonal relationships through nonverbal messages), emotional expressions, and interpersonal deception.

This research offers a new paradigm—using a lens model to specify testable theoretical linkages, employing social science methods to elicit testable stimuli, and applying a rigorous methodology to cross-validate automated tools—for conducting communication research.

This research also provides the prospect of improving our security through the development of automated systems for flagging hostile, deceptive or suspicious communications in organizational, public, and personal discourse, building intelligent systems that can better approximate the ways humans express themselves, and ultimately, achieving a level of recognition and interpretation of subtle behavior patterns that exceeds the ability of human observers.

Acknowledgements

Portions of this research were supported by funding from the U. S. Air Force Office of Scientific Research under the U. S. Department of Defense University Research Initiative (Grant #F49620-01-1-0394) and Department of Homeland Security - Science and Technology Directorate under cooperative agreement NBC2030003. The views, opinions, and/or findings in this report are those of the authors and should not be construed as an official U.S. Government position, policy, or decision.

References

1. Bavelas, J.B.: Situations that lead to disqualification. *Human Communication Research* 9(2), 130–145 (1983)
2. Birdwhistell, R.L.: Background to kinesics. *ETC* 13, 10–18 (1955)
3. Birdwhistell, R.L.: *Kinesics and Context*. University of Pennsylvania Press, Philadelphia (1970)
4. Burgoon, J.K.: Nonverbal signals. In: Knapp, K.L., Miller, G.R. (eds.) *Handbook of interpersonal communication*, Sage, Beverly Hills, CA, pp. 344–390 (1994)

5. Burgoon, J.K., Hale, J.L.: The fundamental topoi of relational messages. *Communication Monographs* 51, 193–214 (1984)
6. Burgoon, J.K., Le Poire, B.A.: Nonverbal cues and interpersonal judgments: Participant and observer perceptions of intimacy, dominance, composure, and formality. *Communication Monographs* 66, 105–124 (1999)
7. Burgoon, J.K., Newton, D.A.: Applying a social meaning model to relational messages of conversational involvement: Comparing participant and observer perspectives. *Southern Communication Journal* 56, 96–113 (1991)
8. Burgoon, J.K., Birk, T., Pfau, M.: Nonverbal behaviors, persuasion, and credibility. *Human Communication Research* 17(1), 140–169 (1990)
9. Burgoon, J.K., Johnson, J.L., Koch, P.T.: The nature and measurement of interpersonal dominance. *Communication Monographs* 65, 308–335 (1998)
10. Burgoon, J.K., Blair, J.P., Qin, T., Nunamaker, J.F.: Detecting Deception Through Linguistic Analysis. In: Chen, H., Miranda, R., Zeng, D.D., Demchak, C.C., Schroeder, J., Madhusudan, T. (eds.) *ISI 2003. LNCS*, vol. 2665, Springer, Heidelberg (2003)
11. Burgoon, J.K., Blair, J.P., Moyer, E.: Effects of Communication Modality on Arousal, Cognitive Complexity, Behavioral Control and Deception Detection During Deceptive Episodes. In: Annual Meeting of the National Communication Association, Miami Beach, Florida (2003)
12. Burgoon, J.K., Adkins, M., Kruse, J., Jensen, M.L., Meservy, T.O., Twitchell, D.P., Deokar, A., Nunamaker, J.F.: An approach for intent identification by building on deception detection. In: *Proceedings of the 38th Annual Hawaii International Conference on System Science (CD/ROM)*. Computer Society Press, Hawaii (2005)
13. Burgoon, J.K., Jensen, M.L., Kruse, J., Meservy, T.O., Nunamaker, J.F.: Deception and intention detection. In: Chen, H., Raghu, T.S., Ramesh, R., Vinze, A., Zeng, D. (eds.) *Handbooks in Information Systems*. Elsevier B.V., Amsterdam (2007)
14. Coker, D.A., Burgoon, J.K.: The nature of conversational involvement and nonverbal encoding patterns. *Human Communication Research* 13(4) (1987)
15. DePaulo, B., Lindsay, J.J., Malone, B.E., Muhlenbruck, L., Charlton, K., Cooper, H.: Cues to deception. *Psychological Bulletin* 129(1), 74–118 (2003)
16. Ekman, P., Friesen, W.V.: Nonverbal leakage and clues to deception. *Psychiatry* 32, 88–106 (1969)
17. Frank, M.G., Feeley, T.H.: To catch a liar: Challenges for research in lie detection training. *Journal of Applied Communication Research* 31(1), 58–75 (2003)
18. Gifford, R.: A lens-mapping framework for understanding the encoding and decoding of interpersonal dispositions. *Journal of Personality and Social Psychology* 66, 398–412 (1994)
19. Jensen, M.L., Meservy, T.O., Kruse, J., Burgoon, J.K., Nunamaker, J.F.: Identification of deceptive behavioral cues extracted from video. In: *International IEEE Conference on Intelligent Transportation Systems*, Vienna, Austria (2005)
20. Levine, T.R., Freeley, T.H., McCornack, S.A., Hughes, M., Harms, C.M.: Testing the Effects of Nonverbal Behavior Training on Accuracy in Deception Detection with the Inclusion of a Bogus Training Control Group. *Western Journal of Communication* 69(3), 203–217 (2005)
21. Lu, S., Tsechpenakis, G., Metaxas, D.N.: Blob analysis of the head and hands: A method for deception detection. In: *Hawaii International Conference on System Science (HICSS 2005)*, Hawaii (2005)

22. Meservy, T.O., Jensen, M.L., Kruse, J., Burgoon, J.K., Nunamaker, J.F.: Automatic extraction of deceptive behavioral cues from video. In: Kantor, P., Muresan, G., Roberts, F., Zeng, D.D., Wang, F.-Y., Chen, H., Merkle, R.C. (eds.) ISI 2005. LNCS, vol. 3495, Springer, Heidelberg (2005)
23. Trager, G.L.: The typology of paralanguage. *Anthropological Linguistics* 3, 17–21 (1961)
24. Scherer, K.R.: Methods of research on vocal communication: Paradigms and parameters. In: KRSaP, E. (ed.) *Handbook of methods in nonverbal behavior research*, pp. 136–198. Cambridge University Press, Cambridge (1982)
25. Schmitt, N.: Uses and abuses of coefficient alpha. *Psychological Assessment* 8(4), 350–353 (1996)
26. Swets, J.A.: Enhancing diagnostic decisions. In: Connolly, T., Arkes, H.R., Hammond, K.R. (eds.) *Judgment and Decision Making: An Interdisciplinary Reader*. Cambridge University Press, Cambridge (2000)
27. Vrij, A.: Detecting lies and deceit: The psychology of lying and implications for professional practice. John Wiley & Sons, Chichester (2000)
28. Vrij, A., Edward, K., Bull, R.: Police officers' ability to detect deceit: The benefit of indirect deception detection measures. *Legal and Criminological Psychology* 6(2), 185–196 (2001)

Suggested Readings

Deception

- Buller DB, Burgoon JK (1996) Interpersonal Deception Theory. *Communication Theory* 6:203-242
An explanation of Interpersonal Deception Theory that views deception as a strategic interaction between deceiver and receiver. During the course of the interaction, the deceiver and receiver may alter strategies as their effectiveness is observed.
- Vrij A (2000) Detecting Lies and Deceit: The Psychology of Lying and the Implications for Professional Practice. West Sussex, John Wiley & Sons Ltd.
An excellent review of the practical aspects of deception detection methods in real-world environments.
- DePaulo B, Lindsay JJ, et al. (2003) Cues To Deception. *Psychological Bulletin* 129(1):74-118.
A comprehensive meta-analysis of deceptive research and indicators of deception.

