



Plan d'études
DATA SCIENCE
2021 - 2022

arrêté par la direction de l'EPFL le 26 mai 2021

Directeur de section	Prof. S. Vaudenay
Adjointe de la section	Mme E. Hazboun
Conseillers d'études : 1ère année cycle master 2ème année cycle master Projet de Master	Prof. R. Guerraoui vacat Prof. P. Thiran
Coordination des stages en industrie	Mme E. Hazboun
Secrétariat de la section	Mme C. Dauphin

Aux cycles bachelor et master, selon les besoins pédagogiques, les heures d'exercices mentionnées dans le plan d'études pourront être intégrées dans les heures de cours ; les scolarités indiquées représentent les nombres moyens d'heures de cours et d'exercices hebdomadaires sur le semestre.

Code	Matières	Enseignants sous réserve de modification	Sections	Semestres						Crédits	Période des épreuves	Type examen	
				MA1			MA2						
				c	e	p	c	e	p				
	Groupe "Core courses et options"										72		
	Groupe 1 "Core courses"										min. 30		
CS-450	Advanced Algorithms	Kapralov	IN				4	3			7	E	écrit
CS-401	Applied data analysis	West	IN	2	2						6	H	écrit
COM-402	Information security and privacy	Hubaux/Pyrgelis	SC	3	1	2					6	H	écrit
COM-406	Foundations of Data Science	Gastpar/Urbanke	SC	4	2						6	H	écrit
CS-433	Machine learning	Jaggi/Flammarion	IN	4	2						7	H	écrit
CS-439	Optimization for Machine Learning	Jaggi/Flammarion	IN				2	2	1		5	E	écrit
MATH-413	Statistics for Data Science	Olhede	MA	4	2						6	H	écrit
CS-449	Systems for Data Science	Kermarrec	IN				2	2	2		6	sem P	
	Groupe 2 "Options"	(la somme des crédits des groupes 1 et 2 doit être de 72 crédits au minimum)											
---	Cours à option	Divers enseignants	Divers										
	Bloc "Projets et SHS" :										18		
COM-412	Semester project in Data Science	divers enseignants	SC	← 2 →						12	sem A ou P		
HUM-nnn	SHS : introduction au projet	divers enseignants	SHS	2		1					3	sem A	
HUM-nnn	SHS : projet	divers enseignants	SHS							3	3	sem P	
	Total des crédits du cycle master										90		

Stage d'ingénieur :

Voir les modalités dans le règlement d'application

Mineurs :

Le cursus peut être complété par un des mineurs figurant dans l'offre de l'EPFL (renseignements à la page sac.epfl.ch/mineurs), à l'exclusion des mineurs "Data Science", "Informatique" et "Systèmes de communication" qui ne peuvent pas être choisis.

Parmi les mineurs offerts par l'EPFL, la section recommande à ses étudiants les mineurs suivants :

- Biocomputing (SIN)
- Computational Science and Engineering (SMA)
- Management de la technologie et entrepreneuriat (SMTE)
- Technologies biomédicales (SMT)
- Technologies spatiales (SEL)

Le choix des cours de tous les mineurs se fait sur conseil de la section de l'étudiant et du responsable du mineur.

Code	Matières	Enseignants sous réserve de modification	Sections	Semestres						Crédits	Nbre places	Période des épreuves	Type examen	Cours biennaux donnés en
				MA1			MA2							
				c	e	p	c	e	p					
COM-501	Advanced cryptography	Vaudenay	SC				2	2			4		E	écrit
COM-417	Advanced probability and applications	Lévêque	SC				3	2			6		E	écrit
CS-523	Advanced topics on privacy enhancing technologies	Troncoso	IN				3	1	2		7		E	écrit
MATH-493	Applied biostatistics	Goldstein	MA				2	2			5		sem P	
CS-456	Artificial neural networks	Gerstner	IN				2	2			5		E	écrit
EE-554	Automatic speech processing	Magimai Doss	EL	2	1						3		H	écrit
BIO-465	Biological modeling of neural networks	Gerstner	IN				2	2			4		E	écrit
MATH-453	Computational linear algebra	Kressner	MA				2	2			5		E	oral
CS-524	Computational complexity	Göös	IN	3	1						4		sem A	
CS-413	Computational Photography	Süsstrunk	SC				2		2		5		sem P	
COM-418	Computers and Music	Prandoni P.	SC				2	1			4		E	écrit
CS-442	Computer vision	Fua	IN				2	1			4		E	écrit
CS-453	Concurrent algorithms	Guerraoui	SC	3	1	1					5		H	écrit
COM-401	Cryptography and security	Vaudenay	SC	4	2						7		H	écrit
COM-480	Data visualization	Vuillon	SC				2		2		4		sem P	
EE-559	Deep learning	Fleuret	EL				2	2			4	500	E	écrit sans retrait
CS-411	Digital education & learning analytics	Dillenbourg/Jermann	IN	2		2					4		H	oral
CS-451	Distributed algorithms	Guerraoui	SC	3	2	1					6		H	écrit
CS-423	Distributed information systems	Aberer	SC	2	1						4		H	écrit
ENG-466	Distributed intelligent systems	Martinoli	SIE				2	3			5		E	oral
CS-550	Formal verification	Kuncak	IN	2	2	2					6		sem A	
CS-457	Geometric Computing	Pauly	IN	3		2					6		sem A	
MATH-360	Graph Theory	Maffucci	MA	2	2						5		H	écrit
CS-486	Interaction design	Pu	IN				2	1	1		4		sem P	
EE-451	Image analysis and pattern recognition	Thiran J.-P.	EL				2		2		4		sem P	
COM-404	Information theory and coding	Telatar	SC	4	2						7		H	écrit
CS-430	Intelligent agents (pas donné en 2021-2022)	Faltings	IN	3	3						6		sem A	
CS-431	Introduction to natural language processing	Chappelier/Rajman	IN	2	2						4		H	écrit
COM-490	Large-scale data science for real-world data	Bouillet/Sarni/Verscheure/Delgado	SC						4		4		sem P	sans retrait
CS-526	Learning theory	Macris/Urbanke	SC				2	2			4		E	écrit
MATH-341	Linear models	Panaretos	MA	2	2						5		H	écrit
CS-421	Machine learning for behavioral data	Käser	IN				2		2		4		E	écrit
COM-516	Markov chains and algorithmic applications	Lévêque/Macris	SC	2	2						4		H	écrit
COM-514	Mathematical foundations of signal processing	Simeoni/Behar Haro/Fageot	SC	3	2						6		H	écrit
EE-556	Mathematics of data: from theory to computation	Cevher	EL	3		3					6		H	écrit
COM-512	Networks out of control	Thiran P./Grossglauser	SC				2	1			4		E	écrit
COM-508	Optional project in Data Science	Divers enseignants	SC	← 2 →						8		sem A ou P		
MATH-447	Risk, rare events and extremes	Davison	MA	2	2						5		H	écrit
MATH-441	Robust and nonparametric statistics (pas donné en 2021-22)	Morgenthaler	MA				2	2			5		E	oral
CS-412	Software security	Payer	IN				3	2	1		6		sem P	
MATH-486	Statistical mechanics and Gibbs measures	Friedli	MA				2	2			5		E	oral
MATH-442	Statistical Theory	Koch	MA	2	2						5		H	écrit
COM-506	Student seminar: security protocols and applications	Vaudenay	SC				2				3		E	écrit
CS-448	Sublinear algorithms for big data analysis	Kapralov	IN	2	1						4		sem A	2021-2022
CS-410	Technology ventures in IC (pas donné en 2021-22)	Bugnion	IN				2		2		4		sem P	
CS-458	The GC Maker Project (pas donné en 2021-22)	Pauly	IN							6	6		sem P	
MATH-342	Time Series	Olhede	MA				2	2			5		E	écrit
CS-455	Topics in theoretical computer science (pas donné en 2021-22)	Kapralov	IN	3	1						4		sem A	2022-2023
CS-444	Virtual reality	Boulic	IN				2	1			4		sem P	
CS-503	Visual Intelligence : Machines and Minds	Zamir	IN	2	2						5		sem A	

**RÈGLEMENT D'APPLICATION DU CONTRÔLE DES
ÉTUDES DE LA SECTION DE SYSTÈMES DE
COMMUNICATION POUR LE MASTER EN DATA
SCIENCE pour l'année académique 2021-2022
du 26 mai 2021**

La direction de l'École polytechnique fédérale de Lausanne

vu l'ordonnance sur la formation menant au bachelor et au master de l'EPFL du 14 juin 2004,
vu l'ordonnance sur le contrôle des études menant au bachelor et au master à l'EPFL du 30 juin 2015,
vu le plan d'études de la section de systèmes de communication pour le master en Data Science.

arrête:

Article premier - Champ d'application

Le présent règlement fixe les règles d'application du contrôle des études de master de la section de systèmes de communication pour le master en Data Science qui se rapportent à l'année académique 2021-2022.

Art. 2 – Étapes de formation

Le master en Data Science est composé de deux étapes successives de formation :

- le cycle master d'une durée de 3 semestres dont la réussite implique l'acquisition de 90 crédits, condition pour effectuer le projet de master.
- le projet de master, d'une durée de 17 semaines à l'EPFL ou de 25 semaines hors EPFL (industrie ou autre haute école) et dont la réussite se traduit par l'acquisition de 30 crédits. Il est placé sous la responsabilité d'un professeur ou MER affilié à la section de systèmes de communication ou d'informatique.

Art 3 – Sessions d'examen

1. Les branches d'examen sont examinées par écrit ou par oral pendant les sessions d'hiver ou d'été. Elles sont mentionnées dans le plan d'études avec la mention H ou E.
2. Les branches de semestre sont examinées pendant le semestre d'automne ou le semestre de printemps. Elles sont mentionnées dans le plan d'études avec la mention sem A ou sem P.
3. Une branche annuelle, c'est à dire dont l'intitulé tient sur une seule ligne dans le plan d'étude, est examinée globalement pendant la session d'été (E).
4. Pour les branches de session, la forme écrite ou orale de l'examen indiquée pour la session peut être complétée par des contrôles de connaissances écrits ou oraux durant le semestre, selon indications de l'enseignant.

Art. 3 – Prérequis

Certains enseignements peuvent exiger des prérequis qui sont mentionnés dans la fiche de cours concerné. Le cours prérequis est validé si les crédits correspondants ont été acquis pour le cours ou par moyenne du bloc.

Art. 4 – Conditions d'admission

1. Les étudiants issus du Bachelor en Informatique ou en Systèmes de communications sont admis automatiquement.
2. Les étudiants issus du Bachelor en Informatique ou en Systèmes de communication qui n'auront pas suivi les cours prérequis durant leur cycle Bachelor devront les prendre en parallèle à leur cycle Master.
3. Pour les autres étudiants, l'admission s'effectue sur dossier.

Art. 5 - Organisation

1. Les enseignements du cycle master sont répartis en deux groupes et un bloc dont les crédits doivent être obtenus de façon indépendante.
2. Le bloc « Projets et SHS » est composé d'un projet de 12 crédits et de l'enseignement SHS.
3. Le groupe 1 « Core courses » est composé des cours de la liste du plan d'études dans la rubrique « Master ».
4. Le groupe 2 « Options » est composé
 - des cours de la liste du groupe 2 « options » du plan d'études dans la rubrique « Master » ;
 - des crédits surnuméraires obtenus dans le groupe 1 « Core courses » ;
 - d'un projet optionnel de 8 crédits ;
 - de cours hors plan d'études suivant l'alinéa 6.
5. Le projet du bloc « Projets et SHS » et le projet optionnel du groupe 2 ne peuvent être effectués dans le même semestre.
6. Des cours, comptant pour un maximum de 15 crédits au total, peuvent être choisis en dehors de la liste des cours du plan d'études dans la rubrique « Master ». Le choix de ces cours doit être accepté préalablement par le directeur de la section qui peut augmenter le maximum de 15 crédits si la demande est justifiée.

Art. 6 - Examen du cycle master

1. Le bloc « Projets et SHS » est réussi lorsque **18 crédits** sont obtenus.
2. Le groupe « Core courses et Options », composé du groupe 1 « Core courses » et du groupe 2 « Options » est réussi lorsque **72 crédits** sont obtenus.
3. Le groupe 1 « Core courses » est réussi lorsqu'**au moins 30 crédits** sont obtenus.

Art. 7 - Enseignement SHS

Les deux branches SHS donnent chacune lieu à 3 crédits. L'enseignement du semestre d'automne introduit à la réalisation du projet du semestre de printemps. Pour autant qu'il considère que le motif est justifié, le Collège des Humanités peut déroger à cette organisation. Il peut également autoriser à ce qu'un étudiant réalise son projet sur un semestre

qui ne suit pas immédiatement celui dans lequel a lieu l'enseignement d'introduction.

Art. 8 – Mineurs

1. Afin d'approfondir un aspect particulier de sa formation ou de développer des interfaces avec d'autres sections, l'étudiant peut choisir la formation offerte dans le cadre d'un mineur figurant dans l'offre de l'EPFL.
2. Le choix des cours qui composent un mineur se fait avec la section de systèmes de communication et avec le responsable du mineur. Les mineurs « Data Science » « Informatique » et « Systèmes de Communication » ne peuvent pas être choisis.
3. L'étudiant annonce le choix d'un mineur à sa section au plus tard à la fin du premier semestre des études de master.
4. Un mineur est réussi quand 30 crédits au minimum sont obtenus parmi les branches avalisées.

Art. 8 – Stage d'ingénieur

1. Les étudiants commençant leur cycle master doivent effectuer un stage d'ingénieur durant leur master :
 - soit un stage d'été de minimum 8 semaines
 - soit un stage de minimum 6 mois en entreprise (en statut stage durant un semestre). Durant la période du COVID-19, la durée du stage peut être adaptée.
 - soit un Projet de Master de 25 semaines en entreprise (valide le stage et le Projet de Master)
2. En règle générale, pour les étudiants issus du Bachelor IC, le stage peut être effectué dès le 2^{ème} semestre du cycle master, mais avant le projet de master. Sur demande de l'étudiant, la section peut l'autoriser à effectuer son stage avant ou pendant le 1^{er} semestre du cycle Master.
3. L'étudiant ne peut pas faire de cours/projet en parallèle à son stage.
4. Le responsable du stage de la section évalue le stage, par l'appréciation « réussi » ou « non réussi ». Sa réussite est une condition pour l'admission au projet de master. En cas de non réussite, il peut être répété une fois, en règle générale dans une autre entreprise.
5. Il est validé avec les 30 crédits du projet de master.
6. Les modalités d'organisation et les critères de validation du stage font l'objet d'une directive interne à la section.

Art. 9 – Spécialisation Enseignement

1. Les étudiants en Master Data Science ont la possibilité de suivre une spécialisation en informatique pour l'enseignement.
2. L'étudiant admis à cette spécialisation ne peut pas suivre de mineur. Le plan d'études est modifié comme suit : (i) Un

nouveau groupe de 30 ECTS de cours à la HEP Vaud est rajouté et le nombre de ECTS du Cycle Master passe de 60 à 30 ECTS ; (ii) les cours SHS sont remplacés par un cours à la HEP Vaud ; (iii) le Projet de Master peut s'étaler sur deux semestres et commencer après que l'étudiant a complété le bloc « Projets et SHS » et le groupe « Core courses » ; (iv) la durée maximale des études ne peut pas dépasser 8 semestres.

3. Au moins 50 ECTS doivent avoir été obtenus pour débiter la spécialisation.

Art. 10 – Procédure d'admission

1. L'admission à cette spécialisation n'est pas automatique. Pour être admis à la spécialisation, le candidat doit être inscrit au Master en Data Science de l'EPFL et répondre aux conditions pour l'admission au Diplôme d'enseignement pour le degré secondaire II fixées par le Règlement d'application de la loi sur la HEP du 3 juin 2009 (RLHEP).
2. L'étudiant s'inscrit auprès de la HEP Vaud selon les conditions et délais de la candidature en ligne et transmet les pièces requises par le RLHEP ainsi qu'une attestation d'immatriculation à l'EPFL.

Au nom de la direction de l'EPFL

Le président, M. Vetterli

Le vice-président académique, J. S. Hesthaven

Lausanne, le 26 mai 2021

CS-450

Advanced algorithms

Kapralov Mikhail

Cursus	Sem.	Type
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Obl.
Data Science	MA2, MA4	Obl.
Informatique et communications		Opt.
Informatique	MA2, MA4	Obl.
Mineur en Data science	E	Opt.
Mineur en Informatique	E	Opt.
Robotique, contrôle et systèmes intelligents		Opt.
SC master EPFL	MA2, MA4	Obl.
Science et ing. computationnelles	MA2, MA4	Opt.

Language	English
Credits	7
Session	Summer
Semester	Spring
Exam	Written
Workload	210h
Weeks	14
Hours	7 weekly
Lecture	4 weekly
Exercises	3 weekly
Number of positions	

Summary

A first graduate course in algorithms, this course assumes minimal background, but moves rapidly. The objective is to learn the main techniques of algorithm analysis and design, while building a repertory of basic algorithmic solutions to problems in many domains.

Content

Algorithm analysis techniques: worst-case and amortized, average-case, randomized, competitive, approximation. Basic algorithm design techniques: greedy, iterative, incremental, divide-and-conquer, dynamic programming, randomization, linear programming. Examples from graph theory, linear algebra, geometry, operations research, and finance.

Keywords

See content.

Learning Prerequisites**Required courses**

An undergraduate course in Discrete Structures / Discrete Mathematics, covering formal notation (sets, propositional logic, quantifiers), proof methods (derivation, contradiction, induction), enumeration of choices and other basic combinatorial techniques, graphs and simple results on graphs (cycles, paths, spanning trees, cliques, coloring, etc.).

Recommended courses

An undergraduate course in Data Structures and Algorithms.
An undergraduate course in Probability and Statistics.

Important concepts to start the course

Basic data structures (arrays, lists, stacks, queues, trees) and algorithms (binary search; sorting; graph connectivity); basic discrete mathematics (proof methods, induction, enumeration and counting, graphs); elementary probability and statistics (random variables, distributions, independence, conditional probabilities); data abstraction.

Learning Outcomes

By the end of the course, the student must be able to:

- Use a suitable analysis method for any given algorithm
- Prove correctness and running-time bounds
- Design new algorithms for variations of problems studied in class
- Select appropriately an algorithmic paradigm for the problem at hand
- Define formally an algorithmic problem

Teaching methods

Ex cathedra lecture, reading

Assessment methods

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes
Others	For details, see the course web page.

Resources

Bibliography

See web page for the course.

Ressources en bibliothèque

- [Randomized Algorithms / Motwani](#)
- [Approximation Algorithms / Vazirani](#)
- [Quantum Computation and Quantum Information / Nielsen](#)
- [Algebraic Complexity Theory / Buegisser](#)
- [Computational Complexity / Papadimitrou](#)

Notes/Handbook

Class notes and references for the running semester will be provided as needed within a few days after each lecture.

Websites

- <http://theory.epfl.ch/courses/AdvAlg/>

COM-501

Advanced cryptography

Vaudenay Serge

Cursus	Sem.	Type
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Mineur en Data science	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

This course reviews some failure cases in public-key cryptography. It introduces some cryptanalysis techniques. It also presents fundamentals in cryptography such as interactive proofs. Finally, it presents some techniques to validate the security of cryptographic primitives.

