CS5100 - Data Analysis

Prerequisites:

Co-ordinator: Vladimir Vovk

Course value: 20 credits

COURSE SUMMARY

The course teaches fundamental facts and skills in data analysis, including machine learning, data mining, and statistics:

- Supervised learning: classification, regression, and ensemble methods.
- Algorithm-independent machine learning.
- Unsupervised learning and clustering. Exploratory data analysis.
- Bayesian methods. Bayes networks and causality.
- Applications, such as information retrieval and natural language processing.

LEARNING OUTCOMES

By the end of the course students should be able to:

- develop, validate, evaluate, and use effectively machine learning models and statistical models
- apply methods and techniques such as clustering, regression, decision trees, and neural networks
- more generally, extract value and insight from data
- implement machine learning algorithms in R and MATLAB

TEACHING AND LEARNING METHODS

The course will normally be presented as three hours of lecture slots per week during ten weeks, and one hour per week of practical classes during ten weeks. Guided independent study consists of sixteen hours per week during ten weeks.

KEY BIBLIOGRAPHY

Core reading material:

- Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, An Introduction to Statistical Learning with Applications in R, Springer, 2013.

Additional reading:


FORMATIVE ASSESSMENT AND FEEDBACK

The lecturers will provide oral feedback during post-practical class discussions. The students will receive written feedback on their coursework assignments.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5110 - Methods of Bioinformatics

Prerequisites:

Co-ordinator: Alberto Paccanaro

Course value: 10 credits

COURSE SUMMARY

To introduce the main approaches currently in use in bioinformatics, with special emphasis on the analysis of DNA, protein sequences and large scale biological networks emerging from genome sequencing projects and genome-wide experimental assays.

Course content:

- Basic molecular biology: introduction to the basic components of living cells, their functions and interactions, and to other concepts essential to understanding the use of computers in biology.
- Dynamic Programming, sequence alignments, substitution matrices.
- Phylogenetic trees.
- Systems biology: gene expression analysis, Protein-Protein Interaction analysis, biological networks, clustering.

LEARNING OUTCOMES

By the end of the course students should be able to:

- assess and evaluate confidently the main approaches currently in use in bioinformatics;
- demonstrate advanced understanding of the analysis of DNA and protein sequences;
- demonstrate a mastery of applying algorithms of bioinformatics and analysing the results.

TEACHING AND LEARNING METHODS

The course will normally be presented as three hours of lecture slots per week during ten weeks. Guided independent study consists of seven hours per week during ten weeks.

KEY BIBLIOGRAPHY

Core reading material:


FORMATIVE ASSESSMENT AND FEEDBACK

The students will receive written feedback on their coursework assignments.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5200 - On-line Machine Learning

Prerequisites: CS5810 Programming for Data Analysis (taken by Big Data students as core in Term 1) or good knowledge of MATLAB.

Co-ordinator: Yuri Kalnishkan

Course value: 20 credits

COURSE SUMMARY

The course addresses the on-line framework of machine learning in which the learning system learns and issues predictions or decisions in real time, perhaps in a changing environment. The course teaches protocols, methods and applications of on-line learning.

The course covers probabilistic on-line models based on Markov chains and their applications (PageRank, Markov chain Monte-Carlo) with a particular focus on hidden Markov models. Time series models are covered and their connections with Kalman filters are explored.

Learning models based on the prequential paradigm are covered in depth, including prediction with expert advice, aggregating algorithm, sleeping and switching experts. Universal algorithms are covered with an application to portfolio theory including Cover’s universal portfolios.

Finally, prediction with confidence framework is covered.

LEARNING OUTCOMES

By the end of the course students should be able to:

- understand and evaluate probabilistic and non-probabilistic on-line learning protocols;
- demonstrate advanced understanding of the prequential and prediction with confidence frameworks;
- demonstrate advanced knowledge of methods of Markov models, prediction with expert advice and prediction with confidence;
- analyse the properties of on-line learning algorithms;
- apply on-line algorithms to real-world data and evaluate the results.

TEACHING AND LEARNING METHODS

The course will be presented as three/four hours of lectures per week during ten weeks,
and one hour per week of practical classes during eight weeks.