Blob Analysis used for Deception Detection

- Lu S, Tsechpenakis G, et al. (2005) Blob Analysis of the Head and Hands: A Method for Deception Detection. Hawaii International Conference on System Science (HICSS'05), Hawaii.
An in-depth, mathematical explanation of blob analysis.
- Meservy TO, Jensen ML, et al. (2005) Deception Detection through Automatic, Unobtrusive Analysis of Nonverbal Behavior. IEEE Intelligent Systems (September/October).
An overview of our approach of deception detection and a comparison of various classification methods including discriminant analysis, alternating decision trees, neural networks, and support-vector machines.

Online Resources

- Center for the Management of Information (CMI):
<http://www.cmi.arizona.edu>

A research center at the University of Arizona that conducts numerous innovative experimental, field, and systems research programs including the Automatic Detection of Deception and Intent. CMI and its partners have developed a number of deception corpuses for various environments and contexts.

- Center for BioImaging and Modeling (CBIM):
<http://cbim.rutgers.edu/>

A research center at Rutgers University conducts novel research in the areas of Computational BioMedicine, Computer Vision and Computer Graphics. CBIM developed the head and hand tracking software we use in our method of deception detection.

Questions for Discussions

1. How common is deception? In your daily life? In different cultures?
2. What are the most common deception detection methods? What are the benefits and drawbacks of each of these methods?
3. How would you categorize deception detection methods? (Invasiveness, cost, accuracy, operational expertise needed, etc.)
4. Many deception detection methods rely on low-level, precise cues. Compare and contrast the benefits and challenges of humans directly using these cues to detect deception and humans using perceptions of involvement, dominance, tenseness, arousal to detect deception?
5. What aspects of communication (involvement, dominance, tenseness, arousal) do you believe are most highly correlated with deception? Why? Does the type of deception matter?
6. What is an acceptable level of deception detection accuracy? What are the trade-offs for misclassifying someone as deceptive or truthful?
7. Assuming an acceptable level of deception detection accuracy, discuss environmental and societal challenges in deploying such a system to a field environment (physical environment, security/privacy concerns, etc.).

Subject Index

A

Abstract State Machine, 337, 340, 351, 352
access control, 1, 3, 8, 9, 10, 24, 124, 127, 275, 383
airport security, 340, 341, 342, 344, 351
anomaly detection, 357, 358, 359, 360, 362, 378, 379
anticipatory event, 97, 98, 101, 103, 104, 108, 117
artificial intelligence, 21, 178, 387, 388, 401
ASM ground model, 341, 342, 347, 351
association rule mining, 140, 180, 213, 306, 307
association strength, 46, 55, 56, 57, 145
Assured Information Sharing, 1, 2, 12
authoritativeness, 275, 286, 294, 295, 296, 297
aviation security, 337, 338, 339, 340, 342, 351

B

Bayesian inference, 250, 251, 253, 255–261, 268, 288, 296
Bayesian network, 249–254, 257, 268, 276, 277, 284, 285, 291, 297, 298, 300, 433
betweenness centrality, 47, 49, 51
binary decision tree, 322, 323, 329, 331
Boolean function, 321, 323, 325, 326, 327, 329

bursty feature, 105–107, 110, 112, 114
bursty feature representation, 105, 106, 107, 114

C

CBP agents, 305, 306, 309, 310
centrality measurement, 47, 51, 64
civil aviation, 337, 339, 342
classification, 38, 38, 79, 81–84, 86–90, 92–94, 97–99, 108, 109, 111, 114, 115, 117, 139, 173, 179, 180, 184, 187, 195, 199, 200, 204, 229, 230, 253, 308, 320–322, 331, 358, 360–362, 365, 366, 370, 373, 379, 391, 393, 395, 400, 408, 410, 418, 422, 430, 432, 433, 437
classification accuracy, 90, 92, 109, 361, 370, 371
classification error, 185, 187
classification method, 84, 93, 109, 117, 366
classification model, 79, 86, 108, 111, 358, 360, 379
closeness centrality, 46, 47, 49, 51
clustering coefficient, 67, 70, 71, 75
collaboration level, 276, 278, 286, 287–289, 294, 296, 300
collaborative inference, 275, 276, 286, 293
collaborative level, 286, 290, 292, 293
collaborative user, 275–277, 286, 287, 289, 290, 300
communication fidelity, 275, 294–297, 300
component plane, 235–237
computer vision, 407, 408

conditional probability, 251, 253, 266, 271, 280–282, 284, 289, 308, 311, 328
 container inspection, 319–321, 323, 324, 328, 332, 332
 context entropy, 34, 35
 crime analyst, 137, 140, 141, 143, 144, 146, 147
 crime investigation, 122, 123, 126, 129, 130, 133, 173, 174, 194
 criminal activity, 135–137, 139, 140, 305, 306, 308, 309, 315
 criminal data, 135, 173, 174, 177–179, 195
 criminal justice, 135, 138
 criminal network, 135, 139

D

Dark Web, 65, 66, 69, 75, 76, 79, 82, 217, 218
 data mining, 3, 6, 7, 9, 12, 86, 90, 122, 168, 173, 174, 175, 177, 179, 194, 195, 204, 211, 227, 228, 244, 269, 277, 306, 360, 379
 data sharing, 1–4, 7–10, 13, 135–138, 146, 153
 database schema, 275–279, 300
 database system, 3, 151, 153, 157, 168, 169, 173, 178, 200, 203, 211, 277, 297
 deception detection, 425–429, 437
 decision function, 321, 322, 326, 327, 329
 decision support system, 251
 decision tree, 79, 87, 90–92, 184, 185, 188, 195, 228, 322, 323–326, 328, 329, 331, 332, 364
 degree centrality, 46, 47, 49, 51
 dependency link, 280, 283, 285, 290, 291
 detection rate, 99, 117, 323, 324, 328, 412
 direct friend, 252–256, 263, 265
 dirty data, 151, 154, 169, 170
 distal cue, 427, 429–434, 437

domain expert, 69, 81, 122, 125, 126, 129, 133, 139, 269, 283, 300, 305, 306, 310, 311, 315, 340

domain knowledge, 29, 99, 178, 179, 185, 199–201, 279

dynamic density, 418

E

edit distance, 152, 154–160, 162–164, 166, 169, 170
 event coding, 19, 22
 event detection, 97, 98, 100, 117
 event extraction, 17, 18, 21, 22, 24, 25, 29, 32, 35, 38, 40
 event transition, 97, 98, 100–102, 103, 105, 108, 110, 111, 117
 event transition graph, 98, 100, 101