Content

1. **The cryptographic zoo:** definitions, cryptographic primitives, math, algorithms, complexity
2. **Cryptographic security models:** security notions for encryption and authentication, game reduction techniques, RSA and Diffie-Hellman security notions
3. **Public-key cryptanalysis:** side channels, low RSA exponents, discrete logarithm, ElGamal signature
4. **Interactive proofs:** NP-completeness, interactive systems, zero-knowledge
5. **Symmetric-key cryptanalysis:** differential and linear cryptanalysis, hypothesis testing, decorrelation
6. **Proof techniques:** random oracles, leftover-hash lemma, Fujisaki-Okamoto transform

Keywords

cryptography, cryptanalysis, interactive proof, security proof

Learning Prerequisites**Required courses**

- Cryptography and security (COM-401)

Important concepts to start the course

- Cryptography
- Mathematical reasoning
- Number theory and probability theory
- Algorithmics
- Complexity

Learning Outcomes

By the end of the course, the student must be able to:

- Assess / Evaluate the security deployed by cryptographic schemes

- Prove or disprove security
- Justify the elements of cryptographic schemes
- Analyze cryptographic schemes
- Implement attack methods
- Model security notions

Teaching methods

ex-cathedra

Expected student activities

- active participation during the course
- take notes during the course
- do the exercises during the exercise sessions
- complete the regular tests and homework
- read the material from the course
- self-train using the provided material
- do the midterm exam and final exam

Assessment methods

Mandatory continuous evaluation:

- homework (30%)
- regular graded tests (30%)
- midterm exam (40%)

Final exam averaged (same weight) with the continuous evaluation, but with final grade between final_exam-1 and final_exam+1.

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	Lecturers and assistants are available upon appointment.

Resources

Bibliography

- Communication security: an introduction to cryptography. Serge Vaudenay. Springer 2004.
- A computational introduction to number theory and algebra. Victor Shoup. Cambridge University Press 2005.
- Algorithmic cryptanalysis. Antoine Joux. CRC 2009.

Ressources en bibliothèque

- [Algorithmic cryptanalysis / Joux](#)
- [A computational introduction to number theory and algebra / Shoup](#)
- [Communication security / Vaudenay](#)

Websites

- <http://lasec.epfl.ch/teaching.shtml>

Moodle Link

- <https://moodle.epfl.ch/course/view.php?id=13913>

COM-417

Advanced probability and applications

Lévêque Olivier

Cursus	Sem.	Type
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Génie électrique		Opt.
Informatique et communications		Obl.
Informatique	MA2, MA4	Opt.
Mineur en Data science	E	Opt.
Robotique, contrôle et systèmes intelligents		Opt.
SC master EPFL	MA2, MA4	Obl.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	5 weekly
Lecture	3 weekly
Exercises	2 weekly
Number of positions	

Summary

In this course, various aspects of probability theory are considered. The first part is devoted to the main theorems in the field (law of large numbers, central limit theorem, concentration inequalities), while the second part focuses on the theory of martingales in discrete time.

Content

- sigma-fields, random variables
- probability measures, distributions
- independence, convolution
- expectation, characteristic function
- random vectors and Gaussian random vectors
- inequalities, convergences of sequences of random variables
- laws of large numbers, applications and extensions
- convergence in distribution, central limit theorem and applications
- moments and Carleman's theorem
- concentration inequalities
- conditional expectation
- martingales, stopping times
- martingale convergence theorems

Keywords

probability theory, measure theory, martingales, convergence theorems

Learning Prerequisites**Required courses**

Basic probability course
Calculus courses

Recommended courses

complex analysis

Important concepts to start the course

This course is NOT an introductory course on probability: the students should have a good understanding and practice of basic probability concepts such as: distribution, expectation, variance, independence, conditional probability.

The students should also be at ease with calculus. Complex analysis is a plus, but is not required.

On the other hand, no prior background on measure theory is needed for this course: we will go through the

basic concepts one by one at the beginning.

Learning Outcomes

By the end of the course, the student must be able to:

- understand the main ideas at the heart of probability theory

Teaching methods

Ex cathedra lectures + exercise sessions

Expected student activities

active participation to exercise sessions

Assessment methods

graded homeworks 20%

midterm 20%

final exam 60%

Resources

Bibliography

Sheldon M. Ross, Erol A. Pekoz, A Second Course in Probability, 1st edition, www.ProbabilityBookstore.com, 2007.

Jeffrey S. Rosenthal, A First Look at Rigorous Probability Theory, 2nd edition, World Scientific, 2006.

Geoffrey R. Grimmett, David R. Stirzaker, Probability and Random Processes, 3rd edition, Oxford University Press, 2001.

Richard Durrett, Probability: Theory and Examples, 4th edition, Cambridge University Press, 2010.

Patrick Billingsley, Probability and Measure, 3rd edition, Wiley, 1995.

Ressources en bibliothèque

- [Sheldon M. Ross, Erol A. Pekoz, A Second Course in Probability, 1st ed](#)
- [Patrick Billingsley, Probability and Measure, 3rd ed](#)
- [Richard Durrett, Probability: Theory and Examples, 4th ed](#)
- [Jeffrey S. Rosenthal, A First Look at Rigorous Probability Theory, 2nd ed](#)
- [Probability and Random Processes](#)

Notes/Handbook

available on the course website

Websites

- <https://moodle.epfl.ch/course/view.php?id=14557>

Prerequisite for

Advanced classes requiring a good knowledge of probability

CS-523

Advanced topics on privacy enhancing technologies

Troncoso Carmela

Cursus	Sem.	Type
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Informatique et communications		Obl.
Informatique	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	7
Session	Summer
Semester	Spring
Exam	Written
Workload	210h
Weeks	14
Hours	6 weekly
Lecture	3 weekly
Exercises	1 weekly
Project	2 weekly
Number of positions	

Summary

This advanced course will provide students with the knowledge to tackle the design of privacy-preserving ICT systems. Students will learn about existing technologies to protect privacy, and how to evaluate the protection they provide.

Content

The course will cover the following topics:

- Privacy definitions and concepts.
- Privacy-preserving cryptographic solutions: anonymous credentials, zero-knowledge proofs, secure multi-party computation, homomorphic encryption, Private information retrieval (PIR), Oblivious RAM (ORAM)
- Anonymization and data hiding: generalization, differential privacy, etc
- Machine learning and privacy
- Protection of metadata: anonymous communications systems, location privacy, censorship resistance.
- Online tracking and countermeasures
- Privacy engineering: design and evaluation (evaluation metrics and notions)
- Legal aspects of privacy

Keywords

Privacy, anonymity, homomorphic encryption, secure multi-party computation, anonymous credentials, ethics

Learning Prerequisites**Required courses**

COM-402 Information Security and Privacy
COM-301 Computer Security

Recommended courses

COM-401 Cryptography

Important concepts to start the course

Basic programming skills; basics of probabilities and statistics; basics of cryptography

Learning Outcomes

By the end of the course, the student must be able to:

- Select appropriately privacy mechanisms
- Develop privacy technologies
- Assess / Evaluate privacy protection
- Reason about privacy concerns

Teaching methods

Lectures and written exercises to deepen understanding of concepts

Programming-oriented assignments to practice use of privacy technologies

Expected student activities

Participation in the lectures. Active participation is encouraged.

Participation in exercise session and complete the exercises regularly

Completion of programming assignments

Assessment methods

Final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

MATH-493

Applied biostatistics

Goldstein Darlene

Cursus	Sem.	Type
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Génie civil & environnement		Opt.
Informatique	MA2, MA4	Opt.
Ing.-math	MA2, MA4	Opt.
Ingénierie des sciences du vivant	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
Mineur en Data science	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	During the semester
Workload	150h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

This course covers topics in applied biostatistics, with an emphasis on practical aspects of data analysis using R statistical software. Topics include types of studies and their design and analysis, high dimensional data analysis (genetic/genomic) and other topics as time and interest permit.

Content

- Types of studies
- Design and analysis of studies
- R statistical software
- Reproducible research techniques and tools
- Report writing
- Exploratory data analysis
- Linear modeling (regression, anova)
- Generalized linear modeling (logistic, Poisson)
- Survival analysis
- Discrete data analysis
- Meta-analysis
- High dimensional data analysis (genetics/genomics applications)
- Additional topics as time and interest permit

Keywords

Data analysis, reproducible research, statistical methods, R, biostatistical data analysis, statistical data analysis

Learning Prerequisites**Required courses**

This course will be very difficult for students with no previous course or experience with statistics. **Previous experience with R is neither assumed nor required.**

Recommended courses

Undergraduate statistics course

Important concepts to start the course

It is useful to review statistical hypothesis testing.

Learning Outcomes

By the end of the course, the student must be able to:

- Interpret analysis results
- Justify analysis plan
- Plan analysis for a given dataset
- Analyze various types of biostatistical data
- Synthesize analysis into a written report
- Report plan of analysis and results obtained

Transversal skills

- Write a scientific or technical report.
- Assess one's own level of skill acquisition, and plan their on-going learning goals.
- Take feedback (critique) and respond in an appropriate manner.
- Use a work methodology appropriate to the task.

Teaching methods

Lectures and practical exercises using R. Typically, each week covers an analysis method in the lecture and then the corresponding exercise session consists of an R practical showing how to implement the methods using R. In each practical, students use R to carry out analyses of the relevant data type for that week.

Expected student activities

Students are expected to participate in their learning by attending lectures and practical exercise sessions, posing questions, proposing topics of interest, peer reviewing of preliminary reports, and interacting with teaching staff regarding their understanding of course material. In addition, there will be a number of short activities in class aimed at improving English for report writing.

Assessment methods

Evaluation is based on written reports of projects analyzing biostatistical data.

Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

CS-401

Applied data analysis

West Robert

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational Neurosciences minor	H	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Obl.
Data science minor	H	Opt.
Digital Humanities	MA1, MA3	Obl.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Energy Science and Technology	MA1, MA3	Opt.
Financial engineering	MA1, MA3	Opt.
Internet of Things minor	H	Opt.
Learning Sciences		Obl.
Life Sciences Engineering	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.
UNIL - Sciences forensiques	H	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

This course teaches the basic techniques, methodologies, and practical skills required to draw meaningful insights from a variety of data, with the help of the most acclaimed software tools in the data science world: pandas, scikit-learn, Spark, etc.

Content

Thanks to modern software tools that allow to easily process and analyze data at scale, we are now able to extract invaluable insights from the vast amount of data generated daily. As a result, both the business and scientific world are undergoing a revolution which is fueled by one of the most sought after job profiles: the data scientist.

This course covers the fundamental steps of the data science pipeline:

Data wrangling

- Data acquisition (scraping, crawling, parsing, etc.)
- Data manipulation, array programming, dataframes
- The many sources of data problems (and how to fix them): missing data, incorrect data, inconsistent representations
- Data quality testing with crowdsourcing

Data interpretation

- Statistics in practice (distribution fitting, statistical significance, etc.)
- Working with "found data" (design of observational studies, regression analysis)
- Machine learning in practice (supervised and unsupervised, feature engineering, more data vs. advanced algorithms, curse of dimensionality, etc.)
- Text mining: vector space model, topic models, word embedding
- Social network analysis (influencers, community detection, etc.)

Data visualization

- Introduction to different plot types (1, 2, and 3 variables), layout best practices, network and geographical data
- Visualization to diagnose data problems, scaling visualization to large datasets, visualizing uncertain data

Reporting

- Results reporting, infographics
- How to publish reproducible results

The students will learn the techniques during the ex-cathedra lectures and will be introduced, in the lab sessions, to the software tools required to complete the homework assignments and the in-class quizzes.

In parallel, the students will embark on a semester-long project, split in agile teams of 3-4 students. The project consists of two parts: (1) replication of a data analysis pipeline from a published scientific paper, (2) a "free-style" component where students propose and execute their own extension of part 1. The outcome of this team effort will be a project portfolio that will be made public (and available as open source).

At the end of the semester, students will also take a 3-hour final exam in a classroom with their own computer, where they will be asked to complete a data analysis pipeline (both with code and extensive comments) on a dataset they have never worked with before.

Keywords

data science, data analysis, data mining, machine learning

Learning Prerequisites

Required courses

The student must have passed an introduction to databases course, OR a course in probability & statistics, OR two separate courses that include programming projects.

Recommended courses

- CS-423 Distributed Information Systems
- CS-433 Machine Learning

Important concepts to start the course

Algorithms, (object-oriented) programming, basic probability and statistics

Learning Outcomes

By the end of the course, the student must be able to:

- Construct a coherent understanding of the techniques and software tools required to perform the fundamental steps of the data science pipeline
- Perform data acquisition (data formats, dataset fusion, Web scrapers, REST APIs, open data, big data platforms, etc.)
- Perform data wrangling (fixing missing and incorrect data, data reconciliation, data quality assessments, etc.)
- Perform data interpretation (statistics, correlation vs. causality, knowledge extraction, critical thinking, team discussions, ad-hoc visualizations, etc.)
- Perform result dissemination (reporting, visualizations, publishing reproducible results, ethical concerns, etc.)

Transversal skills

- Give feedback (critique) in an appropriate fashion.
- Demonstrate the capacity for critical thinking
- Write a scientific or technical report.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.

Teaching methods

- Physical in-class recitations and lab sessions
- Homework assignments
- In-class quizzes
- Course project

Expected student activities

Students are expected to:

- Attend the lectures and lab sessions
- Complete 2-3 homework assignments
- Complete 3 in-class quizzes (held during lab sessions)
- Conduct the class project
- Read/watch the pertinent material before a lecture
- Engage during the class, and present their results in front of the other colleagues

Assessment methods

- 30% continuous assessment during the semester (homework)
- 30% final exam, data analysis task on a computer (3 hours)
- 25% final project, done in groups of 4
- 15% regular online quizzes

Supervision

Others <http://ada.epfl.ch>

CS-456

Artificial neural networks

Gerstner Wulfram

Cursus	Sem.	Type
Biocomputing minor	E	Opt.
Computational Neurosciences minor	E	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical Engineering		Opt.
Financial engineering	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	Written
Workload	150h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

Since 2010 approaches in deep learning have revolutionized fields as diverse as computer vision, machine learning, or artificial intelligence. This course gives a systematic introduction into the main models of deep artificial neural networks: Supervised Learning and Reinforcement Learning.

Content

- *General Introduction: Deep Networks versus Simple perceptrons*
- *Reinforcement Learning 1: Bellman equation and SARSA*
- *Reinforcement Learning 2: variants of SARSA, Q-learning, n-step-TD learning*
- *Reinforcement Learning 3: Policy gradient*
- *Deep Networks 1: BackProp and Multilayer Perceptrons*
- *Deep Networks 2: Regularization and Tricks of the Trade in deep learning*
- *Deep Networks 3: Error landscape and optimization methods for deep networks*
- *Deep Networks 4: Statistical Classification by deep networks*
- *Deep Networks 5: Convolutional networks*
- *Deep reinforcement learning 1: Exploration*
- *Deep reinforcement learning 2: Actor-Critic networks*
- *Deep reinforcement learning 3: Atari games and robotics*
- *Deep reinforcement learning 4: Board games and planning*
- *Deep reinforcement learning 5: Sequences, recurrent networks, partial observability*

Keywords

Deep learning, artificial neural networks, reinforcement learning, TD learning, SARSA,

Learning Prerequisites

Required courses

CS 433 Machine Learning (or equivalent)

Calculus, Linear Algebra (at the level equivalent to first 2 years of EPFL in STI or IC, such as Computer Science, Physics or Electrical Engineering)

Recommended courses

stochastic processes

optimization

Important concepts to start the course

- *Regularization in machine learning,*
- *Training base versus Test base, cross validation.*
- *Gradient descent. Stochastic gradient descent.*
- *Expectation, Poisson Process, Bernoulli Process.*

Learning Outcomes

- Apply learning in deep networks to real data
- Assess / Evaluate performance of learning algorithms
- Elaborate relations between different mathematical concepts of learning
- Judge limitations of algorithms
- Propose algorithms and models for learning in deep networks
- Apply Reinforcement Learning

Transversal skills

- Continue to work through difficulties or initial failure to find optimal solutions.
- Access and evaluate appropriate sources of information.
- Write a scientific or technical report.
- Manage priorities.

Teaching methods

ex cathedra lectures and miniproject. Every week the ex cathedra lectures are interrupted for at least one in-class exercise which is then discussed in classroom before the lecture continues. Additional exercises are given as homework or can be discussed in the second exercise hour.

Expected student activities

work on miniproject

solve all exercises

attend all lectures and take notes during lecture, participate in quizzes.

If you cannot attend a lecture, then you must read the recommended book chapters

Assessment methods

written exam (70 percent) and miniproject (30 percent)

Supervision

Office hours Yes

Assistants	Yes
Forum	Yes
Others	TAs are available during exercise sessions. Every week one of the exercises is run as 'integrated exercise' during the lecture. Choice between two different exercise sessions

Resources

Bibliography

- Textbook: Deep Learning by Goodfellow, Bengio, Courville (MIT Press)
- Textbook: Reinforcement Learning by Sutton and Barto (MIT Press)

Pdfs of the preprint version for both books are available online

Ressources en bibliothèque

- [Reinforcement Learning by Sutton and Barto](#)
- [Deep Learning by Goodfellow, Bengio, Courville](#)

Websites

- <http://for videos and lecture slides https://lcnwww.epfl.ch/gerstner/VideoLecturesANN-Gerstner.html>
- <http://main web page is moodle>

Videos

- <http://yes, for most session.>

EE-554

Automatic speech processing

Magimai Doss Mathew

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	3
Session	Winter
Semester	Fall
Exam	Written
Workload	90h
Weeks	14
Hours	3 weekly
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Summary

The goal of this course is to provide the students with the main formalisms, models and algorithms required for the implementation of advanced speech processing applications (involving, among others, speech coding, speech analysis/synthesis, and speech recognition).

Content

1. Introduction: Speech processing tasks, language engineering applications.
2. Basic Tools: Analysis and spectral properties of the speech signal, linear prediction algorithms, statistical pattern recognition, dynamic programming.
3. Speech Coding: Human hearing properties, quantization theory, speech coding in the temporal and frequency domains.
4. Speech Synthesis: Morpho-syntactic analysis, phonetic transcription, prosody, speech synthesis models.
5. Automatic Speech Recognition: Temporal pattern matching and Dynamic Time Warping (DTW) algorithms, speech recognition systems based on Hidden Markov Models (HMMs).
6. Speaker recognition and speaker verification: Formalism, hypothesis testing, HMM based speaker verification.
7. Linguistic Engineering: state-of-the-art and typical applications

Keywords

speech processing, speech coding, speech analysis/synthesis, automatic speech recognition, speaker identification, text-to-speech

Learning Prerequisites**Required courses**

Basis in linear algebra, signal processing (FFT), and statistics

Important concepts to start the course

Basic knowledge in signal processing, linear algebra, statistics and stochastic processes.

Learning Outcomes

By the end of the course, the student must be able to:

- speech signal properties
- Exploit those properties to speech codign, speech synthesis, and speech recognition

Transversal skills

- Use a work methodology appropriate to the task.
- Access and evaluate appropriate sources of information.
- Use both general and domain specific IT resources and tools

Teaching methods

Lecture + lab exercises

Expected student activities

Attending courses and lab exercises. Read additional papers and continue lab exercises at home if necessary. Regularly answer list of questions for feedback.