KEY BIBLIOGRAPHY

Core reading material:


FORMATIVE ASSESSMENT AND FEEDBACK

The lecturers will provide oral feedback during post-practical class discussions. The students will receive written feedback on their coursework assignments.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5220 - Digital Audio and Applications

Prerequisites:

Co-ordinator: Nuno Barreiro

Course value: 10 credits

COURSE SUMMARY

Digital audio has many applications, ranging from music production to audio enhancement, or from noise cancellation in mobile phones to automatic music generations for games. These applications share common features that will be covered in this course. Starting with the fundamentals of sound (nature, transmission, volume), students will then learn how to convert sound to and from the digital domain. Once in the digital domain, audio can be manipulated using algorithms from Digital Signal Processing. The course covers some basic DSP algorithms, engaging the students both in audio analysis (amplitude, frequency, pan, etc.) and audio synthesis (additive, FM, etc.). By the end of this course, students should be able to write simple applications that analyse, manipulate and generate sound.

LEARNING OUTCOMES

By the end of the course students should be able to:

- master digital audio and the fundamentals of AD/DA conversion
- program Fourier analysis to synthesize and transform sounds in the frequency domain
- demonstrate an understanding of lossy and lossless compression algorithms
- program digital signal processing (post-processing and effects)
- develop applications for digital audio (music production, noise cancellation, audio enhancement, automatic music generation)
- acquire a basic knowledge of sound perception and psychoacoustics

TEACHING AND LEARNING METHODS

The course will be presented as four hours of sessions per week.

KEY BIBLIOGRAPHY

Core reading material:

FORMATIVE ASSESSMENT AND FEEDBACK

The lecturer will provide oral feedback during practical class sessions. The students will receive written feedback on their coursework.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5234 - Large-Scale Data Storage and Processing

Prerequisites: Basic familiarity with fundamentals of computing along with programming experience in Java at the level of CS5800 or equivalent, or passing the relevant departmental test. Adequate knowledge of core database concepts including relational databases and SQL at the level of CS5855 or equivalent, or passing the relevant departmental test.

Co-ordinator: Gregory Chockler

Course value: 20 credits

COURSE SUMMARY

Study underlying principles of storage and processing massive collections of data, typical of today's Big Data systems. Gain hands-on experience in using large and unstructured data sets for analysis and prediction. The topics covered will include techniques and paradigms for querying and processing massive data sets (MapReduce, Hadoop, data warehousing, SQL for data analytics, stream processing), fundamentals of scalable data storage (NoSQL data bases such as MongoDB, Cassandra, HBase), working with dynamic web data (data acquisition, data formats), elements of cloud computing, and applications to real world data analytics and data mining problems (sentiment analysis, social network mining).

LEARNING OUTCOMES

By the end of the course students should be able to:

- Knowledge and understanding of core concepts, theories and principles of large-scale data storage and processing frameworks;

- Sound evaluation of opportunities and challenges related to leveraging those frameworks for building massive scale analytics solutions, and an ability to make recommendations to resolve these challenges;

- Proficient knowledge and use of at least one large-scale data store system, and at least one massive scale processing framework;

- Ability to design, develop, and evaluate an end-to-end analytics solution combining large-scale data storage and processing frameworks.

- Knowledge of cloud computing as a platform for Big Data analytics.

TEACHING AND LEARNING METHODS
The course will be presented as two hours of lectures per week during ten weeks, and two hours per week of practical classes during ten weeks. Guided independent study consists of one hundred and sixty hours during the course.

**KEY BIBLIOGRAPHY**

Core reading material:


- Miner, Shook, MapReduce Design Patterns, O'Reilly, ISBN-10: 1449327176


**FORMATIVE ASSESSMENT AND FEEDBACK**

The lecturers will provide oral feedback during post-practical class discussions. The students will receive written feedback on their coursework assignments.

**SUMMATIVE ASSESSMENT**

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5250 - Visualisation and Exploratory Analysis

Prerequisites:

Co-ordinator: Zhiyuan Luo

Course value: 20 credits

COURSE SUMMARY

The course aims to teach the principles and arts of statistical visualisation and exploratory analysis of data. There are principles, theory, and skills to be acquired.

