Plan d’études

DATA SCIENCE

2024-2025

arrêté par la vice-présidence académique de l'EPFL le 20 juin 2024

Directeur de section	**Prof. K. Aberer**
Adjointe de la section	**Mme E. Hazboun**
Coordinatrice des stages en entreprise	**Mme P. Genet**
Spécialiste administrative	**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.

2024-2025 **DATA SCIENCE** **Cycle Master**

Code	Matières	Enseignants	Semestres						Crédits	Période	Type
		sous réserve	MA1/MA3			MA2/MA4				des	examen**
		de modification	c	e	p	c	e	p		épreuves*	
Groupe "Core courses et options"									**72**		
Groupe 1 "Core courses"									**min. 32**		
CS-450	Algorithms II	Kapralov/Svensson	**4**	**3**					8	H	écrit
CS-401	Applied data analysis	Brbic	**2**	**2**					8	H	écrit
COM-406	Foundations of Data Science	Gastpar/Urbanke	**4**	**2**					8	H	écrit
COM-402	Information security and privacy	Payer	**3**	**1**	**2**				8	H	écrit
CS-433	Machine learning	Jaggi/Flammarion	**4**	**2**	**2**				8	H	écrit
CS-552	Modern natural language processing	Bosselut				**3**	**1**	**2**	8	sem P	
CS-439	Optimization for Machine Learning	Flammarion				**2**	**2**	**1**	8	E	écrit
MATH-413	Statistics for Data Science	Chandaki/Limnios				**4**	**2**		8	E	écrit
CS-460	Systems for data management and data science	Ailamaki/Kermarrec				**2**	**2**	**2**	8	E	écrit
Groupe 2 "Options" (la somme des crédits des groupes 1 et 2 doit être de 72 crédits au minimum)											
	- voir liste cours à option										
Bloc "Projets et SHS" :									**18**		
COM-412	Research project in Data Science	divers enseignants	←		**2**			→	12	sem A ou P	
HUM-nnn	SHS : introduction au projet	divers enseignants	**2**		**1**				3	sem A	
HUM-nnn	SHS : projet	divers enseignants						**3**	3	sem P	
	Total des crédits du cycle master								**90**		

Remarques :

* Cf. l'art. 3 de l'Ordonnance sur le contrôle des études à l'EPFL

** sans retrait = pas de retrait possible après le délai d'inscription

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

2024-2025

DATA SCIENCE - Options

Code	Matières	Enseignants	Semestres						Crédits	Nbre	Période	Type
		sous réserve	MA1/MA3			MA2/MA4				places	des	examen**
		de modification	c	e	p	c	e	p			épreuves*	
Groupe 2 "Options" (la somme des crédits des groupes 1 et 2 doit être de 72 crédits au minimum)												
CS-420	Advanced compiler construction	Schinz				*2*		*2*	6		sem P	
COM-501	Advanced cryptography	Vaudenay				2	2		6		E	écrit
COM-417	Advanced probability and applications	Shkel	4	2					8		H	écrit
CS-523	Advanced topics on privacy enhancing technologies	Troncoso				3	1	2	8		E	écrit
CS-500	AI product management	Kaboli/Zamir	*2*	*2*	*3*				6		sem A	sans retrait
MATH-493	Applied biostatistics	Goldstein				2	2		5		sem P	
EE-554	Automatic speech processing	Magimai Doss	2	2					4		H	écrit
MICRO-452	Basics of mobile robotics	Mondada	1		3				4		H	écrit
MGT-416	Causal inference ****	Kiyavash				2	1		4		sem P	
MATH-352	Causal thinking	Stensrud	2	2					5		H	écrit
MATH-453	Computational linear algebra	Kressner				2	2		5		E	oral
CS-524	Computational complexity	Göös	3	1					6		H	écrit
NX-465	Computational neuroscience : neural dynamics	Gerstner				2	2		5		E	écrit
CS-413	Computational Photography	Süsstrunk				2		2	6		sem P	
COM-418	Computers and Music (pas donné en 2024-25)	Prandoni P.				2	1		6		sem P	écrit
CS-442	Computer vision	Fua				2	1		6		E	écrit
CS-453	Concurrent computing	Guerraoui	2	1	2				6		H	écrit
COM-401	Cryptography and security	Vaudenay	4	2					8		H	écrit
COM-480	Data visualization	Vuillon				2		2	6		sem P	
EE-559	Deep learning	Cavallaro				2	2		4	150	sem P	sans retrait
CS-502	Deep learning in biomedicine (pas donné en 2024-25)	Brbic				2	2	1	6			sem P
CS-456	Deep reinforcement learning	Gulcehre				2	1	1	6		E	écrit
CS-411	Digital education	Dillenbourg/Jermann				2		2	6		E	écrit
CS-451	Distributed algorithms	Guerraoui	2	1	3				8		H	écrit
CS-423	Distributed information systems	Aberer	2	1	1				6		H	écrit
ENG-466	Distributed intelligent systems	Martinoli	2	3					5	40	H	oral sans retrait
DH-415	Ethics and Law of AI	Rochel	2	2					4	100	sem A	sans retrait
CS-550	Formal verification	Kuncak	2	2	2				6		sem A	
CS-457	Geometric Computing	Pauly	3		2				6		H	écrit
MATH-360	Graph Theory	Vérité	2	2					5		H	écrit
EE-451	Image analysis and pattern recognition	Thiran J.-P./Bozorgtabar				2		2	4		sem P	
COM-404	Information theory and coding	Telatar	4	2					8		H	écrit
CS-430	Intelligent agents	Faltings	3	3					6		sem A	
CS-486	Interaction design	Pu				2	1	1	6		sem P	
CS-428	Interactive theorem proving	Barrière/Pit-Claudel				2	1	2	6		sem P	
CS-431	Introduction to natural language processing	Chappelier/Rajman/ Bosselut	2	2					6		H	écrit
COM-490	Large-scale data science for real-world data	Bouillet/Sarni/ Verscheure/Delgado						4	6		sem P	sans retrait
CS-479	Learning in neural networks	Gerstner				2	1	1	6		E	oral
CS-526	Learning theory	Macris				2	2		6		E	écrit
MATH-341	Linear models	Panaretos	2	2					5		H	écrit
CS-421	Machine learning for behavioral data ****	Käser				2		2	6		E	écrit
MGT-427	Management de projet et analyse du risque	Wieser	2	1					4		sem A	sans retrait
COM-516	Markov chains and algorithmic applications (pas donné en 2024-25) ****	Lévêque/Macris	2	1	1				6		H	écrit
EE-556	Mathematics of data: from theory to computation	Cevher	3		3				6		H	écrit
EE-452	Network machine learning	Frossard/Thanou				2		2	4		sem P	
COM-512	Networks out of control (pas donné en 2024-25) ****	Thiran P./Grossglauser				2	1		6		E	écrit
COM-508	Optional research project in Data Science	Divers enseignants	2						8		sem A ou P	
CS-412	Software security	Payer				3	2	1	8		sem P	
MATH-486	Statistical mechanics and Gibbs measures (pas donné en 2024-25)	Friedli				2	2		5		E	oral
PHYS-512	Statistical physics of computation	Erba	2	2					4		H	écrit
MATH-442	Statistical Theory	Zemel	2	2					5		H	écrit
COM-506	Student seminar: security protocols and applications	Vaudenay				2			3		sem P	
CS-448	Sublinear algorithms for big data analysis ****	Kapralov				2	1		6		sem P	
MATH-342	Time Series	Olhede				2	2		5		E	écrit
CS-455	Topics in theoretical computer science (pas donné en 2024-25) ****	Kapralov	3	1					6		sem A	
CS-444	Virtual reality	Boulic				2		1	6		sem P	
CS-503	Visual Intelligence : Machines and Minds	Zamir				2	1	1	6		sem P	

Remarques :
* Cf. l'art. 3 de l'Ordonnance sur le contrôle des études à l'EPFL
** sans retrait = pas de retrait possible après le délai d'inscription
**** cours biennaux donnés une année sur deux

RÈGLEMENT D’APPLICATION DU CONTRÔLE DES ÉTUDES DE LA SECTION DE SYSTÈMES DE COMMUNICATION POUR LE MASTER EN DATA SCIENCE

Année académique 2024-2025

du 20 juin 2024

La Vice-présidence académique

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:

Arti. 1 - Champ d'application

Le présent règlement fixe les règles d’application du contrôle des études de master en Data Science pour l’année académique 2024-2025.

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 ou de 25 semaines, dont la réussite se traduit par l’acquisition de 30 crédits.

Art 3 – Sessions d’examen

1. Les branches d’examen sont examinées pendant les sessions d’hiver ou d’été (mention H ou E dans le plan d’études).

2. Les branches de semestre sont examinées pendant le semestre d’automne ou le semestre de printemps (mention sem A ou sem P).

3 Pour les branches de session, l’examen indiqué pour la session peut être complété par des contrôles de connaissances durant le semestre, selon les indications du personnel enseignant.

Art. 4 – 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. 5 - Organisation

1. Les étudiantes et les étudiants restent soumis à leur plan d’études en vigueur lors de leur entrée au Master.

2. Les enseignements du cycle master sont répartis en un bloc et deux groupes dont les crédits doivent être acquis de façon indépendante.

3. Le bloc « Projets et SHS » est composé d’un projet de recherche et de l’enseignement SHS.

4. Le groupe 1 « Core courses » est composé des cours de la liste du plan d’études dans la rubrique « Master ».

5. 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 de recherche optionnel de 8 crédits suivant l'alinéa 6 ;
- de cours hors plan d'études suivant l'alinéa 7.

6. Le projet de recherche du bloc « Projets et SHS » et le projet de recherche optionnel du groupe 2 ne peuvent pas être effectués dans le même semestre.

7. 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 la direction 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 acquis.

2. Le groupe « Core courses et Options », composé du groupe 1 « Core courses » et du groupe 2 « Options » est réussi lorsque **72 crédits** sont acquis.

3. Le groupe 1 « Core courses » est réussi lorsqu'**au moins 32 crédits** sont acquis.

Art. 7 - Enseignement SHS

1 L'enseignement SHS du semestre d'automne constitue l'introduction à la réalisation du projet SHS du semestre de printemps.

2 Pour autant qu'il considère que le cursus d'un cas individuel le justifie, le Collège des Humanités peut, d'entente avec l'équipe enseignante, déroger à cette organisation en autorisant que le projet soit réalisé au même semestre que le cours d'introduction ou soit réalisé à un semestre ultérieur.

Art. 8 – Mineurs

1 Afin d'approfondir un aspect particulier de sa formation ou de développer des interfaces avec d'autres sections, l'étudiante ou l'étudiant peut choisir la formation offerte dans le cadre d'un mineur figurant dans l'offre de l'EPFL.

2 Les mineurs « Data Science » « Informatique » et « Systèmes de Communication » ne peuvent pas être choisis.

3 L'étudiante ou 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 Le choix des cours qui composent un mineur se fait d'entente avec la section de Systèmes de communication et la personne responsable du mineur.

5 Un mineur est réussi quand 30 crédits au minimum sont acquis parmi les branches avalisées.

Art. 9 – Stage d'ingénierie

1. Un stage d'ingénierie doit être effectué dès la fin du 2e semestre et avant le projet de master. Sur demande, la section peut autoriser les titulaires d'un bachelor EPFL en Informatique ou Systèmes de communication, à réaliser le stage plus tôt.

2 Le stage prend l'une des formes suivantes :

- soit un stage d'été de minimum 8 semaines

- soit un stage de minimum 6 mois en entreprise (en statut stage durant un semestre).