Assessment methods

Written exam without notes

Supervision

Office hours	No
Assistants	Yes
Forum	No

Resources

Ressources en bibliothèque

- [Traitement de la parole / Boite](#)

Websites

- <http://lectures.idiap.ch/>

Cursus	Sem.	Type
Auditeurs en ligne	E	Opt.
Biocomputing minor	E	Opt.
Biomedical technologies minor	E	Opt.
Computational Neurosciences minor	E	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Neuroprosthetics minor	E	Opt.
Neuroscience		Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

In this course we study mathematical models of neurons and neuronal networks in the context of biology and establish links to models of cognition. The focus is on brain dynamics approximated by deterministic or stochastic differential equations.

Content

I. Models of single neurons 1. Introduction: brain vs computer and a first simple neuron model 2. Models on the level of ion current (Hodgkin-Huxley model) 3./4. Two-dimensional models and phase space analysis **II. Neuronal Dynamics of Cognition** 5./6. Associative Memory and Attractor Dynamics (Hopfield Model) 7. Neuronal Populations and networks 8. Continuum models and perception 9. Competition and models of Decision making **III. Noise and the neural code** 10. Noise and variability of spike trains (point processes, renewal process, interval distribution) 11: Variance of membrane potentials and Spike Response Models **IV. Plasticity and Learning** 12. Synaptic Plasticity and Long-term potentiation and Learning (Hebb rule, mathematical formulation) 13. Summary: Fitting Neural Models to Data

Keywords

neural networks, neuronal dynamics, computational neuroscience, mathematical modeling in biology, applied mathematics, brain, cognition, neurons, memory, learning, plasticity

Learning Prerequisites

Required courses

undergraduate math at the level of electrical engineering or physics majors
undergraduate physics.

Recommended courses

Analysis I-III, linear algebra, probability and statistics
For SSV students: Dynamical Systems Theory for Engineers or "Mathematical and Computational Models in Biology"

Important concepts to start the course

Differential equations, stochastic processes,

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze two-dimensional models in the phase plane
- Solve linear one-dimensional differential equations
- Develop a simplified model by separation of time scales
- Analyze connected networks in the mean-field limit
- Formulate stochastic models of biological phenomena
- Formalize biological facts into mathematical models
- Prove stability and convergence
- Apply model concepts in simulations
- Predict outcome of dynamics
- Describe neuronal phenomena

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Collect data.
- Write a scientific or technical report.

Teaching methods

Classroom teaching, exercises and miniproject. One of the two exercise hours is integrated into the lectures.

Expected student activities

- participate in ALL in-class exercises.
- do all homework exercises (paper-and-pencil)
- study video lectures if you miss a class
- study suggested textbook sections for in-depth understanding of material
- submit miniprojects

Assessment methods

Written exam (70%) & miniproject (30%)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	The teacher is available during the breaks of the class. Some exercises are integrated in class in the presence of the teacher and the teaching assistants.

Resources

Bibliography

Gerstner, Kistler, Naud, Pansinski : Neuronal Dynamics, Cambridge Univ. Press 2014

Ressources en bibliothèque

- [Neuronal Dynamics / Gerstner](#)

Notes/Handbook

The textbook is online at: <http://neurondynamics.epfl.ch/>

Videos

- <http://lcn.epfl.ch/~gerstner/NeuronalDynamics-MOOC1.html>
- <http://lcn.epfl.ch/~gerstner/NeuronalDynamics-MOOC2.html>

CS-524

Computational complexity

Göös Mika

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	4 weekly
Lecture	3 weekly
Exercises	1 weekly
Number of positions	

Summary

In computational complexity we study the computational resources needed to solve problems and understand the relation between different types of computation. This course advances the students knowledge of computational complexity, and develop an understanding of fundamental open questions.

Content

- Complexity classes (time, space, nondeterminism)
- Space complexity (Logspace, $L=NL$)
- Boolean circuits and nonuniform computation
- Power of randomness (interactive proofs, $IP=PSPACE$)
- Lower bounds for concrete models of computation: Decision trees, communication protocols, propositional proofs.

Keywords

theoretical computer science
computational complexity

Learning Prerequisites**Recommended courses**

Theory of computation (CS-251)
Algorithms (CS-250)

Learning Outcomes

By the end of the course, the student must be able to:

- Demonstrate an understanding of computational complexity and the P vs NP problem
- Formalize and analyze abstractions of complex scenarios/problems
- Express a good understanding of different concepts of proofs
- Prove statements that are similar to those taught in the course
- Use and understand the role of randomness in computation

- Explain recent exciting developments in theoretical computer science
- Compare different models of computation

Transversal skills

- Demonstrate the capacity for critical thinking
- Summarize an article or a technical report.

Teaching methods

Lecturing and exercises

Expected student activities

Actively attending lectures and exercise sessions. Also homeworks and exam.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

Sanjeev Arora and Boaz Barak: *Computational Complexity: A Modern Approach*, Cambridge University Press.

Stasys Jukna: *Boolean Function Complexity*, Springer

Ressources en bibliothèque

- [Computational Complexity: A Modern Approach / Arora](#)
- [Boolean Function Complexity / Stasys](#)

Websites

- <http://theory.epfl.ch/courses/complexity/>

MATH-453

Computational linear algebra

Kressner Daniel

Cursus	Sem.	Type
Computational science and Engineering	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Ing.-math	MA2, MA4	Opt.
Mathématicien	MA2	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	Oral
Workload	150h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

This course provides an overview of advanced techniques for solving large-scale linear algebra problems, as they typically arise in applications. A central goal of this course is to give the ability to choose a suitable solver for a given application.

Content**Introduction**

Sources of large-scale linear algebra problems. Recap of required linear algebra concepts.

Eigenvalue problems

Krylov subspace methods. Singular value problems. Preconditioned iterative methods.

Linear systems

Direct sparse factorizations. Krylov subspace methods and preconditioners.

Matrix functions

Theory and algorithms.

Keywords

linear systems, eigenvalue problems, matrix functions

Learning Prerequisites**Required courses**

Linear Algebra, Numerical Analysis

Learning Outcomes

By the end of the course, the student must be able to:

- Choose method for solving a specific problem.
- Prove the convergence of iterative methods.
- Interpret the results of a computation in the light of theory.
- Implement numerical algorithms.
- Describe methods for solving linear algebra problems.
- State theoretical properties of numerical algorithms.

Teaching methods

Ex cathedra lecture, exercises in the classroom and with computer

Expected student activities

Attendance of lectures.
Completing exercises.
Completing a miniproject.
Solving problems on the computer.

Assessment methods

Miniproject and oral examination.

Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

Resources

Bibliography

Lecture notes will be provided by the instructor. Complimentary reading:

H. Elman, D. J. Silvester, and A. J. Wathen. Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics. Oxford University Press, 2005.

G. H. Golub and C. Van Loan. Matrix computations. Johns Hopkins University Press, 1996.

Y. Saad. Iterative methods for sparse linear systems. Second edition. SIAM, 2003.

Ressources en bibliothèque

- [Finite elements and fast iterative solvers / Elman](#)
- [Matrix computations / Golub](#)
- [Iterative methods for sparse linear systems / Saad](#)

CS-413

Computational photography

Süsstrunk Sabine

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	During the semester
Workload	150h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

The students will gain the theoretical knowledge in computational photography, which allows recording and processing a richer visual experience than traditional digital imaging. They will also execute practical group projects to develop their own computational photography application.

Content

Computational photography is the art, science, and engineering of creating a great (still or moving) image. Information is recorded in space, time, across visible and invisible radiation and from other sources, and then post-processed to produce the final - visually pleasing - result.

Basics: Human vision system, Light and illumination, Geometric optics, Color science, Sensors, Digital camera systems.

Generalized illumination: Structured light, High dynamic range (HDR) imaging, Time-of-flight.

Generalized optics: Coded Image Sensing, Coded aperture, Focal stacks.

Generalized sensing: Low light imaging, Depth imaging, Plenoptic imaging, Light field cameras.

Generalized processing: Super-resolution, In-painting, Compositing, Photomontages, Panoramas, HDR imaging,

Multi-wavelength imaging, Dynamic imaging.

Generalized display: Stereoscopic displays, HDR displays, 3D displays, Mobile displays.

Deep Learning for image resoration and image enhancement.

Keywords

Computational Photography, Coded Image Sensing, Non-classical image capture, Multi-Image & Sensor Fusion, Mobile Imaging, Machine Learning

Learning Prerequisites**Required courses**

- A basic Signal Processing, Image Processing, and/or Computer Vision course.
- Linear Algebra.

Recommended courses

- Introduction to Computer Vision.
- Signal Processing for Communications.
- Machine Learning.

Important concepts to start the course

- Basic signal/image processing.
- Basic computer vision.
- Basic programming (Python, iOS, Android).

Learning Outcomes

- Identify the main components of a computational photography system.
- Contextualise the main trends in computational optics, sensing, processing, and displays.
- Create a computational photography application.
- Design a computational photography solution to solve a particular imaging task.
- Assess / Evaluate hardware and software combinations for their imaging performance.
- Formulate computational photography challenges that still need to be resolved.

Transversal skills

- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Continue to work through difficulties or initial failure to find optimal solutions.

Teaching methods

The course consists of 2 hours of lectures per week that will cover the theoretical basics. An additional 2 hours per week are dedicated to a group project designing, developing, and programming a computational photography application on a mobile platform (iOS, Android).

Expected student activities

The student is expected to attend the class and actively participate in the practical group project, which requires coding on either Android or iOS platform. The student is also required to read the assigned reading material (book chapters, scientific articles).

Assessment methods

The theoretical part will be evaluated with an oral exam at the end of the semester, and the practical part based on the students' group projects.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

- Selected book chapters
- Course notes (on moodle)
- Links to relevant scientific articles and on-line resources will be given on moodle.

CS-442

Computer vision

Fua Pascal

Cursus	Sem.	Type
Communication systems minor	E	Opt.
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	3 weekly
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Summary

Computer Vision aims at modeling the world from digital images acquired using video or infrared cameras, and other imaging sensors. We will focus on images acquired using digital cameras. We will introduce basic processing techniques and discuss their field of applicability.

Content**Introduction**

- History of Computer Vision
- Human vs Machine Vision
- Image formation

Extracting 2D Features

- Contours
- Texture
- Regions

3D Shape Recovery

- From one single image
- From multiple images

Learning Outcomes

By the end of the course, the student must be able to:

- Choose relevant algorithms in specific situations
- Perform simple image-understanding tasks

Teaching methods

Ex cathedra lectures and programming exercises using matlab.

Assessment methods

With continuous control

Resources

Bibliography

- R. Szeliski, Computer Vision: Algorithms and Applications, 2010.
- A. Zisserman and R. Hartley, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.

Ressources en bibliothèque

- [Computer Vision: Algorithms and Applications / Szeliski](#)
- [Multiple View Geometry in Computer Vision / Zisserman](#)

Websites

- <http://cvlab.epfl.ch/>

Moodle Link

- <http://moodle.epfl.ch/course/view.php?id=472>

COM-418

Computers and music

Prandoni Paolo

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	3 weekly
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Summary

In this class we will explore some of the fundamental ways in which the pervasiveness of digital devices has completely revolutionized the world of music in the last 40 years, both from the point of view of production and recording, and from the point of view of listening and distribution.

Content

- review of digital signal processing: discrete-time signals, spectral analysis, digital filters
- audio measurement standards; A/D and D/A converters; oversampling; sigma-delta
- audio compression; the MP3 standard
- digital synthesizers: oscillators, FM synthesis, samplers
- fundamentals of time-frequency analysis; pitch shifting; time stretching; vocoder
- music production; equalization, compression, reverb
- notions of balancing and mastering; the MIDI and VST standards
- nonlinear system modeling
- deep learning in audio processing

Keywords

DSP, computer music, digital audio

Learning Prerequisites**Required courses**

digital signal processing, programming

Recommended courses

signals and systems, Python, C++

Important concepts to start the course

Digital signals, filters, spectral analysis

Learning Outcomes

By the end of the course, the student must be able to:

- Describe the fundamental techniques in digital audio recording and production
- Be able to avoid unwanted artifacts in sound recording and compression
- Recognize the typical acoustic footprint of classic synthesizers and audio effects
- Write working signal processing code to synthesize sounds and process audio
- Write code that interfaces to existing equipment via industry-standard protocols

Transversal skills

- Access and evaluate appropriate sources of information.
- Summarize an article or a technical report.
- Write a scientific or technical report.
- Demonstrate a capacity for creativity.

Teaching methods

lectures

Expected student activities

- Attending lectures
- Writing code samples
- Solving exercises
- Read technical papers

Assessment methods

mini projects and/or final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

TBD

Notes/Handbook

handouts, papers and code samples

CS-453

Concurrent algorithms

Guerraoui Rachid

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	5 weekly
Lecture	3 weekly
Exercises	1 weekly
Practical work	1 weekly
Number of positions	

Summary

With the advent of multiprocessors, it becomes crucial to master the underlying algorithmics of concurrency. The objective of this course is to study the foundations of concurrent algorithms and in particular the techniques that enable the construction of robust such algorithms.

Content**Model of a parallel system**

A multicore architect
Processes and objects
Safety and liveness

Parallel programming

Automatic parallelism
Mutual exclusion and locks
Non-blocking data structures

Register Implementations

Safe, regular and atomic registers
General and limited transactions
Atomic snapshots

Hierarchy of objects

The FLP impossibility
The consensus number
Universal constructions

Transactional memories

Transactional algorithms
Opacity and obstruction-freedom

Keywords

Concurrency, parallelism, algorithms, data structures

Learning Prerequisites**Required courses**

ICC, Operatings systems

Recommended courses

This course is complementary to the Distributed Algorithms course.

Important concepts to start the course

Processes, threads, data structures

Learning Outcomes

By the end of the course, the student must be able to:

- Reason in a precise manner about concurrency
- Design a concurrent algorithm
- Prove a concurrent algorithm
- Implement a concurrent system

Teaching methods

Lectures, exercises and practical work

Expected student activities

Midterm and final exam

Project

Assessment methods

With continuous control, midterm final exams and project

Supervision

Office hours	Yes
Assistants	Yes
Forum	No

Resources**Notes/Handbook**

Concurrent Algorithms, R. Guerraoui and P. Kouznetsov

Websites

- <http://lpd.epfl.ch/site/education>

COM-401

Cryptography and security

Vaudenay Serge

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Opt.
Financial engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	7
Session	Winter
Semester	Fall
Exam	Written
Workload	210h
Weeks	14
Hours	6 weekly
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

This course introduces the basics of cryptography. We review several types of cryptographic primitives, when it is safe to use them and how to select the appropriate security parameters. We detail how they work and sketch how they can be implemented.

Content

1. **Ancient cryptography:** Vigenère, Enigma, Vernam cipher, Shannon theory
2. **Diffie-Hellman cryptography:** algebra, Diffie-Hellman, ElGamal
3. **RSA cryptography:** number theory, RSA, factoring
4. **Elliptic curve cryptography:** elliptic curves over a finite field, ECDH, ECIES, pairing
5. **Symmetric encryption:** block ciphers, stream ciphers, exhaustive search
6. **Integrity and authentication:** hashing, MAC, birthday paradox
7. **Public-key cryptography:** cryptosystem, digital signature, post-quantum cryptography
8. **Trust establishment:** password-based cryptography, secure communication, trust setups
9. **Case studies:** WiFi, bitcoin, mobile telephony, WhatsApp, EMV, Bluetooth, biometric passport, TLS

Keywords

cryptography, encryption, secure communication

Learning Prerequisites**Required courses**

- Algebra (MATH-310)
- Probabilities and statistics (MATH-232)
- Algorithms (CS-250)

Recommended courses

- Computer security (COM-301)

Important concepts to start the course

- Mathematical reasoning
- Probabilities

- Algebra, arithmetics
- Algorithmics

Learning Outcomes

By the end of the course, the student must be able to:

- Choose the appropriate cryptographic primitive in a security infrastructure
- Judge the strength of existing standards
- Assess / Evaluate the security based on key length
- Implement algorithms manipulating big numbers and use number theory
- Use algebra and probability theory to analyze cryptographic algorithms
- Identify the techniques to secure the communication and establish trust

Teaching methods

ex-cathedra

Expected student activities

- active participation during the course
- take notes during the course
- do the exercises during the exercise sessions
- complete the regular tests and homework
- read the material from the course
- self-train using the provided material
- do the midterm exam and final exam

Assessment methods

Mandatory continuous evaluation:

- homework (30%)
- regular graded tests (30%)
- midterm exam (40%)

Final exam averaged (same weight) with the continuous evaluation, but with final grade between final_exam-1 and final_exam+1.

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	Lecturers and assistants are available upon appointment.

Resources

Bibliography

- Communication security: an introduction to cryptography. Serge Vaudenay. Springer 2004.
- A computational introduction to number theory and algebra. Victor Shoup. Cambridge University Press

2005.

Ressources en bibliothèque

- [A computational introduction to number theory and algebra / Shoup](#)
- [Communication security / Vaudenay](#)

Websites

- <http://lasec.epfl.ch/teaching.shtml>

Moodle Link

- <https://moodle.epfl.ch/course/view.php?id=13671>

Prerequisite for

- Advanced cryptography (COM-401)
- Student seminar: security protocols and applications (COM-506)

COM-480

Data visualization

Vuillon Laurent Gilles Marie

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Learning Sciences		Obl.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

Understanding why and how to present complex data interactively in an effective manner has become a crucial skill for any data scientist. In this course, you will learn how to design, judge, build and present your own interactive data visualizations.

Content**Tentative course schedule**

Week 1: Introduction to Data visualization Web development

Week 2: Javascript

Week 3: More Javascript

Week 4: Data Data driven documents (D3.js)

Week 5: Interaction, filtering, aggregation (UI /UX). Advanced D3 / javascript libs

Week 6: Perception, cognition, color Marks and channels

Week 7: Designing visualizations (UI/UX) Project introduction Dos and don'ts for data-viz

Week 8: Maps (theory) Maps (practice)

Week 9: Text visualization

Week 10: Graphs

Week 11: Tabular data viz Music viz

Week 12: Introduction to scientific visualisation

Week 13: Storytelling with data / data journalism Creative coding

Week 14: Wrap-Up

Keywords

Data viz, visualization, data science

Learning Prerequisites**Required courses**

CS-305 Software engineering (BA)

CS-250 Algorithms (BA)

CS-401 Applied data analysis (MA)

Recommended courses

EE-558 A Network Tour of Data Science (MA)

CS-486 Interaction design (MA)

CS-210 Functional programming (BA)

Important concepts to start the course

Being autonomous is a prerequisite, we don't offer office hours and we won't have enough teaching assistants (you've been warned!).

Knowledge of one of the following programming language such as C++, Python, Scala.

Familiarity with web-development (you already have a blog, host a website). Experience with HTML5, Javascript is a strong plus for the course.

Learning Outcomes

By the end of the course, the student must be able to:

- Judge visualization in a critical manner and suggest improvements.
- Design and implement visualizations from the idea to the final product according to human perception and cognition
- Know the common data-viz techniques for each data domain (multivariate data, networks, texts, cartography, etc) with their technical limitations
- Create interactive visualizations in the browser using HTML5 and Javascript

Transversal skills

- Communicate effectively, being understood, including across different languages and cultures.
- Negotiate effectively within the group.
- Resolve conflicts in ways that are productive for the task and the people concerned.

Teaching methods

Ex cathedra lectures, exercises, and group projects

Expected student activities

- Follow lectures
- Read lectures notes and textbooks
- Create an advanced data-viz in groups of 3.
- Answer questions assessing the evolution of the project.
- Create a 2min screencast presentation of the viz.
- Create a process book for the final data viz.