Course content includes:

Construction of informative bivariate plots. This will cover standard axis transformations and why they are useful, and also the importance of constructing data transformations such that a reference hypothesis presents in the simplest possible form. Smoothing with loess. Visualisation of distributions: histograms, binning, and kernel density estimation; cumulative distributions and QQ plots. These topics will be backed up with plenty of practical examples and coursework.

Visualisation of multivariate data. Linear projections and principal components analysis in primal and dual mode. Kernel PCA. Canonical correlations analysis (CCA) and Kernel CCA.

Dimensional reduction. Non-linear methods, examples such as t-SNE, Isomap, and proxigrams.

Exploratory cluster analysis, and metrics for comparison of clusterings.

Standard methods for visualisation of relational and graph data will be described including practical implementations such as Gephi.

Checking a found pattern by constructing a suitable randomisation of the data; importance of guarding against ‘snooping’.

Principles of selecting modes of presentation for various quantities; basic principles of colour scale design and glyph choice.

LEARNING OUTCOMES

By the end of the course students should be able to:

- perform open-ended exploratory analysis of data, and master the analytical presentation and critical evaluation of the results of statistical analyses;
- construct linear projections of multivariate data and demonstrate an advanced
understanding of non-linear dimension reduction methods;

- demonstrate practical experience of using standard graph visualisation methods and evaluation of results;

- be effective in avoiding data snooping;

- critically evaluate choices in representational mode, glyph design, and colour design for presentation graphics.

TEACHING AND LEARNING METHODS

The course will be presented as two hours of lectures per week during eleven weeks, and one hour per week of practical classes during eleven weeks.

KEY BIBLIOGRAPHY

Core reading material:

- Visualising Data, by William S. Cleveland.

- The Elements of Graphing Data, by William S. Cleveland.

- The Visual Display of Quantitative Data, by Edward Tufte.

- Envisioning Information by Edward Tufte.

FORMATIVE ASSESSMENT AND FEEDBACK

The students will receive written feedback on their coursework assignments.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5490 - Computational Optimization

Prerequisites:

Co-ordinator: Gregory Gutin

Course value: 10 credits

COURSE SUMMARY

This course will:

- provide students with advanced understanding of fundamental models of computational optimisation (CO) and important algorithms for solving CO problems;

- enable students to analyse CO algorithms using theoretical and computational methods;

- discuss and compare available software packages for solving CO problems.

Course content:

- Linear programming (LP) model. Formulating problems as LP problems.

- Graphical solution. Simplex method.

- Duality in LP. Decomposition of LP problems. LP software.


- CO problems. Polynomial-time algorithms. Greedy-type algorithms. Construction heuristics and local search for the TSP.

- Theoretical and computational analysis of heuristics and meta-heuristics.

LEARNING OUTCOMES

By the end of the course students should be able to:

- understand and evaluate fundamental models of computational optimization (CO)

- examine and apply the fundamental algorithms for solving CO problems

- apply theoretical and computational methods to analyse CO algorithms

- further develop algorithmic and mathematical skills
TEACHING AND LEARNING METHODS

The course will be presented as three hours of lecture slots per week during ten weeks. Guided independent study consists of seven hours per week during ten weeks.

KEY BIBLIOGRAPHY

Core reading material:


FORMATIVE ASSESSMENT AND FEEDBACK

The lecturers will provide written feedback based on a mid-term test and exercise sheets.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5504 - Business Intelligence Systems, Infrastructures and Technologies

Prerequisites:

Co-ordinator: Giorgios Koutsoukos

Course value: 20 credits

COURSE SUMMARY

Business Intelligence (BI) refers to the skills, processes, methodologies, technologies, applications, and practices used in order to leverage (gathering, storing, analyzing) an organization’s internal and external information assets to support and improve decision-making. With the advent of Big-Data there is considerably increased demand for skills and knowledge, both conceptual and technological, that can be effectivelly applied to support this new era of Big-Data based decision-making.

This course aims to provide students with

(a) a broad understanding of the information assets and the conceptual and technical architectures of information and business intelligence systems in modern organizations

(b) the necessary background knowledge of, and skills to design, implement and evaluate business intelligence systems and technologies.