- soit un projet de master de 25 semaines en entreprise (art. 12)

3 Effectuer des cours/projet en parallèle au stage n'est pas autorisé.

4. La personne responsable des stages de la section évalue le stage, par l'appréciation « réussi » ou « échoué ». Sa réussite est une condition pour l'admission au projet de master. En cas d'échec, 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'instructions internes à la section.

Art. 10 – Organisation de la Spécialisation en Informatique pour l'enseignement

L'étudiante ou l'étudiant admis à cette spécialisation ne peut pas suivre un mineur. Le plan d'études est modifié comme suit :

(i) Un nouveau groupe de 30 crédits de cours à la HEP Vaud est ajouté et le nombre de crédits du Cycle Master passe de 60 à 30 crédits ;
(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'étudiante ou 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 débuter la spécialisation.

Art. 11 – Procédure d'admission

1. Pour être admis à la spécialisation, la candidate ou 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'étudiante ou 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.

Art. 12 – Projet de master

1. Le projet de master s'étend sur une durée de 17 semaines s'il est effectué à l'EPFL ou de 25 semaines s'il est effectué hors EPFL.

2. Il est placé sous la responsabilité d'une professeure, d'un professeur ou MER affilié à la section d'Informatique ou Systèmes de communication.

3. Sa réussite se traduit par l'acquisition de 30 crédits

Au nom de l'EPFL

Le Vice-président académique, J. S. Hesthaven

Lausanne, le 20 juin 2024

CS-500 AI product management

Kaboli Amin, Zamir Amir

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

Language of teaching	English
Credits	6
Withdrawal	Unauthorized
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**7 weekly**
Lecture	2 weekly
Exercises	2 weekly
Project	3 weekly
Number of positions	

It is not allowed to withdraw from this subject after the registration deadline.

Summary

The course focuses on the development of real-word AI/ML products. It is intended for students who have acquired a theoretical background in AI/ML and are interested in applying that toward developing AI/ML-oriented products.

Content

AI is set to transform several industry sectors, and there is high demand for AI product managers. AI product management is a complex role that requires an understanding of both AI and product management. This course will enable students to identify opportunities for developing new AI products, understand when they should use AI in an existing product/process, manage the development of AI products, and launch AI products successfully. The lectures will introduce general product management to the students, and the guest lectures, by leading figures in AI industries, explain how the general product management skills are applied to the development of AI products.

Module 1: Introduction to AI product management

- Why is this needed?
- Product strategy
- Setting product objectives & identifying opportunities

Module 2: AI Product Discovery

- Understanding customers and problems
- Creating and testing hypothesis
- Defining product requirements
- Establishing product-market fit
- Setting AI product vision, strategy, roadmap

Module 3: AI Product development

- Mastering agile methodologies
- AI product design, tests, and development
- Managing team dynamics and effective communication with stakeholders

Module 4: AI Product Delivery

- Planning and executing a successful AI product launch
- Marketing the AI product
- Release and performance metrics for continuous improvement of AI products

Keywords

Artificial Intelligence (AI), AI product managers, Innovation

Learning Prerequisites

Required courses

CS-233 Introduction to machine learning or CS-433 Machinie learning or equivalent course on the basics of machine learning and deep learning

Important concepts to start the course

- Python programming
- Bascis 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:

- and understand opportunities for an AI product or using AI within an existing product
- the development of AI features
- Launch AI products successfully

Transversal skills

- Demonstrate the capacity for critical thinking
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Communicate effectively, being understood, including across different languages and cultures.
- Set objectives and design an action plan to reach those objectives.
- Chair a meeting to achieve a particular agenda, maximising participation.
- Resolve conflicts in ways that are productive for the task and the people concerned.
- Make an oral presentation.
- Take account of the social and human dimensions of the engineering profession.

Teaching methods

- Formal lectures
- Group activities
- Class discussions
- Simulation games
- Hands-on exercises
- Project-based learning
- Real-world case studies

- Guest lectures by leading academic and industry figuers

Expected student activities

- **Individual** : Case evaluations, self-study, class discussions
- **In-group** : In-class exercises, projects, simulations games
- **Presentation** : Weekly presentations of assignments in coaching sessions

Assessment methods

Continuous evaluation of case reports, proejcts, individual and group presentations, class discussions, during the semester. More precisely :
25% Weekly in-class work and engagement
45% Class assignments, presentations, projects, and case reports
30% Final (final report and presentation and understanding of the case)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

- Cagan, M. (2017). *How to Create Tech Products Customers Love.* Wiley
- Kahneman, D., Sibony, O., & Sunstein, C. R. (2021). *Noise: A flaw in human judgment*. Little, Brown.
- Iansiti, M., & Lakhani, K. R. (2020). *Competing in the age of AI: strategy and leadership when algorithms and networks run the world*. Harvard Business Press.

Ressources en bibliothèque

- How to Create Tech Products Customers Love / Cagan
- Noise / Kahneman
- Competing in the age of AI / Iansiti

Moodle Link

- https://go.epfl.ch/CS-500

CS-420 Advanced compiler construction

Schinz Michel

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

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

Students learn several implementation techniques for modern functional and object-oriented programming languages. They put some of them into practice by developing key parts of a compiler and run time system for a simple functional programming language.

Content

Part 1: implementation of high-level concepts

- functional languages: closures, continuations, tail call elimination,
- object-oriented languages: object layout, method dispatch, membership test.

Part 2: optimizations

- compiler intermediate representations (RTL, SSA, CPS),
- inlining and simple optimizations,
- register allocation.

Part 3: run time support

- interpreters and virtual machines,
- memory management (including garbage collection).

Keywords

compilation, programming languages, functional programming languages, object-oriented programming languages, code optimization, register allocation, garbage collection, virtual machines, interpreters, Scala.

Learning Prerequisites

Recommended courses

CS-320 Computer language processing

Important concepts to start the course

Excellent knowledge of Scala and C programming languages

Learning Outcomes

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

- Assess / Evaluate the quality of a compiler intermediate representation
- Design compilers and run time systems for object-oriented and functional programming languages
- Implement rewriting-based compiler optimizations
- Implement efficient virtual machines and interpreters
- Implement mark and sweep or copying garbage collectors

Teaching methods

Ex Cathedra, mini-project

Assessment methods

Continuous control (mini-project 80%, final exam 20%)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

The Garbage Collection Handbook - The Art of Automatic Memory Management, second edition, Richard Jones, Antony Hosking, Eliot Moss (ISBN 9781032218038).

Ressources en bibliothèque

- The Garbage Collection Handbook / Jones

Websites

- https://cs420.epfl.ch/

COM-501 Advanced cryptography

Vaudenay Serge

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

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
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 cryptograhic 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 contiuous 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
- Communication security / Vaudenay
- A computational introduction to number theory and algebra / Shoup

Websites

- https://lasec.epfl.ch/teaching.php

Moodle Link

- https://go.epfl.ch/COM-501

Videos

- https://mediaspace.epfl.ch/channel/COM-501+Advanced+Cryptography

COM-417

Advanced probability and applications

Shkel Yanina

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

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	4 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 analyisis 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 and flipped 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

- Probability and Random Processes
- 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

Notes/Handbook

available on the course website

Websites

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

Moodle Link

- https://go.epfl.ch/COM-417

Prerequisite for

Advanced classes requiring a good knowledge of probability

CS-523

Advanced topics on privacy enhancing technologies

Troncoso Carmela

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

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
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 prect privacy, and how to evaluate the protection they provide.

Content

The course will cover the following topics :
- Privacy definitions and concepts
- Privacy-preserving cryptographics 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, censorpship 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-paty computation, anonymous credentials, ethics

Learning Prerequisites

Required courses

COM-301 Computer security
COM-402 Information security and privacy

Recommended courses

COM-401 Cryptography and security

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 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

Supervision

Assistants	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-523

CS-450

Algorithms II

Kapralov Michael, Svensson Ola Nils Anders

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	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.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.
Statistics	MA1, MA3	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
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.

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

- midterm (30%)
- homework (30%)
- final exam (40%)

Supervision

Forum	Yes
Others	For details, see the course web page

Resources

Websites

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

Moodle Link

- https://go.epfl.ch/CS-450

MATH-493 Applied biostatistics

Goldstein Darlene

Cursus	Sem.	Type
Computational and Quantitative Biology		Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Ing.-math	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	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
Liniear modeling (regression, anova)
Generalized linear modeling (logistic, Poission)
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
- Synthesize analysis into a written report
- Report plan of analysis and results obtained
- Justify analysis plan
- Plan analysis for a given dataset
- Interpret analysis results
- Analyze various types of biostatistical data

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.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

To be provided during the course.
Pre-recorded lectures (videos) will also be provided.

Moodle Link

- https://go.epfl.ch/MATH-493

CS-401 Applied data analysis

Brbic Maria

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational and Quantitative Biology		Opt.
Computational biology minor	H	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Obl.
Data and Internet of Things minor	H	Opt.
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.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.
Statistics	MA1, MA3	Opt.
UNIL - Sciences forensiques	H	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
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 acqusition (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, evaluation, etc.)
- Text mining: preprocessing steps, vector space model, topic models
- Social network analysis (properties of real networks, working graph data, 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.
In parallel, the students will embark on a semester-long project, split in agile teams of 3-4 students. In the project, students propose and execute meaningful analyses of a real-world dataset, which will require creativity and the application of the tools encountered in the course. 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 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. Programming skills are required (in class we will use mostly Python).

Recommended courses

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

Important concepts to start the course

programming, algorithms, probability and statistics, databases

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, knowledge extraction, critical thinking, team discussions, ad-hoc visualizations, etc.)
- Perform result dissemination (reporting, visualizations, publishing reproducible results, ethical concerns, etc.)
- Construct a coherent understanding of the techniques and software tools required to perform the fundamental steps of the data science pipeline
- Perform data interpretation (statistics, correlation vs. causality, knowledge extraction, critical thinking, team discussions, ad-hoc visualizations, etc.)
- Construct a coherent understanding of the techniques and software tools required to perform the fundamental steps of the data science pipeline

Transversal skills

- Give feedback (critique) in an appropriate fashion.
- 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
- Course project

Expected student activities

Students are expected to:

- Attend the lectures and lab sessions
- Complete 2-3 homework assignments
- Conduct the class project
- Engage during the class, and present their results in front of the other colleagues

Assessment methods

- Homework
- Project
- Final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Websites

- http://ada.epfl.ch

Moodle Link

- https://go.epfl.ch/CS-401

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 of teaching	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 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, speaker recognition).

Content

1. Introduction: Speech processing tasks, Speech science, language engineering applications.

2. Basic Tools: Analysis and spectral properties of the speech signal, linear prediction algorithms, statistical pattern recognition, dynamic programming, speech representation learning

3. Speech Coding: Human hearing properties, quantization theory, speech coding in the temporal and frequency domains.

4. Speech Synthesis: speech synthesis models, concatenative synthesis, hidden Markov models (HMMs) based speech synthesis, Neural speech synthesis.

5. Automatic Speech Recognition: Temporal pattern matching and Dynamic Time Warping (DTW) algorithms, speech recognition systems based on HMMs, Neural networks-based speech recognition.

6. Speaker recognition and speaker verification: Formalism, hypothesis testing, Text-dependent and Text-independent speaker verification, Gaussian mixture models-/HMM-based speaker verification, speaker embeddings based speaker verification, presentation attack detection (antispoofing).