Assessment methods

- Data-viz (35%)
- Technical implementation (15%)
- Website, presentation, screencast (25%)
- Process book (25%)

Supervision

Office hours	No
Assistants	No
Forum	No

Resources

Bibliography

Visualization Analysis and Design by Tamara Munzner, CRC Press (2014). Free online version at EPFL.
Interactive Data Visualization for the Web by Scott Murray O'Reilly (2013) - D3 - Free online version.
The Truthful Art: Data, Charts, and Maps for Communication by Cairo, Alberto. Royaume-Uni, New Riders, (2016).
Data Visualisation: A Handbook for Data Driven Design by Kirk, Andy. Royaume-Uni, SAGE Publications, (2019).

Ressources en bibliothèque

- [Interactive Data Visualization for the Web / Murray](#)
- [Visualization Analysis and Design / Munzner](#)
- [Data Visualisation / Kirk](#)
- [The Truthful Art / Cairo](#)

Notes/Handbook

Lecture notes

Websites

- <https://www.kirellbenzi.com>

Moodle Link

- <https://moodle.epfl.ch/course/view.php?id=15487>

EE-559

Deep learning

Fleuret François

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Financial engineering	MA2, MA4	Opt.
Learning Sciences		Obl.
Life Sciences Engineering	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Withdrawal Session	Unauthorized Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	500

Summary

The objective of this course is to provide a complete introduction to deep machine learning. How to design a neural network, how to train it, and what are the modern techniques that specifically handle very large networks.

Content

The course aims at providing an overview of existing processings and methods, at teaching how to design and train a deep neural network for a given task, and at providing the theoretical basis to go beyond the topics directly seen in the course.

It will touch on the following topics:

- What is deep learning, introduction to tensors.
- Basic machine-learning, empirical risk minimization, simple embeddings.
- Linear separability, multi-layer perceptrons, back-propagation.
- Generalized networks, autograd, batch processing, convolutional networks.
- Initialization, optimization, and regularization. Drop-out, batchnorm, resnets.
- Deep models for Computer Vision.
- Analysis of deep models.
- Auto-encoders, embeddings, and generative models.
- Recurrent and attention models, Natural Language Processing.

Concepts will be illustrated with examples in the PyTorch framework (<http://pytorch.org>).

Keywords

machine learning, neural networks, deep learning, computer vision, python, pytorch

Learning Prerequisites

Required courses

- Linear algebra (vector, matrix operations, Euclidean spaces).
- Differential calculus (Jacobian, Hessian, chain rule).
- Python programming.
- Basics in probabilities and statistics (discrete and continuous distributions, normal density, law of large numbers, conditional probabilities, Bayes, PCA)

Recommended courses

- Basics in optimization (notion of minima, gradient descent).
- Basics in algorithmic (computational costs).
- Basics in signal processing (Fourier transform, wavelets).

Teaching methods

Ex-cathedra with exercise sessions and mini-projects. Possibly invited speakers.

Assessment methods

Mini-projects by groups of students, and one final written exam.

Resources**Notes/Handbook**

Not mandatory: <http://www.deeplearningbook.org/>

Websites

- <https://fleuret.org/ee559/>

CS-411

Digital education & learning analytics

Dillenbourg Pierre, Jermann Patrick

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Learning Sciences		Obl.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Oral
Workload	120h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

This course addresses the relationship between specific technological features and the learners' cognitive processes. It also covers the methods and results of empirical studies on this topic: do student actually learn due to technologies?

Content

- *Learning theories and learning processes.*
- *Types of learning technologies*
- *Instructional design: methods, patterns and principles.*
- *On-line education.*
- *Effectiveness of learning technologies.*
- *Methods for empirical research.*

Keywords

learning, pedagogy, teaching, online education, MOOCs

Learning Outcomes

By the end of the course, the student must be able to:

- Describe the learning processes triggered by a technology-based activity
- Explain how a technology feature influences learning processes
- Elaborate a study that measures the learning effects of a digital environment
- Select appropriately a learning technology given the target audience and the expected learning outcomes
- Apply machine learning methods to educational traces

Transversal skills

- Set objectives and design an action plan to reach those objectives.

Teaching methods

The course will combine participatory lectures with a project around learning analytics

Expected student activities

The project will include several milestones to be delivered along the semester.

Assessment methods

- Project + exam
- 50 / 50

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- <http://moodle.epfl.ch/course/view.php?id=14248>

CS-451

Distributed algorithms

Guerraoui Rachid

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	6 weekly
Lecture	3 weekly
Exercises	2 weekly
Practical work	1 weekly
Number of positions	

Summary

Computing is often distributed over several machines, in a local IP-like network, a cloud or in a P2P network. Failures are common and computations need to proceed despite partial failures of machines or communication links. The foundations of reliable distributed computing will be studied.

Content

Reliable broadcast
 Causal Broadcast
 Total Order Broadcast
 Consensus
 Non-Blocking Atomic Commit
 Group Membership, View Synchrony
 Terminating Reliable Broadcast
 Shared Memory in Message Passing Systems
 Byzantine Fault Tolerance
 Self Stabilization
 Population protocols (models of mobile networks)
 Bitcoin, Blockchain
 Distributed Machine Learning
 Gossip

Keywords

Distributed algorithms, checkpointing, replication, consensus, atomic broadcast, distributed transactions, atomic commitment, 2PC, Machine Learning

Learning Prerequisites**Required courses**

Basics of Algorithms, networking and operating systems

Recommended courses

The lecture is orthogonal to the one on concurrent algorithms: it makes a lot of sense to take them in parallel.

Learning Outcomes

By the end of the course, the student must be able to:

- Choose an appropriate abstraction to model a distributed computing problem
- Specify the abstraction
- Present and implement it
- Analyze its complexity
- Prove a distributed algorithm
- Implement a distributed system

Teaching methods

Ex cathedra

Lectures, exercises and practical work

Assessment methods

Midterm and final exams

Project

Supervision

Office hours Yes

Assistants Yes

Forum Yes

Resources

Ressources en bibliothèque

- [Introduction to reliable and secure distributed programming / Cachin](#)

Notes/Handbook

Reliable and Secure Distributed Programming

Springer Verlag

C. Cachin, R. Guerraoui, L. Rodrigues

Websites

- <http://lpdwww.epfl.ch/education>

CS-423

Distributed information systems

Aberer Karl

Cursus	Sem.	Type
Biocomputing minor	H	Opt.
Civil & Environmental Engineering		Opt.
Communication systems minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Energy Management and Sustainability	MA1, MA3	Opt.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Learning Sciences		Obl.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	3 weekly
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Summary

This course introduces the key concepts and algorithms from the areas of information retrieval, data mining and knowledge bases, which constitute the foundations of today's Web-based distributed information systems.

Content**Information Retrieval**

1. Information Retrieval - Introduction
2. Text-Based Information Retrieval (Boolean, Vector space, probabilistic)
3. Inverted Files
4. Distributed Retrieval
5. Query Expansion
6. Embedding models (LSI, word2vec)
7. Link-Based Ranking

Mining Unstructured Data

1. Association Rule Mining
2. Document Classification (knn, Naive Bayes, Fasttext, Transformer models)
3. Recommender Systems (collaborative filtering, matrix factorization)
4. Mining Social Graphs (modularity clustering, Girvan-Newman)

Knowledge Bases

1. Semantic Web
2. Keyphrase extraction
3. Named entity recognition
4. Information extraction
5. Taxonomy Induction
6. Entity Disambiguation
7. Label Propagation
8. Link Prediction
9. Data Integration

Learning Prerequisites**Recommended courses**

Introductory courses to databases and machine learning are helpful, but not required.
Programming skills in Python are helpful, but not required.

Learning Outcomes

By the end of the course, the student must be able to:

- Characterize the main tasks performed by information systems, namely data, information and knowledge management
- Apply collaborative information management models, like crowd-sourcing, recommender systems, social networks
- Apply knowledge models, their representation through Web standards and algorithms for storing and processing semi-structured data
- Apply fundamental models and techniques of text retrieval and their use in Web search engines
- Apply main categories of data mining techniques, local rules, predictive and descriptive models, and master representative algorithms for each of the categories

Teaching methods

Ex cathedra + programming exercises (Python)

Assessment methods

25% Continuous evaluations with bonus system during the semester

75% Final written exam (180 min) during exam session

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

ENG-466

Distributed intelligent systems

Martinoli Alcherio

Cursus	Sem.	Type
Biocomputing minor	E	Opt.
Civil Engineering	MA2, MA4	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Energy Management and Sustainability	MA2, MA4	Opt.
Energy Science and Technology	MA2, MA4	Opt.
Environmental Sciences and Engineering	MA2, MA4	Opt.
Microtechnics	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	Oral
Workload	150h
Weeks	14
Hours	5 weekly
Lecture	2 weekly
Exercises	3 weekly
Number of positions	

Summary

The goal of this course is to provide methods and tools for modeling distributed intelligent systems as well as designing and optimizing coordination strategies. The course is a well-balanced mixture of theory and practical activities.

Content

- Introduction to key concepts such as self-organization and tools used in the course
- Examples of natural, artificial and hybrid distributed intelligent systems
- Modeling methods: sub-microscopic, microscopic, macroscopic, multi-level; spatial and non-spatial; mean field, approximated and exact approaches
- Machine-learning methods: single- and multi-agent techniques; expensive optimization problems and noise resistance
- Coordination strategies and distributed control: direct and indirect schemes; algorithms and methods; performance evaluation
- Application examples in distributed sensing and action

Keywords

Artificial intelligence, swarm intelligence, distributed robotics, sensor networks, modeling, machine-learning, control

Learning Prerequisites**Required courses**

Fundamentals in analysis, probability, and programming for both compiled and interpreted languages

Recommended courses

Basic knowledge in statistics, programming language used in the course (C, Matlab, Python), and signals and systems

Learning Outcomes

By the end of the course, the student must be able to:

- Design control algorithms
- Formulate a model at different level of abstraction for a distributed intelligent system
- Analyze a model of a distributed intelligent system
- Analyze a distributed coordination strategy/algorithm
- Design a distributed coordination strategy/algorithm
- Implement code for single robot and multi-robot systems
- Carry out systematic performance evaluation of a distributed intelligent system
- Apply modeling and design methods to specific problems requiring distributed sensing and action
- Optimize a controller or a set of possibly coordinated controllers using model-based or data-driven methods

Transversal skills

- Demonstrate a capacity for creativity.
- Access and evaluate appropriate sources of information.
- Collect data.
- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Write a scientific or technical report.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.

Teaching methods

Ex-cathedra lectures, assisted exercises, and homework in team

Expected student activities

Attending lectures, carrying out exercises and the course project, and reading handouts.

Assessment methods

Oral exam (60%) with continuous assessment during the semester (40%).

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Lecture notes, selected papers and book chapters distributed at each lecture.

Websites

- https://disal.epfl.ch/teaching/distributed_intelligent_systems/

Moodle Link

- <https://moodle.epfl.ch/course/view.php?id=15472>

Prerequisite for

R&D activities in engineering

COM-596

Engineering internship credited with Master project (master in Data science)

Profs divers *

Cursus	Sem.	Type
Data Science	MA1, MA2, MA3, MA4, PME, PMH	Obl.

Language	English
Credits	0
Session	Winter, Summer
Semester	Fall
Exam	Term paper
Workload	0h
Weeks	
Practical work	320 weekly
Number of positions	

Summary

The engineering internship is an integral part of the curriculum for master's students. They join companies in Switzerland or abroad to carry out an internship or their master's project in a field of activity where the skills of the futur Engineer are highlighted.

Content

Internship represent an important experience for students, allowing them to achieve the following goals :

- Immerse themselves in the professional world
- Highlight the importance of teamwork
- Consider the imperatives of a company in its processes
- Put into practice the knowledge acquired from the study plan

The following three forms of internship are possible as part of the masters' study plan :

- 8 weeks internship during the summer only
- 6 monts (long) internship. The student is on leave for one semester
- Master project in the industry (25 weeks)

Learning Prerequisites

Required courses

- Have completed one full semester for students who have obtained their Bachelor at EPFL
- Have completed two full semesters for students coming from another university

Learning Outcomes

By the end of the course, the student must be able to:

- Be aware of the importance of legal procedures and the ethical code of the profession
- Communicate effectively and be understood
- Self-assess the level of acquired skills and plan the next goals
- Manage the priorities
- Receive and give feedback (criticism) and respond appropriately

Expected student activities

The student agrees to do this internship with professionalism

Assessment methods

- Short internship (8 weeks): electronic evaluation at the end of the internship
- Long internship (6 months): electronic evaluation at the end of the intership
- Master's project in the industry : see the master's project course book

Supervision

Others

Industry supervisor
EPFL supervisor

Resources

Notes/Handbook

Internship guidelines available on the internship website

Websites

- <https://www.epfl.ch/schools/ic/internships/>
- <https://www.epfl.ch/about/recruiting/recruiting/internships/>

CS-550

Formal verification

Kuncak Viktor

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	6 weekly
Lecture	2 weekly
Exercises	2 weekly
Practical work	2 weekly
Number of positions	

Summary

We introduce formal verification as an approach for developing highly reliable systems. Formal verification finds proofs that computer systems work under all relevant scenarios. We will learn how to use formal verification tools and explain the theory and the practice behind them.

Content

Topics may include (among others) some of the following:

- Importance of Reliable Systems. Methodology of Formal Verification. Soundness and Completeness in Modeling and Tools. Successful Tools and Flagship Case Studies
- Review of Sets, Relations, Computability, Propositional and First-Order Logic Syntax, Semantics, Sequent Calculus.
- Completeness and Semi-Decidability for First-Order Logic. Inductive Definitions and Proof Trees. Higher-Order Logic and LCF Approach.
- State Machines. Transition Formulas. Traces. Strongest Postconditions and Weakest Preconditions.
- Hoare Logic. Inductive Invariants. Well-Founded Relations and Termination Measures
- Linear Temporal Logic. System Verilog Assertions. Monitors
- SAT Solvers and Bounded Model Checking
- Model Checking using Binary Decision Diagrams
- Loop Invariants. Hoare Logic. Statically Checked Function Contracts. Relational Semantics and Fixed-Point Semantics
- Symbolic Execution. Satisfiability Modulo Theories
- Abstract Interpretation and Predicate Abstraction
- Information Flow and Taint Analysis
- Verification of Security Protocols
- Dependent and Refinement Types

Learning Prerequisites**Recommended courses**

Computer Language Processing / Compilers

Important concepts to start the course

Discrete Mathematics

Learning Outcomes

By the end of the course, the student must be able to:

- Formalize specifications
- Synthesize loop invariants
- Specify software functionality
- Generalize inductive hypothesis
- Critique current software development practices

Teaching methods

Instructors will present lectures and exercises and supervise labs on student laptops.

Expected student activities

Follow the course material and complete and explain projects during the semester.

Assessment methods

The grade is based on the code, documentation, and explanation of projects during the semester. There are no written exams.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

- **Harrison, J. (2009). *Handbook of Practical Logic and Automated Reasoning*. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511576430**
- **Aaron Bradley and Zohar Manna: *The Calculus of Computation - Decision Procedures with Applications to Verification*, Springer 2007.**
- Michael Huth and Mark Ryan: *Logic in Computer Science - Modelling and Reasoning about Systems*. Cambridge University Press 2004.
- *Handbook of Model Checking*, <https://www.springer.com/de/book/9783319105741> Springer 2018. Including Chapter Model Checking Security Protocols by David Basin.
- Tobias Nipkow, Gerwin Klein: *Concrete Semantics with Isabelle/HOL*. <http://concrete-semantics.org/concrete-semantics.pdf>
- Nielson, Flemming, Nielson, Hanne R., Hankin, Chris: *Principles of Program Analysis*. ISBN 978-3-662-03811-6. Springer 1999.
- Peter B. Andrews: *An Introduction to Mathematical Logic and Type Theory (To Truth Through Proof)*, Springer 2002.
- <http://logitext.mit.edu/tutorial>

Ressources en bibliothèque

- [Handbook of model checking / Clarke](#)

- [Introduction to mathematical logic and type theory / Andrews](#)
- [Principles of Program Analysis / Flemming](#)
- [The Calculus of Computation / Bradley](#)
- [Logic in Computer Science / Huth](#)
- [Handbook of model checking : Model Checking Security Protocols / Bassin](#)

Notes/Handbook

- <https://lara.epfl.ch/w/fv>

Websites

- <https://lara.epfl.ch/w/fv>

Videos

- <https://tube.switch.ch/channels/f2d4e01d>

Prerequisite for

MSc thesis in the LARA group

COM-406

Foundations of Data Science

Gastpar Michael C., Urbanke Rüdiger

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computer and Communication Sciences		Obl.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Obl.
Data science minor	H	Opt.
Digital Humanities	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	6 weekly
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

We discuss a set of topics that are important for the understanding of modern data science but that are typically not taught in an introductory ML course. In particular we discuss fundamental ideas and techniques that come from probability, information theory as well as signal processing.

Content

This class presents basic concepts of Information Theory and Signal Processing and their relevance to emerging problems in Data Science and Machine Learning.

A tentative list of topics covered is:

1. Information Measures
2. Signal Representations
3. Detection and Estimation
4. Multi-arm Bandits
5. Distribution Estimation, Property Testing, and Property Estimation
6. Exponential Families
7. Compression and Dimensionality Reduction
8. Information Measures and Generalization Error

Keywords

Information Theory, Signal Processing, Statistical Signal Processing, Machine Learning, Data Science.

Learning Prerequisites**Required courses**

COM-300 Modèles stochastiques pour les communications

Recommended courses

Statistics

Important concepts to start the course

Solid understanding of linear algebra and probability as well as real and complex analysis.

Learning Outcomes

By the end of the course, the student must be able to:

- Formulate the fundamental concepts of signal processing such as basis representations and sampling
- Formulate the fundamental concepts of information theory such as entropy and mutual information

- Analyze problems in statistical settings using fundamental bounds from information theory
- Formulate problems using robust and universal techniques

Teaching methods

Ex cathedra lectures, exercises, and small projects.

Expected student activities

Follow lectures; independent work on problems (homework and small projects).

Assessment methods

Written final exam during the exam session.

Homework Problem Sets during the semester.

10% homework, 30% midterm, 60% final exam; (if for some reason the course has to be given over zoom then we will skip the midterm and the course will be evaluated by 10% homework and 90% final)

Resources

Bibliography

Cover and Thomas, Elements of Information Theory (Second Edition), Wiley, 2006.

Ressources en bibliothèque

- [Elements of Information Theory / Cover](#)

Notes/Handbook

Lectures notes will be available on the course web page.

CS-457

Geometric computing

Pauly Mark

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	5 weekly
Lecture	3 weekly
Practical work	2 weekly
Number of positions	

Summary

This course will cover mathematical concepts and efficient numerical methods for geometric computing. We will develop and implement algorithms to simulate and optimize 2D and 3D geometric models with an emphasis towards computational design for digital fabrication.

Content

- Overview of modern digital fabrication technology
- Discrete geometric models for curves, surfaces, volumes
- Basics of finite element modeling
- Physics-based simulation methods
- Forward and inverse design optimization methods
- Shape Optimization

Keywords

geometry, simulation, shape optimization, digital fabrication

Learning Prerequisites**Recommended courses**

CS-328 : Numerical Methods for Visual Computing and ML

Important concepts to start the course

Undergraduate knowledge of linear algebra, calculus, and numerical methods; programming experience (e.g. Python, C/C++, Java, Scala)

Learning Outcomes

By the end of the course, the student must be able to:

- Model and formalize geometric shape design & optimization problems
- Design and implement computational methods for shape processing, physics-based simulation, and numerical optimization based on discrete geometry representations
- Apply geometric abstraction principles to reduce the complexity of shape optimization problems
- Assess / Evaluate geometry processing algorithms for their suitability for specific digital fabrication technologies

Transversal skills

- Demonstrate a capacity for creativity.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Use both general and domain specific IT resources and tools
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.