Course content:

- Introduction to Information Systems & Business Intelligence: Overview of Information Systems and BI Systems, Information Systems and BI Technical Architectures (Logical & Physical aspects), Acquisition models and Business cases.

- Data Warehousing and Dimensional Modelling (Definitions, Concepts, Architectures, Design Processes, Implementation Aspects)

- BI Applications: OLAP, Reports, Dashboards, Data Mining, Visualization and UI design

- Practical Sessions (Surgeries, Labs) on Dimensional Modelling, Reports and Dashboards using BI tools.

LEARNING OUTCOMES

By the end of the course students should be able to:

- have a holistic view of business intelligence systems and their role in the IT environment of modern organizations

- understand and evaluate the concepts, terminology and architectures of Data
Warehouses and BI solutions

- understand Data Modelling concepts and provide design solutions using Dimensional Modeling
- know the important elements of business intelligence applications such as Data Analysis, Data Mining and Dashboards; understand and evaluate BI Visualization aspects and the relationship of BI solutions to CRM and ERP systems.
- have hands-on experience with industrial business intelligence tools

TEACHING AND LEARNING METHODS

The course will be taught in weekly sessions over the course of a term.

KEY BIBLIOGRAPHY

Core reading material:


FORMATIVE ASSESSMENT AND FEEDBACK

The lecturers will provide oral feedback during practical sessions and written feedback to coursework.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5580 - Advanced Data Communications

Prerequisites:

Co-ordinator: Zhiyuan Luo

Course value: 10 credits

COURSE SUMMARY

The course teaches important topics in modern data communications including text, image, audio and video transfer over networking infrastructures and recent trends in QoS management, network security and network management.

Course content:


- Multimedia communications: multimedia information representation, coding and compression, applications and standards, quality of service (QoS) requirements.

- Internet technologies: IP addressing, routing algorithms and routing protocols, RIP, OSPF, the Internet multicast model, scheduling and queue management.

- Security in communication networks: security issues, security mechanisms, secure protocols

- Network management: network management issues, infrastructure and framework for Internet management.

LEARNING OUTCOMES

By the end of the course students should be able to:

- show advanced understanding of multimedia communications

- evaluate coding and compressing algorithms for text, audio, image and video

- analyse Internet technologies in terms of supporting Quality of Service

- examine modern network security mechanisms and their applications

- evaluate Internet network management issues and explain possible solutions

TEACHING AND LEARNING METHODS
The course will be presented as three hours of lecture slots per week during ten weeks. Guided independent study consists of seven hours per week during ten weeks.

**KEY BIBLIOGRAPHY**

Core reading material:


- Fred Halsall: Multimedia Communications: applications, networks, protocols and standards, Addison-Wesley


**FORMATIVE ASSESSMENT AND FEEDBACK**

The students will receive written feedback on their coursework assignments.

**SUMMATIVE ASSESSMENT**

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5800 - Computation with Data

Prerequisites:

Co-ordinator: Gregory Chockler

Course value: 10 credits

COURSE SUMMARY

The course will be structured around a series of programming problems and assignments designed to teach students basics of algorithmic thinking and problem solving using programming. The students will be introduced to the Java programming language features and constructs as well as basics of object-oriented programming, which will be put in the context of solving specific algorithmic tasks.

LEARNING OUTCOMES

By the end of the course students should be able to:

- understand procedural and object-oriented programming concepts;
- apply understanding to solve programming tasks;
- evaluate programming solutions.

TEACHING AND LEARNING METHODS

The course will be presented as one hour of lectures per week during ten weeks, and one hour per week of practical classes during ten weeks. Guided independent study consists of eight hours per week during ten weeks.

KEY BIBLIOGRAPHY

Core reading material:


FORMATIVE ASSESSMENT AND FEEDBACK

The lecturers will provide oral feedback during post-practical class discussions.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5810 - Programming for Data Analysis

Prerequisites:

Co-ordinator: Alberto Paccanaro

Course value: 10 credits

COURSE SUMMARY

This is a practical course and it aims to give students advanced understanding of:

- MATLAB, a programming language widely used for data analysis;
- Weka, an environment which provides a collection of machine learning algorithms for data mining tasks.

