Autonomous Intelligent Machines & Systems

EPSRC Pioneering research and skills

EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems (AIMS)

Course Syllabus 2016

The following courses will be offered during Michaelmas 2016 and Hilary Term 2017.

Michaelmas Term

- Data Estimation and Inference - Week 2
- Machine Learning - Week 3
- Signal Processing - Week 4
- Optimization - Week 5
- Embedded Systems Programming - Week 6
- Introduction to Modern Control - Week 7
- Learning from Big Data - Week 8

Hilary Term

- Computer Vision - Week 1
- Systems Verification - Week 2
- Privacy & Security in Wireless Networks - Week 3
- Computational Game Theory - Week 4
- Sensor and Actuator Networks – Weeks 6 and 7
- Mobile Robotics- Weeks 7/ 8

There will also be various one day courses and one week courses which will happen out of term time, and also in Trinity Term. You will be notified about these in due course.

Michaelmas Term

Data Estimation and Inference
Pedro Pinies

Introduction

We are implicit in an increasingly dense web of data: the rise of big data presents unprecedented research opportunities across science and engineering. Modelling such data presents acute challenges; its complexity demands principled management of uncertainty. As such, this course will provide an introduction to probabilistic inference and modern computational statistics.

Objectives

- Understanding how to manage and transform probability distributions as a tool for characterising knowledge and ignorance in light of data.
- The practical use of decision theory to describe how to take action in an uncertain world.
- The selection of appropriate approximations to permit the implementation of probabilistic systems within computational constraints.

Contents

- Basic properties of distributions: independence, joint and conditional distributions, marginals. Bayes' theorem. The extension from univariate to multivariate distributions.
- The representation of a distribution as a belief network.
- Gaussian processes as a means of inferring functions.
- Bayesian Decision Theory: the specification of appropriate loss functions, and the minimisation of expected loss to select actions. Classifiers and Decision Surfaces, particularly the discriminant function derived from Normal distributions.
- Techniques of approximation: Maximum Likelihood; Maximum A-Posteriori; Laplace Approximations

Other Sources

- K. P. Murphy, 'Machine Learning: A Probabilistic Perspective', 2012, MIT Press
- David Barber, `Bayesian Reasoning and Machine Learning <http://web4.cs.ucl.ac.uk/staff/D.Barber/pmwiki/pmwiki.php?n=Brml.Online>', (pdf available for free), 2012, Cambridge University Press. Chapters: 1; 3.1, 3.3; 7.1-7.2; 8; 9.1-9.2; 10; 17.1-17.4; 24.1-24.4.
- Phil Gregory, `Bayesian Logical Data Analysis for the Physical Sciences', 2010, Cambridge University Press. Chapters: 1; 3.1-3.4; 5.1-5.12; 9.
- David MacKay, Information Theory, Inference, and Learning Algorithms<http://www.inference.phy.cam.ac.uk/itprnn/book.html>' (pdf available for free), 2003, Cambridge University Press. Chapters: 2.1-2.4; 3; 36.

Machine Learning
Frank Wood

Introduction

Machine learning is a broad field that spans reinforcement learning, deep learning, Bayesian nonparametrics, graphical models, probabilistic programming, and much more. In this course we will focus on a central theme: probabilistic inference and, to a lesser extent, modelling. Participants will leave well versed in the fundamentals of computational approaches to approximate inference in probability models.

Objectives

- Understand several flavors of approximate inference and trade-offs between them
- Gain experience with models such as Gaussian mixture models (GMM), latent Dirichlet allocation (LDA), linear dynamical systems (LDS), Dirichlet/Gaussian (D/GP) process-based models, etc.

