Jeremy's University Courses

Jeremy Barisch-Rooney

April 4, 2019

Contents

1 Department of Computational Science

1.1 Introduction to Computational Science (UvA)

1.1.1 Objectives

The aim of this course is to provide an overview of Computational Science and modelling techniques available to the Computational Scientist. We will do this by considering one specific application (spreading of infectious diseases) and formulating different types of models and simulations, thus not only exploring the rich set of modelling paradigms available to us, but also diving deep in the specific application field. You will also be exposed to recent research results from the Computational Science Lab, to see other examples of Computational Science in action.

1.1.2 Contents

Modelling and Simulation, spreading of infectious disease, SIR model and variants, ODE, PDE, CAs, Networks, ABM. Acquired knowledge will be applied in in a number of lab assignments, covering the whole chain from model description and simulation, model verification and validation, to data gathering and analysis .

1.2 Numerical Algorithms (UvA)

1.2.1 Objectives

To get an overview of basic Numerical Algorithms.

1.2.2 Contents

The course gives a broad overview of basic numerical methods for solving Linear Systems, Least Squares problems, Eigenvalue problems, Nonlinear equations, Optimization problems, Interpolation and Quadrature problems, and Ordinary Differential Equations. These methods form the basis for many numerical algorithms used in computational science and engineering.

1.3 Agent-based Modelling (UvA)

1.3.1 Objectives

At the end of the course, the student will be able to

- Define what Agent-Based models (ABMs) are.
- Compare different methods for developing ABMs
- Properly analyse the output of an ABM
- Summarize and Compare existing ABM
- Develop your own Agent-based Model
- Properly report on your ABM (Scientific paper)

1.3.2 Contents

The course provides a broad introduction to the field of ABM. It includes a series of 6 2 hour lectures covering the following topics:

- Introduction and Classic Models (Epstein, Schelling, Axtell)
- Agents and AI, a view of agency and agents as defined by the field of Artificial Intelligence
- Game Theory & Agents, covering basic game theory and evolutionary game theory (Iterated & Evolutionary Prisoners Dilemma).

- Discrete Choice Theory for ABM - Logit, Probit Models and more.
- Sensitivity Analysis Methods for ABM - OFAT, Regression methods and Sobol.
- Validation for ABM (covering methodologies and challenges in validating ABM).

The course also includes practical lectures using (MESA & Python), these practical lectures include exercises and questions required each week.

1.4 Complex System Simulation (UvA)

1.4.1 Objectives

The student will be able to:

- Explain the concept of emergence;
- Name and reason about different types of emergent phenomena, such as chaos, phase transitions, network connectivity, and complexity;
- Name and reason about different types of computational models used to study these phenomena;
- Implement these models to reproduce a given emergent phenomenon;
- Compare model outcomes with predictions from (mean-field) theory;
- Interpret and use the model outcomes in terms of practical applications;
- Implement and study interventions and what-if scenarios to improve/optimize with respect to a practical application.

1.4.2 Contents

The course has two major components, the first two weeks will include lectures covering concepts and methods related to Complex System Simulation. The second two weeks will be group projects (2-3 students), where each group will develop a simulation of a complex system and conduct research using that simulation.

The group project should reflect the concepts and methods covered in the first two weeks, however students are also encouraged to self study and suggest alternative ideas.

Topics to be covered (may change):

- Introduction to Cellular Automata (1D/2D CA, rule codes, phenomenological studies, behaviour classes)
- Self-Organized Criticality (Sandpile Model);
- Random Networks, Scale Free Networks, Complex Networks;
- Phase Transitions;
- (Deterministic) chaos;
- Complexity;
- Algorithmic information theory.

1.5 Scientific Computing (UvA)

1.5.1 Objectives

- Knowledge. The student has basic knowledge about solving (examples of) partial diferential equations and is able to analyze stability and accuracy of a numerical scheme
- Skills: The student is able to develop a computational model for solving examples of partial differential equations (e.g wave equation, diffusion equations) in (simple) applications

1.5.2 Contents

The focus is on developing numerical algorithms (the emphasis is on finite differencing) to solve prototypical partial differential equations. Examples are the wave equation, the time-dependent diffusion equation, the Laplace equation and reaction-diffusion equations. The validation and verification of numerical schemes for solving these equation will discusssed, Methods for analyzing the stability and accuracy of the numerical scheme will be discussed.