7. Paralinguistic speech processing: fundamentals and applications (e.g., emotion recognition, pathological speech detection, depression detection), hand-crafted feature based approaches, neural approaches.

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. Basic knowledge in machine learning/statistical pattern recognition is not a must but would be helpful in .

Learning Outcomes

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

- Analyze speech signal properties
- Exploit those properties for speech coding, speech synthesis, speech recognition, speaker recognition and paralinguistic speech processing
- Formulate speech processing problems
- Choose appropriate methods for target speech processing tasks

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. Regulary answer list of questions for feedback.

Assessment methods

Written exam without notes

Resources

Bibliography

Fundamentals of Speech Recognition / Rabiner and Juang
Speech and Language Processing / Dan Jurafsky and Daniel Martin (2nd Edition)
Spoken language processing: A Guide to Theory, Algorithm and System Development / Xuedong Huang, Alex Acero and Hsiao-Wuen Hon
Speech and Audio Signal Processing: Processing and Perception of Speech and Music / Ben Gold, Nelson Morgan and Dan Ellis
Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing / Björn Schuller and Anton Batliner

Ressources en bibliothèque

- Speech and Language Processing / Dan Jurafsky and Daniel Martin
- Fundamentals of Speech Recognition / Rabiner and Juang
- Spoken language processing: A Guide to Theory, Algorithm and System Development / Xuedong Huang, Alex Acero and Hsiao-Wuen Hon
- Speech and Audio Signal Processing: Processing and Perception of Speech and Music / Ben Gold, Nelson Morgan and Dan Ellis
- Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing / Björn Schuller and Anton Batliner

Moodle Link

- https://go.epfl.ch/EE-554

MICRO-452

Basics of mobile robotics

Mondada Francesco

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Mechanical engineering	MA1, MA3	Opt.
Microtechnics	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Obl.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	1 weekly
Practical work	1 weekly
Project	2 weekly
Number of positions	

Summary

The course teaches the basics of autonomous mobile robots. Both hardware (energy, locomotion, sensors) and software (signal processing, control, localization, trajectory planning, high-level control) will be tackled. The students will apply the knowledge to program and control a real mobile robot.

Content

- Sensors
- Perception, feature extraction
- Modeling
- Markov localization: Bayesian filter, Monte Carlo localization, extended Kalman filter
- Navigation: path planning, obstacle avoidance
- Control architectures and robotic frameworks
- Locomotion principles and control
- Sustainability

Keywords

mobile robots, sensing, perception, localisation, navigation, locomotion.

Learning Prerequisites

Required courses

Introduction to automatic control (catching up possible with extra effort)
Introduction to signal processing

Recommended courses

Microinformatique (SMT)

Important concepts to start the course

Embedded system programming
Basics of automatic control

Basics of signal processing

Learning Outcomes

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

- Choose the right methods to design and control a mobile robot for a particular task.
- Integrate approriate methods for sensing, cognition and actuation
- Justify design choices for a robotic system
- Implement perception, localisation/navigation and control methods on a mobile robot
- Choose the right methods to design and control a mobile robot for a particular task.

Transversal skills

- 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.
- Use a work methodology appropriate to the task.
- Assess progress against the plan, and adapt the plan as appropriate.
- Chair a meeting to achieve a particular agenda, maximising participation.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Negotiate effectively within the group.
- Resolve conflicts in ways that are productive for the task and the people concerned.

Teaching methods

Ex cathedra, case studies, exercises, work on mobile robots, group project

Expected student activities

- weekly lectures
- studying provided additional materials
- attend case study discussions
- lab exercises with practical components
- project at the end of the semester

Assessment methods

Project during the semester (60% of the grade). The project takes place during the semester and the report and presentation are done before the end of the semester, following the specific planning given by the teacher at the beginning of the semester.
Written exam (40% of the grade)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Bibliography

Introduction to Autonomous Mobile Robots R. Siegwart, and I. Nourbakhsh, MIT Press, 2004

Autonomous Robots: From Biological Inspiration to Implementation and Control G.A. Bekey, MIT Press, 2005
Probabilistic Robotics S. Thrun, W. Burgard and D. Fox, MIT Press, 2005
Handbook of Robotics (chapter 35) B. Sicilian, and O. Khatib (Eds.), Springer, 2008
Elements of Robotics M. ben-Ari and F. Mondada, Spinger, 2017.
additional literature provided on Moodle

Ressources en bibliothèque

- Handbook of Robotics / Sicilian
- Elements of Robotics / Ben-Ari
- Probabilistic Robotics / Thrun
- Introduction to Autonomous Mobile Robots / Siegwart
- Autonomous Robots / Bekey

Notes/Handbook

Lecture slides are continously provided on Moodle during the course.
Introduction to Autonomous Mobile Robots R. Siegwart, and I. Nourbakhsh, MIT Press, 2004
Probabilistic Robotics S. Thrun, W. Burgard and D. Fox, MIT Press, 2005

Moodle Link

- https://go.epfl.ch/MICRO-452

MGT-416

Causal inference

Kiyavash Negar

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.
Financial engineering	MA2, MA4	Opt.
Management, Technology and Entrepreneurship minor	E	Opt.
Managmt, tech et entr.	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Systems Engineering minor	E	Opt.

Language of teaching	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

Students will learn the core concepts and techniques of network analysis with emphasis on causal inference. Theory and application will be balanced, with students working directly with network data throughout the course.

Content

- Introduction: What is causal inference?
- Review of Useful Probability concepts
- Random variable, predictors, divergences
- Introduction to Applications
- Computational neuroscience
- Financial markets
- Social networks
- Pearl Causality
- Causal Bayesian Networks (CBNs)
- Learning CBNS: Faithfulness and identifiably
- Algorithms
- Potential Outcome Model
- Counterfactuals and identification problems
- Graphical causal models
- Randomized Experiments
- Identification of causes in randomized experiments
- Effect modification
- Causality in Times Series
- Granger causality
- More general linear predictors
- Beyond linear models and Granger causality
- Directed information graphs
- Efficient algorithms
- Concrete Applications
- Computational neuroscience
- Financial markets
- Social networks

Keywords

Causality, structure learning, network inference

Learning Prerequisites

Required courses

This course attempts to be as self contained as possible, but it does approach the topic from a quantitative point of view and, as such, students should be comfortable with the basics of (i.e. have taken at least one course in) the following topics before enrolling:
- Statistics
- Probability Theory
- Linear Algebra
- Calculus (integral and differential)
- Programing in Pythor and Matlab

As course work will be largely computational, experience with at least one programming language is also required.

Recommended courses

Knowledge of probability and calculus as well as programming is a must.

Learning Outcomes

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

- Identify situations in which a problem/data can be thought of as a network.
- Analyze data appropriately using a variety of network causal inference techniques
- Interpret the results of applying causal discovery or inference algorithms

Transversal skills

- Continue to work through difficulties or initial failure to find optimal solutions.
- Demonstrate the capacity for critical thinking
- Use both general and domain specific IT resources and tools
- Access and evaluate appropriate sources of information.

Teaching methods

In class with supporing problem solving sessions.

Expected student activities

TA problem solving sessions, homework, exams, projects

Assessment methods

Regular individual assignments: 30%
Midterm project: 30%
Final project: 40%

Supervision

Office hours	Yes
Assistants	Yes
Forum	No

Resources

Notes/Handbook

course notes

Moodle Link

- https://go.epfl.ch/MGT-416

MATH-352

Causal thinking

Stensrud Mats Julius

Cursus	Sem.	Type
Chemistry	BA5	Opt.
Computational and Quantitative Biology		Obl.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	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

This course will give a unified presentation of modern methods for causal inference. We focus on concepts, and we will present examples and ideas from various scientific disciplines, including medicine, computer science, engineering, economics and epidemiology.

Content

Association vs. causation
Definitions of causal effects
- Causal models
- Counterfactuals and potential outcomes
- Individual level causal effects vs. average causal effects
- Population causal effects

Study design
- Randomisation and experiments
- Observational studies

Causal graphs
- Causal Directed Acyclic Graphs
- Single World Intervention Graphs

Identification of causal effects
- Identifiability assumptions
- SWIGs

Causal mechanisms
- Mediation and path specific effects
- Instrumental variables

Applications
- Medical interventions, including pharmaceuticals
- Experiments in technology industry and engineering
- Experiments in life sciences
- Causal effects and mechanisms in the social sciences.

Estimation of causal effects
- Estimation using classical statistical models
- Estimation using machine learning

Keywords

Causality; Causal inference; Randomisation; Design of experiments; Observational studies; Causal Graphs

Learning Prerequisites

Required courses

The course is intended for students from a range of different disciplines, including computer science, engineering, life science and physics. The students are expected to know the basics of statistical theory and probability theory (such as the second year courses in probability and statistics for engineers).

Recommended courses

Courses in statistical inference.

Important concepts to start the course

Familiarity with basic concepts in probability and statistics.

Learning Outcomes

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

- Design experiments that can answer causal questions.
- Describe the fundamental theory of causal models.
- Critique assess causal assumptions and axioms.
- Distinguish between interpretation, identification and estimation.
- Describe when and how causal effects can be identified and estimated from non- experimental data.
- Estimate causal parameters from observational data

Teaching methods

Classroom lectures, where I will use a digital blackboard and slides.

Assessment methods

Final written exam. 1-2 graded homeworks.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Hernan, M.A. and Robins, J.M., 2020. Causal inference: What if?
Imbens, G.W. and Rubin, D.B., 2015. Causal inference in statistics, social, and biomedical sciences. Cambridge University Press.
Pearl, J., 2009. Causality. Cambridge university press.

Ressources en bibliothèque

- Causal inference in statistics, social, and biomedical sciences / Imbens
- Causality / Pearl
- Causal Inference / Hernan

Moodle Link

- https://go.epfl.ch/MATH-352

CS-524 Computational complexity

Göös Mika

Cursus	Sem.	Type
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Minor in Quantum Science and Engineering	H	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 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 vs NL)
- Boolean circuits and nonuniform computation
- Power of randomness
- 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

- Illustrate a basic understanding of probabilistically checkable proofs and their characterization of the class NP (the PCP-Theorem)
- 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.

Assessment methods

Three homeworks and final exam

Resources

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 / Jukna

Moodle Link

- https://go.epfl.ch/CS-524

MATH-453 Computational linear algebra

Kressner Daniel

Cursus	Sem.	Type
Computational and Quantitative Biology		Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Data Science	MA2, MA4	Opt.
Ing.-math	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	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

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
Theory foundations. Krylov subspace methods. Probabilistic analysis. 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

Lectures + exercise sessions

Expected student activities

Students are expected to attend lectures and participate actively in class and exercises. Exercises will include both theoretical work and programming assignments. Students also complete a substantial project (possibly in small groups) that likewise include theoretical and numerical work.

Assessment methods

20% of the grade will be based on projects, 80% of the grade will be based on a written exam

Resources

Moodle Link

- https://go.epfl.ch/MATH-453

NX-465

Computational neurosciences: neuronal dynamics

Gerstner Wulfram

Cursus	Sem.	Type
Auditeurs en ligne	E	Opt.
Biomedical technologies minor	E	Opt.
Computational and Quantitative Biology		Opt.
Computational biology minor	E	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	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.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Neuroscience		Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	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

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, computers, 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. Associative Memory and Attractor Dynamics (Hopfield Model)
6. Neuronal Populations and mean-field methods
7. Continuum models and perception
8. Competition and models of Decision making
III. Noise and the neural code
9. Noise and variability of spike trains (point processes, renewal process, interval distribution)
10: Variance of membrane potentials and Spike Response Models
11. Population dynamics: Fokker-Planck equation
12. Fitting Neural Models to Data
13. Neural Manifolds and low-dimensional dynamics
14. Summary

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, Linear equations,

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
- Predict outcome of dynamics
- Prove stability and convergence
- Describe neuronal phenomena
- Test model concepts in simulations

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Collect data.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Write a scientific or technical report.

