SECTION DE SYSTEMES DE COMMUNICATION

DE L'ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE

Master en Data Science

LIVRET DES COURS

ANNEE ACADEMIQUE 2020/2021

TABLE DES MATIERES Page

Ordonnance sur le contrôle des études menant au bachelor et au master ………………………. 3

Plan d'études

- Cycle master ……………………………………………………………………… 15
- Mineur en Data Science 17

Règlement d'application du contrôle des études …………………………………………... 18

Descriptifs des enseignements de la section de Systèmes de Communication

- Cycle Master ………………………………………………………………………. 23 à 132

Index des cours (par ordre alphabétique………………………………………………………. 133
Index par enseignants-es (par ordre alphabétique)……………………………………………. 134

Ordonnance sur le contrôle des études menant au bachelor et au master à l'École polytechnique fédérale de Lausanne

(Ordonnance sur le contrôle des études à l'EPFL)

du 30 juin 2015 (Etat le 1er juin 2019)

La Direction de l'École polytechnique fédérale de Lausanne (Direction de l'EPFL),

vu l'art. 3, al. 1, let. b, de l'ordonnance du 13 novembre 2003 sur l'EPFZ et l'EPFL[1],

arrête:

Chapitre 1 Dispositions générales

Section 1 Objet et champ d'application

Art. 1 Objet

La présente ordonnance arrête les règles de base du contrôle des études à l'EPFL.

Art. 2 Champ d'application

1 La présente ordonnance s'applique à la formation menant au bachelor et au master de l'EPFL.

2 Dans la mesure où la direction de l'EPFL n'a pas édicté de règles particulières, les art. 8, 10, 12, 14, 15 et 18 à 20 s'appliquent également:

a. aux examens d'admission;
b. aux examens du cours de mathématiques spéciales (CMS);
c. aux examens du cours de mise à niveau;
d. aux examens de doctorat;
e. aux examens des programmes doctoraux;
f. aux examens de la formation continue et de la formation approfondie.

Section 2 Définitions générales

Art. 3 Branche

1 Une branche est une matière d'enseignement faisant l'objet d'une ou de plusieurs épreuves.

RO **2015** 2525

1 RS **414.110.37**

[2] Une branche dite de semestre est une branche dont les épreuves se déroulent pendant la période de cours.

[3] Une branche dite de session est une branche dont une épreuve se déroule en session d'examens. Elle peut comporter des épreuves se déroulant pendant la période de cours.

[4] Une branche de semestre peut consister en un stage.

Art. 4 Crédits et coefficients

À toute branche est associé un nombre de crédits ECTS (European Credit Transfer and Accumulation System) (crédits) ou, pour le cycle propédeutique, un coefficient, qui indiquent son poids dans la formation.

Section 3
Dispositions communes aux études de bachelor et de master

Art. 5 Plans d'études et règlements d'application

Des plans d'études et des règlements d'application sont édictés pour chaque cycle d'études de chaque domaine. Ils définissent en particulier:

a. les branches de semestre et les branches de session;
b. le semestre ou la session pendant lesquels ces branches peuvent être présentées;
c. la forme (écrite ou orale) de l'épreuve en session;
d. la composition des blocs et des groupes de branches;
e. les coefficients ou les crédits attribués à chaque branche;
f. le nombre de crédits ou le coefficient à acquérir dans chaque bloc et chaque groupe;
g. les conditions applicables aux prérequis (art. 25);
h. les conditions de réussite particulières;
i. les études d'approfondissement, de spécialisation ou interdisciplinaires;
j. les éventuels régimes transitoires applicables aux modifications des plans d'études et des règlements d'application.

Art. 6 Blocs et groupes de branches

[1] Les branches sont rassemblées en bloc ou en groupe. Chaque branche ne peut faire partie que d'un seul bloc ou d'un seul groupe. Un bloc peut être constitué d'une seule branche.

2 Un bloc est réputé réussi:

a. lorsque la somme des crédits acquis par branche est égale ou supérieure au nombre requis, ou

b. lorsque la somme des crédits acquis pour les branches présentées atteint le nombre requis et que la moyenne du bloc (art. 8, al. 5) est égale ou supérieure à 4,00; dans ce cas, la totalité des crédits des branches présentées est acquise.

3 Un groupe est réputé réussi lorsque les crédits des branches qui le composent ont été accumulés jusqu’au nombre requis; aucune compensation n’est possible entre les notes des branches du groupe.

Art. 7 Fiches de cours

Les fiches de cours publiées indiquent en particulier, pour chaque branche:

a. les objectifs de formation;

b. un bref descriptif de la matière;

c. les épreuves composant la note finale, avec leur pondération et leur forme;

d. les éventuels prérequis (art. 25);

e. la langue d’enseignement.