2 Artificial Intelligence

2.1 Evolutionary Computing (VU)

2.1.1 Objectives

This course has a threefold objective: 1) To learn about computational methods based on Darwinian principles of evolution. 2) To illustrate the usage of such methods as problem solvers and as simulation tools. 3) To gain hands-on experience in performing computational experiments with evolutionary algorithms.

2.1.2 Content

The course is treating various algorithms based on the Darwinian evolution theory. Driven by natural selection (survival of the fittest), an evolution process is being emulated and solutions for a given problem are being ”bred”. During this course all ”dialects” within evolutionary computing are treated (genetic algorithms, evolution strategies, evolutionary programming, genetic programming). Applications in optimisation, constraint handling, machine learning, and robotics are discussed. Specific subjects handled include: various genetic structures (representations), selection techniques, sexual and asexual variation operators, (self-)adaptivity. Special attention is paid to methodological aspects, such as algorithm design and tuning. If time permits, subjects in Artificial Life will be handled. Hands-on- experience is gained by a compulsory programming assignment.

2.2 Computational Intelligence (UvA)

2.2.1 Objectives

The overall aim of this course is to provide knowledge about concepts, theory, and techniques used in computational intelligence and the know-how to employ these for making intelligent machines. In particular, to enable students to:

- gain profound understanding of fundamental computational intelligence concepts, algorithms, and their implementation;
- understand the theoretical background of proposed solutions;
- develop skills in the use of computational intelligence and to demonstrate this in physical robots or virtual creatures;

- appreciate relevant current research topics in the theory and practice of computational intelligence.

2.2.2 Contents

Computational intelligence can be positioned as the research area that follows a bottom-up approach to developing systems that exhibit intelligent behavior in complex environments. It is often contrasted to the top-down approach followed by traditional artificial intelligence. Typically, sub-symbolic and nature-inspired methods are adopted that tolerate incomplete, imprecise and uncertain knowledge. As a consequence, the resulting approaches allow for approximate, manageable, robust and resource-efficient solutions.

This course covers nature-inspired techniques such as neural networks, evolutionary algorithms and swarm intelligence. Special attention is paid to using such techniques for making autonomous and adaptive machines.

2.3 Data Mining Techniques (VU)

2.3.1 Objectives

The aim of the course is that students acquire data mining knowledge and skills that they can apply in a business environment. How the aims are to be achieved: Students will acquire knowledge and skills mainly through the following: an overview of the most common data mining algorithms and techniques (in lectures), a survey of typical and interesting data mining applications, and practical assignments to gain "hands on" experience. The application of skills in a business environment will be simulated through various assignments of the course.

2.3.2 Contents

The course will provide a survey of basic data mining techniques and their applications for solving real life problems. After a general introduction to Data Mining we will discuss some "classical" algorithms like Naive Bayes, Decision Trees, Association Rules, etc., and some recently discovered methods such as boosting, Support Vector Machines, and co-learning. A number of successful applications of data mining will also be discussed: marketing, fraud detection, text and Web mining, possibly bioinformatics. In addition to lectures, there will be an extensive practical part, where students will experiment with various data mining algorithms and data sets. The grade for the course will be based on these practical assignments (i.e., there will be no final examination).

3 Computer Science

3.1 Performance of Networked Systems (VU)

3.1.1 Contents

Students will acquire basic knowledge of:

- performance aspects of networked systems, consisting of servers, services, and clients

- performance engineering principles and methods,

- quantitative models for predicting and optimizing the performance

of networked systems,

- quantitative models for planning capacity of networked systems. Students will gain experience in engineering and planning performance of networked systems, and will learn how to tackle practical performance problems arising in the ICT industry.

3.1.2 Objectives

Over the past few decades, information and communication technology (ICT) has become ubiquitous and globally interconnected. As a consequence, our information and communication systems are expected to process huge amounts of (digital) information, which puts a tremendous burden on our ICT infrastructure. At the same time, our modern society has become largely dependent on the well-functioning of our ICT systems; large-scale system failures and perceivable Quality of Service (QoS) degradation may completely disrupt our daily lives and have huge impact on our economy. Motivated by this, the course will focus on performance-related issues of networked systems. In the first part, we study capacity planning and modeling for server systems and networks. In the second part, we study the client side of performance while focusing on web applications for both desktop and mobile devices. We address questions like:

- How can we design and engineer networked systems for performance?