Teaching methods

- Video lectures, combined with classroom teaching, classroom discussions, exercises and miniproject. One of the two exercise hours is integrated into the lectures.
- Short mooc-style videos are available as support
- Textbook available as support

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%)
The miniproject is done in teams of 2 students.

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: from single neurons to networks and models of cognition / Wulfram Gerstner, Werner M. Kistler, Richard Naud, Liam Paninski

Websites

- https://neuronaldynamics.epfl.ch/
- https://lcnwww.epfl.ch/gerstner/NeuronalDynamics-MOOCall.html

Moodle Link

- https://go.epfl.ch/NX-465

Videos

- https://lcnwww.epfl.ch/gerstner/NeuronalDynamics-MOOCall.html

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.
Minor in Imaging	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
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

- Computer Vision.
- Signal Processing
- Machine Learning.

Important concepts to start the course

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

Learning Outcomes

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

- 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 on a mobile platform.
- 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.
- Create a computational photography application.

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 compuational photography application on a mobile plateform (iOS, Android).

Expected student activities

The studens is expected to attend the class and actively participate in the practical group project, which requires coding on either Android or iOS plateform. 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

Resources

Bibliography

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

Moodle Link

- https://go.epfl.ch/CS-413

CS-442 Computer vision

Fua Pascal

Cursus	Sem.	Type
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.
Minor in Imaging	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
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 Python.

Assessment methods

With continuous control

Resources

Bibliography

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

Ressources en bibliothèque

- Multiple View Geometry in Computer Vision / Zisserman
- Computer Vision: Algorithms and Applications / Szeliski

Moodle Link

- https://go.epfl.ch/CS-442

COM-418 Computers and music

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Minor in digital humanities, media and society	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Oral
Workload	180h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Remark

Pas donné en 2024-25

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

Recommended courses

Signals and systems, Python, C++

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

Moodle Link

- https://go.epfl.ch/COM-418

CS-453 Concurrent computing

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 of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	2 weekly
Number of positions	

Summary

With the advent of modern architectures, 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
Multicore and multiprocessors architecture

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
Counters General and limited operations
Atomic counters and snapshots

Hierarchy of objects
The FLP impossibility
The consensus number
Universal constructions

Transactional memories
Transactional algorithms
Opacity and obstruction-freedom

Anonymous computing

Fault-tolerant shared-memory computing

Keywords

Concurrency, parallelism, algorithms, data structures

Learning Prerequisites

Required courses

ICC, Operating systems

Recommended courses

This course is complementary to the Distributed Algorithms course

Important concepts to start the course

Processes, threads, datas 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

Final exam
Project

Assessment methods

Final exam (theory) and project (practice)

Resources

Notes/Handbook

Algorithms for Concurrent Systems, R. Guerraoui and P. Kouznetsov

Moodle Link

- https://go.epfl.ch/CS-453

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.
Minor in Quantum Science and Engineering	H	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
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

MATH-310 Algebra
MATH-232 Probability and statistics for IC
CS-250 Algorithms I

Recommended courses

COM-301 Computer security and privacy

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 contiuous evaluation, but with final grade between final_exam-1 and final_exam+1.

Supervision

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 Classical Introduction to Cryptography / Vaudenay
- A computational introduction to number theory and algebra / Shoup

Websites

- https://lasec.epfl.ch/teaching.php

Moodle Link

- https://go.epfl.ch/COM-401

Videos

- https://mediaspace.epfl.ch/channel/COM-401+Cryptography+and+security

Prerequisite for

- Advanced cryptography (COM-501)
- Student seminar: security protocols and applications (COM-506)

COM-480 Data visualization

Vuillon Laurent Gilles Marie

Cursus	Sem.	Type
Computational and Quantitative Biology		Obl.
Computational biology 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.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Financial engineering	MA2, MA4	Opt.
Learning Sciences		Opt.
Minor in digital humanities, media and society	E	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
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-250 Algorithms I (BA)
CS-401 Applied data analysis (MA)

Recommended courses

CS-486 Interaction design (MA)
CS-214 Software construction (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 progrmaming language such as C++, Python, Scala.
Familiarity with web-development (you already have a blog, host a webiste). 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 int he browser using HTM5 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%)

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

- Data Visualisation / Kirk
- Visualization Analysis and Design / Munzner
- The Truthful Art / Cairo
- Interactive Data Visualization for the Web / Murray [2 ed. 2017]

Notes/Handbook

Lecture notes

Moodle Link

- https://go.epfl.ch/COM-480

EE-559 Deep learning

Cavallaro Andrea

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering 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.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Financial engineering	MA2, MA4	Opt.
Learning Sciences		Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Minor in Quantum Science and Engineering	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Quantum Science and Engineering	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	4
Withdrawal	Unauthorized
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	**150**

It is not allowed to withdraw from this subject after the registration deadline.

Summary

This course explores how to design reliable discriminative and generative neural networks, the ethics of data acquisition and model deployment, as well as modern multi-modal models.

Content

This course equips students with a comprehensive foundation of modern deep learning, enabling students to design and train discriminative and generative neural networks for a wide range of tasks. Topics include:

- Deep learning applications (natural language processing, computer vision, audio processing, biology, robotics, science), principles and regulations
- Loss functions, data and labels, data provenance
- Training models: gradients and initialization
- Generalization and performance
- Transformers
- Graph neural networks
- Generative adversarial networks
- Variational autoencoders
- Diffusion models
- Multi-modal models
- Interpretability, explanations, bias and fairness

Keywords

machine learning, neural networks, deep learning, python

Learning Prerequisites

Required courses

- Basics in probabilities and statistics
- Linear algebra
- Differential calculus
- Python programming

Recommended courses

- Basics in optimization
- Basics in algorithmic
- Basics in signal processing

Important concepts to start the course

Discrete and continuous distributions, normal density, law of large numbers, conditional probabilities, Bayes, PCA, vector, matrix operations, Euclidean spaces, Jacobian, Hessian, chain rule, notion of minima, gradient descent, computational costs, Fourier transform, convolution.

Learning Outcomes

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

- Interpret the performance of a deep learning model
- Analyze the limitations of a deep learning model
- Justify the choices for training and testing a deep learning model
- Propose new solutions for a given application

Transversal skills

- Respect relevant legal guidelines and ethical codes for the profession.
- Take account of the social and human dimensions of the engineering profession.
- Design and present a poster.
- Make an oral presentation.
- Demonstrate the capacity for critical thinking

Teaching methods

Ex-cathedra lectures, class discussion, exercises (using python), group project.

Expected student activities

Attendance to lectures, participation in discussions, completing exercises, completing a project, reading written material (scientific papers and books).

Assessment methods

Excercises and group project.

Resources

Références suggérées par la bibliothèque

- Deep learning, Goodfellow, MIT Press, 2016

Notes/Handbook

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

Moodle Link

- https://go.epfl.ch/EE-559

CS-502 Deep learning in biomedicine

Cursus	Sem.	Type
Computational biology minor	E	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Minor in life sciences engineering	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	2 weekly
Number of positions	

Remark

Pas donné en 2024-25

Summary

Deep learning offers potential to transform biomedical research. In this course, we will cover recent deep learning methods and learn how to apply these methods to problems in biomedical domain.

Content

The goal of this course is to cover recent deep learning methods and demonstrate how they can be applied to biomedical data. The course will cover ongoing advances in deep learning research for different input data types (e.g., convolutional neural networks for images, graph convolutional neural networks for graph structured data, transformers for sequence data). We will start with a standard supervised learning setting and then cover the ongoing developments in methodologies that allow us to learn using scarcely labeled datasets by transferring knowledge across tasks (e.g., transfer learning, meta-learning). These settings have particular importance in the biomedical domain in which it is often very difficult to obtain labeled datasets. Recent papers from the literature that apply these methods to problems in biomedicine will be presented and discussed.

In assignments, students will work with popular deep learning software frameworks. They will be evaluated on their ability to understand and implement the methods learned in a class. In the project, students will choose a real-world problem in the biomedical domain and develop a solution for the problem of their choice. They will be evaluated on the ability to propose and develop a suitable model to solve the task, propose suitable evaluation, provide analysis and extract insights from the developed models, write a project report and present project results.

This course is of interest to MS/PhD students interested in recent deep learning methods and their applications to real-world problems in the biomedical domain.

Learning Prerequisites

Required courses

CS-433 Machine learning

Recommended courses

CS-233 Introduction to machine learning

Important concepts to start the course

- Python programming

- Probability and statistics
- Linear Algebra
- Machine learning

Learning Outcomes

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

- Understand and implement deep learning methods covered in the course
- Understand benefits and shortcomings of the methods covered in the course
- Understand common problems in the biomedical domain and know which methods are suitable for solving these problems
- Review academic research papers and understand their contributions according to concepts covered in the course
- Complete a project that applies learned algorithms to a real-world problem in the biomedical domain

Teaching methods

- Lectures
- Paper reading
- Course project

Expected student activities

- Attend lectures and participate in class
- Complete homework assignments
- Complete a deep learning project in a group. This includes preparing a project proposal, implementing the method, submitting final project report and presenting project results

Assessment methods

- Assignments during the semester (50%)
- Project (50%)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Goodfello, Bengio, Courville. Deep Learning. MIT Press (2016)

Ressources en bibliothèque

- Deep Learning / Goodfellow

Moodle Link

- https://go.epfl.ch/CS-502

CS-456 Deep reinforcement learning

Gulcehre Caglar

Cursus	Sem.	Type
Computational biology minor	E	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering 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.
Electrical Engineering		Obl.
Financial engineering	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Quantum Science and Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	1 weekly
Number of positions	

Summary

This course provides an overview and introduces modern methods for reinforcement learning (RL.) The course starts with the fundamentals of RL, such as Q-learning, and delves into commonly used approaches, like PPO and DQN. The course will introduce students to practical applications of RL.

Content

- *Introduction and Overview (What is RL?)*

- *An overview of neural networks and deep learning approaches*

- *Deep learning frameworks*

- *Supervised learning of behaviors (behavior cloning)*

- *Value function methods and related theory*
- *Policy gradient methods and related theory*
- *Actor-Critic Algorithms (A2C, A3C)*
- *Deep RL with Q functions (DQN, R2D2)*
- *Deep Policy Gradient and Optimization methods (PPO, TRPO, Impala, MPO)*
- *Model-based RL and Planning (Alphago, Alphazero, Dreamer)*
- *Exploration and credit assignment in Deep RL*
- *Offline RL (BVE, CQL, CRR, ...)*
- *Deep Imitation learning and Learning from demonstrations (DAGGER, DQFD, R2D3, Learning from play, Third person imitation*
- *RL from human feedback and alignment (InstructGPT, DPO, ReST, etc.)*
- *Advanced continuous control approaches (DDPG, D4PG, SAC)*

- *A selection of extra topics from:*

  - MPO, IMPALA
  - Distributional RL
  - Multi-agent RL (Centralized Training, Decentralized Execution)

Keywords

Deep learning, reinforcement learning, TD learning, SARSA, Actor-Critic Networks, policy gradients, alphago, alphastar, planning, alignment, RLHF, PPO

Learning Prerequisites

Required courses

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

Recommended courses

- Introduction to machine learning (CS-233)
- Machine learning (CS-433)

Important concepts to start the course

- *Regularization in machine learning,*
- *Gradient descent. Stochastic gradient descent.*
- *Expectation, statistics*
- *Linear algebra and probabilities*
- *programming*

Learning Outcomes

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

- Apply Understand and define basic problems and tasks in reinforcement learning (like Markov decision process, model-based and model-free RL, on-policy vs off-policy RL)
- Assess / Evaluate Formulate a real-world problem as an RL setting to apply the approaches taught in the class.
- Elaborate Implement standard deep RL algorithms.
- Judge Understand the failure modes of these models and learning algorithms.
- Propose Read and review academic papers to understand their contributions and learn how to evaluate them critically.
- Apply Students gain the skills and knowledge necessary to tackle complex problems in autonomous robotics, game-playing, and other domains through lectures, hands-on coding exercises, practical applications, and course projects.