Lectures (and contents):

M: Bayes nets, factor graphics, belief propagation
Tu: EM, Variational inference, GMM
W: Markov chain Monte Carlo, LDA
Th: Hidden Markov model, forward/backward, LDS/Switching LDS, sequential Monte Carlo
F: Nonparametrics: DP, GP, HDP

Labs

M: Implement belief propagation, run on example Bayes net
Tu: Implement expectation maximization for Gaussian mixture model
Wed: Implement collapsed Gibbs sampler for LDA
Th: Implement particle filter for switching linear dynamical system
F: Use probabilistic programming to implement hierarchical Dirichlet process

Prerequisites

Calculus, linear algebra, probability and statistics, familiarity with programming

Other Sources

- C. M. Bishop, 'Pattern Recognition and Machine Learning', 2006, Springer
- K. P. Murphy, 'Machine Learning: A Probabilistic Perspective', 2012, MIT Press
- A. Gelman, 'Bayesian Data Analysis', 1996, Chapman & Hall
- D. Barber, Bayesian Reasoning and Machine Learning', (pdf available for free), 2012, Cambridge University Press. Chapters: 1; 3.1, 3.3; 7.1-7.2; 8; 9.1-9.2; 10; 17.1-17.4; 24.1-24.4.
- D. MacKay, ' Information Theory, Inference, and Learning Algorithms' (pdf available for free), 2003, Cambridge University Press. Chapters: 2.1-2.4; 3; 36.

Signal Processing
Steve Roberts

Introduction

Data which changes with time is ubiquitous in the modern world, from finance to climate, from communications to smart-grids and in the scientific domain it underpins knowledge extraction from astronomy to zoology. This course provides an underpinning foundation to dealing with such time-varying signals.

Objectives

- Present the basics of timeseries modelling
- Provide an overview of filters and stochastic models
- Present the concepts and practice of state-space models and adaptive systems
- Present the underpinning concepts of spectral estimation and timeseries forecasting

Contents

- Basics of signals and systems
- The Fourier domain
- Filters and stochastic models
- State-space models
- Source separation
- Spectral estimation & timeseries forecasting

Prerequisites

- Probability and estimation
- Linear algebra

Other Sources

- Lynn. An introduction to the analysis and processing of signals. Macmillan.
- Oppenhein & Shafer. Digital signal processing. Prentice Hall
- Orfanidis. Introduction to Signal Processing. Prentice Hall.
- Proakis & Manolakis. Digital Signal Processing: Principles, Algorithms and Applications.

Exercises

- Coding methods and testing on real timeseries

Contents

Optimization
Pawan Mudigonda and Philip Torr

Syllabus

- Introduction
- Discrete optimization fundamentals and motivating applications
- Continuous optimization fundamentals and motivating applications
- Linear and quadratic programming
- Duality
- Optimization techniques
- Dynamic programming
- Shortest paths
- Tree-structured graphical models
- Graph cuts
- Minimum cut and maximum flow
- Local search for NP-hard problems
- Linear programming relaxation
- Tight relaxations for minimum cut
- Accurate relaxations for NP-hard problems

Practicals

Three x 2hrs (Tue, Thu, Fri)

- Unconstrained continuous optimization and linear programming
- Graph cuts and local search for sparse CRFs
- Convex relaxations for dense CRFs

Reading

- Convex Optimization by Stephen Boyd & Lieven Vandenberghe, CUP, 2004 (http://stanford.edu/~boyd/cvxbook)
- The design of approximation algorithms by David P. Williamson and David B. Shmoys (http://www.designofapproxalgs.com)

Embedded Systems Programming
Daniel Kroening and Alessandro Abate

Introduction

This course covers the bare basics of programming techniques for typical embedded systems, including programming language, tools and workflow. The module features a substantial practical component.

Objectives

- Arrive at starting point required to do a basic embedded programming project
- Knowledge of the most commonly used design and programming tools
- Consciousness of the implications of limited hardware resources in embedded programming.

Contents

- Brief summary of embedded micro-architectures, assembler
- Basics of C
- Basics of C++ and object-orientation, including related UML diagram types
- Basics of programming
- Basics of using Simulink for generating embedded software

Prerequisites: none

Other Sources: textbooks to be selected

Introduction to Modern Control
Paul Goulart, Alessandro Abate and James Anderson

Introduction

This module will introduce students to modern control theory based on state space methods and optimization. The focus will be primarily on modelling, analysis and controller design of continuous time, Linear Time Invariant (LTI) systems. The course will emphasise, through examples how to apply modern control techniques to system models using the MATLAB and Simulink environments.