- How can we plan server capacity in networked systems?
- How can web applications improve performance across wired and wireless networks?

3.2 Experimental Design & Data Analysis (VU)

3.2.1 Objectives

In this course the student will get acquainted with the most common experimental designs and regression models, nonparametric Vrije Universiteit Amsterdam - Faculteit der Exacte Wetenschappen - M Computer Science (joint degree) - 2017-2018 18-7-2018 - Pagina 39 van 73 tests and bootstrap methods will be discussed. On completion of this course the student should be able to:

- design experiments and analyse the results according to the design, - analyse data using the common ANOVA designs,
- analyse data using linear regression or a generalized linear regression model,
- perform basic nonparametric tests,
- perform bootstrap and permutation tests.

3.2.2 Contents

Regression models try to explain or predict a dependent variable using measured independent variables. Statistical methods are needed if there is random variation in the dependent variables. We will discuss multiple linear regression, analyses of variance (ANOVA), generalized linear regression models. All methods will be illustrated with practical examples. Especially in the case of ANOVA it is necessary that the study is well designed in order to draw sound conclusions from an experiment or survey. In this course a few well known designs (completely randomized, randomized block etc.) and the associated analyses of variance are discussed. The remainder of the course is be dedicated to non-parametric testing methods and bootstrap methods:

- Wilcoxon test for (one and two samples), - Kolmogorov-Smirnov test (two samples), - rank correlation tests,
- permutation and bootstrap tests.

All analyses are carried out by using the statistical package R.

3.3 The Social Web (VU)

3.3.1 Objectives

In this course the students will learn theory and methods concerning communication and interaction in a Web context. The focus is on distributed user data and devices in the context of the Social Web. Course content

3.3.2 Contents

This course will cover theory, methods and techniques for:

- personalization for Web applications;
- Web user & context modelling;
- user-generated content and metadata;
- multi-device interaction;
- usage of social-web data.

3.4 Distributed Algorithms (VU)

3.4.1 Objectives

The main objective is to provide students with an algorithmic frame of mind for solving fundamental problems in distributed computing. They obtain insight into concurrency concepts, and are offered a bird's-eye view on a wide range of algorithms for basic and important challenges in distributed systems.

Characteristic of the course is that correctness arguments and complexity calculations of distributed algorithms are provided in an intuitive fashion and by means of examples and exercises.

3.4.2 Contents

The following topics are treated in the course: Logical clocks, snapshots, graph traversal, termination detection, garbage collection, deadlock detection, routing, election, minimal spanning trees, anonymous networks, checkpointing, fault tolerance, failure detection, consensus, mutual exclusion, self-stabilization, blockchains, database transactions

4 Bioinformatics

4.1 Algorithms in Sequence Analysis (VU)

4.1.1 Objectives

Have you ever wondered how we can track a gene across 3 billion years of evolution? Sequence alignment can be used to compare genes from humans and bacteria, using a dynamic programming algorithm. In this course we focus on algorithms for biological sequences that can be applied to real scientific problems in biology.

Students will obtain in-depth knowledge about the theory of sequence analysis methods. They will also develop understanding and skills to apply the algorithms to protein and DNA sequences. We would like to stress that no biological knowledge is required to enter this course.

Goals

- At the end of the course, the student will be aware of the major

issues, methodology and available algorithms in sequence analysis.

- At the end of the course, the student will have hands-on experience in

tackling biological problems using sequence analysis algorithms and applying the general statistical framework of Hidden Markov Models.

- At the end of the course, the student will be able to implement

several of the most important algorithms in sequence analysis. Course content

4.1.2 Contents

Theory:

- Dynamic programming, database searching, pairwise and multiple

alignment, probabilistic methods including hidden markov models, pattern matching, entropy measures, evolutionary models, and phylogeny.

Practical:

- Programming (in Python) own alignment algorithm based on dynamic

programming

- Reverse translation and dynamic programming

- Homology searching and pattern recognition using biological and

disease examples

- Multiple alignment of biological sequences
- Entropy-based functional residues prediction
- Programming (in Python) own implementation of Hidden Markov Models and

using it to predict protein domain structure