Teaching methods

Lectures, interactive demos, exercises, practical work sessions

Expected student activities

Attend and participate in lectures, study provided reading material, solve theory exercises and implementation homeworks, design and fabricate (with support) physical models

Assessment methods

Graded theory and implementation homeworks

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

MATH-360

Graph theory

Maffucci Riccardo Walter

Cursus	Sem.	Type
Data Science	MA1, MA3	Opt.
Mathematics	BA5	Opt.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

The course aims to introduce the basic concepts and results of modern Graph Theory with special emphasis on those topics and techniques that have proved to be applicable in theoretical computer science and in practice.

Content

1. Graphic sequences
2. Connectivity
3. Planarity
4. Methods from linear algebra
5. Forests and spanning trees
6. Eulerian and Hamiltonian graphs
7. Colourings
8. Extremal Graph Theory

Keywords

Graph, isomorphism, complement, complete, bipartite, product, graphic sequence, connected, path, circuit, cycle, block, planar, maximal planar, polyhedron, dual, tree, spanning, Eulerian, Hamiltonian, colouring, forbidden subgraph, extremal graph.

Learning Prerequisites**Recommended courses**

Mandatory for IN/SC: Analyse III, Physique générale I, Physique générale II, Probability and statistics

Learning Outcomes

By the end of the course, the student must be able to:

- Illustrate simple examples of graphs satisfying certain properties
- State definitions and results of graph theory
- Verify hypotheses of theorems for applications
- Implement algorithms of graph theory
- Prove theorems and other properties
- Justify the main arguments rigorously
- Apply relevant results to solve problems.

Assessment methods

WRITTEN EXAM

Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

Resources**Bibliography**

- Diestel : Graph Theory (Springer)
- Bollobas : Modern Graph Theory (Springer)
- Harris, Hirst, Mossinghoff : Combinatorics and Graph Theory (Springer)
- Harary : Graph Theory (Addison-Wesley).

Ressources en bibliothèque

- [Graph Theory / Diestel](#)
- [Modern Graph Theory / Bollobas](#)
- [Graph Theory / Harary](#)
- [\(electronic version\)](#)
- [Combinatorics and Graph Theory / Harris, Hirst & Mossinghoff](#)
- [\(electronic version\)](#)

EE-451

Image analysis and pattern recognition

Thiran Jean-Philippe

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Data Science	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Practical work	2 weekly
Number of positions	

Summary

This course gives an introduction to the main methods of image analysis and pattern recognition.

Content**Introduction**

Digital image acquisition and properties.

Pre-processing: geometric transforms, linear filtering, image restoration.

Introduction to Mathematical Morphology

Examples and applications

Segmentation and object extraction

Thresholding, edge detection, region detection.

Segmentation by active contours. Applications in medical image segmentation.

Shape representation and description

Contour-based representation, region-based representation. Morphological skeletons

Shape recognition

Statistical shape recognition, Bayesian classification, linear and non-linear classifiers, perceptrons, neural networks and unsupervised classifiers.

Applications.

Practical works on computers**Learning Prerequisites****Recommended courses**

Introduction to signal processing, Image processing

Learning Outcomes

- Use Image Pre-processing methods
- Use Image segmentation methods
- Choose shape description methods appropriate to a problem
- Use classification methods appropriate to a problem

Transversal skills

- Use a work methodology appropriate to the task.
- Make an oral presentation.
- Identify the different roles that are involved in well-functioning teams and assume different roles, including leadership roles.
- Assess one's own level of skill acquisition, and plan their on-going learning goals.
- Summarize an article or a technical report.

Teaching methods

Ex cathedra and practical work and oral presentation by the students

Assessment methods

Continuous control

Resources

Bibliography

Reconnaissance des formes et analyse de scènes / Kunt
Image processing, Analysis and Machine Vision / Sonka

Ressources en bibliothèque

- [Reconnaissance des formes et analyse de scènes / Kunt](#)
- [Image processing, Analysis and Machine Vision / Sonka](#)

Prerequisite for

Semester project, Master project, doctoral thesis

COM-402

Information security and privacy

Hubaux Jean-Pierre, Pyrgelis Apostolos

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Obl.
Data science minor	H	Opt.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Obl.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	6 weekly
Lecture	3 weekly
Exercises	1 weekly
Project	2 weekly
Number of positions	

Summary

This course provides an overview of information security and privacy topics. It introduces students to the knowledge and tools they will need to deal with the security/privacy challenges they are likely to encounter in today's Big Data world. The tools are illustrated with relevant applications.

Content

- Overview of cyberthreats
- Exploiting vulnerabilities
- Authentication, access control, compartmentalization
- Basic applied cryptography
- Operational security practices and failures
- Machine learning and privacy
- Data anonymization and de-anonymization techniques
- Privacy enhancing technologies
- Blockchain and decentralization

Keywords

security, privacy, protection, intrusion, anonymization, cryptography

Learning Prerequisites**Required courses**

COM-301 Computer security
Basic Python programming or better
Basic networking knowledge

Learning Outcomes

By the end of the course, the student must be able to:

- Understand the most important classes of information security/privacy risks in today's "Big Data" environment
- Exercise a basic, critical set of "best practices" for handling sensitive information
- Exercise competent operational security practices
- Understand the key technical tools available for security/privacy protection

Expected student activities

Attending lectures, solving assigned problems and "hands-on" exercises, reading and demonstrating understanding of provided materials.

Assessment methods

- continuous control : 30% of the grade
- final exam : 70% of the grade

COM-404

Information theory and coding

Telatar Emre

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	7
Session	Winter
Semester	Fall
Exam	Written
Workload	210h
Weeks	14
Hours	6 weekly
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

The mathematical principles of communication that govern the compression and transmission of data and the design of efficient methods of doing so.

Content

1. Mathematical definition of information and the study of its properties.
2. Source coding: efficient representation of message sources.
3. Communication channels and their capacity.
4. Coding for reliable communication over noisy channels.
5. Multi-user communications: multi access and broadcast channels.
6. Lossy source coding : approximate representation of message sources.
7. Information Theory and statistics

Learning Outcomes

By the end of the course, the student must be able to:

- Formulate the fundamental concepts of information theory such as entropy, mutual information, channel capacity
- Elaborate the principles of source coding and data transmission
- Analyze source codes and channel codes
- Apply information theoretic methods to novel settings

Teaching methods

Ex cathedra + exercises

Assessment methods

With continuous control

Resources**Ressources en bibliothèque**

- [Elements of Information Theory / Cover](#)

Websites

- <http://moodle.epfl.ch/enrol/index.php?id=14593>

CS-430

Intelligent agents

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Data science minor	H	Opt.
Energy Management and Sustainability	MA1, MA3	Opt.
Financial engineering minor	H	Opt.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Obl.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	6 weekly
Lecture	3 weekly
Exercises	3 weekly
Number of positions	

Remark

pas donné en 2021-22

Summary

Software agents are widely used to control physical, economic and financial processes. The course presents practical methods for implementing software agents and multi-agent systems, supported by programming exercises, and the theoretical underpinnings including computational game theory.

Content

The course contains 4 main subject areas:

1) Basic models and algorithms for individual agents:

Models and algorithms for rational, goal-oriented behavior in agents: reactive agents, reinforcement learning, exploration-exploitation tradeoff, AI planning methods.

2) Multi-agent systems:

multi-agent planning, coordination techniques for multi-agent systems, distributed algorithms for constraint satisfaction.

3) Self-interested agents:

Models and algorithms for implementing self-interested agents motivated by economic principles: elements of computational game theory, models and algorithms for automated negotiation, social choice, mechanism design, electronic auctions and marketplaces.

4) Implementing multi-agent systems:

Agent platforms, ontologies and markup languages, web services and standards for their definition and indexing.

Learning Prerequisites**Recommended courses**

Intelligence Artificielle or another introductory course to AI

Learning Outcomes

By the end of the course, the student must be able to:

- Choose and implement methods for rational decision making in software agents, based on decision processes and AI

planning techniques

- Choose and implement methods for efficient rational decision making in teams of multiple software agents
- Model scenarios with multiple self-interested agents in the language of game theory
- Evaluate the feasibility of achieving goals with self-interested agents using game theory
- Design, choose and implement mechanisms for self-interested agents using game theory
- Implement systems of software agents using agent platforms

Teaching methods

Ex cathedra, practical programming exercises

Expected student activities

Lectures: 3 hours

Reading: 3 hours

Assignments/programming: 4 hours

Assessment methods

Mini-projects and exercises 40%, final exam 60%

Resources

Bibliography

Michael Wooldridge : An Introduction to MultiAgent Systems - Second Edition, John Wiley & Sons, 2009
Stuart Russell and Peter Norvig: Artificial Intelligence: A Modern Approach (2nd/3rd Edition), Prentice Hall Series in Artificial Intelligence, 2003/2009.

Ressources en bibliothèque

- [Artificial Intelligence: A Modern Approach / Russell](#)
- [An Introduction to MultiAgent Systems / Wooldridge](#)

Websites

- <http://liawww.epfl.ch/>
- <http://moodle.epfl.ch/>

CS-486

Interaction design

Pu Pearl

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	1 weekly
Project	1 weekly
Number of positions	

Summary

This course focuses on goal-directed design and interaction design, two subjects treated in depth in the Cooper book (see reference below). To practice these two methods, we propose a design challenge, which is to be carried out by a team of three students.

Content**Design methods for HCI**

What is HCI: its aims and goals

Design thinking

Goal-directed Design

Mental model and different types of users

Qualitative research and user interviews

User modeling: persona and empathy diagram

Scenarios, requirements and framework design

Visual design

Information Visualization design

Basic prototyping methods for HCI

Storyboarding

Context scenario

Interactive prototype

Video prototype

Human computer interaction evaluation methods

Cognitive walkthrough

Heuristic evaluation

Evaluation with users

Keywords

Interaction design, design thinking, design for playfulness, rapid prototyping techniques, evaluation with users.

Learning Prerequisites**Required courses**

Introduction to Visual Computing

Recommended courses

Open to students enrolled in the Master and PhD programs in IC.

Important concepts to start the course

Goal-direction design

Learning Outcomes

By the end of the course, the student must be able to:

- Interview users and elicit their needs using the goal-directed design method
- Design and implement interfaces and interactions
- Project management: set objectives and devise a plan to achieve them
- Group work skills: discuss and identify roles, and assume those roles including leadership
- Communication: writing and presentation skills

Teaching methods

Lectures, exercises, hands-on practice, design review

Expected student activities

Lectures, readings, design project, quiz

Assessment methods

Group project, presentation, mid-term exam

Resources

Bibliography

About Face 3: The Essentials of Interaction Design by Alan Cooper et al. (available as e-book at NEBIS)

Ressources en bibliothèque

- [About Face 3](#)

CS-431

Introduction to natural language processing

Chappelier Jean-Cédric, Rajman Martin

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Data science minor	H	Opt.
Digital Humanities	MA1, MA3	Opt.
Learning Sciences		Obl.
SC master EPFL	MA1, MA3	Opt.
UNIL - Sciences forensiques	H	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

The objective of this course is to present the main models, formalisms and algorithms necessary for the development of applications in the field of natural language information processing. The concepts introduced during the lectures will be applied during practical sessions.

Content

Several models and algorithms for automated textual data processing will be described: (1) morpho-lexical level: electronic lexica, spelling checkers, ...; (2) syntactic level: regular, context-free, stochastic grammars, parsing algorithms, ...; (3) semantic level: models and formalisms for the representation of meaning, ...

Several application domains will be presented: Linguistic engineering, Information Retrieval, Text mining (automated knowledge extraction), Textual Data Analysis (automated document classification, visualization of textual data).

Keywords

Natural Language Processing; Computational Linguistics; Part-of-Speech tagging; Parsing

Learning Outcomes

By the end of the course, the student must be able to:

- Compose key NLP elements to develop higher level processing chains
- Assess / Evaluate NLP based systems
- Choose appropriate solutions for solving typical NLP subproblems (tokenizing, tagging, parsing)
- Describe the typical problems and processing layers in NLP
- Analyze NLP problems to decompose them in adequate independent components

Teaching methods

Ex cathedra ; practical work on computer

Expected student activities

attend lectures and practical sessions, answer quizzes.

Assessment methods

4 quiz during semester 25%, final exam 75%

Supervision

Office hours	No
Assistants	No
Forum	No

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

1. M. Rajman editor, "*Speech and Language Engineering*", EPFL Press, 2006.
2. Daniel Jurafsky and James H. Martin, "*Speech and Language Processing*", Prentice Hall, 2008 (2nd edition)
3. Christopher D. Manning and Hinrich Schütze, "*Foundations of Statistical Natural Language Processing*", MIT Press, 2000
4. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, "*Introduction to Information Retrieval*", Cambridge University Press. 2008
5. Nitin Indurkha and Fred J. Damerau editors, "*Handbook of Natural Language Processing*", CRC Press, 2010 (2nd edition)

Ressources en bibliothèque

- [Handbook of Natural Language Processing / Indurkha](#)
- [Introduction to Information Retrieval / Manning](#)
- [Speech and Language Processing / Jurafsky](#)
- [Speech and Language Engineering / Rajman](#)
- [Foundations of Statistical Natural Language Processing / Manning](#)

Websites

- <http://coling.epfl.ch>

COM-490

Large-scale data science for real-world data

Bouillet Eric Pierre, Delgado Pamela, Sarni Sofiane, Verscheure Olivier

Cursus	Sem.	Type
Data Science	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.

Language	English
Credits	4
Withdrawal Session	Unauthorized Summer
Semester Exam	Spring During the semester
Workload	120h
Weeks	14
Hours	4 weekly
Practical work	4 weekly
Number of positions	

Summary

This hands-on course teaches the tools & methods used by data scientists, from researching solutions to scaling up prototypes to Spark clusters. It exposes the students to the entire data science pipeline, from data acquisition to extracting valuable insights applied to real-world problems.

Content**1. Crash-course in Python for data scientists**

- Main Python libraries for data scientists
- Interactive data science with web-based notebooks
- Reusable compute environments for reproducible science
- Homework: Curating data from a network of CO2 sensors

2. Distributed data wrangling at scale

- Understand the main constituents of an Apache Hadoop distribution
- Put Map-Reduce into practice
- Focus on HDFS, Hive and HBase and associated data storage formats
- Homework: Big data wrangling with massive travel data from SBB/CFF

3. Distributed processing with Apache Spark

- RDDs and best practices for order of operations, data partitioning, caching
- Data science packages in Spark: GraphX, MLlib, etc.
- Homework: Uncovering world events using Twitter hashtags

4. Real-time big data processing using Apache Spark Streaming

- Window-based processing of unbounded data
- Homework: Geospatial analysis and visualization of real-time train geolocation data from the Netherlands

5. Final project - Summing it all up

- Robust Journey Planning on the Swiss multimodal transportation network - Given a desired departure, or arrival time, your route planner will compute the fastest route between two stops within a provided uncertainty tolerance expressed as interquartiles. For instance, $\tilde{c}_{Q\%}$ what route from A to B is the fastest at least Q% of the time if I want to leave from A (resp. arrive at B) at instant t ?

Keywords

Data Science, IoT, Machine Learning, Predictive Modeling, Big Data, Stream Processing, Apache Spark, Hadoop, Large-Scale Data Analysis

Learning Prerequisites

Required courses

Students must have prior experience with Python

Recommended courses

Students must have prior experience with at least one general-purpose programming language.

Important concepts to start the course

It is recommended that students familiarize themselves with concepts in statistics and standard methods in machine learning.

Learning Outcomes

By the end of the course, the student must be able to:

- Use standard Big Data tools and Data Science libraries
- Carry out real-world projects with a variety of real datasets, both at rest and in motion
- Design large scale data science and engineering problems
- Present tangible solution to a real-world Data Science problem

Transversal skills

- Demonstrate a capacity for creativity.
- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Write a scientific or technical report.

Teaching methods

- Hands-on lab sessions
- Homework assignments
- Final project

... using real-world datasets and Cloud Compute & Storage Services

Expected student activities

- STUDY : Attend the lab sessions
- WORK : Complete homework assignments
- ENGAGE : Contribute to the interactive nature of the class
- COLLABORATE : Work in small groups to provide solutions to real-world problems
- EXPLAIN : Present ideas and results to the class

Assessment methods

- 60% continuous assessment during the semester

- 40% final project, done in small groups

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

- Python Data Science Handbook: Essential Tools for Working with Data by Jake VanderPlas, O'Reilly Media, November 2016
- pyGAM - <https://github.com/dswah/pyGAM>

A list of additional readings will be distributed at the beginning of the course

Websites

- <https://dslab2020.github.io>

CS-526

Learning theory

Macris Nicolas, Urbanke Rüdiger

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

Machine learning and data analysis are becoming increasingly central in many sciences and applications. This course concentrates on the theoretical underpinnings of machine learning.

Content

- Basics : statistical learning framework, Probably Approximately Correct (PAC) learning, learning with a finite number of classes, Vapnik-Chervonenkis (VC) dimension, non-uniform learnability, complexity of learning.
- Neural Nets : representation power of neural nets, learning and stability, PAC Bayes bounds.
- Graphical model learning.
- Non-negative matrix factorization, Tensor decompositions and factorization.
- Learning mixture models.

Learning Prerequisites**Recommended courses**

- Analysis I, II, III
- Linear Algebra
- Machine learning
- Probability
- Algorithms (CS-250)

Learning Outcomes

By the end of the course, the student must be able to:

- Explain the framework of PAC learning
- Explain the importance basic concepts such as VC dimension and non-uniform learnability
- Describe basic facts about representation of functions by neural networks
- Describe recent results on specific topics e.g., graphical model learning, matrix and tensor factorization, learning mixture models

Teaching methods

- Lectures
- Exercises

Expected student activities

- Attend lectures
- Attend exercises sessions and do the homework

Assessment methods

Final exam and graded homeworks

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes
Others	Course website

MATH-341

Linear models

Panaretos Victor

Cursus	Sem.	Type
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Mathematics	BA5	Opt.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

Regression modelling is a basic tool of statistics, because it describes how a random variable of interest may depend on other variables. The aim of this course is to familiarize students with the basis of regression modelling, and of some related topics.

Content

- Properties of the multivariate Gaussian distribution and related quadratic forms.
- Gaussian linear regression: likelihood, least squares, geometrical interpretation.
- Distribution theory, confidence and prediction intervals.
- Gauss-Markov theorem.
- Model checking and validation: residual diagnostics, outliers and leverage points.
- Analysis of variance.
- Model selection: bias/variance tradeoff, stepwise procedures, information-based criteria.
- Multicollinearity and penalised estimation: ridge regression, LASSO.
- Robust regression and M-estimation.
- Nonparametric regression and smoothing splines.

Learning Prerequisites**Recommended courses**

Analysis, Linear Algebra, Probability, Statistics

Learning Outcomes

By the end of the course, the student must be able to:

- Recognize when a linear model is appropriate to model dependence
- Interpret model parameters both geometrically and in applied contexts
- Estimate the parameters determining a linear model from empirical observations
- Test hypotheses related to the structural characteristics of a linear model
- Construct confidence bounds for model parameters and model predictions
- Analyze variation into model components and error components
- Contrast competing linear models in terms of fit and parsimony
- Construct linear models to balance bias, variance and interpretability

- Assess / Evaluate the fit of a linear model to data and the validity of its assumptions.
- Prove basic results related to the statistical theory of linear models

Teaching methods

Lectures ex cathedra, exercises in class, take-home projects

Assessment methods

Continuous control, final exam.