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

- Lectures
- Lab sessions
- Individual course projects
- Paper reading
- Group projects

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
- Work on a project

Assessment methods

- Written final exam (25%)
- Assignments (25%)
- Course project (50%)

Supervision

Office hours	No
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.

Resources

Bibliography

• Textbook: Reinforcement Learning by Sutton and Barto (MIT Press). Pdfs of the preprint version of the book are availble online

Ressources en bibliothèque

- Reinforcement Learning / Sutton

Moodle Link

- https://go.epfl.ch/CS-456

CS-411

Digital education

Dillenbourg Pierre, Jermann Patrick

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Learning Sciences		Opt.
Minor in digital humanities, media and society	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Winter, Summer
Semester	Spring
Exam	Written
Workload	180h
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: do student actually learn due to technologies? I will be given twice in 2025 (Spring + Fall)

Content

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

Keywords

learning, pedagogy, teaching, online education, maker spaces

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

Transversal skills

- Set objectives and design an action plan to reach those objectives.
- Identify the different roles that are involved in well-functioning teams and assume different roles, including leadership roles.
- Take account of the social and human dimensions of the engineering profession.

- Write a scientific or technical report.

Teaching methods

The course will combine participatory lectures with a project on designing a learning environment, using it in a controlled experiment, analysing the learning effects with statistical methods and writing a report.

Expected student activities

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

Assessment methods

- Project + exam
- 50 / 50

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-411

CS-451 Distributed algorithms

Guerraoui Rachid

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

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	3 weekly
Number of positions	

Summary

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

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, ditributed 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 cathedera
Lectures, exercises and practical work

Assessment methods

Final exam (theory)
Project (practice)

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

Moodle Link

- https://go.epfl.ch/CS-451

CS-423

Distributed information systems

Aberer Karl

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

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	1 weekly
Number of positions	

Summary

This course introduces the foundations 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. Document Classification (knn, Naive Bayes, Fasttext, Transformer models)
2. Recommender Systems (collaborative filtering, matrix factorization)
3. 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

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 projects (Python)

Assessment methods

60% Continuous evaluations of projects with bonus system during the semester
40% Final written exam (180 min) during exam session

Resources

Moodle Link

- https://go.epfl.ch/CS-423

ENG-466

Distributed intelligent systems

Martinoli Alcherio

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Energy Science and Technology	MA1, MA3	Opt.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Mechanical engineering	MA1, MA3	Opt.
Microtechnics	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	5
Withdrawal	Unauthorized
Session	Winter
Semester	Fall
Exam	Oral
Workload	150h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Project	1 weekly
Labs	2 weekly
Number of positions	**40**

It is not allowed to withdraw from this subject after the registration deadline.

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

Swarm intelligence, artificial intelligence, machine-learning, distributed robotics, swarm robotics, multi-robot systems, sensor networks, modeling, control, optimization

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 course project in teams

Expected student activities

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

Assessment methods

Oral exam (50%) with continuous assessment during the semester based on course project (50%).

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

Yes

Bibliography

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

Websites

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

Moodle Link

- https://go.epfl.ch/ENG-466

Videos

- https://mediaspace.epfl.ch/channel/ENG-466+Distributed+intelligent+systems+-+Spring+2022/29956

Prerequisite for

R&D activities in engineering

DH-415 Ethics and law of AI

Rochel Johan Robert

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Minor in digital humanities, media and society	H	Opt.
Neuro-X	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	4
Withdrawal	Unauthorized
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	**100**

It is not allowed to withdraw from this subject after the registration deadline.

Summary

This master course enables students to sharpen their proficiency in tackling ethical and legal challenges linked to Artificial Intelligence (AI). Students acquire the competence to define AI and identify ethical and legal questions linked to its conception and increased use in society.

Content

AI is used as shortcut-concept to identify a number of computational systems producing intelligent behavior, i.e., complex behavior conducive to reaching goals. AI systems are increasingly used across society. They raise conceptual issues (how to define AI?), technological-ethical issues (how should AI systems be conceived?), legal issues (how to define the responsibility of an AI system? how to regulate AI?) and social-political issues (which justice questions does the deployment of AI raise?)
The following issues will be dealt with:
- What is ethics?
- What is an AI system?
- Who is responsible for the actions of an AI system?
- What are the most pressing ethical questions in the phase of conception of AI systems?
- How should we design AI system in order to overcome ethical-legal challenges?
- Should we regulate AI?
- How should we address the consequences of the wide deployment of AI systems?

Keywords

artificial intelligence, ethics, law, data, regulation, responsibility

Learning Prerequisites

Required courses

No pre-requirement

Learning Outcomes

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

- Define the concept of AI
- Assess / Evaluate the contexts in which AI is deployed
- Systematize general principles (law and ethics)
- Analyze the different senses/conceptions/interpretations of agency, autonomy and responsibility
- Develop principles for the conception of AI system
- Distinguish legal and ethical arguments

Transversal skills

- Demonstrate the capacity for critical thinking
- Take account of the social and human dimensions of the engineering profession.
- Respect relevant legal guidelines and ethical codes for the profession.
- Use a work methodology appropriate to the task.

Teaching methods

The course will be organized as an interactive and participative course. For the weekly course: students have to read texts and to be ready for critical discussion. For the weekly exercise: students have to engage in group discussions. The course requires reading complex texts in English.

Expected student activities

Weekly reading of preparatory texts
Active participation in class, both course and exercise

Assessment methods

Students will be assessed in the following way :
- Mid-term: students will have to answer 2 questions during class (compulsory, no grading)
- Open book written exam during the term (100% of the grade)

Supervision

Office hours	No
Assistants	Yes
Others	Upon appointment with Dr Rochel

Resources

Bibliography

All resources will be made available on moodle.
To start with: *AI Ethics* (Mark Coekelbergh, MIT 2020)

Ressources en bibliothèque

- AI Ethics / Mark Coeckelbergh

Moodle Link

- https://go.epfl.ch/DH-415

CS-550

Formal verification

Kuncak Viktor

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

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Project	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
- Set theory for verification

Learning Prerequisites

Recommended courses

CS-320 Computer language processing

Important concepts to start the course

Discrete Mathematics (e.g. Kenneth Rosen: Discrete Mathematics and Its Applications)

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 materials, take mid-term, and complete and explain projects during the semester.

Assessment methods

The grade is based on the written mid-term, as well as code, documentation, and explanation of projects during the semester. Specific percentages will be communicated in the first class.

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 Rayan: 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 Practical Logic and Automated Reasoning / Harrison
- Handbook of model checking / Clarke
- [chapter] Model Checking Security Protocols / Bassin
- Principles of Program Analysis / Flemming
- The Calculus of Computation / Bradley
- Logic in Computer Science / Huth
- Introduction to mathematical logic and type theory / Andrews

Notes/Handbook

https://lara.epfl.ch/w/fv

Websites

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

Moodle Link

- https://go.epfl.ch/CS-550

COM-406

Foundations of Data Science

Gastpar Michael C., Urbanke Rüdiger

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

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
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.

Moodle Link

- https://go.epfl.ch/COM-406

CS-457 Geometric computing

Pauly Mark

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer science	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Exercises	1 weekly
Labs	1 weekly
Number of positions	

Summary

This course will cover mathematical concepts and efficient numerical methods for geometric computing. We will explore the beauty of geometry and develop 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 homeworks, final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-457

MATH-360 Graph theory

Vérité Mathieu Jean-Paul

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

Language of teaching	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
- (electronic version) Diestel
- (electronic version) Bollobas
- Combinatorics and Graph Theory / Harris, Hirst & Mossinghoff
- Graph Theory / Harary

Moodle Link

- https://go.epfl.ch/MATH-360

EE-451

Image analysis and pattern recognition

Bozorgtabar Behzad, Thiran Jean-Philippe

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational and Quantitative Biology		Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Minor in Imaging	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Physics of living systems minor	E	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.

Language of teaching	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 and mini-project on computers

Keywords

image processing, image analysis, image segmentation, feature extraction, introduction to machine learning, pattern recognition.

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

Teaching methods

Ex cathedra and practical work and oral presentation by the students

Assessment methods

Continuous control : oral exam during the semester + graded reports and mini-poject

Resources

Moodle Link

- https://go.epfl.ch/EE-451

Prerequisite for

Semester project, Master project, doctoral thesis

COM-402 Information security and privacy

Payer Mathias

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering 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	Obl.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Opt.
SC master EPFL	MA1, MA3	Obl.
Statistics	MA1, MA3	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
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 world. The tools are illustrated with relevant applications.

Content

- Overview of cyberthreats
- Basic exploitation of 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 systems programming (in C/C++) or better
Basic networking knowledge
Good scripting knowledge (Python)

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 in their home and professional lives
- Understand at overview level the key technical tools available for security/privacy protection
- Understand the key technical tools available for security/privacy protection
- Exercise competent operational security practices

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

Resources

Moodle Link

- https://go.epfl.ch/COM-402

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.
Minor in Quantum Science and Engineering	H	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
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 fundamenal 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

Websites

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

Moodle Link

- https://go.epfl.ch/COM-404

CS-430

Intelligent agents

Faltings Boi

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.
Financial engineering minor	H	Opt.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	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	

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

Midterm and quizzes 30%, final exam 70%

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/4th Edition), Prentice Hall Series in Artificial Intelligence, 2003/2009/2015.

Ressources en bibliothèque

- An Introduction to MultiAgent Systems / Wooldridge
- Artificial Intelligence: A Modern Approach / Russell

Moodle Link

- https://go.epfl.ch/CS-430

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 of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
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 further divided into mini-projects evenly spaced throughout the semester.

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, user interviews, ideation, storyboard, context scenarios, digital mockup, user evaluation, video prototyping.

Learning Prerequisites

Required courses

Interaction personne-système

Recommended courses

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

Important concepts to start the course

Goal-directed design, design thinking, user needs assessment, user interviews & observation, ideation, prototyping, evaluation

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 device 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, flipped classroom lectures, exercises, hands-on practice, case studies

Expected student activities

Participation in lectures, exercises, user interviews, ideation sessions, readings, design project, project presentation

Assessment methods

The assessments consist of five in-class open-book exercises, each lasting one hour. Three of these exercises will be randomly selected for grading. Additionally, there will be two mini-projects that will be graded based on group performance. Furthermore, students' individual engagement in group activities, including user interviews, ideation, prototyping, and peer evaluation, will also be evaluated to determine individual performance.
30% open-book exercises (done in class, open notes, open book) - individual performance
20% individual engagement in group activities such as user interviews - individual performance
50% project - group performance

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 / Cooper

Moodle Link

- https://go.epfl.ch/CS-486

Videos

- https://mediaspace.epfl.ch/channel/CS-486%2BInteraction%2BDesign/29793?&&

CS-428

Interactive theorem proving

Barrière Aurèle, Pit-Claudel Clément

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

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	2 weekly
Number of positions	

Summary

A hands-on introduction to interactive theorem proving, computer-checked mathematics, compiler verification, proofs as programs, dependent types, and proof automation. Come learn how to write computer-checked proofs and certified bug-free code!

