NWI-IMC042 Natural Computing

General information

Teachers:

Prof. Elena Marchiori (coordinator)
Room: M1.02.05A
Email: elenam@cs.ru.nl

Ir. Gijs van Tulder
Email: gijs.vantulder@ru.nl

Dr. Johannes Textor
Email: johannes.textor@radboudumc.nl

Teaching assistants:

Ankur Ankan (assistant)
Email: Ankur.Ankan@radboudumc.nl

Inge Wortel (assistant)
Email: Inge.Wortel@radboudumc.nl

Credits

6 EC

Place in the curriculum

The course is an elective course in the master Computing and Information Sciences, in particular the Data Science master specialization, and in the Master Artificial Intelligence. This is a project-oriented course; therefore students are expected to have basic programming skills. The course assumes knowledge and skills on a bachelor level, and the bachelor course Data Mining as pre-requisite.

Contents / Description

Natural computing concerns computational techniques inspired by nature, such as processes of evolution and natural phenomena of social interaction among organisms. The aim of natural computing is to develop tools either for solving complex optimization problems or for studying emerging properties of complex systems through simulations. Natural computing can be divided into three main branches:

- Computing inspired by nature: nature as source of inspiration for the design of algorithms.
- Simulation of nature: a synthetic process aimed at creating emerging patterns, forms, behaviours, and organisms through evolution.
- Computing with natural materials: the use of natural materials to perform computation as substitute or supplement to classical computers.

This course deals with selected topics from the first two branches: computing inspired by nature, simulation of nature.

Goals

The main goals of this course are to introduce students to various algorithms in the area of Natural Computing and show how they have been applied to various kinds of problems, notably in data science. On completion of the course students should be able to:

- Understand various natural computing algorithms.
- Design an experiment using natural computing algorithms.
- Develop, implement and assess methods based on natural computing.
- Report on project progress.
- Write a research report.

Organization

The lectures will be used to give an overview on some topics in natural computing and to jointly zoom in on advanced aspects. There will be compulsory home exercises to be done by each team in order to get familiar with the topics of the course. The rest of the course will be dedicated to a team project, with meetings between each team and the teacher(s) for discussion and feedback.

Students will work in teams of 3 students.

Each team can choose a research or applied problem.

Project: key components

Proposal: 1/2 page document setting the basis of the project. The project proposal should be structured as follows.

- State your hypothesis/problem.
- Give a rough overview of the methods you will use.
- Tell why such methods will allow you to tackle your problem in a proper way.
- Include at least 3 recent references from the literature and briefly describe them.

Final Report: max 10 pages document. The report should include:

- Specific goals of the work.
- Review of related work.
- Description of your testing procedures.
- Description of the algorithms used. Ensure that the work is reproducible given your description.
- Results. Ensure to perform a correct experimental analysis (e.g. repeated experiments, statistical analysis of results).
- Conclusions. Is your line of work worth pursuing? What additional enhancements could be made?

Requirement: The bibliography should contain at least 5 references.

Implementation: Implementation of algorithms developed and used during the project.

- Source code, together with all data necessary to run the program and a short manual describing how to use/run the program.
- A sample run, screenshot, or other indication of system behavior.

Grading

Grading will be based on the home exercises (40%) and project (60%).

Schedule

Weekly planning and deadlines are available in Brightspace.

Important: get started early enough to discover and overcome issues related to your project.

Suggested study material and software

The following pointers cover material we'll discuss during the lectures and more information useful for your project and exercises. Part of the below material offers more in-depth reading and background on the topics covered in the course.

Evolutionary Computing:

- An introduction to genetic algorithms and evolution strategies. M Dianati, I Song, M Treiber. Technical report, University of Waterloo, Ontario, N2L 3G1, Canada. https://pdfs.semanticscholar.org/79ea/bba1c148c7ac3c33e50895bec4d41a5fed2b.pdf
- Wikipedia https://en.wikipedia.org/wiki/Evolutionary_computation
- Keith Downing video on EC https://www.youtube.com/watch?v=D3zUmfDd79s
- Evolutionary self-adaptation: a survey of operators and strategy parameters. Oliver Kramer. Evolutionary Intelligence, August 2010, Volume 3, Issue 2, pp 51-65.
- Evolutionary Strategies. https://hal.inria.fr/hal-01155533/document
- Darrell Whitley, An overview of evolutionary algorithms: practical issues and common pitfalls, Information and Software Technology, Volume 43, Issue 14, 15 December 2001, Pages 817-831, ISSN 0950-5849.
- http://www.genetic-programming.org/

Implementations of GA' s and GP frameworks:

- Software for distributed EA in Python: DEAP https://en.wikipedia.org/wiki/DEAP_%28software%29
- Genetic algorithms in Matlab: http://nl.mathworks.com/discovery/genetic-algorithm.html
- Sferes2 project is a "framework for evolutionary computation": https://github.com/sferes2/sferes2
- Software in C++ : Galib : http://lancet.mit.edu/ga/,
- EO: http://eodev.sourceforge.net/
- ECJ - Evolutionary Computation/Genetic Programming research system (Java)
- GPE - Framework for conducting experiments in Genetic Programming (.NET)
- PyGP - Python Genetic Programming Project (Python)
- DGPF - simple Genetic Programming research system (Java)
- PMDGP - object oriented framework for solving genetic programming problems (C++)