F

face database, 384, 385, 395, 399
 face image, 383–386, 388–396, 397, 399–403, 406
 facial expression, 384–386, 395, 397, 400, 401, 402, 403, 405, 406, 426
 feature extraction, 200, 363, 386, 397, 430, 437, 438
 feature selection, 87, 105, 119, 238, 239, 308, 367
 feature space, 107, 114, 231, 357, 358, 360, 363, 365–368, 373, 374, 378, 379
 filter refinement, 199, 201, 204, 207
 fisheye view, 52–55, 60, 62, 64
 flow model, 338, 340, 342, 344, 351
 footage area, 407, 409, 410, 418, 422
 forum member, 123–127, 129, 130, 133
 fractal view, 52, 55–57, 59, 60, 62, 64

G

giant component, 65, 70, 71, 73, 75
 Global Salafi Jihad, 45, 46, 59, 60
 goal programming, 330
 Google Acquisition, 110, 111, 115, 116

Gray Web Forum, 121–126, 131, 133

H

Hamburg Cell, 59, 60
hiding friend relationships, 265, 266
homeland security, 135, 156, 166,
 169, 305, 315, 352, 357, 438
honesty score, 428, 430, 435–437
hot thread, 121, 127–130, 133
human coder, 431, 432–434
human expert, 32–34, 82, 89, 110, 290

I

identification, 8, 13, 38, 81, 89, 123,
 125, 134, 139, 203, 228, 305, 315,
 322, 357, 384–386, 393
image database, 384, 385, 388, 394,
 395, 403
incident database, 17, 18, 20, 40
inference accuracy, 255–259, 261,
 268
inference channel, 275–279, 283, 284,
 287–289, 291–293, 295, 298–300
inference detection, 275–277, 297,
 300
inference probability, 275, 284, 286,
 287, 289, 290, 297, 300
inference violation detection, 275,
 276, 285, 286, 290, 297, 300
information extraction, 7, 17, 21, 26
information retrieval, 17, 21, 83, 90,
 177–179, 181, 215, 216
inheritance strength, 255, 257–259,
 262, 263, 264, 269
inspection cost, 322, 324–326, 330,
 332
intelligence analysis, 139, 174, 175,
 177, 199, 216, 219–224
intelligence and security informatics,
 97
intelligence task, 215, 216, 220–224
intelligence work, 215, 221–224
intelligent face recognition system,
 383, 384, 387, 388, 395, 398, 402,
 403

intelligent system, 173, 178, 195, 383,
 387, 392, 393, 399, 402, 403, 438
intelligent systems, 383
interaction boundary, 410, 413, 422
inter-site link, 73–75
investigation clue, 173, 179, 181, 189,
 195

K

key phrase, 129–131, 133, 308
knowledge base, 18, 19, 29, 38, 41
knowledge discovery, 45
knowledge management, 178

L

large database, 151, 153, 154, 164,
 165, 175, 178, 297
law enforcement, 80, 121, 124, 129,
 135, 136, 146, 151–154, 166, 169,
 170, 173–175, 177, 178, 306, 307,
 309, 310, 312, 315
law enforcement agencies, 122,
 135, 139, 153, 156, 199, 200,
 305, 309
law-enforcement agencies, 153
level of corruption, 227, 228, 234,
 237–239, 241, 242, 244
local face recognition, 384, 395, 397,
 402
LS-SVM model, 239–242

M

machine learning, 17, 21, 23, 24, 32,
 90, 97, 208, 219, 220, 228, 360,
 365, 367, 384, 387
Middle-Eastern network, 70–76
model checking, 337–339, 341, 348,
 351
motif expression, 360, 362, 369, 378
moving object, 357, 358, 360, 378,
 379, 407–413, 417, 418, 422
mutation strength, 255, 256, 262, 269,
 270
mutual information, 305–308, 310,
 311, 313, 315

N

- name matching, 151–156, 163, 164, 166, 169
- name searching, 152, 153, 165, 169, 170
- named entities, 29, 32, 36, 99, 102, 104, 105, 109, 112
- natural language, 19, 22, 97, 222, 308, 351
- network structure, 56, 67, 139, 250, 262, 268, 269, 297
- neural network, 178, 228, 383, 387–393, 395, 397

P

- pattern averaging, 383, 384, 388, 389, 391, 393, 395
- Personal Information Management, 215, 218
- PIM system, 216, 219, 222
- pirated CD, 124, 126, 127, 129, 133
- planar homography, 407, 409–413, 415, 416, 422
- police contact, 309–315
- police department, 141, 178, 309, 315, 316
- police record, 135, 136, 138, 142, 147, 169, 312, 314
- port-of-entry inspection, 320–322, 325, 329, 331
- posterior probability, 253–255, 262, 265–267, 272
- posterior probability variation, 265–267
- precision, 19, 22, 32, 38, 39, 60, 101, 114–116, 129–131, 133, 153, 156, 163, 165, 166, 170, 209, 210, 341, 369, 419, 433, 437
- PREFIX algorithm**, 156–158, 160, 162, 164
- prior probability, 255–259, 261–263, 267
- privacy, 2–4, 7–9, 12, 136, 139, 142, 146, 147, 153, 249, 257, 267–269
- privacy protection, 250, 258, 259, 261–268

- private information, 7, 139, 249, 250, 268
- probabilistic model, 337, 338, 341, 342, 348, 351
- projection plane, 410–413, 416–418, 421, 422
- protection rule, 249, 250, 262–264, 266, 267, 269
- proximal percept, 427–434, 436, 437
- public safety, 124, 126, 138

R

- real-time processing, 2, 3, 7, 9
- record-level feedback, 204, 205, 209
- relational database, 7, 178, 277, 279, 282
- relevance feedback, 201, 202, 204, 205
- routine activity, 176, 191