Content

- Intro to the Coq proof assistant (logic, higher-order functions, tactics)
- Functional programming (inductive types and fixpoints)
- Structural induction (data structures and verified algorithms)
- Interpreter-based program semantics (intro to compiler verification)
- Inductive relations (predicates, rule induction)
- Automation and tactics I (bottom-up reasoning and logic programming)
- Operational program semantics (small-and big-step semantics)
- Program logics (hoare triples)
- Automation and tactics II (top-down reasoning)
- Type systems (Simply-typed lambda calculus)
- Dependent types and equality proofs
- Automation and tactics III (proofs by reflection)
- Real-world theorem proving (various topics)

Learning Prerequisites

Recommended courses

This course assumes no knowledge of programming language theory. The following courses may be useful, but are not required:

- CS-320 Computer language processing (to introduce the concept of interpreter)
- CS-425 Foundations of software (to introduce type systems and the lambda calculus)
- CS-550 Formal verification (for a different perspective on theorem proving)

Important concepts to start the course

- Functional programming

Learning Outcomes

- Implement purely-functional algorithms in the Gallina language
- Translate informal requirements about software into precise mathematical properties
- Plan and carry out mechanized proofs in Coq (e.g. maths, algorithms, compilers, type systems)
- Automate repetitive proof tasks by crafting simple custom decision procedures

Teaching methods

- Lectures
- Live-coding sessions

Expected student activities

- Lectures
- Programming and verification assignments
- Project (proposal, check-in, presentation, report)

Assessment methods

- Take-home programming and verification assignments: 40% of the final grade (3 or 4 labs)
- Formal verification project: 60% of the final grade (~10 weeks, in teams of 1 to 4 students)

Supervision

Office hours	Yes
Assistants	No
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-511

CS-431

Introduction to natural language processing

Bosselut Antoine, Chappelier Jean-Cédric, Rajman Martin

Cursus	Sem.	Type
Computer science minor	H	Opt.
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		Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.
UNIL - Sciences forensiques	H	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
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: morpho-lexical level: n-gram and language models, spell checkers, ...; semantic level: models and formalisms for the representation of meaning, embedings, ...
Several application domains will be presented: Linguistic engineering, Information Retrieval, Textual Data Analysis (automated document classification, visualization of textual data).

Keywords

Natural Language Processing; Computationnal Linguisitics; Part-of-Speech tagging

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, ...)
- Describe the typical problems and processing layers in NLP
- Analyze NLP problems to decompose them in adequate independant components

Teaching methods

Flipped classroom (reviews and supervised "hands-on" in class) ; practical work on computer

Expected student activities

attend lectures and practical sessions, answer quizzes.

Assessment methods

4 quiz during semester 16%, final exam 84%

Resources

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 Indurkhya and Fred J. Damerau editors, "*Handbook of Natural Language Processing*", CRC Press, 2010 (2nd edition)

Ressources en bibliothèque

- Handbook of Natural Language Processing / Indurkhya
- Introduction to Information Retrieval / Manning
- Foundations of Statistical Natural Language Processing / Manning
- Speech and Language Engineering / Rajman
- Speech and Language Processing / Jurafsky

Websites

- https://coling.epfl.ch/

Moodle Link

- https://go.epfl.ch/CS-431

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.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Managmt, tech et entr.	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Withdrawal	Unauthorized
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Practical work	4 weekly
Number of positions	

It is not allowed to withdraw from this subject after the registration deadline.

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 expressedas interquartiles. For instance, 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 ?.

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

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 librairies
- Carry out out 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

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 ineractive 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 is small groups

Supervision

Office hours	Yes

Assistants	Yes
Forum	Yes

Resources

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

Ressources en bibliothèque

- Python Data Science Handbook / VanderPlas

Moodle Link

- https://go.epfl.ch/COM-490

CS-479

Learning in neural networks

Gerstner Wulfram

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Oral
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	1 weekly
Number of positions	

Summary

Artificial Neural Networks are inspired by Biological Neural Networks. One big difference is that optimization in Deep Learning is done with the BackProp Algorithm, whereas in biological neural networks it is not. We show what biologically plausible learning algorithms can do (and what not).

Content

- Why BackProp is biologically not plausible. Biological two-factor rules and neuromorphic hardware
- Hebbian Learning (two-factor rules) for PCA and ICA
- Two-factor rules for dictionary learning (k-means/competitive learning/winner-takes-all)
- Three-factor rules and neuromodulators (theory and neuroscience)
- Three-factor rules for reward-based learning (theory)
- Three-factor rules for TD reinforcement-learning (algorithmic formulations)
- Actor-critic networks
- Reinforcement learning in the brain
- Learning by surprise and novelty: exploration and changing environments (algorithmic)
- Surprise and novelty in the brain
- Learning representations in multi-layer networks (algorithms without backprop)
- Learning to find a goal: a bio-plausible model with place cells and rewards
- Neuromorphic hardware and in-memory computing

Keywords

- Hebbian learning and two-factor rules
- distributed local algorithms,
- Principal Component Analysis/Independent Component Analysis (PCA and ICA)
- Reinforcement Learning (RL)
- surprise and novelty
- three-factor rules
- neuromorphic hardware

Learning Prerequisites

Required courses

Linear Algebra AND Analysis.
Machine learning

Recommended courses

Signal processing

Important concepts to start the course

Optimization, Gradient Descent, Filtering, Loss function, Eigenvalues,

Learning Outcomes

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

- Translate concepts from machine learning and signal processing into bio-plausible algorithms
- Translate neuroscience of learning into algorithms
- Explain differences between and similarities of various algorithms
- Discriminate imitations and advantages of various learning algorithms for implementation in biology or hardware

Transversal skills

- 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.
- Manage priorities.
- Continue to work through difficulties or initial failure to find optimal solutions.

Teaching methods

Ex cathedra, Exercises, and Miniproject

Expected student activities

Participation in Class, Solution of Exercises, Miniproject.

Assessment methods

Oral exam (70 percent) plus miniproject (30 percent). If more than 45 students participate, the oral exam is replaced by a written exam.

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-479

CS-526

Learning theory

Macris Nicolas

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

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
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).
- Bias-variance tradeoff and modern double descent phenomena.
- Neural Nets : representation power of neural nets.
- Stochastic gradient descent, modern aspects: mean field approach, neural tangent kernel.
- Matrix factorization, Tensor decompositions and factorization, Jenrich's tehorem, Alternating least squares, Tucker decompositions.
- Applications: e.g. Learning mixture models, topic modeling.

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, bias-variance tradeoff and double descent
- Describe basic facts about representation of functions by neural networks
- Describe recent results on specific topics e.g., 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.
Life Sciences Engineering	MA1, MA3	Opt.
Mathematics	BA5	Opt.

Language of teaching	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 fundamental tool of statistics, because it describes how the law of a random variable of interest may depend on other variables. This course aims to familiarize students with linear models and some of their extensions, which lie at the basis of more general regression model

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

Moodle Link

- https://go.epfl.ch/MATH-341

CS-433 Machine learning

Flammarion Nicolas, Jaggi Martin

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Communication systems minor	H	Opt.
Computational biology minor	H	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	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		Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**8 weekly**
Lecture	4 weekly
Exercises	2 weekly
Project	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 I (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 problems: Regression, classification, clustering, dimensionality reduction, time-series
- 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
- Define the following basic machine learning models: Regression, classification, clustering, dimensionality reduction, neural networks, time-series analysis

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

- The elements of statistical learning : data mining, inference, and prediction / Friedman
- Pattern Recognition and Machine Learning / Bishop
- Understanding Machine Learning / Shalev-Shwartz
- Machine Learning / Murphy

Références suggérées par la bibliothèque

- Neural Networks and Deep Learning / Nielsen

Notes/Handbook

https://github.com/epfml/ML_course

Websites

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

Moodle Link

- https://go.epfl.ch/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.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Learning Sciences		Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Remark

Cours biennal

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 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 (seemless 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-233 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 disavantages and explain thedifferences 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%)

Resources

Moodle Link

- https://go.epfl.ch/CS-421

MGT-427

Management de projet et analyse du risque

Wieser Philippe

Cursus	Sem.	Type
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Génie civil	MA1, MA3	Obl.
Informatique	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
Microtechnique	MA1, MA3	Opt.
Mineur en Management, technologie et entrepreneuriat	H	Opt.
Robotique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Langue d'enseignement	français
Crédits	4
Retrait	Non autorisé
Session	Hiver
Semestre	Automne
Examen	Pendant le semestre
Charge	120h
Semaines	14
Heures	**3 hebdo**
Cours	2 hebdo
Projet	1 hebdo
Nombre de places	

Il n'est pas autorisé de se retirer de cette matière après le délai d'inscription.

Résumé

Ce cours a pour objectif de présenter l'approche générale du management de projet en intégrant la gestion du risque dans toutes les étapes du projet.

Contenu

Le projet est un processus consistant à réaliser un système dans le but de satisfaire le besoin de futurs utilisateurs. Il sera ainsi aborder dans ce cours la présentation des différentes phases du projet, les organisations de projet, les moyens et méthodes de développement des variantes de projets, le choix multicritère de la variante à réaliser, la planification et le suivi dans la phase de réalisation. Une approche « risque » sera abordée à chaque étape du projet par une méthodologie probabiliste.

- Les principes généraux du management de projet
- Les phases d'un projet
- L'approche stratégique et opérationnel d'un projet
- Le management des risques
- Les formes organisationnelles de projets
- Evaluation physique de variantes de projets
- Evaluation comportementale de variantes de projets par simulation numérique
- Evaluation économique de projet
- choix de variante(s), analyse multicritère
- Planification et ordonnancement de projet
- Exemples et études de cas
- Applications informatiques

Mots-clés

Management de projet, risques, évaluation économique, planification, analyse multicritère

Compétences requises

Cours prérequis obligatoires

Néant

Cours prérequis indicatifs

Notions de base de statisques

Concepts importants à maîtriser

Ouvert, curieux et capable d'aborder un domaine complexe multidisciplinaire et multiculturel.
L'étudiant devra avoir une vision transversale des processus et être capable de raisonner de manière systémique

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Diriger une équipe de projet
- Elaborer des variantes de projet
- Planifier les variantes de projet
- Sélectionner ou choisir la variante retenue
- Anticiper les risques
- Organiser
- Restituer et communiquer
- Implémenter

Compétences transversales

- Planifier des actions et les mener à bien de façon à faire un usage optimal du temps et des ressources à disposition.
- Communiquer efficacement et être compris y compris par des personnes de languages et cultures différentes.
- Etre responsable des impacts environnementaux de ses actions et décisions.
- Dialoguer avec des professionnels d'autres disciplines.
- Recevoir du feedback (une critique) et y répondre de manière appropriée.
- Recueillir des données.
- Faire une présentation orale.