Art. 8 Notation

1 Une épreuve est notée de 1,00 à 6,00. Les notes en dessous de 4,00 sanctionnent des prestations insuffisantes. L’épreuve est notée 0 lorsque l’étudiant ne se présente pas, ne répond à aucune question ou ne respecte pas les délais.

2 La note finale de la branche se compose des notes de ses épreuves. Elle est arrêtée au quart de point. Lorsqu’elle est inférieure à 1,00, la branche est considérée comme non acquise et notée NA (non acquis). L’appréciation NA compte comme tentative de réussite.

3 Le règlement d’application peut prévoir qu’une branche est notée au moyen des appréciations R (réussi) ou E (échec).

4 Lorsque la branche est répétée, la note retenue est celle de la seconde tentative.

5 Les moyennes sont calculées en pondérant chaque note finale chiffrée de branche par son coefficient ou son nombre de crédits. Elles sont arrêtées au centième. Les appréciations NA et E empêchent l’obtention d’une moyenne, sauf dans les cas visés à l’art. 6, al. 2, let. b, et 3.

Art. 9 Organisation des sessions et des épreuves et inscriptions aux branches

1 Deux sessions d’examens sont organisées par année académique. Elles ont lieu entre les semestres.

2 Les délais d'inscription aux branches, les délais de retrait, les horaires et les dates des épreuves, ainsi que les autres modalités sont communiqués aux étudiants.

3 À l'échéance des délais, les inscriptions aux branches et les retraits sont définitifs.

4 Lorsque l'étudiant répète une branche, celle-ci est régie par les dispositions en vigueur au moment de la répétition, à moins que l'école n'en ait disposé autrement.

Art. 10 Incapacité

1 L'étudiant qui se prévaut d'un motif d'incapacité à se présenter à une épreuve doit l'annoncer à l'école dès la survenance de ce motif.

2 Il lui présente en outre les pièces justificatives au plus tard trois jours après la survenance du motif d'incapacité. Par pièces justificatives, on entend notamment un certificat médical ou une attestation d'une obligation légale de servir.

3 Invoquer un motif d'incapacité après s'être présenté à l'épreuve ne justifie pas l'annulation d'une note.

Art. 11 Langue des épreuves

1 Les épreuves se déroulent dans la langue de l'enseignement de la branche.

2 L'étudiant a le droit de répondre en français à une épreuve en anglais. Sur demande écrite de sa part, l'enseignant peut lui accorder de répondre en anglais si l'épreuve est en français.

Art. 12 Étudiants en situation de handicap

1 Si un candidat en situation de handicap en fait la demande au début de l'année académique, l'école fixe un déroulement d'épreuve adapté à son handicap et décide de l'utilisation de moyens auxiliaires ou de l'assistance personnelle nécessaires.

2 Les objectifs de l'épreuve doivent être garantis.

Art. 13 Tâches de l'enseignant

1 L'enseignant remplit notamment les tâches suivantes:

a. donner les informations nécessaires sur ses matières d'enseignement pour qu'elles soient publiées dans la fiche de cours;

b. informer les étudiants, s'il y a lieu, du contenu des matières et du déroulement des épreuves;

c. conduire les épreuves;

d. prendre des notes de chaque épreuve orale, qu'il peut être appelé à produire auprès de la conférence d'examen ou des autorités de recours;

e. attribuer les notes des épreuves, ainsi que la note finale de branche;

f. conserver pendant six mois après la fin du cycle concerné (chap. 2 à 4) les épreuves écrites et les notes prises durant les épreuves orales; en cas de recours, ce délai est prolongé jusqu'au terme de la procédure.

2 S'il est empêché de remplir ses tâches, le directeur de section désigne un remplaçant.

Art. 14 Observateur

1 Un observateur désigné par le directeur de section assiste à l'épreuve orale ayant lieu en session d'examens, dans le but de veiller à son déroulement régulier.

2 Il prend, pour chaque candidat, des notes sur le déroulement de l'épreuve et les conserve conformément à l'art. 13, al. 1, let. f.

Art. 15 Consultation des épreuves

L'étudiant peut consulter son épreuve dans les 6 mois qui suivent la communication du résultat.

Art. 16 Commissions d'évaluation

Des commissions d'évaluation peuvent être mises sur pied pour les branches de semestre. Outre l'enseignant et un expert, les commissions d'évaluation peuvent comprendre les assistants et les chargés de cours qui ont participé à l'enseignement, ainsi que d'autres professeurs.