S

- schema link, 280, 281, 283, 284, 290, 291
- scoring model, 204, 205, 208, 209
- screening operation, 342, 344, 345, 347–350
- search engine, 34, 66, 124, 163, 216, 221, 223, 268
- security attribute, 294, 295, 297
- security control, 337, 340, 342–344
- security measure, 337, 339, 340, 342–344, 352
- security policies, 3, 4, 7–10, 12, 13
- security procedure, 338, 340, 342, 344, 351
- self organizing map, 227–229, 234, 240, 244
- semantic inference graph, 275, 278, 283–285, 291, 300
- semantic inference model, 275–280, 290, 291, 300
- semantic knowledge, 275, 277–280, 290, 300
- semantic link, 277, 280, 282–285, 291
- semantic relation, 32, 279, 281, 282, 284

semantic relationship, 269, 277, 279–282, 284
semantic web, 5–7, 12, 13, 19
sensitive data, 10, 136, 275, 276, 289, 290
sensitive information, 4, 11, 139, 254, 275, 276, 278, 285, 294, 299, 300
sensitive node, 284–288
sensitivity analysis, 276, 297, 298, 300, 323, 332
sensor, 5, 200, 217, 319–325, 327–332, 357, 386, 407, 408, 415, 427
sensor reading, 319, 320, 322, 324, 325, 328, 329–331
sentence classifier, 108, 114
shortest path, 47, 49, 51, 56, 57, 60, 67, 68, 70, 139, 144, 146
similarity predicate, 201, 202, 205
site context, 407, 414, 418, 422
situational awareness, 357, 407–409, 418, 422
social network, 13, 45–47, 51–52, 54–56, 59, 60, 62, 64, 67, 136, 139, 142, 146, 249–253, 255, 256, 263, 264, 268
social network analysis, 13, 45, 47, 67, 136, 139, 142, 146, 268
society openness, 255, 256, 258, 261
standard deviation, 113, 153, 166, 371, 373
streaming data, 199–202, 208, 211
structure mining, 66–68
support vector machine, 108, 227, 228, 230, 232, 365, 387, 432, 433

T

terrorist attack, 18, 19, 27, 65, 80, 97
terrorist content, 79, 81, 82, 93

terrorist group, 18, 60, 62, 65, 68, 69, 73, 75, 76, 81
terrorist social network, 45, 51, 52–54, 55, 59, 60, 62, 64
terrorist web site, 65, 66, 68, 69, 73, 75, 76, 79–81, 89, 91
timing constraint, 3, 8–10, 346
topic posting, 129–131
training data, 111, 179, 184, 230, 233, 238, 239, 241, 244, 297, 358, 365–367, 387, 413
transportation system, 319, 408, 413, 417

U

user preference, 101, 103, 108, 110, 111, 308

V

video surveillance, 407–409, 416, 422
violent event, 19, 25, 32, 35, 41

W

web document, 34, 66, 79, 81, 82–87, 89, 91, 93, 297
web page, 12, 25, 65–67, 71, 72, 75, 80, 81, 121, 122, 124, 125
web site, 7, 20, 25, 65, 66, 68, 69, 71, 73, 74–76, 79–82, 89, 91, 92, 94, 122, 124, 131, 297, 341
World Wide Web, 7, 65, 68, 79, 82, 121

X

x-ray machine, 345, 347

Author Index

A

Abadi, M. 269
Abbasi, A. 94
Abbott, R.D. 333
Aberer, K. 300
Abramowicz, W. 42
Abrial, J.R. 352
Adkins, M. 439
Ågerfalk, P.J. 353
Aggarwal, J.K. 422
Agrawal, R. 316, 379
Ahmed, A. 94
Ahuja, N. 404
Albert, R. 76
Alesina, A. 245
Alhojami, E. 246
Alidaee, B. 333
Alison, L. 77
Aljlayl, M. 94
Allan, J. 118
Allen, D. 300
Altenhofen, M. 352
Alvarez, S.A. 317
Anand, S. 333
Antonini, G. 422
Aoe, J. 170
Appelt, D. 41
Armour, T. 225
Arquilla, J. 94
Ashraful, A. 14, 15
Atabakhsh, H. 147, 148, 171, 195,
317
Awad, M. 13, 14
Axford, S.J. 171
Azcarraga, A. 245
Azfar, O. 246

B

Badia, A. 225, 215
Baesens, B. 227, 245

Baeza-Yates, R. 225
Bailey, K.T. 333
Bakiras, S. 380
Ballesteros, L. 94
Bang, S.Y. 404
Barabási, A.L. 76
Barclay, P. 195
Barlow, J.P. 133
Barnett, A. 335
Barnett, V. 380
Bar-Shalom, Y. 422
Battiti, R. 316
Bavelas, J.B. 438
Bederson, B. 226
Belhumeur, P.N. 403
Bellovin, S.M. 265
Ben-Dov, Y. 333
Berners-Lee, T. 14
Berrar, D. 316
Bert, D. 353
Bertino, E. 14, 302, 380
Best, C. 41, 42
Bi, H.H. 195
Bierlaire, M. 422
Bikel, D.M. 118
Bilenko, M. 170
Birdwhistell, R.L. 438
Birk, T. 439
Black, J. 423
Blackler, K. 41, 42
Blair, J.P. 196, 439
Blair, W. 422
Block, C.R. 197
Bock, W. 133
Boghosian, B.A. 423
Bohara, A. 245
Bohlen, M.H. 380
Bolivar, A. 118
Bonati, L.M. 196
Bond, D. 41

Börger, E. 352, 354

Boros, E. 319, 333

Bowen, J.E. 196

Brahan, J.W. 196

Brantingham, P.J. 195, 197

Brantingham, P.L. 195, 197, 352

Brants, T. 118

Breiman, L. 245

Breuer, P.T. 352

Brin, S. 76, 269, 302

Brockett, P.L. 245

Brockhausen, P. 118

Brodie, C. 225

Brooks, C.C. 225

Brown, D.E. 316

Brown, M.H. 64

Buckley, C. 118

Buckley, J. 195

Bull, R. 440

Buller, D.B. 196

Bunke, H. 95

Burges, C.J.C. 245, 380

Burgoon, J.K. 196, 197, 425, 438,
439

Burris, V. 76

Bussler, C. 352

Butterworth, R. 333

C

Cai, Q. 422

Cameron-Jones, R.M. 381

Canter, D. 77

Cao, H. 380

Carbonell, J. 119

Carlson, J.R. 196

Carminati, B. 14

Caronni, G. 301

Carroll, R.J. 333

Carter, C.L. 316

Carthy, J. 118

Celikel, E. 14

Cercone, N. 316

Chabrow, E. 147

Chalupsky, H. 148

Chan, H. 196, 301, 15

Chan, P.K. 302

Chan, P.P. 196

Chang, C.L. 333

Chang, K. 94, 97, 118

Chang, M. 333

Charlton, K. 439

Chau, M. 76, 94, 148, 195, 196

Chaudhuri, S. 211

Chavira, M. 300, 311

Chen, F. 118

Chen, H. 64, 65, 76, 77, 94, 95, 121,
133, 134, 147, 148, 171, 195, 196,
197, 269, 301, 302, 305, 316, 317,
333, 353, 404, 439