Méthode d'enseignement

Ex cathedra, projet avec présentation orale

Travail attendu

Suivi des cours et étude des documents de cours distribués
Réalisation d'un projet durant le semestre. Ce projet est réalisé en groupe et présenté dans les dernières séances du cours

Méthode d'évaluation

- 50% projet durant le semestre
- 50% examen final

Encadrement

Office hours	Oui
Assistants	Oui
Forum électronique	Non

Autres	Disponibilité de l'enseignant par email, téléphone ou visite à son bureau

Ressources

Bibliographie

Indiquée en rapport avec chaque chapitre du cours

Polycopiés

Cours et copies des slides de présentation envoyés à chaque étudiant sous format électronique

Liens Moodle

- https://go.epfl.ch/MGT-427

Préparation pour

Travaux de semestre
Projet de Master

COM-516 Markov chains and algorithmic applications

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

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	1 weekly
Number of positions	

Remark

Pas donné en 2024-25

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

Moodle Link

- https://go.epfl.ch/COM-516

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.

EE-556 Mathematics of data: from theory to computation

Cevher Volkan

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Data Science	MA1, MA3	Opt.
Data science minor	H	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
MNIS	MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
Robotics	MA1, MA3	Opt.
Statistics	MA1, MA3	Opt.

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

Summary

This course provides an overview of key advances in continuous optimization and statistical analysis for machine learning. We review recent learning formulations and models as well as their guarantees, describe scalable solution techniques and algorithms, and illustrate the trade-offs involved.

Content

The course consists of the following lectures (2h each)
Lecture 1: Introduction. The role of models and data. Maximum-likelihood formulations. Error decomposition for estimation and prediction.
Lecture 2: Generalized linear models. Logistic regression.
Lecture 3: Linear algebra reminders. Computation of Gradients. Reading convergence plots.
Lecture 4: The role of computation. Challenges to optimization algorithms. Optimality measures. Structures in optimization. Gradient descent. Convergence rate of gradient descent for smooth functions.
Lecture 5: Optimality of convergence rates. Lower bounds. Accelerated gradient descent. Concept of total complexity. Stochastic gradient descent.
Lecture 6: Concise signal models. Compressive sensing. Sample complexity bounds for estimation and prediction. Challenges to optimization algorithms for non-smooth optimization. Subgradient method.
Lecture 7: Introduction to proximal-operators. Proximal gradient methods. Linear minimization oracles. Conditional gradient method for constrained optimization.
Lecture 8: Time-data trade-offs. Variance reduction for improving trade-offs.
Lecture 9: Introduction to deep learning. Generalization through uniform convergence bounds. Rademacher complexity.
Lecture 10: Double descent curves and over-parameterization. Implicit regularization. Generalization bounds using stability.
Lecture 11: Escaping saddle points. Adaptive gradient methods.
Lecture 12: Adversarial machine learning and generative adversarial networks (GANs). Wasserstein GAN. Difficulty of minimax optimization. Pitfalls of gradient descent-ascent approach.
Lecture 13: Primal-dual optimization-I: Fundamentals of minimax problems. Fenchel conjugates. Duality.
Lecture 14: Primal-dual optimization-II: Extra gradient method. Chambolle-Pock algorithm. Stochastic primal-dual methods.
Lecture 15: Primal-dual III: Lagrangian gradient methods. Lagrangian conditional gradient methods.

Learning Prerequisites

Required courses

Previous coursework in calculus, linear algebra, and probability is required. Familiarity with optimization is useful.
Familiarity with python, and basic knowledge of one deep learning framework (Pytorch, TensorFlow, JAX) is needed.

Resources

Moodle Link

- https://go.epfl.ch/EE-556

CS-552 Modern natural language processing

Bosselut Antoine

Cursus	Sem.	Type
Computational and Quantitative Biology		Opt.
Computer and Communication Sciences		Opt.
Computer science minor	E	Obl.
Computer science	MA2, MA4	Opt.
Data Science	MA2, MA4	Obl.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	During the semester
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	1 weekly
Project	2 weekly
Number of positions	

Summary

Natural language processing is ubiquitous in modern intelligent technologies, serving as a foundation for language translators, virtual assistants, search engines, and many more. In this course, students will learn algorithmic tools for tackling problems in modern NLP.

Content

This course includes lectures, exercises, a midterm exam, and a project. In lectures, we will cover the foundations of modern methods for natural language processing, such as word embeddings, recurrent neural networks, transformers, pretraining, and how they can be applied to important tasks in the field, such as machine translation and text classification. We will also cover issues with these state-of-the-art approaches (such as robustness, interpretability, sensitivity), identify their failure modes in different NLP applications, and discuss analysis and mitigation techniques for these issues.
In assignments, students will be evaluated on their ability to implement methods learned in class on closed-form problems developed by the course staff. In their project, students will be expected to apply techniques learned in lecture to a problem of the course staff's choosing. They will formulate the problem as an NLP task, propose a suitable evaluation to measure their progress, develop a model to solve the task, and provide analysis of the strengths and weaknesses of their method.
This course is of interest to MS / PhD student interested in modern methods and issues in natural language processing, both from a research and applied perspective.

Learning Prerequisites

Recommended courses

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

Important concepts to start the course

- Python programming
- Probability and Statistics
- Linear Algebra
- Machine Learning concepts

Learning Outcomes

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

- Define basic problems and tasks in natural language processing (e.g., machine translation, summarization, text classification, language generation, sequence labeling, informaiton extraction, question answering)
- Implement common modern approaches for tackling NLP problems and tasks (embeddings, recurrent neural models, attentive neural models) and how to train them
- Understand failure modes of these models and learning algorithms (e.g., robustness, interpretability/explainability, bias, evaluation)
- Review academic research papers and understand their contributions, strengths, and weaknesses according to the principles learned in lecture
- Complete a project that applies these algorithms to a real-world NLP problem, where they will define a task, evaluation, model implementation, and analyze the shortcomings of their approach

Teaching methods

- Lectures
- Lab sessions
- Midterm Examination
- Course project

Expected student activities

- Attend lectures and participate in class
- Complete homework assignments
- Pass midterm exam
- Complete a project (set by the course supervisor) : complete a project proposal outlining topic and evaluation plan; submit two project milestones; submit final project report; present project findings to committee of course instructor and TAs.

Assessment methods

- Midterm examination (30%)
- Project (70%)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-552

EE-452

Network machine learning

Frossard Pascal, Thanou Dorina

Cursus	Sem.	Type
Computational and Quantitative Biology		Opt.
Computational biology minor	E	Opt.
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 of teaching	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

Fundamentals, methods, algorithms and applications of network machine learning and graph neural networks

Content

Context
In the last decade, our information society has mutated into a data society, where the volume of worldwide data grows increasingly fast. An increasing amount of this data is structured on networks of different forms. How to make sense of such tremendous volume of data? Developing effective techniques to extract meaningful information from large-scale and high-dimensional network datasets has become essential for the success of business, government and science.

Objective
The goal of this course is to provide a broad introduction to effective methods algorithms in data science, network analysis and network machine learning. A major effort will be given to show that existing data analysis techniques can be defined and enhanced on graphs. Graphs can encode complex structures like cerebral connection, stock exchange, and social network. Strong mathematical tools have been developed based on statistics, or linear and non-linear graph spectral harmonic analysis to advance the standard data analysis algorithms. At the same time, modern machine learning tools such as neural networks have been adapted to process data defined on network structures. The objective of the class is to develop fundamentals and review algorithms that permit to develop modern network data analysis methods. The main topics of the course are networks, network data analysis, unsupervised and supervised learning on graphs and networks, graph generative models, sparse representation, multi-resolution analysis, graph neural networks.

Structure
The course is organized into two parts: lectures (2 hours) and lab assignements and projects (1 hour). The essential objective of the exercises and lab assignements is to apply the theory on real-world cases. The objective of the projects is to study practical network machine learning cases, and develop effective solutions based on tools studied in the class.

Evaluation
Evaluation will be conducted on a continuous basis: homeworks and coding assignments.

Keywords

graph representation learning, machine learning, network science

Learning Prerequisites

Required courses

Fundamentals of Machine Learning, or equivalent
Signal Processing, or equivalent
Introduction to Statistics, or equivalent
Python programming

Learning Outcomes

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

- Apply modern machine learning techniques to network data
- Analyze network properties, network data distributions, and properties of the most common network machine learning algorithms
- Propose solutions for network data analysis problems

Transversal skills

- Use a work methodology appropriate to the task.
- Give feedback (critique) in an appropriate fashion.
- Communicate effectively, being understood, including across different languages and cultures.

Resources

Moodle Link

- https://go.epfl.ch/EE-452

COM-512 Networks out of control

Cursus	Sem.	Type
Communication systems minor	E	Opt.
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 of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal - Pas donné en 2024-25

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: Erdös-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 stastistics; 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%.

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
- Percolation / Grimmett
- Networks, Crowds and Markets / Easley
- Poisson Approximation / Barbour
- Random Graph Dynamics / Durrett

Notes/Handbook

Class notes will be available on the course website.

Moodle Link

- https://go.epfl.ch/COM-512

CS-439 Optimization for machine learning

Flammarion Nicolas

Cursus	Sem.	Type
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Obl.
Data science minor	E	Opt.
Electrical Engineering		Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Quantum Science and Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	2 weekly
Labs	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

Resources

Websites

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

Moodle Link

- https://go.epfl.ch/CS-439

Videos

- https://www.youtube.com/playlist?list=PL4O4bXkI-fAeYrsBqTUYn2xMjJAqlFQzX

COM-508

Optional research project in Data Science

Profs divers *

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

Language of teaching	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 approval of the section. Only for 2nd year Master students. Supervision by an IC authorized professor

Summary

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

Content

Supervisor and subject to be chosen among the themes proposed on the web site :
Projects by laboratory

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 proejct 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.

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.

Resources

Websites

- https://www.epfl.ch/schools/ic/education/master/master-project/

Moodle Link

- https://go.epfl.ch/COM-508

COM-412 Research project in Data Science

Profs divers *

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

Language of teaching	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 :
Projects by laboratory

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/

Moodle Link

- https://go.epfl.ch/COM-412

CS-412 Software security

Payer Mathias

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
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 of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	During the semester
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	2 weekly
Labs	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 (or an equivalent security course)
- A systems programming course (with focus on C/C++)
- An operating systems course

Important concepts to start the course

Basic computer literacy like system administration, build systems, 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 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. Homework exercises consist mostly of paper questions involving the analysis, critical review, and occasional correction of software. They include a reading, writing, and presentation assignment.
The labs focus on practical software security aspects and during the course the students will be assessed through their

completion of several challenging "hands on" labs.

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 pracitcal assignments in the form of several labs and theoretical quizzes throughout the semester. The labs will account for 50%, the quizzes and tests to 50%.
The exact dates of the labs/quizzes will be communicated at the beginning of the class.

Resources

Notes/Handbook

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

Moodle Link

- https://go.epfl.ch/CS-412

MATH-486 Statistical mechanics and Gibbs measures

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

Language of teaching	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

Pas donné en 2024-25. Cours donné en alternance une année sur deux.

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 occuring 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 1et 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.

Supervision

Office hours	No
Assistants	No
Forum	No

Resources

Virtual desktop infrastructure (VDI)

No

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

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

Websites

- http://www.unige.ch/math/folks/velenik/smbook/

Moodle Link

- https://go.epfl.ch/MATH-486

PHYS-512

Statistical physics of computation

Erba Vittorio

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

Language of teaching	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 students understand tools from the statistical physics of disordered systems, and apply them to study computational and statistical problems in graph theory, discrete optimisation, inference and machine learning.