Seconde tentative : Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Ressources en bibliothèque

- [Statistical Models / Davison](#)
- [Applied Regression Analysis / Draper](#)
- [Methods and Applications of Linear Models / Hocking](#)

CS-433

Machine learning

Flammarion Nicolas, Jaggi Martin

Cursus	Sem.	Type
Biocomputing minor	H	Opt.
Civil & Environmental Engineering		Opt.
Communication systems minor	H	Opt.
Computational Neurosciences minor	H	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Obl.
Data science minor	H	Opt.
Digital Humanities	MA1, MA3	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Obl.
Life Sciences Engineering	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	7
Session	Winter
Semester	Fall
Exam	Written
Workload	210h
Weeks	14
Hours	6 weekly
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

Machine learning methods are becoming increasingly central in many sciences and applications. In this course, fundamental principles and methods of machine learning will be introduced, analyzed and practically implemented.

Content

1. *Basic regression and classification concepts and methods: Linear models, overfitting, linear regression, Ridge regression, logistic regression, k-NN, SVMs and kernel methods*
2. *Fundamental concepts: cost-functions and optimization, cross-validation and bias-variance trade-off, curse of dimensionality.*
3. *Neural Networks: Representation power, backpropagation, activation functions, CNN, regularization, data augmentation, dropout*
4. *Unsupervised learning: k-means clustering, gaussian mixture models and the EM algorithm. Basics of self-supervised learning*
5. *Dimensionality reduction: PCA and matrix factorization, word embeddings*
6. *Advanced methods: Adversarial learning, Generative adversarial networks*

Keywords

- *Machine learning, pattern recognition, deep learning, neural networks, data mining, knowledge discovery, algorithms*

Learning Prerequisites**Required courses**

- Analysis I, II, III
- Linear Algebra
- Probability and Statistics (MATH-232)
- Algorithms (CS-250)

Recommended courses

- *Introduction to machine learning (CS-233)*
- *...or similar bachelor lecture from other sections*

Important concepts to start the course

- *Basic probability and statistics (conditional and joint distribution, independence, Bayes rule, random variables, expectation, mean, median, mode, central limit theorem)*
- *Basic linear algebra (matrix/vector multiplications, systems of linear equations, SVD)*
- *Multivariate calculus (derivative w.r.t. vector and matrix variables)*
- *Basic Programming Skills (labs will use Python)*

Learning Outcomes

By the end of the course, the student must be able to:

- Define the following basic machine learning models: Regression, classification, clustering, dimensionality reduction, neural networks, time-series analysis
- Explain the main differences between them
- Implement algorithms for these machine learning models
- Optimize the main trade-offs such as overfitting, and computational cost vs accuracy
- Implement machine learning methods to real-world problems, and rigorously evaluate their performance using cross-validation. Experience common pitfalls and how to overcome them
- Explain and understand the fundamental theory presented for ML methods
- Conduct a real-world interdisciplinary machine learning project, in collaboration with application domain experts

Teaching methods

- Lectures
- Lab sessions
- Course Projects

Expected student activities

Students are expected to:

- attend lectures
- attend lab sessions and work on the weekly theory and coding exercises
- work on projects using the code developed during labs, in small groups

Assessment methods

- Written final exam
- Continuous control (Course projects)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

- Christopher Bishop, Pattern Recognition and Machine Learning
- Kevin Murphy, Machine Learning: A Probabilistic Perspective
- Shai Shalev-Shwartz, Shai Ben-David, Understanding Machine Learning
- Michael Nielsen, Neural Networks and Deep Learning
- (Jerome Friedman, Robert Tibshirani, Trevor Hastie, The elements of statistical learning : data mining, inference, and prediction)

Ressources en bibliothèque

- [Linear algebra and learning from data](#)
- [The elements of statistical learning : data mining, inference, and prediction / Friedman](#)
- [Understanding Machine Learning / Shalev-Shwartz](#)
- [Neural Networks and Deep Learning / Nielsen](#)
- [Machine Learning: A Probabilistic Perspective / Murphy](#)
- [Pattern Recognition and Machine Learning / Bishop](#)

Notes/Handbook

https://github.com/epfml/ML_course

Websites

- <https://www.epfl.ch/labs/mlo/machine-learning-cs-433/>

CS-421

Machine learning for behavioral data

Käser Tanja

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

Computer environments such as educational games, interactive simulations, and web services provide large amounts of data, which can be analyzed and serve as a basis for adaptation. This course will cover the core methods of user modeling and personalization, with a focus on educational data.

Content

The users of computer environments such as intelligent tutoring systems, interactive games, and web services are often very heterogeneous and therefore it is important to adapt to their specific needs and preferences.

This course will cover the core methods of adaptation and personalization, with a focus on educational data. Specifically we will discuss approaches to the task of accurately modeling and predicting human behavior within a computer environment. Furthermore, we will also discuss data mining techniques with the goal to gain insights into human behavior. We will cover the theories and methodologies underlying the current approaches and then also look into the most recent developments in the field.

1. 'Cycle' of adaptation : representation, prediction, intervention (e.g. recommendation)
2. Data Processing and Interpretation (missing data, feature transformations, distribution fitting)
3. Performance evaluation (cross-validation, error measures, statistical significance, overfitting)
4. Representation & Prediction (probabilistic graphical models, recurrent neural networks, logistic models, clustering-classification approaches)
5. Recommendation (collaborative filtering, content-based recommendations, multi-armed bandits)
6. Stealth Assessment (seamless detection of user traits)
7. Multimodal analytics (represent & analyze data from non-traditional sources. i.e. sensors, classroom analytics, human-robot interaction)

Learning Prerequisites**Required courses**

The student must have passed a course in probability and statistics and a course including a programming project

Recommended courses

- CS-433 Machine learning or
- CS-233a / CS-233b Introduction to machine learning

Important concepts to start the course

Probability and statistics, basic machine learning knowledge, algorithms and programming, Python

Learning Outcomes

By the end of the course, the student must be able to:

- Explain the main machine learning approaches to personalization, describe their advantages and disadvantages and explain the differences between them
- Implement algorithms for these machine learning models
- Apply them to real-world data
- Assess / Evaluate their performance
- Explain and understand the fundamental theory underlying the presented machine learning models

Teaching methods

- Lectures
- Weekly lab sessions
- Course project

Expected student activities

- Attend the lectures
- Attend the lab sessions and work on the homework assignments
- Project work

Assessment methods

- Project work (50%)
- Final exam (50%)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

COM-516

Markov chains and algorithmic applications

Lévêque Olivier, Macris Nicolas

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Electrical Engineering		Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

The study of random walks finds many applications in computer science and communications. The goal of the course is to get familiar with the theory of random walks, and to get an overview of some applications of this theory to problems of interest in communications, computer and network science.

Content

Part 1: Markov chains (~6 weeks):

- basic properties: irreducibility, periodicity, recurrence/transience, stationary and limiting distributions,
- ergodic theorem: coupling method
- detailed balance
- convergence rate to the equilibrium, spectral gap, mixing times
- cutoff phenomenon

Part 2: Sampling (~6 weeks)

- classical methods, importance and rejection sampling
- Markov Chain Monte Carlo methods, Metropolis-Hastings algorithm, Glauber dynamics, Gibbs sampling
- applications: function minimization, coloring problem, satisfiability problems, Ising models
- coupling from the past and exact simulation

Keywords

random walks, stationarity, ergodic, convergence, spectral gap, mixing time, sampling, Markov chain Monte Carlo, coupling from the past

Learning Prerequisites**Required courses**

Basic probability course
Basic linear algebra and calculus courses

Recommended courses

Stochastic Models for Communications (COM-300)

Important concepts to start the course

Good knowledge of probability and analysis.
Having been exposed to the theory of Markov chains.

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze the behaviour of a random walk
- Assess / Evaluate the performance of an algorithm on a graph
- Implement efficiently various sampling methods

Teaching methods

ex-cathedra course

Expected student activities

active participation to exercise sessions and implementation of a sampling algorithm

Assessment methods

midterm (20%), mini-project (20%), final exam (60%)

Resources

Bibliography

Various references will be given to the students during the course, according to the topics discussed in class.

Ressources en bibliothèque

- [Probability and random processes / Grimmett](#)

Notes/Handbook

Lecture notes will be provided

Websites

- <https://moodle.epfl.ch/course/view.php?id=15016>

Prerequisite for

This course is not so to speak a prerequisite for other courses, but could complement well the course COM-512 on Networks out of control, as well as other courses in statistics.

COM-514

Mathematical foundations of signal processing

Bejar Haro Benjamin, Fageot Julien René Pierre, Simeoni Matthieu

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA1, MA3	Opt.
Systems Engineering minor	H	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	5 weekly
Lecture	3 weekly
Exercises	2 weekly
Number of positions	

Summary

Signal processing tools are presented from an intuitive geometric point of view which is at the heart of all modern signal processing techniques. The student will develop the mathematical depth and rigor needed for the study of advanced topics in signal processing and approximation theory.

Content

Sequences, Discrete-Time Systems, Functions and Continuous-Time Systems (review of discrete-time Fourier transform; DFT; Fourier transform and Fourier series).

From Euclid to Hilbert: Linear Algebra Fundamentals for Representation Theory (vector spaces; Hilbert spaces; approximations, projections and decompositions; bases and frames; linear operators; adjoint; generalized inverses; matrix representations; computational aspects)

Sampling and Interpolation (sampling and interpolation with normal and non orthogonal vectors, sequences and functions; sampling and interpolation of bandlimited sequences and functions)

Polynomial and Spline Approximation (Legendre and Chebyshev polynomials; Lagrange interpolation; minimax approximation; Taylor expansions; B-splines)

Regularized Inverse Problems (regularized convex optimisation; Tikhonov regularisation; penalised basis pursuit; proximal algorithms; pseudo-differential operators and L-splines; representer theorems for continuous inverse problems with Tikhonov penalties)

Learning Prerequisites**Required courses**

Signal processing for communications (or Digital signal processing on Coursera)
Linear Algebra I and II (or equivalent).

Recommended courses

Signals and Systems

Important concepts to start the course

Good knowledge of linear algebra concepts. Basics of Fourier analysis and signal processing. Good knowledge of Python and its scientific packages (Numpy, Scipy).

Learning Outcomes

By the end of the course, the student must be able to:

- Master the right tools to tackle advanced signal and data processing problems
- Develop an intuitive understanding of signal processing through a geometrical approach
- Get to know the applications that are of interest today
- Learn about topics that are at the forefront of signal processing research
- Identify and implement the algorithm best suited to solve a given convex optimisation problem
- Assess the computational cost and numerical stability of a numerical solver

Transversal skills

- Collect data.
- Write a scientific or technical report.
- Use a work methodology appropriate to the task.
- Demonstrate the capacity for critical thinking
- Use both general and domain specific IT resources and tools

Teaching methods

Ex cathedra with exercises, homeworks and practicals.

Expected student activities

Attending lectures, completing exercises.

Assessment methods

homeworks and project assignement 50%, final exam (written) 50%

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

M. Vetterli, J. Kovacevic and V. Goyal, "*Signal Processing: Foundations*", Cambridge U. Press, 2014.
Available in open access at <http://www.fourierandwavelets.org>

Ressources en bibliothèque

- [Signal Processing: Foundations / Vetterli](#)

Moodle Link

- <http://moodle.epfl.ch/course/view.php?id=13431>

EE-556

Mathematics of data: from theory to computation

Cevher Volkan

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
MNIS	MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	6 weekly
Lecture	3 weekly
Practical work	3 weekly
Number of positions	

Summary

This course reviews recent advances in continuous optimization and statistical analysis along with models. We provide an overview of the emerging learning formulations and their guarantees, describe scalable solution techniques, and illustrate the role of parallel and distributed computation.

Content

The course consists of the following topics

Lecture 1: Introduction. The role of models and data. Maximum-likelihood formulations. Sample complexity bounds for estimation and prediction.

Lecture 2: The role of computation. Challenges to optimization algorithms. Optimality measures. Structures in optimization. Gradient descent. Convergence rate of gradient descent.

Lecture 3: Optimality of convergence rates. Accelerated gradient descent. Concept of total complexity. Stochastic gradient descent.

Lecture 4: Concise signal models. Compressive sensing. Sample complexity bounds for estimation and prediction. Challenges to optimization algorithms for non-smooth optimization.

Lecture 5: Introduction to proximal-operators. Proximal gradient methods. Linear minimization oracles. Conditional gradient method for constrained optimization.

Lecture 6: Time-data trade-offs. Variance reduction for improving trade-offs.

Lecture 7: A mathematical introduction to deep learning. Double descent curves and over-parameterization. Implicit regularization.

Lecture 8: Structures in non-convex optimization. Optimality measures. Escaping saddle points. Adaptive gradient methods.

Lecture 9: Adversarial machine learning and generative adversarial networks (GANs). Wasserstein GAN. Difficulty of minimax optimization.

Lecture 10: Primal-dual optimization-I: Fundamentals of minimax problems. Pitfalls of gradient descent-ascent approach.

Lecture 11: Primal-dual optimization-II: Extra gradient method. Chambolle-Pock algorithm. Stochastic primal-dual methods.

Lecture 12: Primal-dual III: Lagrangian gradient methods. Lagrangian conditional gradient methods.

Recitation 1: Generalized linear models. Logistic regression.

Recitation 2: Computation of Gradients. Reading convergence plots. Helpful definitions on linear algebra.

Recitation 3: Activation functions in neural networks. Backpropagation. Introduction to pytorch.

Keywords

Machine Learning. Signal Processing. Optimization. Statistical Analysis. Linear and non-linear models. Algorithms. Data and computational trade-offs.

Learning Prerequisites

Required courses

Previous coursework in calculus, linear algebra, and probability is required. Familiarity with optimization is useful.

Important concepts to start the course

Previous coursework in calculus, linear algebra, and probability is required. Familiarity with optimization is useful.

Learning Outcomes

By the end of the course, the student must be able to:

- Choose an appropriate convex formulation for a data analytics problem at hand
- Estimate the underlying data size requirements for the correctness of its solution
- Implement an appropriate convex optimization algorithm based on the available computational platform
- Decide on a meaningful level of optimization accuracy for stopping the algorithm
- Characterize the time required for their algorithm to obtain a numerical solution with the chosen accuracy

Assessment methods

Homework assignments. (Continuous control)

COM-512

Networks out of control

Grossglauser Matthias, Thiran Jean-Philippe

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Electrical Engineering		Opt.
SC master EPFL	MA2, MA4	Opt.
Systems Engineering minor	E	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	3 weekly
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal

Summary

The goal of this class is to acquire mathematical tools and engineering insight about networks whose structure is random, as well as learning and control techniques applicable to such network data.

Content

- Random graph models: Erdős-Renyi, random regular, geometric, percolation, small worlds, stochastic block model
- Learning graphs from data: centrality metrics, embeddings, Hawkes processes, network alignment
- Control of processes on graphs: epidemics, navigation

Keywords

Random graphs, network data, machine learning, graph processes.

Learning Prerequisites**Required courses**

Stochastic models in communication (COM-300), or equivalent.

Important concepts to start the course

Basic probability and statistics; Markov chains; basic combinatorics.

Teaching methods

Ex cathedra lectures, exercises, mini-project

Expected student activities

Attending lectures, bi-weekly homeworks, mini-project incl. student presentation at the end of semester, final exam.

Assessment methods

1. Homeworks 10%
2. Mini-project 40%

3. Final exam 50%.

Supervision

Office hours	Yes
Assistants	Yes
Forum	No

Resources

Bibliography

- A. D. Barbour, L. Holst and S. Janson, Poisson Approximation, Oxford Science Publications, 1992.
- B. Bollobas, Random Graphs (2nd edition), Cambridge University Press, 2001.
- R. Durrett, Random Graph Dynamics, Cambridge University Press, 2006 (electronic version).
- D. Easley, J. Kleinberg. Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010 (electronic version).
- G. Grimmett, Percolation (2nd edition), Springer, 1999.
- S. Janson, T. Luczak, A. Rucinski, Random Graphs, Wiley, 2000.
- R. Meester and R. Roy, Continuum Percolation, Cambridge University Press, 1996.

Ressources en bibliothèque

- [Random Graphs / Bollobas](#)
- [Random Graphs / Janson](#)
- [Continuum Percolation / Meester](#)
- [Random Graph Dynamics / Durrett](#)
- [Networks, Crowds and Markets / Easley](#)
- [Poisson Approximation / Barbour](#)
- [Percolation / Grimmett](#)

Notes/Handbook

Class notes will be available on the course website.

Websites

- <http://icawww1.epfl.ch/class-nooc/>

CS-439

Optimization for machine learning

Flammarion Nicolas, Jaggi Martin

Cursus	Sem.	Type
Computational science and Engineering	MA2, MA4	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Obl.
Data science minor	E	Opt.
Electrical Engineering		Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	Written
Workload	150h
Weeks	14
Hours	5 weekly
Lecture	2 weekly
Exercises	2 weekly
Practical work	1 weekly
Number of positions	

Summary

This course teaches an overview of modern optimization methods, for applications in machine learning and data science. In particular, scalability of algorithms to large datasets will be discussed in theory and in implementation.

Content

This course teaches an overview of modern optimization methods, for applications in machine learning and data science. In particular, scalability of algorithms to large datasets will be discussed in theory and in implementation.

Fundamental Contents:

- Convexity, Gradient Methods, Proximal algorithms, Stochastic and Online Variants of mentioned methods, Coordinate Descent Methods, Subgradient Methods, Non-Convex Optimization, Frank-Wolfe, Accelerated Methods, Primal-Dual context and certificates, Lagrange and Fenchel Duality, Second-Order Methods, Quasi-Newton Methods, Gradient-Free and Zero-Order Optimization.

Advanced Contents:

- Non-Convex Optimization: Convergence to Critical Points, Saddle-Point methods, Alternating minimization for matrix and tensor factorizations
- Parallel and Distributed Optimization Algorithms, Synchronous and Asynchronous Communication
- Lower Bounds

On the practical side, a graded **group project** allows to explore and investigate the real-world performance aspects of the algorithms and variants discussed in the course.

Keywords

Optimization, Machine learning

Learning Prerequisites**Recommended courses**

- CS-433 Machine Learning

Important concepts to start the course

- Previous coursework in calculus, linear algebra, and probability is required.

- Familiarity with optimization and/or machine learning is useful.

Learning Outcomes

By the end of the course, the student must be able to:

- Assess / Evaluate the most important algorithms, function classes, and algorithm convergence guarantees
- Compose existing theoretical analysis with new aspects and algorithm variants.
- Formulate scalable and accurate implementations of the most important optimization algorithms for machine learning applications
- Characterize trade-offs between time, data and accuracy, for machine learning methods

Transversal skills

- Use both general and domain specific IT resources and tools
- Summarize an article or a technical report.

Teaching methods

- Lectures
- Exercises with Theory and Implementation Assignments

Expected student activities

Students are expected to:

- Attend the lectures and exercises
- Give a short scientific presentation about a research paper
- Read / watch the pertinent material
- Engage during the class, and discuss with other colleagues

Assessment methods

- Continuous control (course project)
- Final Exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)
No

Websites

- https://github.com/epfml/OptML_course

Videos

- <https://www.youtube.com/playlist?list=PL4O4bXkl-fAeYrsBqTUYn2xMjJAqlFQzX>

COM-508

Optional project in data science

Profs divers *

Cursus	Sem.	Type
Data Science	MA1, MA2, MA3, MA4	Opt.
Data science minor	E, H	Opt.