Chen, P.S. 173, 196

Chen, Q. 301

Chen, Y. 275, 301

Cheng, Q.S. 404

Chiang, C.P. 77

Chiang, K. 301

Chick, S. 354

Chickering, D.M. 269

Chilton, R.J. 196

Chipman, H.A. 333

Chiu, B. 380

Choi, C.H. 317

Chow, G. 301

Chruch, K.W. 316

Chu, H. 335

Chu, W.W. 249, 269, 275, 301, 302,
303

Chua, K. 118

Chung, W. 196

Clarke, E.M. 352

Clarke, R.V. 195, 196

Claude, S. 316

Coady, W.F. 147

Coffman, T. 148

Cohen, L.E. 196

Cohen, W. 170

Coifman, B. 422

Coker, D.A. 439

Coleman, C. 196

Collins, M. 302

Conly, C. 197

Cooper, H. 439

Coppola, B. 42

Corera, G. 94

Cornelli, F. 301

Cornish, D.B. 196

Cox Jr., L.A. 333

Coyne, E. 14

Criminisi, A. 422

Cunningham, H. 41

Cuttrell, E. 225

D

Daboub, M. 197

Dagan, I. 42

Damiani, E. 301

Dart, P. 171

Darwiche, A. 300, 301

Dash, M. 380

Date, C.J. 301

David, F. 94

David, R. 14

Davis, L.S. 423

Day, W. 211

Dayal, U. 317

De Brabanter, J. 245, 246

De Capitani di Vimercati, S. 301

de Haan, J. 246

De Moor, B. 246

Debat, A. 94

Deboeck, G. 245

Dechter, R. 301

Delugach, H.S. 301, 302

Demchak, C.C. 148, 197, 439

Demers, A. 225

Denis, F. 380

Deokar, A. 439

DePaulo, B. 439, 440

Derrig, R. 245

Despotovic, Z. 300

Deter, J. 64

Dobra, A. 225

Domingos, P. 211, 269

Donath, J.S. 133

Donzeau-Gouge, V. 353

Doreian, P. 269

Dringus, L.P. 133

Drożdżyński, W. 41

Dubitzky, W. 316

Dubois, C. 353

Duffuaa, S.O. 334

Duma, C. 301

Dumais, S. 225

Durham, I. 170

Duric, Z. 423

Durie, B. 403

Dzeroski, S. 302

E

Eades, P. 64

Eastman, C.M. 301, 302

Edelsbrunner, H. 211

Edward, K. 440

Ehrgott, M. 334

Eils, R. 316

Ekman, P. 439

Elison, W. 133

Elisseeff, A. 380

Elkan, C.P. 171

Ellis, T. 167, 133, 423

Elmagarmid, A. 225

Elsayed, E.A. 319, 335

Erdos, P. 76

Erickson, T. 225

Erjavec, T. 41

Erwig, M. 380

Eschenauer, H. 334

Eskin, E. 302

Etzioni, O. 76

Evans, G.W. 353

F

Fagin, R. 212

Faloutsos, C. 77

Fan, W. 302

Fan, X. 404

Fang, H. 317

Fano, R.M. 316

Farahat, A. 118

Farahbod, R. 352

Farkas, C. 301, 302

Farrington, P.A. 353

Faust, K. 77

Fazlollahi, B. 196

Fedzhora, L. 332

Feeley, T.H. 439

Felson, M. 196

Feng, A. 118

Ferrari, E. 14

- Ferrer, J. 335
 Ferrin, D. 354
 Fienberg, S. 170
 Fleming, P.J. 334
 Fleuret, F. 316
 Fonseca, M. 334
 Ford, W. 302
 Forlizzi, L. 380
 Forslund, A.C. 42
 Fox, D. 380
 Frank, M.G. 439
 Franklin, M. 225
 Franz, M. 118
 Freeley, T.H. 439
 Freeman, L.C. 76, 269
 Freeman, W.J. 404
 Fregly, S. 197
 Frieder, O. 94
 Friedman, J.H. 245
 Friedman, N. 269, 301, 302
 Friesen, W.V. 439
 Fu, T. 121
 Fu, Y. 316
 Fuart, F. 42
 Fuchs, W.K. 333
 Fuhr, N. 171
 Fujiyoshi, H. 423
 Furnas, G.W. 64
- G**
- Gaffney, S. 380
 Gandhi, T. 423
 Gandibleux, X. 334
 Garad, A. 404
 Garcia, T. 41
 Garcia-Molina, H. 302
 Garofalakis, M. 225
 Garvey, T.D. 301
 Gatersleben, M.R. 353
 Gavrila, D. 422
 Ge, S. 405
 Geason, S. 196
 Gehrke, J. 148, 225
 Geiger, D. 269
 George, E.I. 333
 George, J.F. 196
- Georgopoulos, M. 151
 Gerring, J. 245
 Gerstenfeld, P.B. 77
 Getoor, L. 269, 301, 302
 Ghafoor, A. 302
 Ghani, R. 95
 Gibbons, P.B. 77
 Gibson, D. 77
 Gifford, R. 439
 Ginsparg, P. 148
 Giordano, P.C. 196
 Glässer, U. 337, 352, 353, 354
 Goldstein, H. 148
 Goldstein, J. 41
 Gonzalez, H. 357
 Gordon, J.S. 196
 Gotzhein, R. 353
 Gowda, H. 317
 Grant, D.R. 77
 Granum, E. 423
 Granzow, M. 316
 Gravano, L. 211
 Greenblatt, S. 148
 Greenfield, R.S. 133
 Greiner, R. 334
 Grumberg, O. 352
 Gunopulos, D. 381
 Guntzer, U. 316
 Gupte, S. 422
 Gustavson, A.T. 77
 Gütting, R.H. 380
 Guyon, I. 380
- H**
- Hagen, S. 316
 Hal Feinstein, H. 14
 Hale, J.L. 439
 Halevy, A. 225
 Haller, A. 352
 Hamilton, H.J. 148, 316
 Han, E.H. 317
 Han, J. 94, 95, 316, 380, 3811
 Hanks, P. 316
 Harding, B. 94
 Haritaoglu, I. 423
 Harms, C.M. 439

Harris, D. 14
 Harris, P.M. 196
 Harwood, D. 423
 Hauck, R.V. 196
 He, J. 266, 302
 He, Q. 118
 He, X. 404
 Heckerman, D. 296, 302
 Hendlar, J. 14
 Hendriawan, D. 76
 Henson, T.K. 196
 Herman, I. 64
 Hermanns, H. 353
 Hershkop, S. 302
 Hespanha, J.P. 403
 Hilderman, R.J. 148, 316
 Hindle, D. 316
 Hinke, T.H. 301, 302
 Hinton, A. 353
 Hipp, J. 316
 Hong, J.H. 316
 Hong, K. 64
 Honkela, T. 245
 Hope, T. 196
 Horby, D. 41
 Horvitz, E. 225
 Hou, X.W. 404
 Hovy, E. 317
 Hsieh, M. 245
 Hu, Y. 404
 Huang, K. 423
 Huang, L.L. 404
 Huang, M. 76
 Huang, Z. 316
 Hughes, M. 439
 Hulten, G. 211
 Hussain, F. 380
 Huttunen, S. 42
 Huysmans, J. 227
 Hwang A.Y. 301