Art. 17 Conférence d'examen

1 La conférence d'examen siège à l'issue de chaque session. Elle est composée du vice-provost pour la formation, qui la préside, du directeur de section et du chef du service académique. Les membres de la conférence d'examen peuvent se faire représenter par leur suppléant.

2 La conférence d'examen se prononce sur les cas particuliers conformément aux dispositions légales.

Art. 18 Fraude

1 Par fraude, on entend toute forme de tricherie en vue d'obtenir pour soi-même ou pour autrui une évaluation non méritée.

2 En cas de fraude, de participation à la fraude ou de tentative de fraude, le règlement disciplinaire du 15 décembre 2008 concernant les étudiants de l'École polytechnique fédérale de Lausanne[2] s'applique.

Art. 19 Notification des résultats et communications

1 La décision de réussite ou d'échec pour le cycle d'études est notifiée à l'étudiant.

2 RS **414.138.2**

2 Elle fait mention des notes obtenues et des crédits acquis.

3 La notification de la décision ainsi que les communications ont lieu par voie électronique ou postale.

Art. 20 Demande de nouvelle appréciation et recours administratif

1 La décision peut faire l'objet d'une demande de nouvelle appréciation auprès de l'école dans les 10 jours qui suivent sa notification. L'art. 63, al. 1, 3 et 4, de la loi fédérale du 20 décembre 1968 sur la procédure administrative[3] est applicable.

2 Elle peut également faire l'objet d'un recours administratif auprès de la commission de recours interne des EPF, dans les 30 jours qui suivent sa notification.

Chapitre 2 Examens du cycle propédeutique

Art. 21 Conditions de réussite

1 L'étudiant qui, à l'issue du premier semestre du cycle propédeutique et de la session d'examens afférente, a atteint une moyenne pondérée (art. 8, al. 5) d'au moins 3,50 pour le premier bloc au sens du règlement d'application est admis au second semestre du cycle.

2 À réussi le cycle propédeutique l'étudiant qui, conformément au plan d'études et au règlement d'application:

a. a présenté toutes les branches, et

b. a obtenu une moyenne égale ou supérieure à 4,00 dans chacun des blocs et, le cas échéant, les coefficients requis dans un groupe.

Art. 22 Échec et élimination

1 Constituent un échec, au niveau du cycle propédeutique:

a. la non-atteinte d'une moyenne pondérée d'au moins 3,50 pour le premier bloc, à l'issue du premier semestre et de la session d'examens afférente;

b. la non-atteinte d'une moyenne pondérée d'au moins 4,00 par bloc ou la non-atteinte du nombre de coefficients requis dans un groupe, à l'issue du cycle propédeutique, ou

c. le fait de ne pas avoir présenté toutes les branches du cycle propédeutique, sous réserve de l'art. 23, al. 4.

2 L'étudiant qui suit le cycle propédeutique en première tentative et se trouve dans la situation visée à l'al. 1, let. a, suit au second semestre le cours de mise à niveau de l'EPFL.

3 Est assimilé à un échec au cycle propédeutique de l'EPFL un échec ou une absence de réussite subi dans une autre haute école à un niveau comparable au cycle propé-

3 RS **172.021**

deutique, si la majorité des branches sont considérées par l'EPFL comme étant analogues.

4 Constitue un échec définitif un second échec au niveau du cycle propédeutique ou le non-respect de la durée maximale de deux ans pour réussir le cycle.

5 Constituent un motif d'exclusion définitive de toute formation de bachelor à l'EPFL la non-atteinte d'une moyenne pondérée d'au moins 4,00 à l'issue du cours de mise à niveau ou le non-respect de l'obligation de le suivre.

Art. 23 Répétition

1 L'étudiant qui est en situation d'échec, en première tentative, selon l'art. 22, al. 1, let. b et c, ou qui a atteint une moyenne d'au moins 4,00 au cours de mise à niveau est admis une seconde fois au premier semestre du cycle propédeutique de l'année académique qui suit.

1bis L'étudiant qui, après avoir réussi le cours de mise à niveau, échoue le cycle propédeutique à l'issue du second semestre, peut répéter le second semestre l'année suivante, en dérogation à l'art. 22, al. 4, de la présente ordonnance et à l'art. 7, al. 3, de l'ordonnance du 14 juin 2004 sur la formation à l'EPFL[4].[5]

2 Les branches d'un bloc ou d'un groupe réussis (art. 21, al. 2, let. b) sont acquises et ne peuvent pas être répétées.

3 La répétition des autres branches non réussies est impérative. La répétition des branches réussies est facultative, sauf pour les étudiants issus de la situation visée à l'art. 22, al. 1, let. a, pour lesquels elle est obligatoire. Le règlement d'application peut toutefois prévoir que certaines branches de semestre réussies ne peuvent pas être répétées.