Objectives

- Understanding the basis results in state-space analysis of LTI systems
- Learn fundamental control design architectures
- Understand the role of optimisation in controller synthesis
- Be exposed to research challenges and modern applications in control engineering

Contents

- Introduction and outline of the course. Basic maths primer, some linear algebra and ODEs. Modelling, simulation and linearization. Link to data, system ID. Illustrate with inverted pendulum or simple aircraft model. Examples and exercises.
- Introduce LTI systems. Matrix exponential, SVD, analytic solutions for LTI systems. Observability, controllability. Observer design (possibly Kalman Filter). Examples and exercises.
- Control synthesis. Define the control problem. Pole placement. State feedback, output feedback. Optimal control: LQR. Link to MPC. Examples and exercises.
- Linear Matrix Inequalities and the KYP Lemma. Convex optimization primer, duality theory. Linear matrix inequalities, conic programming. KYP lemma. Link to Lyapunov, SOS, barrier functions. Examples and exercises.
- Research frontiers: multi agent systems, decentralized control, synchronization. Applications in internet congestion control. Applications in biology. Examples and exercises.

Prerequisites

- Qualitative theory of ordinary differential equations
- Linear algebra
- Basics in optimisation

Lecture Notes

- Astrom and Murray, Feedback Systems.

Learning from Big Data
Andrew Zisserman and Andrea Vedaldi

Contents

- Discriminative learning
- Energy based models and max-margin learning.
- Advanced SVMs and kernel methods
- Basic stochastic optimization
- Random forests
- Neural networks, backprop, max-margin revisited for transfer learning
- Advanced stochastic optimization
- Convnets and their application to language, vision and speech
- Structured output CRFs and SVMs.

Prerequisites

- MATLAB
- Basic linear algebra

Other Sources

- Christopher M. Bishop, "Pattern Recognition and Machine Learning" , Springer (2006), ISBN 0-38-731073-8.
- Hastie, Tibshirani, Friedman, "Elements of Statistical Learning", Second Edition, Springer, 2009. Pdf available online.

Exercises

- Practical on large scale classification
- Practical on Convnets and their applications

Hilary Term

Computer Vision
Andrew Zisserman, Andrea Vedaldi and David Murray

Introduction

Computer vision is empowering cutting-edge applications in search, smart sensing, medical imaging, human-machine interaction, and many other areas. In this course the students will be introduced to the fundamental theory and practice of this rapidly evolving technology and will learn the fundamentals required to make use of it in their own research projects.

Objectives

- To introduce state-of-the-art methods for computer vision
- To familiarise the student with the theory and practice of image matching and indexing
- To reveal the geometry underpinning the formation of images from observations of the 3D world
- To supply computer vision software that can be used in subsequent research

Contents

- Object recognition
    - Image transformations and matching
    - Image indexing and search
    - Sliding-window object detectors
- Multi-view geometry
    - Camera models
    - Image correspondences
    - Triangulation
    - 3D reconstruction of motion and structure
- Differential motion
    - optical flow
    - object tracking
- Segmentation
    - edges
    - superpixels

Prerequisites

- MATLAB
- Basic linear algebra

Other Sources

- Computer Vision: A Modern Approach by David Forsyth and Jean Ponce (2nd ed.)
- Computer Vision: Algorithms and Applications by Richard Szeliski (PDF available online)

- Multiple View Geometry in Computer Vision by Richard Hartley and Andrew Zisserman

Exercises

- Practical on image matching and indexing
- Practical on sliding window object detection
- Practical on multi-view geometry

Systems Verification
Daniel Kroening, Marta Kwiatkowska and Alessandro Abate

Introduction

In February 2014, Toyota has recalled 1.9 million of Prius cars due to a programming glitch in their hybrid engine system. In limited cases, the engine could shut down, causing the vehicle to stop, possibly while it is being driven. Although no incidents have been reported in this case, such design flaws can have catastrophic consequences that can be prevented by employing formal verification. The aim of systems verification is to automatically check whether a given property, for example "the car will eventually stop when the brake pedal is activated" or "the probability that the airbag fails to deploy is tolerably small", holds for a system model, and if not provide a counterexample in the form of a trace to error. With emphasis on safety-critical systems, it can further provide correct-by-design synthesis of control actions (decisions) on the model under study. This module will give an introductory overview of the main verification techniques, including quantitative verification, which can be used to ensure the safety, reliability and efficiency of autonomous systems, supported by practical exercises and informed by case studies.