I

Ignat, C. 42
 Imielinski, T. 316
 Ingalls, R.G. 353

J

Jabri, S. 423
 Jackman, R. 246
 Jain, A.K. 404
 Jain, N. 317
 Jajodia, S. 301
 James, A.P. 171
 James, C. 94
 Jamshidi, M. 352
 Jansen, R.W.J. 197
 Jaro, M.A. 170
 Javed, A. 316
 Jensen, C.S. 380, 381
 Jensen, F.V. 302
 Jensen, M.L. 425, 439, 440
 Jin, C. 119
 Jin, H. 118
 Joachimss, T. 118
 Johnson, J.L. 439
 Jokipii, L. 42
 Jones, A. 14
 Jones, D.F. 334
 Jones, G.A. 423
 Jones, R. 42
 Jones, W. 225, 226
 Joyce, W.B. 334
 Jung, S.Y. 316

K

Kacprzyk, J. 94
 Kadane, J.B. 334
 Kak, A.C. 404
 Kalbfleisch, J.D. 334
 Kalnis, P. 380
 Kamber, M. 94, 196
 Kamvar, S.D. 302
 Kandel, A. 79, 94, 95
 Kantarcioglu, M. 14
 Kantardzic, M. 196
 Kantor, P.B. 333
 Karalic, A. 245
 Karat, C.M. 2285
 Karat, J. 225
 Karger, D. 225
 Karypis, G. 94
 Kaski, S. 245

- Kautz, H. 269, 302, 380
Kaza, S. 148, 305, 317, 318
Keffe, J.O. 302
Kek, A. 118
Kennedy, J.M. 171
Keogh, E. 380, 381
Khan, L. 13, 14
Khashman, A. 383, 404
Khokhar, A. 316
Ki, M. 404
Kim, D. 404
Kim, J. 14
Kim, S. 357
Kim, T.S. 316
Kim, W. 171
Kimler, M. 42
Kinney, B. 352
Kleinberg, J. 77, 118, 148
Klein-Seetharaman, J. 317
Klerks, P. 148
Kloos, C.D. 352
Klosgen, W. 245
Knack, S. 246
Knapp, K.L. 438
Knorr, E. 380
Koch, P.T. 439
Kohavi, R. 245
Kohonen, T. 245
Koike, H. 64
Koller, D. 269, 301, 302
Kong, G. 303
Koperski, K. 380
Koski, J. 334
Kostov, V. 380
Kowalski, R. 334
Kraiem, N. 353
Krieger, H.-U. 41
Kriegman, D.J. 403, 404
Krizan, L. 225
Krohn, M.D. 196
Kruse, J. 439, 440
Kudoh, T. 380
Kuehner, W. 333
Kumar, S.R. 77
Kumaran, G. 118
Kuramochi, M. 94
Kursun, O. 151
Kushner, H. 334
Kwak, N. 317
Kwiatkowska, M. 353, 354
- L**
- Laguna, M. 333
Lagus, K. 245
Lai, G. 77, 95, 134
Laleau, R. 353
Lam, K.P. 196
Lamb, D. 170
Lambert, D. 212
Lambsdorff, J. 245
Lan, K.K. 333
Larkey, L.S. 94
Larson, C. 147, 301
Laskey, K.B. 302
Lassila, O. 14
Last, M. 94, 95
Lauritzen, S.L. 302
Lavee, G. 14
Lavrac, N. 302
Lavrenko, V. 118
Lawless, J.F. 335
Layfield, R. 14
Le Poire, B.A. 439
Ledru, Y. 353
Lee, W. 302, 317
Lee, Y. 246
Leenders, R.T. 269
Leite, C. 245
Lemcke, J. 352
Lemoine, M. 353
Leung, W. 196
Leung, Y.W. 334
Leutenegger, S.T. 381
Levenshtein, V.L. 171
Levin, A. 404
Levine, T.R. 439
Levitt, K.N. 303
Lewis, T. 380
Li, C. 212
Li, G. 404
Li, J. 318
Li, M.J. 118

- Li, N. 302
 Li, S.Z. 404
 Li, W. 318
 Li, X. 357
 Li, Z.W. 118
 Liao, L. 380
 Lichtblau, E. 94
 Lim, E.P. 97, 118, 148, 318
 Lin, C. 148
 Lin, H.M. 121
 Lin, S.D. 148
 Lin, W. 317
 Lindsay, J.J. 439
 Ling, C. 212
 Lipton, A.J. 423
 Lipton, E. 94
 Liu, H. 380
 Liu, L. 14, 303
 Liu, N. 64
 Liu, Z. 269, 302
 Livny, M. 381
 Lo, B. 423
 Loader, B. 77
 Lonardi, S. 380
 Lopez, M.A. 381
 Lorentzos, N.A. 380
 Lotem, A. 212
 Lowd, D. 269
 Lu, K. 404
 Lu, S. 439
 Lu, X. 317, 404
 Lunt, T.F. 301
 Lyall, S. 94
- M**
- Ma, W.Y. 118
 Ma, Y. 199
 MacQueen, J.B. 270
 Madhusudan, T. 148, 197, 439
 Madigan, D. 333, 334
 Magerman, D.M. 317
 Mahadev, N.V.R. 334
 Maier, D. 225
 Makkonen, J. 118
 Makris, D. 423
- Malone, B.E. 439
 Mamdani, A. 302
 Mammone, R. 333
 Mamoulis, N. 380
 Manojit, S. 64
 Marcella, A.J. 133
 Marcus, M.P. 317
 Marcus, S. 148
 Markov, A. 79, 94
 Marshall, B. 135, 148, 317
 Marshall, M.S. 64
 Marti, S. 302
 Martin, R.F.K. 422
 Martinez, A.M. 404
 Masand, B. 212
 Masoud, O. 422
 Massey, J.L. 96
 Masud, M. 14
 Matusov, J.B. 334
 Maurer, J. 14
 Mauro, P. 245
 Maxwell, T. 171
 Maynard, D. 41
 McAndrew, D. 77
 McCaghy, C.H. 196
 McCallum, A. 42, 212
 McCarley, J.S. 118
 McCornack, S.A. 439
 McCulloch, R.E. 333
 McGill, M. 95
 McGloin, J.M. 197
 McIllwain, J.S. 77
 McKenna, S.J. 423
 Meehan, J.R. 64
 Mehrotra, S. 199, 269, 301, 302,
 317, 333, 353, 404
 Meier, R.F. 197
 Melancon, G. 64
 Melbin, M. 133
 Meltzer, B. 334
 Mena, J. 133, 225
 Meng, F. 303
 Merkl, D. 246
 Merritt, M. 269
 Merz, C.J. 212
 Meservy, T.O. 425, 439, 440
 Miethe, T.D. 197