Course content:

Matlab:

- Using Matlab as calculator. Using the help system.
- Data input and output. Vectors, arrays and matrices.
- Data visualization (including plots in 2 and 3 dimensions, scatter plots, barplots, histograms).
- Implementing concepts from Linear Algebra and Statistics (including probability distributions, matrix decompositions).
- Programming: loops, conditional executions, string manipulations, data structures, etc.
- Writing functions, debugging the code. Using packages and toolboxes

WEKA:

- Data Input: concepts, instances, attributes. Feature selection
- Using machine learning schemes (including decision trees, naive Bayes classifiers, clustering methods)
- Training and Testing, predicting generalization performance, cross-validation

LEARNING OUTCOMES

By the end of the course students should be able to:

- have advanced understanding of the general principles of the Matlab language
- apply MATLAB to implement algorithms for data analysis; evaluate efficiency of implementations
• apply WEKA for data analysis

TEACHING AND LEARNING METHODS

The course will be presented as one hour of lectures per week during ten weeks, and two hours per week of practical classes during ten weeks. Guided independent study consists of seven hours per week during ten weeks.

KEY BIBLIOGRAPHY

Core reading material:


FORMATIVE ASSESSMENT AND FEEDBACK

The lecturers will provide oral feedback during post-practical class discussions.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5840 - Interconnected Devices

Prerequisites: Programming experience (professional or academic), preferably in C/C++

Co-ordinator: Sara Bernardini

Course value: 10 credits

COURSE SUMMARY

The course will present the Internet of Things (IoT) by covering the applications of IoT in society, the components of typical IoT systems and the trends for the future.

The course will expose students to IoT design considerations, constraints and interfacing between the physical world and the IoT devices.

After this stage, the course will focus on how the Arduino platform works in terms of the physical board, the libraries and the IDE (integrated development environment). It will cover how to program the Arduino via C/C++ code and how to access the pins on the board via the software to control external devices.

Finally, the course will provide hands-on knowledge on how to plug shields into the main Arduino board to perform other functions such as sensing and actuating.

LEARNING OUTCOMES

By the end of this course a student should be able to:

- Design, build, and test a simple microcontroller-based embedded system
- Using C/C++ code to program an embedded system
- Critically analyse requirements and make design trade-off decisions between hardware, software and networking
- Connect multiple devices to achieve planned goals

TEACHING AND LEARNING METHODS

Up to 22 hours of lectures and laboratory classes.

KEY BIBLIOGRAPHY

- Michael Margolis, Arduino Cookbook, 3rd edition
- John Hughes, Arduino in a Nutshell
FORMATIVE ASSESSMENT AND FEEDBACK

Verbal feedback on progress will be given during the weekly laboratory sessions.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5855 - Databases

Prerequisites:

Co-ordinator: Magnus Wahlström

Course value: 10 credits

COURSE SUMMARY

The aim of this course is to teach students a number of database concepts and techniques. This ranges from the specification and modelling stages to the implementation of relational databases. The course also introduces students to the usage of databases from software applications.

The content of the course includes:

- Data modelling: views, subschema, data dictionary, data independence, entity relationship model. The relational model: relations, attributes, domains, relational algebra. Database design: normalisation, normal forms, entities and attributes SQL: basic SQL, correspondence between the relational model and SQL commands, simple queries, combination and sub-queries Administration and implementation: integrity, recovery from failure, concurrency, deletion and updating, forms, report writing.

LEARNING OUTCOMES

By the end of this course a student should be able to:

- explain the issues involved in database design and the theory of the relational view of data
- describe the crucial issues concerning database integrity and recovery from failure
- write SQL queries
- design and implement a database, from the user specifications to the final design

TEACHING AND LEARNING METHODS

Lecture based delivery supported by laboratory sessions. Normally 3 hours of sessions per week.

KEY BIBLIOGRAPHY


FORMATIVE ASSESSMENT AND FEEDBACK

Verbal feedback will be given during the laboratory sessions.
Details of coursework submission deadlines will be published on the department website at the start of term.
CS5860 - Advanced Distributed Systems

Prerequisites: Programming experience (professional or academic)

Co-ordinator: Peter Robinson

Course value: 20 credits

COURSE SUMMARY

The course will cover fundamental principles of building modern distributed systems, for example in the context of the Internet of Things. The specific emphasis will be on the two central components of the IoT reference architecture: cloud infrastructure and wireless networking.