- JGAP - Java Genetic Algorithms and Genetic Programming, an open-source framework (Java)

Cellular Automata

Background reading on Cellular automata:

- "Cellular Automata: A Selected Review" by Melanie Mitchell, https://melaniemitchell.me/PapersContent/ca-review.pdf
- Steven Wolfram, "A new kind of Science", Wolfram Media, 2002

Complexity and Pattern Formation:

- "The Chemical Basis of Morphogenesis" by Alan Turing, http://www.dna.caltech.edu/courses/cs191/paperscs191/turing.pdf
- "Studying Artificial Life with Cellular Automata" by Christopher G Langton, https://doi.org/10.1016/0167-2789(86)90237-X
- "Simulation of biological cell sorting using a two-dimensional extended Potts model" by Graner and Glazier, https://link.aps.org/doi/10.1103/PhysRevLett.69.2013
- "Crawling and Gliding: A Computational Model for Shape-Driven Cell Migration" by Niculescu, Textor, and de Boer, https://doi.org/10.1371/journal.pcbi.1004280

Interactive implementations of Cellular Automata models:

- https://bitstorm.org/gameoflife/ -- the Game of Life
- http://www.langtonant.com – Langton's Ant
- https://mrob.com/pub/comp/xmorphia/ogl/index.html -- the Gray-Scott model of pattern formation in reaction/diffusion systems
- https://github.com/jtextor/cpmjs -- a JavaScript library containing multiple interactive examples of cellular Automata and especially the cellular Potts model

Ensemble Learning

- Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). The Elements of Statistical Learning (2nd ed.). Springer. Chapter 10, 15
- Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). Foundations of Machine Learning. The MIT Press. Chapter 6
- Kuncheva, L. I. (2004). Combining Pattern Classifiers. Methods and Algorithms. Wiley, Chichester.
- Ho, T. K. (1998). The random subspace method for constructing decision forests. IEEE transactions on pattern analysis and machine intelligence, 20(8), 832-844.
- Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
- Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems. Journal of Machine Learning Research, 15(1), 3133-3181.
- https://www.microsoft.com/en-us/research/publication/real-time-human-pose-recognition-in-parts-from-a-single-depth-image/ Paper describing how Microsoft used random forests for the Kinect system

The following implementations may be helpful during the exercises:

- http://scikit-learn.org/stable/modules/ensemble.html The ensemble learning section of the sk-learn library covering both bagging and boosting.
- https://cran.r-project.org/web/packages/randomForest/index.html The standard R implementation based on the original code by Breiman and Cutler.
- (https://github.com/dmlc/xgboost The popular xgboost gradient boosting implementation)

Swarm Intelligence

ACO

- Dorigo M, Di Caro G, Gambardella LM. "Ant algorithms for discrete optimization", Artif Life. 1999 Spring;5(2):137-72.
- M. Dorigo, M. Birattari, T. Stützle, "Ant Colony Optimization – Artificial Ants as a Computational Intelligence Technique", IEEE Computational Intelligence Magazine, 2006.
- M. Dorigo K. Socha, "An Introduction to Ant Colony Optimization", T. F. Gonzalez, Approximation Algorithms and Metaheuristics, CRC Press, 2007.
- M. Dorigo T. Stützle, "The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances", Handbook of Metaheuristics, 2002.
- http://www.math.uwaterloo.ca/tsp/ (software, benchmarks data and other resources for the TSP).

PSO

- PSO applied to computer games http://link.springer.com/chapter/10.1007/978-81-322-2220-0_21#page-1
- PSO applied to full model selection in machine learning http://www.jmlr.org/papers/v10/escalante09a.html
- A tutorial and other resources on PSO http://www.swarmintelligence.org/tutorials.php
- See http://en.wikipedia.org/wiki/Particle_swarm_optimization for more references and information on PSO.

Artificial Immune Systems

- http://www.artificial-immune-systems.org (Somewhat outdated) website summarizing many AIS algorithms.
- http://johannes-textor.name/negativeselection.html contains an implementation of the negative selection algorithm.
- https://en.wikipedia.org/wiki/Artificial_immune_system Wikipedia AIS article.
- http://www.cs.unm.edu/~immsec/data-sets.htm Stephanie Forrest's older page on applying negative selection in computer security.
- https://msdn.microsoft.com/en-us/magazine/jj883960.aspx Microsoft's toy implementation of an AIS (in visual basic)
- "Artificial Immune Systems: A New Computational Intelligence Approach" by Leandro de Castro and Jonathan Timmis, Springer, 20022
- "Computer Immunology" by Catherine Beauchemin and Stephanie Forrest, Immunological Reviews 216/2007, available at http://cs.unm.edu/~mfricke/CS523_2017spring/Readings/Forrest2007.pd