Objectives

- To familiarise with the main concepts in systems modelling and property specification notations.
- To explain the fundamentals of explicit and symbolic algorithms in qualitative and quantitative verification.
- To gain practical experience of verification tools and how they are applied through examples of autonomous systems.
- To give appreciation of relevant research topics in systems verification and synthesis.

Contents

- Basics of verification: transition system models, temporal logics CTL and LTL, examples of system models and quantitative requirements. Fundamentals of algorithmic verification (aka model checking).
- Basic idea of propositional satisfiability (SAT) checking, SMT solving. Bounded model checking (BMC). Unbounded verification with SAT/SMT and inductive reasoning. Application to software.
- Introduction to quantitative verification: Markov chain models, time and rewards, probabilistic temporal logics. Overview of algorithms, including statistical model checking. Examples.
- Quantitative verification for systems with non-determinism/decisions: Markov decision processes, issues of time and rewards, generalisation of the probabilistic temporal logic. Stochastic games if time allows. Main algorithms for verification and strategy/controller synthesis. Examples.
- Hybrid systems. From discrete to continuous models: syntax and semantics. The role of stochasticity. Classical analysis of hybrid systems (stability, reachability). Formal abstractions for automated verification and symbolic controller synthesis. Software tools for verification and synthesis.

Prerequisites

- Prior familiarity with logic, probability and dynamical systems (the latter two topics are covered in MT modules).

Other Sources

- Principles of Model Checking, Christel Baier and Joost-Pieter Katoen, MIT Press, 2008.
- Decision procedures, Daniel Kroening and Ofer Strichman, Springer, 2008.
- Stochastic Model Checking (www.prismmodelchecker.org/bibitem.php?key=KNP07a), Marta Kwiatkowska, Gethin Norman and David Parker.
- Automated Verification Techniques for Probabilistic Systems (www.prismmodelchecker.org/bibitem.php?key=FKNP11), Vojtěch Forejt, Marta Kwiatkowska, Gethin Norman and David Parker.
- Model Checking for Probabilistic Timed Automata (http://www.prismmodelchecker.org/bibitem.php?key=NPS13). Gethin Norman, David Parker and Jeremy Sproston. Formal Methods in System Design, 43(2), pages 164-190, Springer. September 2013
- P. Tabuada, Verification and Control of Hybrid Systems, A symbolic approach, Springer, 2009.
- www.prismmodelchecker.org
- http://sourceforge.net/projects/faust2
- https://sites.google.com/a/cyphylab.ee.ucla.edu/pessoa/home

Exercises

- Problem sheets and practical exercises to be completed during lab sessions in the afternoon.

Security in Wireless and Mobile Networks
Ivan Martinovic

Introduction

The purpose of this course is to familiarise students with threats, vulnerabilities, and countermeasures of the existing and upcoming wireless technologies. The covered security topics include an introduction to the main cryptographic concepts (such as symmetric and asymmetric cryptographic primitives and protocols) and their application in securing wireless and mobile networks and systems.

Objectives

The successful participant will

- understand the main security principles, cryptographic primitives, and threat models;
- be able to reason about security protocols and protection techniques, discuss proposed solutions and their limitations;
- have an overview of the recent advances regarding wireless security research, including lightweight authentication, key management for wireless networks, secure localization and location verification, PHY layer attacks, etc.