The course will discuss major challenges found in these environments (such as massive scales, wide distribution, decentralisation, unreliable communication links, component failures and network partitions) and general approaches for dealing with these challenges.

The topics covered will include:

- abstract models (such as the synchronous and asynchronous distributed computing models, models for wireless networks);
- algorithmic techniques (such as distributed coordination, fault-tolerant design of distributed algorithms, synchronization techniques);
- practical case studies.

The students will also have an opportunity to apply the studied material for implementing various components of a realistic distributed system through a series of formative coursework assignments, lab practicals, and a final project.

LEARNING OUTCOMES

- Display a mastery of core concepts, theories and principles of distributed systems
- Demonstrate knowledge of algorithmic techniques for solving problems in a distributed environment.
- Explain the practical aspects of implementing various components of a distributed system
• Implement various components of a realistic distributed system

TEACHING AND LEARNING METHODS

Normally 4 hrs per week of lectures and laboratory sessions.

KEY BIBLIOGRAPHY


FORMATIVE ASSESSMENT AND FEEDBACK

Oral feedback in laboratory sessions.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5870 - Wireless Sensor and Actuator Networks

Prerequisites: CS5840 or equivalent

Co-ordinator: Sara Bernardini

Course value: 20 credits

COURSE SUMMARY

The course will combine formal lectures that will focus on the algorithms and the protocols behind wireless sensor and actuator networks and lab classes that will focus on how to build wireless sensor and actuator networks for a variety of applications.

The lectures will cover critical design factors for WSANs, the protocol stack, models and algorithms for WSANs, routing protocols and more advanced open research problems, such as topology control and mobility.

The practical classes will cover how to design and build wireless sensor networks and intelligent interactive devices with the ZigBee wireless networking protocol.

LEARNING OUTCOMES

By the end of this course a student should be able to:

- Design, build, and test wireless sensor and actuator networks for a variety of application setups
- Explore the design space and conduct trade-off analysis between performance and resources
- Design and implement simple protocols and algorithms for wireless sensor and actuator networks
- Think critically about open research problems

TEACHING AND LEARNING METHODS

Up to 44 hours of lectures and laboratory classes.

KEY BIBLIOGRAPHY

- Building Wireless Sensor Networks, Robert Faludi (without kit)
- Algorithms and protocols for wireless sensor networks, Azzedine Boukerche

FORMATIVE ASSESSMENT AND FEEDBACK
Verbal feedback will be given during the laboratory sessions

**SUMMATIVE ASSESSMENT**

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5920 - Machine Learning

Prerequisites:

Co-ordinator: Vladimir Vovk

Course value: 10 credits

COURSE SUMMARY

The course will teach main ideas of machine learning with a particular emphasis on kernel methods.

Course content:

- Nearest Neighbours for classification and regression; interesting distances.
- Discriminant analysis.
- Ridge regression and Lasso.
- Support vector machines for classification and regression.
- Kernel trick and its applications to the algorithms covered so far.
- Practically useful kernels, including string kernels.

LEARNING OUTCOMES

By the end of the course students should be able to:

- Demonstrate the advanced knowledge of the theoretical foundations of machine learning;
- Understand to an advanced level the main advantages and limitations of various approaches to machine learning and specific machine-learning algorithms;
- Have a mastery of several machine-learning algorithms;
- Understand, at an advanced level, ways to apply the ideas and algorithms of machine learning in industry, medicine, and other fields.

TEACHING AND LEARNING METHODS

The course will be presented as two/three hours of lecture slots per week during ten weeks, and one hour per week of practical classes during nine weeks. Guided independent study consists of seven hours per week during ten weeks.
KEY BIBLIOGRAPHY

Core reading material:


FORMATIVE ASSESSMENT AND FEEDBACK

Students will have access to help and oral feedback during practical classes.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5930 - Methods of Computational Finance

Prerequisites:

Co-ordinator: Yuri Kalnishkan

Course value: 10 credits

COURSE SUMMARY

Pricing derivatives (and associated strategies of dynamic hedging) will be the main topic of this course. The course is called "computational finance" because many derivatives are too complicated to be priced and hedged using simple mathematical formulae and therefore advanced computational models are required. The second major topic is risk exposure including the concept of value of risk.