Miglio, R. 334
 Mihov, S. 171
 Milgram, S. 77, 270
 Miller, G.R. 438
 Miller, M. 302
 Minock, M. 301
 Miranda, R. 148, 197, 439
 Mitchell, N. 245
 Mitchell, T.M. 95
 Mitchie, D. 334
 Mittal, S. 334
 Mitten, L.G. 334
 Mittendorff, C. 245
 Mobasher, B. 317
 Moeslund, T.B. 423
 Monge, A.E. 171
 Montinola, G. 246
 Mooney, R. 170
 Morik, K. 118
 Morimoto, K. 170
 Morrice, D.J. 354
 Morris, B. 423
 Moyer, E. 196, 439
 Moynihan, J. 196
 Mhlenbruck, L. 439
 Müller, W. 352
 Murase, H. 404
 Murphy, P. 212

N

Nakhaeizadeh, G. 316
 Nallapati, R. 118
 Namkung, J. 404
 Naor, M. 212
 Nardelli, E. 380
 Nascimento, M.A. 381
 Natarajan, K.S. 334
 Navarro, G. 171
 Nayar, S.K. 404
 Needham, R. 269
 Nembhard, H.B. 353
 Neto, R. 212
 Neuhold, E. 317
 Newcombe, H.B. 171
 Newman, M.E. 77, 270
 Newton, D.A. 439

Ng, R. 380
 Ng, T.D. 64
 Ng, W.K. 318
 Nguyen, T. 335
 Niewiadomski, A. 94
 Nigam, K. 42
 Nilsson, N.J. 334
 Niyogi, P. 404
 Norman, G. 353, 354
 Numrych, K. 225
 Nunamaker, J.F. 425, 439, 440

O

Oellinger, T. 42
 Ogihara, M. 318
 Ogrodnik, L. 197
 Olesen, K.G. 302
 Oliver, N.M. 423
 Olivier, M.S. 14
 Olsen, R.A. 245
 Omiecinski, E.R. 317
 Ong, T. 317
 Ong, W.S. 118
 Osyczka, A. 334
 Oyama, T. 354
 Ozawa, J. 380

P

Padmanabhan, B. 148
 Page, L. 76, 269, 302
 Paiement, R. 197
 Paik, J.K. 404
 Pakut, A. 334
 Palmer, C.R. 77
 Palsberg, J. 353
 Pan, S. 245
 Panangaden, P. 354
 Pang, S. 404
 Pantel, P. 317
 Papanikopoulos, N.P. 422
 Paraboschi, S. 301
 Park, C. 404
 Park, K. 170
 Park, S. 407, 423
 Parker, D. 353, 354

- Parthasarathy, S. 318
 Pathak, S. 333
 Patil, R.S. 423
 Paul, R. 14
 Pazzani, M. 380
 Pearl, J. 302
 Pedersen, J.O. 119
 Pekce, A. 334
 Peled, D.A. 352
 Pendergraft, D.R. 353
 Peng, F. 118
 Pentland, A.P. 404, 423
 Peters, B.A. 353
 Petersen, T. 147, 148, 317
 Peureux, F. 353
 Pfau, M. 439
 Pfeffer, A. 269, 301, 302
 Pfeifer, U. 171
 Pfoser, D. 380
 Philpot, A. 317
 Piatetsky-Shapiro, G. 212
 Pierce, T. 119
 Pinheiro, J.C. 212
 Piskorski, J. 17, 41, 42
 Pliant, L. 197
 Ploetner, J. 423
 Poersch, T. 171
 Pohl, I. 334
 Pollock, S.M. 335
 Poole, D. 270, 303
 Popp, R. 225
 Porter, M. 95
 Potchak, M.C. 197
 Pouliquen, B. 17, 42
 Prabhakar, S. 148
 Prentice, R.L. 334
 Prinz, A. 353
 Psaila, G. 379
 Pu, X. 405
- Q**
- Qin, J. 77, 95, 134
 Qin, T. 439
 Qiu, Y. 333
 Quain, X. 301
 Quinlan, J.R. 95, 245, 381

- R**
- Raghavan, P. 77
 Raghu, T.S. 439
 Rajagopalan, S. 77
 Rajasekaran, S. 317
 Ralyté, J. 353
 Ramakrishnan, R. 381
 Ramesh, R. 439
 Raouf, A. 334
 Rastkar, S. 337, 353
 Rastogi, R. 225
 Rauber, A. 246
 Rauramo, A. 42
 Ravikumar, P. 170
 Reid, E. 77, 95, 134
 Reid, I. 422
 Remagnino, P. 423
 Renyi, A. 76
 Reuter, A. 317
 Reynolds, K. 151
 Rhodes, W.M. 197
 Ribeiro-Neto, B. 225
 Rice, K.J. 197
 Richardson, J. 148
 Richardson, M. 269
 Riedewald, M. 225
 Riloff, E. 42
 Ripplinger, B. 95
 Roberts, F.S. 333, 334, 335
 Robertson, C.V. 353
 Rocchio, J. 212
 Romney, K. 269
 Ronfeldt, D. 77
 Rosario, B. 423
 Rossetti, M.D. 353
 Rossmo, D.K. 197
 Rothkopf, M.H. 335
 Roweis, S.T. 404
 Roy, N. 212
 Ruiz, C. 317
 Ruppert, D. 333
 Rusinkiewicz, M. 225
 Rutten, J. 354
- S**
- Saeger, K. 333, 335
 Sageman, M. 64, 77, 95, 226