Contents

- Security principles
  - General security objectives: authentication, confidentiality, integrity, availability
  - Principles of the symmetric-key and public-key cryptography
- Security protocols
  - Key distribution, key management, PKI
  - Analysis of some concrete cryptographic protocols
  - Performance vs. security tradeoffs
- Security in wireless networks and systems
  - Security challenges and threats
  - Architectures and protocols: 2g/3g cellular radio networks, WiFi networks, NFC/RFIDs, sensor networks, etc.
- Advanced wireless security topics
  - Attacks on wireless communication
  - Jamming/Anti-jamming techniques
  - Secure localisation and positioning, distance bounding, multilateration

Prerequisites: none.

Other Sources: will be provided.

Exercises: will be provided.

Computational Game Theory
Michael Wooldridge

Introduction

Game theory is the mathematical theory of strategic interactions between self-interested agents. Game theory provides a range of models for representing strategic interactions, and associated with these, a family of solution concepts, which attempt to characterise the rational outcomes of games. Game theory is important to computer science for several reasons: First, interaction is a fundamental topic in computer science, and if it is assumed that system components are self-interested, then the models and solution concepts of game theory seems to provide an appropriate framework with which to model such systems. Second, the problem of computing with the solution concepts proposed by game theory raises important challenges for computer science, which test the boundaries of current algorithmic techniques. This course aims to introduce the key concepts of game theory for a computer science audience, emphasising both the applicability of game theoretic concepts in a computational setting, and the role of computation in game theoretic problems.

Objectives

The aims of this module are threefold:

1. to introduce the key models and solution concepts of non-cooperative and cooperative game theory;
2. to introduce the issues that arise when computing with game theoretic solution concepts, and the main approaches to overcoming these issues, and to illustrate the role that computation plays in game theory;
3. to introduce a research-level topic in computational game theory.

Contents

Upon completing this module, a student will:

1. understand the key concepts of preferences, utility, and decision-making under certainty and uncertainty, and the key computational issues in representing and manipulating representations of preferences and utility;
2. understand and be able to apply the key models and solution concepts of non-cooperative game theory, including both strategic form and extensive form games, and the key computational issues that arise when applying these models;
3. understand and be able to apply the key models and solution concepts of cooperative game theory, including TU and NTU games, and the
4. understand a contemporary research-level topic at the intersection between game theory and computer science.

Prerequisites

The course assumes no prior knowledge of game theory.

Outline Syllabus

1. Preferences, Utility, and Goals:

- Preference relations and their interpretation; utility as a numeric model of preference.
- Decision-making under uncertainty: preferences over lotteries; von Neumann and Morgenstern utility functions; expected utility and expected utility maximisation.
- Paradoxes of expected utility maximisation; framing effects and prospect theory.
- Compact representations for preference relations (e.g., CP-NETS).
- Dichotomous preferences and goals. Representations for specifying goals (e.g., weighted formula representations for combinatorial domains); expressiveness and computational issues.

2. *Strategic Form Non-Cooperative Games:*

- The basic model; solution concepts: pure strategy Nash equilibrium; dominant strategies; notable games (e.g., Prisoner's Dilemma; Game of Chicken; Stag Hunt); coordination games and focal points; complexity of pure strategy Nash equilibrium.
- Measuring social welfare; utilitarian social welfare; egalitarian social welfare.
- Mixed strategies; Nash's theorem; c-Nash equilibrium.
- Computing mixed strategy Nash equilibria: the Lemke-Howson algorithm.
- Zero sum games; the Minimax Theorem.
- Compact representations for strategic form games; Boolean games; congestion games.

3. Iterated Games:

- Finitely repeated games and backward induction.
- Infinitely repeated games; measuing utility over infinite plays; modelling strategies as finite state machines with output (Moore machines); the folk theorems; implications of using bounded automata to model strategies.
- Iterated Boolean games.
- Axelrod's tournament; the Hawk-Dove game; evolutionary game theory; evolutionarily stable strategies.

4. Extensive Form Non-Cooperative Games:

- Extensive form games of perfect information; Zermelo's algorithm and backward induction; P-completeness of Zermelo's algorithm; subgame perfect equilibrium.
- Win-lose games; Zermelo's theorem.
- Compact representations for extensive form games; the PEEK games and EXPTIME-completeness results; the Game Description Language (GDL).
- Imperfect information games; information sets; solution concepts for imperfect information games.
- Compact representations for imperfect information games; PEEK games with incomplete information; undecidability results.