Course content:

- Introduction: financial markets; the "rules of the game".
- Taxonomy of securities: main kinds of derivative securities and underlying markets.
- Mathematical techniques: Wiener process; diffusion processes as mathematical models of price dynamics; stochastic differential equations; computer simulations.
- Pricing and hedging in the Black-Scholes world: risk-neutral valuation; the Black-Scholes equation and analytic formulae; the "Greek letters" and their use.
- Beyond the Black-Scholes world: application issues; computational models; fractals and their use in finance.
- Risk management: Value at Risk. Analytical and computational techniques.
- Efficient markets hypothesis: theory vs empirical evidence.

LEARNING OUTCOMES

By the end of the course students should be able to:

- demonstrate advanced understanding of mathematical and computational models of underlying and derivative securities;
- evaluate and apply techniques for pricing derivatives and for dynamic hedging;
- demonstrate advanced understanding of mathematical and computational models of risk exposure;
- evaluate and apply techniques for calculating value at risk;
- demonstrate advanced understanding of the market efficiency hypothesis and
apply it to examine financial techniques.

TEACHING AND LEARNING METHODS

The course will be presented as three hours of lecture slots per week during ten weeks. Guided independent study consists of seven hours per week during ten weeks.

KEY BIBLIOGRAPHY

Core reading material:

- J.Hull, Options, Futures, and Other Derivatives. Prentice Hall, 8th edition, 2011 (or any other edition)

FORMATIVE ASSESSMENT AND FEEDBACK

The students will receive written feedback on their coursework assignments.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5940 - Intelligent Agents and Multi-agent Systems

Prerequisites:

Co-ordinator: Kostas Stathis

Course value: 10 credits

COURSE SUMMARY

The course aims to teach the concept of an agent and multi-agent system, and the main applications for which they are appropriate; the main issues surrounding the design of intelligent agents and multi-agent society; a contemporary platform for implementing agents and multi-agent systems.

Course content:

- Introduction: what is an agent: agents and objects; agents and expert systems; agents and distributed systems; typical application areas for agent systems.

- Intelligent Agents: (a) abstract architectures for agents; tasks for agents. (b) the design of intelligent agents - reasoning agents (e.g., Agent0), (c) agents as reactive systems (e.g., subsumption architecture); (d) hybrid agents (e.g., PRS); layered agents (e.g., Interrap).

- Multi-Agent Systems: (a) classifying multi-agent interactions - cooperative versus non-cooperative; (b) zero-sum and other interactions; what is cooperation? how cooperation occurs - the Prisoner’s dilemma and Axelrod’s experiments; (d) interactions between self-interested agents: auctions systems; negotiation; argumentation; (e) interaction languages and protocols: speech acts, KQML/KIF, the FIPA framework, ontologies, coordination languages.(f) interactions between benevolent agents: cooperative distributed problem solving (CDPS), partial global planning; coherence and coordination;

- Applications of intelligent agents and multi-agent systems.

LEARNING OUTCOMES

By the end of the course students should be able to:

- understand the notion of an agent, explain how agents are distinct from other software paradigms (e.g., objects), and analyse the characteristics of applications that lend themselves to an agent-oriented solution;

- appraise the key issues associated with constructing agents capable of intelligent autonomous action, and the main approaches taken to developing such agents;

- assess the key issues in designing societies of agents that can effectively cooperate.
in order to solve problems, including an evaluation of the key types of multi-agent interactions possible in such systems;

- evaluate the main application areas of agent-based solutions, and develop a meaningful agent-based system using a contemporary agent development platform

TEACHING AND LEARNING METHODS

The course will be presented as two hours of lecture slots per week during ten weeks, and one hour per week of tutorials during ten weeks. Guided independent study consists of seven hours per week during ten weeks.