Language	English
Credits	8
Session	Winter, Summer
Semester	Fall
Exam	During the semester
Workload	240h
Weeks	14
Hours	2 weekly
Project	2 weekly
Number of positions	

Remark

for students doing a minor in Data Science : Registration upon authorization of the section. Only for 2nd year Master students. Supervision by an IC professor

Summary

Individual research during the semester under the guidance of a professor or an assistant.

Content

Subject to be chosen among the themes proposed on the web site :
<https://www.epfl.ch/schools/ic/education/master/data-science/projects-lab-ds/>

Learning Outcomes

By the end of the course, the student must be able to:

- Organize a project
- Assess / Evaluate one's progress through the course of the project
- Present a project

Transversal skills

- Write a literature review which assesses the state of the art.
- Write a scientific or technical report.

Teaching methods

Individual and independant work, under the guidance of a professor or an assistant.

Expected student activities

Written report due within the allotted time.

Information on the format and the content of the report is provided by the proeject supervisor.

Assessment methods

Autumn : The written report must be returned to the laboratory no later than **the Friday of the second week** after the end of the classes.

Spring : The written report must be returned to the laboratory no later than **the Friday of the first week** after the end of

the classes.

The oral presentation is organized by the laboratory.

MATH-447

Risk, rare events and extremes

Davison Anthony

Cursus	Sem.	Type
Data Science	MA1, MA3	Opt.
Data science minor	H	Opt.
Financial engineering	MA1, MA3	Opt.
Ing.-math	MA1, MA3	Opt.
Mathématicien	MA1, MA3	Opt.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

Modelling of rare events, such as stock market crashes, storms and catastrophic structural failures, is important. This course will describe the special models and methods that are relevant to such modelling, including the mathematical bases, statistical tools and applications.

Content

- **Mathematical bases:** behaviour of maxima and threshold exceedances in large samples, both for independent and dependent data. Poisson process modelling.
- **Statistical methods:** modelling using the GEV and GP distributions, for independent and dependent data. Likelihood and Bayesian inference. Non-stationarity. Extremal coefficients. Multivariate extreme-value distributions. Max-stable processes.
- **Applications:** Environmental, financial, and engineering applications. Use of R for extremal modelling.

Learning Prerequisites**Important concepts to start the course**

Probability and statistics at the level of second-year bachelor (mathematics), plus further knowledge of statistics and stochastic processes.

Learning Outcomes

By the end of the course, the student must be able to:

- Recognize situations where statistical analysis of extrema is appropriate
- Manipulate mathematical objects related to the study of extrema
- Analyze empirical data on extremes using appropriate statistical methods
- Construct appropriate statistical models for extremal data
- Interpret such models in terms of underlying phenomena
- Infer properties of real systems in terms of probability models for extremes

Teaching methods

Lectures, theoretical and computational exercises in class and at home.

Assessment methods

Mini-project, final exam.

Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

Supervision

Assistants Yes

Resources

Bibliography

Coles, S. G. (2001) An Introduction to the Statistical Modelling of Extreme Values. Springer.
Beirlant, J, Goegebeur. Y., Teugels. J. and Segers. J. (2004) Statistics of Extremes: Theory and Applications. Wiley.

MATH-441

Robust and nonparametric statistics

Cursus	Sem.	Type
Data Science	MA2, MA4	Opt.
Ing.-math	MA2, MA4	Opt.
Mathématicien	MA2	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	Oral
Workload	150h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Remark

Cours donné en alternance tous les deux ans (pas donné en 2021-22)

Summary

In the decades from 1930 to 1950, many rank-based statistics were introduced. These methods were received with much interest, because they worked under weak conditions. Starting in the late 1950, a theory of robustness was added. The course gives an overview of these two approaches to data analysis.

Content**I. Robust Statistics**

- Global and local robustness indicators: Breakdown point, influence function
- Hampel's lemma
- Huber's theory: M-estimators, L-estimators
- Robust tests
- Robust regression

II. Linear Rank Tests

- Test of Mann-Whitney-Wilcoxon and general linear rank tests: asymptotic theory, R-estimators
- Rank correlations
- U-statistics
- Comparison of tests: Pitman efficacy
- Permutation tests

III. Estimation of smooth functions

- Curve fitting: polynomial regression, splines
- Smoothing: non parametric estimation, degree of smoothness, bias vs. variance, penalization
- Kernel estimators: definition, properties
- Smoothing splines
- Local regression
- Wavelets

Learning Prerequisites

Required courses

Introduction to Probability, Introduction to Statistics

Learning Outcomes

By the end of the course, the student must be able to:

- Expound the content of the course.
- Apply the statistical methods explained in the course.
- Sketch the proofs of the theoretical results given in the course.
- Choose the appropriate robust or non parametric methods for a given data analysis problem.
- Differentiate between robust and non-parametric methods.
- Generalize the tools treated in the course to other problems.
- Apply spline and kernel smoothers.
- Apply M-estimators in a variety of situations.

Transversal skills

- Assess one's own level of skill acquisition, and plan their on-going learning goals.
- Manage priorities.

Teaching methods

Ex cathedra lecture and exercises in the classroom

Expected student activities

Do all the exercises. Prepare each week for the course. Participate actively in the course.

Assessment methods

Oral exam.

Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

Resources

Bibliography

Introduction to the theory of nonparametric statistics by R.H. Randles and D.A. Wolfe, Wiley.

All of nonparametric statistics by L. Wasserman, Springer.

Robust Statistics: The approach based on influence functions by F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, W.A. Stahel, Wiley.

Robust Statistics by P.J. Huber, Wiley (second edition).

Robust Statistics: Theory and Methods by D.R. Martin, M. Salibian-Barrera, R.A. Maronna, V.J. Yohai, Wiley.

Ressources en bibliothèque

- [Introduction to the theory of nonparametric statistics / Randles & Wolfe](#)
- [Robust Statistics / Hampel & al.](#)
- [Robust Statistics / Martin & al.](#)
- [All of nonparametric statistics / Wasserman](#)

COM-412

Semester project in Data Science

Profs divers *

Cursus	Sem.	Type
Data Science	MA1, MA2, MA3, MA4	Obl.

Language	English
Credits	12
Session	Winter, Summer
Semester	Fall
Exam	During the semester
Workload	360h
Weeks	14
Hours	2 weekly
Project	2 weekly
Number of positions	

Summary

Individual research during the semester under the guidance of a professor or an assistant.

Content

Subject to be chosen among the themes proposed on the web site :
<https://www.epfl.ch/schools/ic/education/master/data-science/projects-lab-ds/>

Learning Outcomes

By the end of the course, the student must be able to:

- Organize a project
- Assess / Evaluate one's progress through the course of the project
- Present a project

Transversal skills

- Write a literature review which assesses the state of the art.
- Write a scientific or technical report.

Expected student activities

Written report due within the allotted time.

Information on the format and the content of the report is provided by the project supervisor.

Assessment methods

Autumn : The written report must be returned to the laboratory no later than **the Friday of the second week** after the end of classes.

Spring : The written report must be returned to the laboratory no later than **the Friday of the first week** after the end of classes.

The oral presentation is organized by the laboratory.

Resources**Websites**

- <https://www.epfl.ch/schools/ic/education/master/semester-project-msc/>

CS-412

Software security

Payer Mathias

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	6 weekly
Lecture	3 weekly
Exercises	2 weekly
Practical work	1 weekly
Number of positions	

Summary

This course focuses on software security fundamentals, secure coding guidelines and principles, and advanced software security concepts. Students learn to assess and understand threats, learn how to design and implement secure software systems, and get hands-on experience with security pitfalls.

Content

This course focuses on software security fundamentals, secure coding guidelines and principles, and advanced software security concepts. Students will learn to assess and understand threats, learn how to design and implement secure software systems, and get hands-on experience with common security pitfalls.

Software running on current systems is exploited by attackers despite many deployed defence mechanisms and best practices for developing new software. In this course students will learn about current security threats, attack vectors, and defence mechanisms on current systems. The students will work with real world problems and technical challenges of security mechanisms (both in the design and implementation of programming languages, compilers, and runtime systems).

- Secure software lifecycle: design, implementation, testing, and deployment
- Basic software security principles
- Reverse engineering : understanding code
- Security policies: Memory and Type safety
- Software bugs and undefined behavior
- Attack vectors: from flaw to compromise
- Runtime defense: mitigations
- Software testing: fuzzing and sanitization
- Focus topic : Web security
-

Focus topic : Mobile security

Keywords

Software security, mitigation, software testing, sanitization, fuzzing

Learning Prerequisites

Required courses

- COM-402 Information security and privacy

Important concepts to start the course

Basic computer literacy like system administration, build systems, basic C/C++ programming skills, debugging, and development skills. Understanding of virtual machines and operating systems.

Learning Outcomes

By the end of the course, the student must be able to:

- Explain the top 20 most common weaknesses in software security and understand how such problems can be avoided in software.
- Identify common security threats, risks, and attack vectors for software systems.
- Assess / Evaluate current security best practices and defense mechanisms for current software systems. Become aware of limitations of existing defense mechanisms and how to avoid them.
- Identify security problems in source code and binaries, assess the associated risks, and reason about their severity and exploitability.
- Assess / Evaluate the security of given source code or applications.

Transversal skills

- Identify the different roles that are involved in well-functioning teams and assume different roles, including leadership roles.
- Keep appropriate documentation for group meetings.
- Summarize an article or a technical report.
- Access and evaluate appropriate sources of information.
- Write a scientific or technical report.
- Make an oral presentation.

Teaching methods

The lectures are denser early in the semester, then tapering off before the end. They may be peppered with occasional short surprise quizzes that are not mandatory but may earn points for successful participants. They are backed up by PDF files of all the lecture material, as well as a few textbook recommendations.

The exercises sessions start slowly early in the semester but pick up and occupy all time towards the end. They consist mostly of paper questions involving the analysis, critical review, and occasional correction of software. They include a reading, writing, and presentation assignment.

Expected student activities

Students are encouraged to attend lectures and exercise sessions. In addition to normal studying of the lecture and practice of the exercises, the reading assignment consists of analyzing a few suggested scientific papers on a large selection of topics; the presentation assignment consists of holding a 15-minute presentation on the selected topic; and the writing assignment of documenting what was learned in a term paper due at the end of the semester.

Assessment methods

The grade will continuously be evaluated through a combination of practical assignments in the form of several labs and theoretical quizzes and assignments throughout the semester. The labs will account for 70%, the quizzes and assignments to 30%.

Supervision

Office hours	Yes
Assistants	Yes
Forum	No

Resources

Notes/Handbook

Software Security: Principles, Policies, and Protection (SS3P, by Mathias Payer)
<http://nebelwelt.net/SS3P/>

MATH-486

Statistical mechanics and Gibbs measures

Friedli Sacha

Cursus	Sem.	Type
Data Science	MA2, MA4	Opt.
Ing.-math	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	Oral
Workload	150h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

This course provides a rigorous introduction to the ideas, methods and results of classical statistical mechanics, with an emphasis on presenting the central tools for the probabilistic description of infinite lattice systems.

Content

The goals of this course are to present

- the probabilistic description of large systems with interacting components,
- the mathematical description of phase transitions occurring in certain discrete models (Curie-Weiss, Ising model, long-range models, etc.)
- the general theory of infinite-volume Gibbs measures (the so-called Dobrushin-Lanford-Ruelle approach)

If times permits, and depending on the interest of the participants, we consider the peculiar properties of certain models with an underlying continuous symmetry (Gaussian free field, Mermin-Wagner Theorem for $O(n)$ models).

This course is companion to the course "lattice models", where discrete models are also considered, but with an emphasis on different aspects.

The lectures will be largely based on the book *Statistical mechanics of lattice systems; a concrete mathematical introduction*, by S. Friedli and Y. Velenik (Cambridge University Press, 2017)

Keywords

statistical mechanics, phase transitions, Gibbs measures, entropy, Ising model, Gaussian Free Field

Learning Prerequisites**Required courses**

- Analyse 1 et 2
- Théorie de la Mesure
- Probabilités

Assessment methods

Examen oral.

Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

Resources

Bibliography

Statistical mechanics of lattice systems; a concrete mathematical introduction, by S. Friedli and Y. Velenik (Cambridge University Press, 2017)

Gibbs Measures and Phase Transitions, by H.-O. Georgii (De Gruyter Studies in Mathematics Vol. 9. Berlin: de Gruyter 1988)

Ressources en bibliothèque

- [Gibbs Measures and Phase Transitions / Georgii](#)
- [\(electronic version\)](#)
- [Statistical mechanics of lattice systems / Friedli & Velenik](#)

MATH-442

Statistical theory

Koch Erwan

Cursus	Sem.	Type
Data Science	MA1, MA3	Opt.
Ing.-math	MA1, MA3	Opt.
Mathématicien	MA1, MA3	Opt.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

The course aims at developing certain key aspects of the theory of statistics, providing a common general framework for statistical methodology. While the main emphasis will be on the mathematical aspects of statistics, an effort will be made to balance rigor and intuition.

Content

- Stochastic convergence and its use in statistics: modes of convergence, weak law of large numbers, central limit theorem.
- Formalization of a statistical problem : parameters, models, parametrizations, sufficiency, ancillarity, completeness.
- Point estimation: methods of estimation, bias, variance, relative efficiency.
- Likelihood theory: the likelihood principle, asymptotic properties, misspecification of models, the Bayesian perspective.
- Optimality: decision theory, minimum variance unbiased estimation, Cramér-Rao lower bound, efficiency, robustness.
- Testing and Confidence Regions: Neyman-Pearson setup, likelihood ratio tests, uniformly most powerful (UMP) tests, duality with confidence intervals, confidence regions, large sample theory, goodness-of-fit testing.

Learning Prerequisites**Recommended courses**

Real Analysis, Linear Algebra, Probability, Statistics.

Learning Outcomes

By the end of the course, the student must be able to:

- Formulate the various elements of a statistical problem rigorously.
- Formalize the performance of statistical procedures through probability theory.
- Systematize broad classes of probability models and their structural relation to inference.
- Construct efficient statistical procedures for point/interval estimation and testing in classical contexts.
- Derive certain exact (finite sample) properties of fundamental statistical procedures.
- Derive certain asymptotic (large sample) properties of fundamental statistical procedures.
- Formulate fundamental limitations and uncertainty principles of statistical theory.
- Prove certain fundamental structural and optimality theorems of statistics.

Teaching methods

Lecture ex cathedra using slides as well as the blackboard (especially for some proofs). Examples/exercises presented/solved at the blackboard.

Assessment methods

Final written exam.

Dans le cadre de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Ressources en bibliothèque

- [Mathematical Statistics / Knight](#)
- [Mathematical Statistics \(e-book\)](#)

Notes/Handbook

The slides will be available on Moodle.

MATH-413

Statistics for data science

Olhede Sofia Charlotta

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Data Science	MA1, MA3	Obl.
Data science minor	H	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	6 weekly
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

Statistics lies at the foundation of data science, providing a unifying theoretical and methodological backbone for the diverse tasks encountered in this emerging field. This course rigorously develops the key notions and methods of statistics, with an emphasis on concepts rather than techniques.

Content**Keywords**

Data science, inference, likelihood, regression, regularisation, statistics.

Learning Prerequisites**Required courses**

Real analysis, linear algebra, probability.

Recommended courses

A first course in statistics.

Important concepts to start the course

Students taking the course will need a solid grasp of notions from analysis (limits, sequences, series, continuity, differential/integral calculus) and linear algebra (linear subspaces, bases, dimension, eigendecompositions, etc). Though the course will cover a rapid review of probability, a first encounter with the subject is necessary (random variables, distributions/densities, independence, conditional probability). Familiarity with introductory level notions of statistics would be highly beneficial but not necessary.

Learning Outcomes

By the end of the course, the student must be able to:

- Derive properties of fundamental statistical procedures
- Estimate model parameters from empirical observations
- Test hypotheses related to the structural characteristics of a model
- Construct confidence bounds for model parameters and predictions
- Contrast competing models in terms of fit and parsimony

Teaching methods

Slides and whiteboard.

Assessment methods

Final exam and a midterm counting for 15%.

Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

Supervision

Office hours	No
Assistants	Yes
Forum	No

Resources

Bibliography

Davison, A.C. (2003). Statistical Models, Cambridge.

Panaretos, V.M. (2016). Statistics for Mathematicians. Birkhäuser.

Wasserman, L. (2004). All of Statistics. Springer.

Friedman, J., Hastie, T. and Tibshirani, R. (2010). Elements of Statistical Learning. Springer

Ressources en bibliothèque

- [Elements of Statistical Learning](#)
- [All of Statistics.](#)
- [Statistics for Mathematicians](#)
- [Statistical Models](#)

Moodle Link

- <https://moodle.epfl.ch/course/view.php?id=15506>

COM-506

Student seminar: security protocols and applications

Vaudenay Serge

Cursus	Sem.	Type
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	3
Session	Summer
Semester	Spring
Exam	Written
Workload	90h
Weeks	14
Hours	2 weekly
Lecture	2 weekly
Number of positions	

Summary

This seminar introduces the participants to the current trends, problems, and methods in the area of communication security.

Content

We will look at today's most popular security protocols and new kinds of protocols, techniques, and problems that will play an emerging role in the future. Also, the seminar will cover methods to model and analyze such security protocols. This course will be held as a seminar, in which the students actively participate. The talks will be assigned in the first meeting to teams of students, and each team will have to give a 45 minutes talk, react to other students' questions, and write a 3-4 pages summary of their talk.

Keywords

network security, security protocols, cryptography

Learning Prerequisites**Required courses**

- Computer security (COM-301)
- Cryptography and security (COM-401)

Learning Outcomes

By the end of the course, the student must be able to:

- Synthesize some existing work on a security protocol
- Analyze a security protocol
- Present a lecture

Transversal skills

- Make an oral presentation.
- Summarize an article or a technical report.

Expected student activities

- prepare a lecture (presentation and a 4-page report)
- present the lecture
- attend to others' lectures and grade them
- do the final exam

Assessment methods

- lecture and attendance to others' lectures (50%)
- final exam (50%)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	Lecturers and assistants are available upon appointment.

Resources

Websites

- <http://lasec.epfl.ch/teaching.shtml>

Moodle Link

- <https://moodle.epfl.ch/course/view.php?id=13965>

CS-448

Sublinear algorithms for big data analysis

Kapralov Mikhail

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	3 weekly
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal, donné les années impaires

Summary

In this course we will define rigorous mathematical models for computing on large datasets, cover main algorithmic techniques that have been developed for sublinear (e.g. faster than linear time) data processing. We will also discuss limitations inherent to computing with constrained resources.

Content

The tentative list of topics is:

Streaming: given a large dataset as a stream, how can we approximate its basic properties using a very small memory footprint? Examples that we will cover include statistical problems such as estimating the number of distinct elements in a stream of data items, finding heavy hitters, frequency moments, as well as graphs problems such as approximating shortest path distances, maximum matchings etc.;

Sketching: what can we learn about the input from a few carefully designed measurements (i.e. a 'sketch') of the input, or just a few samples of the input? We will cover several results in sparse recovery and property testing that answer this question for a range of fundamental problems;

Sublinear runtime: which problems admit solutions that run faster than it takes to read the entire input? We will cover sublinear time algorithms for graph processing problems, nearest neighbor search and sparse recovery (including Sparse FFT);

Communication: how can we design algorithms for modern distributed computation models (e.g. MapReduce) that have low communication requirements? We will discuss graph sketching, a recently developed approach for designing low communication algorithms for processing dynamically changing graphs, as well as other techniques.