- Sahar, S. 148
 Saltenis, S. 381
 Salton, G. 95, 118, 212
 Samarati, P. 301
 Sanchez, P.J. 354
 Sanders, W.B. 197
 Sandhu, R. 14
 Sarin, R. 225
 Sarkar, M. 64
 Saul, L.K. 404
 Saxe, J. 170
 Schäfer, U. 41
 Schenker, A. 95
 Schepses, E. 197
 Scherer, K.R. 404
 Schlosser, M.T. 302
 Schmid, J. 354
 Schmitt, N. 440
 Schneider, M. 380, 381
 Schölkopf, B. 246
 Schrodte, P. 42
 Schroeder, J. 148, 195, 197, 439
 Schroepfer, C. 335
 Schulz, K.U. 171
 Schwartz, R. 118
 Scniederjans, M.J. 335
 Sebastiani, F. 95
 Segala, R. 353
 Seid, D.Y. 199
 Seldadyo, H. 246
 Selman, B. 269, 302
 Senator, T. 225
 Setiono, R. 245
 Shafer, J.C. 319
 Shafiq, B. 302
 Shah, M. 269, 302
 Shahmehri, N. 301
 Shashua, A. 404
 Shelton, C. 381
 Sherkat, D.E. 77
 Sheth, A. 225
 Shi, W. 333
 Shi, X.D. 119
 Shihab, A.I. 423
 Shimizu, A. 404
 Shishibori, M. 170
 Shrader, S. 353
 Silberschatz, A. 148
 Silva, J.R.O. 381
 Singh, K. 352
 Slagle, J.R. 333
 Slattery, S. 95
 Smith, E. 76
 Smith, J.S. 353
 Smith, W.R. 197
 Smola, A. 246
 Smyth, P. 148, 380
 Soffritti, G. 334
 Son, S. 14
 Song, M. 317
 Sparrow, M.K. 77, 148
 Spiegelhalter, D.J. 302
 Spiegelman, C.H. 333
 Spitzner, L. 14
 Sproston, J. 353
 Srikanth, R. 316, 317
 Srivastava, J. 148, 317
 Stärk, R. 352, 354
 Statnikov, R.S. 335
 Stefanski, L.A. 333
 Steinberger, R. 17, 42
 Stepanova, M. 245
 Stickel, M. 301
 Stokes, N. 118
 Stolfo, S.J. 302, 317
 Stone, C.J. 245
 Stonebraker, M. 317
 Strahm, A. 76
 Strogatz, S.H. 77, 270
 Stroud, P. 333, 335
 Su, H. 318
 Subbiah, G. 13
 Sultan, A.M. 335
 Sun, J. 423
 Suykens, J. 245, 246
 Swami, A. 316
 Swamy, A. 246
 Sweeney, L. 270
 Swets, J.A. 440
 Szczepaniak, P.S. 94
 Szpektor, I. 42
- T**
- Tablan, V. 41
 Taipale, K.A. 171

- Takakuwa, S. 354
 Talbot, D. 95
 Tamiz, M. 334, 335
 Tan, C.L. 380
 Tanev, H. 17, 42
 Tao, T. 317
 Taskar, B. 301
 Teevan, J. 225, 226
 Temnikova, I. 42
 Templeman, A.B. 335
 Thacker, S. 245
 Theodoridis, Y. 380, 381
 Thuraisingham, B. 1, 14, 15, 148,
 226, 269, 301, 302, 317, 333, 353,
 404
 Thuraisingham, B.M. 302
 Thuraisingham, N. 14
 Toland, T.S. 301, 302
 Tomkins, A. 77
 Trager, G.L. 440
 Treisman, D. 246
 Trivedi, M.M. 407, 423
 Tsechpenakis, G. 439
 Tsfati, Y. 133
 Tsoukatos, I. 381
 Tsyblikov, N. 15
 Turk, M. 404
 Tuzhilin, A. 148
 Twitchell, D.P. 197, 439
- U**
 Ünlüyurt, T. 333, 335
- V**
 Vajihollahi, M. 337, 352, 353
 Valera, M. 423
 van Breugel, F. 354
 van der Goot, E. 41
 van der Weij, S.W. 353
 Van Gestel, T. 245, 246
 van Koppen, P.J. 197
 van Rijsbergen, C.J. 171
 Vandewalle, J. 246
 Vanthienen, J. 227, 245
 Vapnik, V.N. 246, 381
 Vatis, M. 14
 Vaz, E.W. 196
- Vazirgiannis, M. 380
 Velarde, J.L.G. 333
 Velastin, S.A. 423
 Verma, B. 404
 Vesanto, J. 246
 Viaene, S. 245
 Vicencio-Silva, M.A. 423
 Vignes, S. 353
 Vinze, A. 439
 Violette, C. 147, 148, 317
 Vrij, A. 440
 Vu, Q. 317
- W**
 Wade, C. 118
 Waltz, E. 226
 Wang, B. 118
 Wang, F.-Y. 77, 94, 95, 148, 269,
 301, 302, 317, 333, 353, 404, 440,
 Wang, G. 148, 171, 196
 Wang, J.H. 64, 121, 134
 Wang, T. 317
 Wang, Y. 317, 334, 404
 Ward, T. 118
 Wasserman, S. 77
 Watts, D.J. 77, 270
 Watts, G.S. 316
 Weaver, E. 316
 Wechsler, H. 423
 Weder, B. 245
 Wei, C. 64
 Wei, L. 381
 Weidmann, J. 245
 Weimann, G. 42, 77, 95, 133, 134
 Weischedel, R.M. 118
 Weisser, D. 317
 Wellman, M.P. 302
 Whine, M. 77
 Whinn-Yates, T. 195
 White, C.H. 197
 White, D.R. 269
 White, S. 148
 Widiger, A. 42
 Wilcox, J. 171, 197
 William, O. 94
 Wilson, D.L. 354
 Wilson, P. 196
 Wimmers, E.L. 379

- Winkler, W.E. 171
Winsborough, W. 302
Winslett, M. 303
Wolf, R. 302
Wolfson, O. 380
Wong, A. 95
Wooley, D.R. 134
Woon, Y.K. 318
Wren, J.D. 318
- X**
- Xi, X. 381
Xia, X. 245
Xie, M. 319
Xiong, L. 303
Xu, F. 41
Xu, J. 65, 77, 148, 195, 196, 317
- Y**
- Yan, S. 404
Yan, X. 95
Yang, C.C. 45, 64, 94, 119
Yang, C.S. 95
Yang, H. 301
Yang, M.H. 404
Yang, Y. 95, 119
Yangarber, R. 42
Yates, R.B. 212
Yi, G.Y. 335
Yin, X. 381
Yip, R.W. 303
Yoshihara, H. 64
Yoshioka, M. 380
Youman, C. 14
- Yu, T. 303
Yu, Y. 335
- Z**
- Zaghouani, W. 42
Zait, M. 379
Zaki, M.J. 318
Zeng, D. 76, 94, 439
Zeng, D.D. 77, 94, 95, 148, 197,
269, 301, 302, 317, 333, 353, 404,
439, 440
Zgoba, K.M. 197
Zhai, C.X. 317
Zhang, B. 405
Zhang, G. 303
Zhang, H. 335, 405
Zhang, H.J. 404
Zhang, J. 119, 195, 302, 404
Zhang, N.L. 270, 303
Zhang, T. 381
Zhao, J. 404
Zhao, J.L. 148
Zheng, Z. 405
Zhou, L. 197, 246
Zhou, W. 405
Zhou, Y. 77, 95, 134
Zhu, J. 17
Zhu, W. 118
Zisserman, A. 422
Zizka, J. 42
Zobel, J. 171
Zytkow, J. 245