Content

Interest in the methods and concepts of statistical physics is rapidly growing in fields as diverse as theoretical computer science, probability theory, machine learning, discrete mathematics, optimization, signal processing and others. Large part of the related work has relied on the use of message-passing algorithms and their connection to the statistical physics of glasses and spin glasses.

This course covers this active interdisciplinary research landscape. Specifically, we will review the statistical physics approach to problems ranging from graph theory (e.g. community detection) to discrete optimization and constraint satisfaction (e.g. satisfiability or coloring) and to inference and machine learning problems (learning in neural networks, clustering of data and of networks, compressed sensing or sparse linear regression, low-rank matrix factorization).

We will expose theoretical methods of analysis (replica, cavity, ...) algorithms (message passing, spectral methods, etc), discuss concrete applications, highlight rigorous justifications as well as present the connection to the physics of glassy and disordered systems.

This is an advanced theoretical course that is designed for students with background in mathematics, electrical engineering, computer science or physics. This course exposes advanced theoretical concepts and methods, with exercises in the analytical methods and usage of the related algorithms.

Learning Prerequisites

Important concepts to start the course

For physics students Statistical physics I and II (or equivalent) is required.
This lecture is accessible to students in mathematics, electrical engineering, computer science without any previous training in statistical physics. Those students are expected to have strong interest in theory, probabilistic approaches to analysis of algoriths, high-dimensional statistics or probabilistic signal processing.

Learning Outcomes

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

- Analyze theoretically a range of problems in computer science and learning.
- Derive algorithms for a range of computational problems using technics stemming from statistical physics.

Teaching methods

2h of lecture + 2h of excercise

Assessment methods

Final written exam counting for 50% and graded homework during the semester counting for the other 50%.

Resources

Bibliography

Information, Physics and Computation (Oxford Graduate Texts), 2009, M. Mézard, A. Montanari
Statistical Physics of inference: Thresholds and algorithms, Advances in Physics 65, 5 2016, L. Zdeborova & F. Krzakala, available at https://arxiv.org/abs/1511.02476

Ressources en bibliothèque

- Information, Physics and Computation / Mézard

Références suggérées par la bibliothèque

- Statistical Physics of inference: Thresholds and algorithms / Zdeborova & Krzakala

Notes/Handbook

Policopié "Statistical Physics methods in Optimization & Machine Learning" by L. Zdeborova & F. Krzakala, available athttps://sphinxteam.github.io/EPFLDoctoralLecture2021/Notes.pdf

Moodle Link

- https://go.epfl.ch/PHYS-512

MATH-442

Statistical theory

Zemel Yoav

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

Language of teaching	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

-

Resources

Moodle Link

- https://go.epfl.ch/MATH-442

MATH-413 Statistics for data science

Chandak Rajita Ramesh, Limnios Myrto

Cursus	Sem.	Type
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Data Science	MA2, MA4	Obl.
Data science minor	E	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Managmt, tech et entr.	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
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 enountered 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
- Statistical Models
- Statistics for Mathematicians
- All of Statistics

Moodle Link

- https://go.epfl.ch/MATH-413

COM-506 Student seminar: security protocols and applications

Vaudenay Serge

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 of teaching	English
Credits	3
Session	Summer
Semester	Spring
Exam	During the semester
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

Assessment methods

- lecture and attendance to others' lectures

Supervision

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

Resources

Websites

- https://lasec.epfl.ch/teaching.php

Moodle Link

- https://go.epfl.ch/COM-506

Videos

- https://mediaspace.epfl.ch/channel/COM-506+Student+Seminar+on+Security+Protocols+and+Applications

CS-448

Sublinear algorithms for big data analysis

Kapralov Michael

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

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
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

Resources

Moodle Link

- https://go.epfl.ch/CS-448

CS-460 Systems for data management and data science

Ailamaki Anastasia, Kermarrec Anne-Marie

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	E	Opt.
Computer science	MA2, MA4	Obl.
Cybersecurity	MA2, MA4	Obl.
Data Science	MA2, MA4	Obl.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Labs	2 weekly
Number of positions	

Summary

This course is intended for students who want to understand modern large-scale data analysis systems and database systems. The course covers fundamental principles for understanding and building systems for managing and analyzing large amounts of data. It covers a wide range of topics and technologi

Content

Topics include large-scale data systems design and implementation, and specifically :

- Distributed data management systems
- Data management : locality, accesses, partitioning, replication
- Modern storage hierarchies
- Query optimization, database tuning
- Transaction management
- Data structures : File systems, Key-value stores, DBMS
- Consistency models
- Large-scale data analytics infrastructures
- Parallel Processing
- Data stream and graph processing

Learning Prerequisites

Recommended courses

- CS-107 Introduction to programming
- CS-214 Software construction
- CS-300 Data-Intensive Systems
- CS-202 Computer systems
- CS-452 Foundations of software

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 recommanded.
- Basic knowledge or computer networking and distributed systems.

Learning Outcomes

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

- Understand how to design big data analytics systems using state-of-the-art infrastructures for horizontal scaling, e.g., Spark
- Implement algorithms and data structures for streaming data analytics
- Decide between different storage models based on the offered optimizations enabled by each modl and the expected query workload
- Compare concurrency control algorithms, and algorithms for distributed data management
- Configure systems parameters, data layouts, and application designs for database systems
- Develop data-parallel analytics programs that make us of modern clusters and cloud offerings to scale up to very large workloads
- Analyze the trade-offs between various approaches to large-scale data management and analytics, depending on efficiency, scalability, and latency needs

Teaching methods

Lectures, project, homework, exercises and practical work

Expected student activities

- Attend lectures and participate in class
- Complete a project as per the guidelines posted by the teaching team

Assessment methods

- Project
- Midterm (as needed)
- Final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

J. Hellerstein & M. Stonebraker, Readings in Database Systems, 4th Edition, 2005
R. Ramakrishnan & J. Gehrke: "Database Management Systems", McGraw-Hill, 3rd Edition, 2002.
A. Rajaraman & J. Ullman: "Mining of Massive Datasets", Cambridge Univ. Press, 2011.

Ressources en bibliothèque

- Mining of Massive Datasets / Rajaraman
- Database Management Systems / Ramakrishnan
- Readings in Database Systems / Hellerstein

Moodle Link

- https://go.epfl.ch/CS-460

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.
Statistics	MA2, MA4	Opt.

Language of teaching	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 & mid term assessed coursework - counts 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

Assistants	Yes
Forum	No

Resources

Bibliography

Lecturenotes available at https://moodle.epfl.ch/course/view.php?id=15393

Ressources en bibliothèque

- Spectral Analysis for Physical Applications / Percival
- Analysis of Financial Time Series / Tsay
- Introduction to Time Series and Forecasting / Brockwell & Davis
- (electronic version) Analysis of Financial Time Series
- Time Series Analysis and its Applications, with R Examples / Shumway & Stoffer
- (electronic version) Time Series Analysis and its Applications, with R Examples
- (electronic version) Introduction to Time Series and Forecasting

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.

Moodle Link

- https://go.epfl.ch/MATH-342

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 of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	3 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal - pas donné en 2024-25

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

Moodle Link

- https://go.epfl.ch/CS-455

CS-444 Virtual reality

Boulic Ronan

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Learning Sciences		Opt.
Minor in digital humanities, media and society	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Project	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 programming techniques for achieving efficient VR applications

Content

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

- Visual display
- Interaction devices and sensors
- Software environment (UNITY3D, programming in C#)

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 ?
- VR, cognitive science and true experimental design

The group project focuses on CODING a 3D VR game. So we recommend to rather attend DH-414 "Game design & prototying" in case of lack of coding background as this latter does not require coding skills.

Spring 2024-25 will be the last instance of CS-444 in the study plan.

Keywords

3D interaction, display, sensors, immersion, presence, embodiment

Learning Prerequisites

Required courses

Mastering an Object-Oriented programming language

Recommended courses

(CS-341) Computer graphics

Important concepts to start the course

1) Object Oriented programming lies at the core of the project development in C# with Unity3D. Some programming experience with this approach is compulsory as all students will be assessed on the individual coding of some features of the project.
2) from Computer Graphics:
- perspective transformations
- representation and manipulation of 3D 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
- Describe how the human perception-action system is exploited in VR
- Apply the concepts of immersion, presence and flow
- Give an example of applications of VR in different inustrial 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
- Give an example of applications of VR in different industrial sectors

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 (in C# within Unity3D). 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. The project development will have to be conducted with git.

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 theoretical oral and the project oral

Assessment methods

Scientific paper study : summary of contributions and citation analysis (around week6 of the semester)
Theoretical oral exam (last week of the semester)
Project assessment through code repository, report and oral exam around the end of the semester

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
- Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 2017. 3D User Interfaces: Theory and Practice. Second edition, Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.
- J. Jerald, The VR Book, ACM Press 2015
- Parisi, Learning Virtual Reality, O'Reilly 2015

Ressources en bibliothèque

- 3D User Interfaces / Bowman
- Learning Virtual Reality / Parisi
- The VR book / Jerald

Notes/Handbook

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

Websites

- http://www.thevrbook.net/
- http://gitlab.epfl.ch

Moodle Link

- https://go.epfl.ch/CS-444

CS-503

Visual intelligence : machines and minds

Zamir Amir

Cursus	Sem.	Type
Civil & Environmental Engineering		Obl.
Computer science	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Minor in Imaging	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	1 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 -- with a primary focus on embodied intelligence and vision 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 inspiration around the missing capabilities and how to approach them, we will turn to visual perception in biological organisms.

The course includes lectures and projects. There will be a heavy emphasis on the *projects* and *hands-on experience*. The course project will be around designing, implementing, and testing a solution to a (preferably open) problem pertinent to visual perception. The students are encouraged to work in groups, self-propose a project that excites them, 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 their course projects. In particular, the lectures will discuss the following:

1. A recap of basic computer vision concepts: classification, detection, grouping, image transformations, optical flow, 3D from X, etc., and recent successful neural network architectures, such as Transformers.
2. Psychology/physiology of the visual system.
3. Perception-action loop: active perception and embodied vision.

The course interests masters/PhD students interested in research in computer vision, machine learning, and perceptual robotics, as well as senior undergraduate students interested in understanding state-of-the-art computer vision.

Keywords

Computer vision, Machine learning, Embodied intelligence, Robotics, Cognition, Neural networks, AI.

Learning Prerequisites

Required courses

- Machine Learning (CS-433) or Introduction to Machine Learning (CS-233) or equivalent course on the basics of machine learning.
- Deep Learning (EE-559) or Artificial Neural Networks (CS-456) or equivalent course on the basics of deep learning.

Recommended courses

- Computer vision (CS-442) or equivalent undergraduate/masters course on the basics of computer.

Important concepts to start the course

- Deep learning and machine learning.
- Python programming.
- 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, demonstrate continuous progress throughout the semester, and provide a final written report and presentation.

Assessment methods

- Project (70%) [distributed over the project proposal, milestone reports, final report and presentation]

- Homeworks (30%)

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.
- Animal Eyes, Michael Land and Dan-Eric Nilsson, 2012.

Ressources en bibliothèque

- Animal Eyes / Land
- An immense world / Yong (added)
- Vision Science / Palmer
- Ecological Approach to Visual Perception / Gibson
- Computer Vision / Szeliski

Notes/Handbook

The reference reading of different lectures will be from different books (the main ones listed above) and occasionally from papers. Resources will be provided in class. Full-text books are not mandatory.

Moodle Link

- https://go.epfl.ch/CS-503