Keywords

streaming, sketching, sparse recovery, sublinear algorithms

Learning Prerequisites**Required courses**

Bachelor courses on algorithms, complexity theory, and discrete mathematics

Important concepts to start the course

Discrete probability; mathematical maturity

Learning Outcomes

By the end of the course, the student must be able to:

- Design efficient algorithms for variations of problems discussed in class
- Analyze space/time/communication complexity of randomized algorithms
- Prove space/time/communication lower bounds for variations of problems discussed in class
- Choose an appropriate algorithmic tool for big data problem at hand

Teaching methods

Ex cathedra, homeworks, final

Assessment methods

Continuous control

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

CS-449

Systems for data science

Kermarrec Anne-Marie

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational science and Engineering	MA2, MA4	Opt.
Data Science	MA2, MA4	Obl.
Data science minor	E	Opt.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	6 weekly
Lecture	2 weekly
Exercises	2 weekly
Project	2 weekly
Number of positions	

Summary

The course covers fundamental principles for understanding and building systems for managing and analyzing large amounts of data.

Content

Big data systems design and implementation :

- *Distributed systems for data science*
- *Data management : locality, accesses, partitioning, replication*
- *Distributed Machine Learning Systems : federated learning/parameter server/decentralized learning*
- *Massively parallel processing operations*

Large-scale storage systems :

- *Data structures : File systems, Key-value stores, DBMS*
- *Consistency models. The CAP theorem. NoSQL and NewSQL systems*
- *Transactions*

Large-scale processing :

- *Parallel processing*
- *Streaming Processing*
- *Online Processing*
- *Graph Processing*

Keywords

Distributed systems, Parallel programming, Large-scale storage systems, Large-scale data management

Learning Prerequisites**Recommended courses**

CS-322 *Introduction to database systems*
 CS-323: *Introduction to operating systems*

CS-206 Parallelism and concurrency

Important concepts to start the course

- *Algorithms and data structures.*
- *Scala and/or Java programming languages will be used throughout the course. Programming experience in one of these languages is strongly recommended.*
- *Basic knowledge or computer networking and distributed systems*

Learning Outcomes

By the end of the course, the student must be able to:

- Choose systems parameters, data layouts, and application designs for database systems and applications.
- Develop data-parallel analytics programs that make use of modern clusters and cloud offerings to scale up to very large workloads.
- Analyze the trade-offs between various approaches to large-scala data management and analytics, depending on efficiency, scalability, and latency needs
- Choose the most appropriate existing systems architecture and technology for a task

Teaching methods

Lectures, exercises and practical work

Expected student activities

During the semester, the students are expected to:

- attend the lectures in order to ask questions and interact with the professor,
- attend the exercises session to solve and discuss exercises,
- solve practical homeworks and/or finish a project during the semester,
- take the exams during the semester

Assessment methods

Homeworks, written examinations, project. Continuous control

Supervision

Others Office ours by appointment

Resources

Bibliography

Relevant resources (textbook chapters, articles, and videos) posted on moodle page.

CS-410

Technology ventures in IC

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Project	2 weekly
Number of positions	

Remark

Pas donné en 2021-22

Summary

This hands-on class gives graduate students in IC interested in startups the opportunity to learn and put in practice the fundamental skills required to assess a technology concept in the context of a business opportunity. This class is focused only on business opportunities where high-technology

Content

Working in teams, students will learn the fundamentals of:

- *Opportunity assessment*
- *Customer development and validation*
- *Business model alternatives*
- *Intellectual Property*
- *Strategy and Financial planning*
- *Go-to-market, launch, and growth*

This is a hands-on class where students start the class with their own technology venture concept (e.g. the work done as part of their PhD, or some well-formed idea, maybe with a prototype). During the class, they convert their concept into a integrated business plan.

Keywords

Entrepreneurship, startups, technology transfer, intellectual property

Learning Prerequisites**Required courses**

- *None – but available to MS and Ph.D. students only*

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze a business plan
- Create a business plan

Teaching methods

- Short ex-cathedra presentations of each topic
- Hands-on seminar with many short student presentations
- Presentations from invited guests, in particular industry executives and entrepreneurs
- Discussion and case studies

Assessment methods

- In-class participation (30%)
- In-class presentations (30%)
- Final pitch (40%)

Supervision

Office hours	Yes
Assistants	No
Forum	Yes

CS-458

The GC maker project

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	6 weekly
Practical work	6 weekly
Number of positions	

Remark

pas donné en 2021-22

Summary

The GC Maker Project is an interdisciplinary project course where students work in teams towards solving real-world challenges by leveraging geometric computing methods and digital fabrication technologies.

Content

At the beginning of the course we will identify 3-4 interdisciplinary teams with complementary skills and expertise. Each team will work on a specific computational design challenge chosen by the team members in consultation with the teachers. The main focus will be on topics that combine geometry, computing, engineering, and digital fabrication to achieve the project goals. We will follow a design thinking methodology and develop a suitable project plan for each team.

Geometric and algorithmic foundations and implementations will be discussed on demand when identified during project development as necessary to achieve specific project goals.

Students will have access to a variety of digital fabrication machines, such as laser cutters, CNC milling machines, or 3D printers, and will receive appropriate training to explore different prototyping options. This will enable a cycle of ideation, code development, rapid prototyping and evaluation to progressively solve the chosen design challenge. We will define a suitable format to present project outcomes in a public forum in the final week of the course.

Learning Prerequisites**Recommended courses**

CS-457 Geometric Computing is highly recommended

Important concepts to start the course

This course is a project course with limited capacity for 20 students. Students will be selected based on a 1-page letter that needs to be submitted before the semester starts describing their background and laying out their motivation for taking the course.

Learning Outcomes

By the end of the course, the student must be able to:

- Apply a design thinking methodology in a computational fabrication project
- Evaluate how to best integrate computational methods and digital fabrication tools to achieve project goals
- Develop and implement geometric computing algorithms relevant for the project goals
- Assess own project progress and devise adaptations of the project plan if necessary

- Provide constructive feedback on other groups' projects
- Communicate effectively with collaborators from different disciplines
- Design a suitable format and material for public presentation of project outcomes

Transversal skills

- Assess progress against the plan, and adapt the plan as appropriate.
- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Set objectives and design an action plan to reach those objectives.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Give feedback (critique) in an appropriate fashion.

Teaching methods

- Tutoring throughout the design cycle
- Hands-on tutorials on digital fabrication technologies
- Regular project critiques
- Interspersed lectures to deep-dive into specific topics, such as theoretical concepts, algorithmic foundations, engineering background, digital fabrication technologies

Expected student activities

- Coordinate project team and engage in collaborative problem solving
- Implement/adapt geometric computing algorithms
- Fabricate and evaluate prototypes
- Discuss project progress in class
- Provide constructive criticism and feedback to other groups
- Present project outcome in a public forum

Assessment methods

project assement throught the semester, final presentation

MATH-342

Time series

Olhede Sofia Charlotta

Cursus	Sem.	Type
Data Science	MA2, MA4	Opt.
Financial engineering minor	E	Opt.
Financial engineering	MA2, MA4	Opt.
Mathematics	BA6	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	Written
Workload	150h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

A first course in statistical time series analysis and applications.

Content

- Motivation; basic ideas; stochastic processes; stationarity; trend and seasonality.
- Autocorrelation and related functions.
- Stationary linear processes: theory and applications.
- ARIMA, SARIMA models and their use in modelling.
- Prediction of stationary processes.
- Spectral representation of a stationary process: theory and applications.
- Financial time series: ARCH, GARCH models.
- State-space models: Kalman filter.
- VAR and other simple multivariate time series models
- Other topics as time permits.

Learning Prerequisites**Required courses**

Probability and Statistics

Recommended courses

Probability and Statistics for mathematicians. A course in linear models would be valuable but is not an essential prerequisite.

Important concepts to start the course

The material from first courses in probability and statistics.

Learning Outcomes

By the end of the course, the student must be able to:

- Recognize when a time series model is appropriate to model dependence
- Manipulate basic mathematical objects associated to time series
- Estimate parameters of basic time series models from data

- Critique the fit of a time series model and propose alternatives
- Formulate time series models appropriate for empirical data
- Distinguish a range of time series models and understand their properties

Teaching methods

Ex cathedra lectures and exercises in the classroom and at home.

Assessment methods

final exam

Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

Supervision

Assistants	Yes
Forum	No

Resources

Bibliography

Lecture notes available at <https://moodle.epfl.ch/course/view.php?id=15393>

Ressources en bibliothèque

- [Dynamic Linear Models with R / Petris, Petrone & Campagnoli](#)
- [Analysis of Financial Time Series / Tsay](#)
- [Introduction to Time Series and Forecasting / Brockwell & Davis](#)
- [\(electronic version\)](#)
- [Time Series Analysis and its Applications, with R Examples / Shumway & Stoffer](#)
- [\(electronic version\)](#)
- [\(electronic version\)](#)
- [\(electronic version\)](#)

Notes/Handbook

- Brockwell, P. J. and Davis, R. A. (2016) Introduction to Time Series and Forecasting. Third edition. Springer.
- Shumway, R. H. and Stoffer, D. S. (2011) Time Series Analysis and its Applications, with R Examples. Third edition. Springer.
- Tsay, R. S. (2010) Analysis of Financial Time Series. Third edition. Wiley.

- Percival, D.P. and Walden A. T. (1994) Spectral Analysis for Physical Applications. CUP.

CS-455

Topics in theoretical computer science

Cursus	Sem.	Type
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	4 weekly
Lecture	3 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal - pas donné en 2021-22

Summary

The students gain an in-depth knowledge of several current and emerging areas of theoretical computer science. The course familiarizes them with advanced techniques, and develops an understanding of fundamental questions that underlie some of the key problems of modern computer science.

Content

Examples of topics that will be covered include:

- Laplacians, random walks, graph sparsification: It is possible to compress graphs while approximately preserving their spectral properties (in particular, properties of random walks)? We will cover the main results from the recent influential line of work on spectral sparsification that provides such compression schemes.
- Laplacian system solvers: given a linear system $Ax=b$, how quickly can we find x ? We will cover nearly linear time algorithms for solving $Ax=b$ when A is a symmetric diagonally dominant matrix (a common scenario in practice) that crucially rely on spectral graph sparsification.
- Spectral clustering: given a graph, can we find a partition of the graph into k vertex disjoint parts such that few edges cross from one part to another? This is the fundamental graph clustering problem that arises in many applications. We will cover several results on spectral graph partitioning, where one first embeds vertices of the graph into Euclidean space using the bottom few eigenvectors of the graph Laplacian, and then employs Euclidean clustering primitives to find the partition.
- Local clustering with random walks: Given a very large graph and a seed node in it, can we find a small cut that separates the seed node from the rest of the graph, without reading the entire graph? We will cover local clustering algorithms, which identify such cuts in time roughly proportional to the number of vertices on the small side of the cut, by carefully analyzing distributions of random walks in the graph.

Keywords

spectral graph theory, sparsification, clustering, random walks

Learning Prerequisites**Required courses**

Bachelor courses on algorithms and discrete mathematics, mathematical maturity.

Learning Outcomes

By the end of the course, the student must be able to:

- Design efficient algorithms for variations of problems discussed in class;
- Analyze approximation quality of spectral graph algorithms;

Teaching methods

Ex cathedra, homeworks, reading

Expected student activities

Attendance at lectures, completing exercises, reading written material

Assessment methods

- Continuous control

Supervision

Office hours	Yes
Assistants	Yes
Others	Electronique forum : Yes

Resources

Bibliography

There is no textbook for the course. Notes will be posted on the course website.

Ressources en bibliothèque

- [Randomized Algorithms / Motwani](#)

CS-444

Virtual reality

Boulic Ronan

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Learning Sciences		Obl.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	3 weekly
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Summary

The goal of VR is to embed the users in a potentially complex virtual environment while ensuring that they are able to react as if this environment were real. The course provides a human perception-action background and describes the key techniques for achieving efficient VR applications.

Content

The first lectures focus more on the technical means (hw & sw) for achieving the hands-on sessions:

- Visual display
- Interaction devices and sensors
- Software environment (UNITY3D)

The proportion of more theoretical VR and Neuroscience background increases over the semester:

- Key Human perception abilities, Cybersickness, Immersion, presence and flow
- Basic 3D interaction techniques: Magic vs Naturalism
- The perception of action
- Haptic interaction
- What makes a virtual human looking alive ?
- Motion capture for full-body interaction
- VR, cognitive science and true experimental design

Keywords

3D interaction, display, sensors, immersion, presence

Learning Prerequisites**Required courses**

(CS 341) Introduction to Computer Graphics

Recommended courses

(CS 211) Introduction to Visual Computing

Important concepts to start the course

from Computer Graphics:

- perspective transformations
- representation of orientation

- 3D modelling hierarchy
- matrix algebra: translation, orientation, composition

Learning Outcomes

By the end of the course, the student must be able to:

- Describe how the human perception-action system is exploited in VR
- Apply the concepts of immersions, presence and flow
- Give an example of applications of VR in different industrial sectors
- Choose a method of immersion suited for a given 3D interaction context
- Explain the possible causes of cybersickness in a given VR system configuration
- Design a VR system involving 3D interactions

Transversal skills

- Set objectives and design an action plan to reach those objectives.
- Assess one's own level of skill acquisition, and plan their on-going learning goals.

Teaching methods

Ex cathedra + Hands-on sessions on VR devices in the first half of the semester,
A mini-project in groups of 2-3 persons will have to integrate various components of 3D real-time interaction. The group will submit their project proposal to the course responsible TAs who will assess whether it meets the key specifications and is original enough. The proposal will include the use of some VR devices that the IIG research group will lend during the mini-project period.

Expected student activities

exploit citation analysis tools to evaluate a scientific paper
combine 3D interaction components to produce an original 3D experience
experiment the hands-on practical work in the lab
synthesize the knowledge acquired in course and hands-on in the quizzes and final oral

Assessment methods

Throughout semester: 1 paper citation study (20%), 1 project (50%), 1 theoretical oral (30%)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

- Course notes will be updated and made available after each course, with links to key sites and on-line documents
- J. Jerald, The VR Book, ACM Press 2015
- Parisi, Learning Virtual Reality, O'Reilly 2015
- Le Traité de Réalité Virtuelle (5 vol.) Presses des Mines, ParisTech, 2006-2009, available on-line, free for student upon registration.

- Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 2004. 3D User Interfaces: Theory and Practice. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

Ressources en bibliothèque

- [3D User Interfaces: Theory and Practice / Bowman](#)
- [Le Traité de Réalité Virtuelle / Fuchs](#)
- [The VR Book / Jerald](#)
- [Learning Virtual Reality / Parisi](#)

Notes/Handbook

pdf of slides are made visible after the ex-cathedra courses

Websites

- <http://www.thevrbook.net/>

Moodle Link

- <http://moodle.epfl.ch/course/view.php?id=6841>

CS-503

Visual intelligence : machines and minds

Zamir Amir

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	During the semester
Workload	150h
Weeks	14
Hours	4 weekly
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

The course will discuss classic material as well as recent advances in computer vision and machine learning relevant to processing visual data. The primary focus of the course will be on embodied intelligence and perception for active agents.

Content

Visual perception is the capability of inferring the properties of the external world merely from the light reflected off the objects therein. This is done beautifully well by simple (e.g. mosquitoes) or complex (e.g. humans) biological organisms. They can see and understand the complex environment around them and act accordingly; all done in an efficient and astonishingly robust way. Computer vision is the discipline of replicating this capability for machines. The progress in computer vision has brought about successful applications, such as face detection/recognition or handwriting recognition. However, a large gap to sophisticated perceptual capabilities, such as those exhibited by animals, remains.

The goal of this course is to discuss what is possible in computer vision today, and what is not. We will overview the basic concepts in computer vision and recent advances in machine learning relevant to processing visual data and active perception. For inspirations around what the missing capabilities are and how to approach them, we will turn to visual perception in biological organisms.

The course has a heavy emphasis on projects and hands-on experience. The course project will be around designing, implementing, and testing a solution to an open problem pertinent to visual perception. The students are encouraged to work in groups, self-propose a project that makes them excited, and go for ambitious yet feasible projects. The course staff will provide support throughout the semester with the projects. In the lectures, the students will learn about the principles of computer vision, the current limits, and the visual perception in humans and animals, which will help them with formulating and executing their course projects. In particular, the lectures will discuss:

1) A recap of basic computer vision concepts: classification, detection, segmentation, transformations, optical flow, 3D from X, etc, 2), What/why/how of visual representations. Supervised, self-supervised, unsupervised learning of representations. 3), Psychology of the visual system. 4), Physiology of the visual system. 5), Perception-action loop: active perception and embodied intelligence.

The course is of interest to MS/PhD students interested in research in computer vision, machine learning, and perceptual robotics as well as senior undergraduate students interested in gaining an advanced understanding of SOTA computer vision.

Keywords

Computer vision, machine learning, cognition, embodied intelligence, robotics, neural networks, AI.

Learning Prerequisites**Required courses**

Introduction to Machine Learning (CS-233) or Machine Learning (CS-433) or equivalent course on the basics of machine learning and deep learning.

Recommended courses

Computer vision (CS-442) or equivalent undergraduate course on the basics of computer vision.

Important concepts to start the course

- Python programming.
- Basics of deep learning and machine learning.
- Basics of probability and statistics.

Learning Outcomes

By the end of the course, the student must be able to:

- Define the basic concepts in computer vision, such as detection, segmentation, 3D from X, covered in the lectures.
- Explain the range of theories in psychology around visual perception, covered in the lectures.
- Design and implement computer vision/machine learning algorithms to address problems with real- world complexity.
- Design and implement proper evaluation pipelines for computer vision/machine learning algorithms to assess their performance in the real-world.
- Assess the limits and performance pitfalls of a given computer vision/machine learning algorithm, especially when facing real-world complexity

Transversal skills

- Write a scientific or technical report.
- Make an oral presentation.
- Assess progress against the plan, and adapt the plan as appropriate.
- Demonstrate the capacity for critical thinking

Teaching methods

Lectures. Lab sessions. Project Tutoring. Course Project.

Expected student activities

- In regard to the lectured material, the students are expected to study the provided reading material, actively participate in the class, engage in the discussions, and answer homework questions. In regard to the course project, the students are expected to formulate and implement an in-depth project and demonstrate continuous progress throughout the semester.

Assessment methods

- Project (70%) [Project proposal, Project checkpoint reports, Final project report and presentation]
- Homeworks (20%)
- Class attendance and engagement (10%)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources**Bibliography**

- Vision Science: Photons to Phenomenology, Steven Palmer, 1999.
- The Ecological Approach to Visual Perception, Jame Gibson, 1979.
- Computer Vision: Algorithms and Applications, Richard Szeliski, 2020

Ressources en bibliothèque

- [Vision science : photons to phenomenology / Palmer](#)
- [The Ecological Approach to Visual Perception /Gibson](#)
- [Computer Vision: Algorithms and Applications / Szeliski \(2022? ; online drafts\)](#)

Notes/Handbook

The reference reading of different lectures will be from different books (main ones listed above) and occasionally from papers. Resources will be provided in class. Full-text books are not mandatory.