5. Cooperative Games:

- Transferable utility (TU) characteristic function games; the basic model; stability & fairness solution concepts: the core; the kernel; the Nucleolus; the cost of stability; the Shapley value; the Banzhaf index.
- Compact representations for TU games; induced subgraph representation; marginal contribution nets.
- Simple TU games; swap and trade robustness; weighted voting games; vector weighted voting games; network flow games.
- NTU games and representations for them; hedonic games.
- Coalition structure formation; exact and approximation algorithms.

Lecture Notes

See: http://www.cs.ox.ac.uk/people/michael.wooldridge/pubs/imas/resources.html

Other Sources

Recommended Reading

- Michael Maschler, Eilon Solan, Shmuel Zamir. *Game Theory,* Cambridge UP, 2013.

The best contemporary overview of game theory.

- Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory.* MIT Press, 1994.

An excellent introduction to game theory, freely available from:

http://books.osborne.economics.utoronto.ca

- Y. Shoham and K. Leyton-Brown. *Multiagent Systems*. Cambridge UP, 2009.

Freely available from:

http://www.masfoundations.org/

- G. Chalkiadakis, E. Elkind, and M. Wooldridge. *Computational Aspects of Cooperative Game Theory*. Morgan & Claypool, 2011.

The book for cooperative games.

Sensor and Actuator Networks
Niki Trigoni, Andrew Markham and Alex Rodgers

Introduction

Recent years have witnessed the advent of wireless mobile and sensor technologies and the proliferation of application scenarios whereby large numbers of pervasive computing devices are connected to a wireless networking infrastructure. This course will cover communication, localisation and coordination protocols for sensor and actuator networks.

Objectives

To identify applications and challenges of emerging networked environments
To explore communication, localisation and coordination protocols for such networks
To experiment with different types of sensors, actuators, and networks

Contents

- Introduction to applications of sensor and actuator networks, and the Internet of Things
- Overview of communication (physical, medium access control and routing) protocols
- Indoor / underground positioning systems (WiFi-based, IMU-based, magnetic-based)
- Data management and coordination protocols for fixed / mobile / delay-tolerant networks

Prerequisites
- Embedded systems programming
- Data estimation and inference

Mobile Robotics
Paul Newman and Ingmar Posner

Introduction

In the future, autonomous vehicles will play an important part in our lives. They will come in a variety of shapes and sizes and undertake a diverse set of tasks on our behalf. We want smart vehicles to carry, transport, labour for and defend us. We want them to be flexible, reliable and safe. This course will to introduce you to the key components of mobile robotics in a very real sense (you will end up in field with one)

Objectives

- To understand the interplay between localisation, planning and control in an embedded system and to execute the design of such a system
- To understand and experience the pros and cons of infrastructure dependent and infrastructure free localisation
- To gain practical experience in safely operating autonomous vehicles in the field and debugging errant behaviour
- To understand the theoretical underpinnings of planning-by-search, visual odometry and visual localisation, lidar based navigation and obstacle detection

Prerequisites

- Ability to program in C++
- CDT-gleaned knowledge on computer vision and estimation

Lecture Notes and Teaching Methods

- each day will begin with a brief lecture before carrying on with the practical
- the practical will be supported by 5 members of the robotics research group to ensure you have the attention and problem solving resource you will need.

Other Sources

The mobile robotics group will provide you with a working robot (a Husky or Kuku Youbot) and some key software and hardware components for example a visual odometry system, a low level controller, a communications manager and a visualisation tool kit, a dense stereo library and a location sensing device.

Exercises

The course will be built around a single extensive practical (long days of great enjoyment). You will be split into two teams, each given a mobile robot (100kg) and at the end of the week you will compete against each other in a mobile robot challenge (for example finding an object in the University Parks).

Assessment Mode

You will give a presentation of your results and critique of your system to the mobile robotics research group on the last day of the week. Each team member will be asked to write a 3 page report on their contribution and evaluate the overall performance of their implementation.