KEY BIBLIOGRAPHY

Core reading material:


FORMATIVE ASSESSMENT AND FEEDBACK

The lecturers will provide oral feedback during tutorials. The students will receive written feedback on their coursework assignments.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5945 - Semantic Web

Prerequisites:

Co-ordinator: Iddo Tzameret

Course value: 10 credits

COURSE SUMMARY

The Web, as it exists today, primarily supports human understanding and the interpretation of the vast information space it encompasses. However the Web was originally designed with a goal to support not only human-human communication but also as one that would enable automated machine processing of data with minimal human intervention. The Semantic Web is Tim Berners-Lee’s vision of a machine understandable and unambiguously computer interpretable Web. The rationale behind such a system is that most of the data currently posted on the web is buried in HTML files suitable for human reading and not for computers to manipulate meaningfully. The semantic Web, an extension of the current web, can be thought of as a globally linked database where information is given well-defined meaning using metadata for better enabling computers and humans to work in close cooperation. The realisation of a Semantic Web will thus make machine reasoning more ubiquitous and powerful, creating an environment where intelligent software agents can roam, carrying out sophisticated tasks for their users.

Though the original motivation of the semantic web was to constitute the next generation of the WWW, the standards and technologies developed in the process have been found useful in specific realm enterprises as well. From this perspective the Semantic Web can be viewed as a semantically-rich data model that is more expressive than the usual relational data model used in standard databases systems, and is also more adequate to distributed and incomplete resources.

This course is about the notions, concepts, technologies and modelling techniques that constitute the Semantic Web, whose key distinguishing characteristics will be the support for and use of semantics in new, more effective, more intelligent, ways of managing information and supporting applications.

LEARNING OUTCOMES

By the end of the course students should be able to:

- demonstrate a mastery of the fundamental concepts, and standards of the semantic web
- demonstrate an advanced understanding of the use of standards such as RDF, RDFS and OWL for modeling different scenarios and reasoning
- demonstrate an advanced understanding of underlying logical theory behind the semantic web, for example Description Logic
TEACHING AND LEARNING METHODS

The course will be presented as two hours of lectures per week, and one hour per week of practical classes during five weeks.

KEY BIBLIOGRAPHY

Core reading material:


- A Semantic Web Primer (Cooperative Information Systems), 3rd Ed., by Grigoris Antoniou (Author), Paul Groth (Author) et al., 2012.


FORMATIVE ASSESSMENT AND FEEDBACK

The students will receive oral feedback in laboratory sessions.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.
CS5950 - Deep Learning

Prerequisites: Course unit providing good knowledge of MATLAB or sufficient past experience.

Co-ordinator: Chris Watkins

Course value: 20 credits

COURSE SUMMARY

The aim of the course is to give students an introduction to deep learning that covers neural network optimisation by gradient descent from first principles, and which also gives a broader introduction to a range of advanced architectures, with hands-on implementation.

The course starts by considering models of artificial neural networks for supervised learning, and introduces notions of activation function, loss function, and computation of loss-grads using back-propagation with the chain rule. Neural network learning with back-propagation and different gradient descent algorithms will be covered in detail, and visualised in lab-sessions. Next, the ‘disappearing gradient’ problem in deep architectures will be raised, and methods for resolving this problem will be discussed. A range of deep architectures will be described for discriminative learning, generative learning and learning of representations, and for reinforcement learning. Students will implement a deep architecture using a toolkit in a project assignment at the end of the course.

LEARNING OUTCOMES

By the end of this course a student should be able to:

- Demonstrate an advanced understanding of the notions of an artificial neural network, and of learning by minimising a loss function, using training, validation, and test data sets.

- Master computation of loss-grads for different neural network architectures, and of a range of algorithms for optimisation by gradient descent, as applied in neural networks.

- Demonstrate a good comparative understanding of a range of deep learning architectures.

- Master the basic techniques necessary for gradient optimisation of deep networks, and of some diagnostics needed for determining whether gradient descent is working correctly.

- Apply deep learning algorithms to real-world data and evaluate the results.

- Implement and run deep learning algorithms using appropriate tool-kits.
TEACHING AND LEARNING METHODS

Lecture based delivery supported by laboratory sessions. Normally 4 hours of sessions per week.

KEY BIBLIOGRAPHY

- Deep Learning, by Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press Dec 2016 (already available on web)

- The course will make extensive use of notes, handouts, and freely available material on the web, which is abundant and rapidly changing.

FORMATIVE ASSESSMENT AND FEEDBACK

Verbal feedback will be given during the laboratory sessions.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.