4 En cas d'absence justifiée au sens de l'art. 10, l'école examine s'il est raisonnablement exigible de l'étudiant qu'il complète le cycle propédeutique à la session ordinaire correspondante de l'année suivante ou si l'étudiant doit être considéré comme ayant échoué.

Chapitre 3 Examens du cycle bachelor et du cycle master

Art. 24 Crédits

Les crédits de la branche sont attribués lorsque la note obtenue est égale ou supérieure à 4,00 ou que la moyenne du bloc de branches à laquelle elle appartient est égale ou supérieure à 4,00.

4 RS **414.132.3**
5 Introduit par le ch. I de l'O de la Direction de l'EPFL du 20 août 2019, en vigueur depuis le 1er juin 2019 (RO **2019** 2641).

Art. 25 Prérequis

Le règlement d'application ou la fiche de cours définit les branches dont l'étudiant doit avoir acquis les crédits afin d'être admis à suivre d'autres branches.

Art. 26 Conditions de réussite

1 Les crédits requis du cycle bachelor et du cycle master doivent être acquis conformément à la présente ordonnance, à l'ordonnance du 14 juin 2004 sur la formation à l'EPFL[6] et au règlement d'application.

2 Dans le cycle bachelor, 60 crédits au moins doivent être acquis par tranche de deux ans.

Art. 27 Répétition

1 Si, dans un bloc ou un groupe, le nombre de crédits requis n'est pas acquis, les branches dont la note est inférieure à 4,00 peuvent être répétées une fois, impérativement à la session ordinaire de l'année qui suit.

2 L'étudiant qui échoue deux fois à une branche optionnelle peut en présenter une nouvelle conformément au plan d'études.

Art. 28 Échec définitif

Si l'étudiant n'acquiert pas les crédits requis conformément à la présente ordonnance et au règlement d'application, dans le respect des durées maximales fixées par l'ordonnance du 14 juin 2004 sur la formation à l'EPFL[7], il se trouve en situation d'échec définitif.

Art. 29 Admission conditionnelle au cycle consécutif

1 Peut être admis conditionnellement au cycle master consécutif l'étudiant qui:

a. n'a pas plus de 10 crédits manquants sur ceux requis par le plan d'études de dernière année du cycle bachelor de l'EPFL, et

b. n'est pas en situation d'échec définitif.

2 L'étudiant admis conditionnellement au cycle master consécutif a l'obligation d'acquérir les crédits manquants du bachelor dans l'année de son admission conditionnelle, sous peine d'être exclu du cycle.

3 Peut être admis conditionnellement au projet de master l'étudiant qui:

a. n'a pas plus de 8 crédits manquants sur ceux requis pour le cycle master y compris les études visées à l'art. 5, let. i;

b. n'est pas en situation d'échec définitif.

6 RS **414.132.3**
7 RS **414.132.3**

Chapitre 4 Projet de master

Art. 30 Déroulement

1 Le sujet du projet de master est fixé ou approuvé par le professeur ou le maître d'enseignement et de recherche qui en assume la direction.

2 Sur demande, le directeur de section peut confier la direction du projet de master à un professeur ou un maître d'enseignement et de recherche rattaché à une autre section ou à un collaborateur scientifique.

3 L'examen du projet de master consiste en une évaluation de sa présentation finale suivie d'une interrogation orale devant l'enseignant qui a dirigé le projet et un expert externe à l'EPFL désigné par l'enseignant en accord avec le directeur de section. Seul l'enseignant peut inviter d'autres personnes à l'interrogation orale; celles-ci ne participent pas à l'évaluation.

4 Si la qualité rédactionnelle du projet est jugée insuffisante, l'enseignant peut exiger que l'étudiant y remédie dans un délai de deux semaines à compter de l'interrogation orale.

Art. 31 Conditions de réussite

1 Le projet de master est réputé réussi lorsque la note attribuée est égale ou supérieure à 4,00.

2 Si le règlement d'application prévoit un stage associé au projet de master, celui-ci doit avoir été réussi préalablement.

Art. 32 Répétition

1 En cas d'échec, un nouveau projet de master peut être présenté dans le respect de la durée maximale prévue par l'ordonnance du 14 juin 2004 sur la formation à l'EPFL[8].

2 Un second échec constitue un échec définitif.

Chapitre 5 Dispositions finales

Art. 33 Abrogation

L'ordonnance du 14 juin 2004 sur le contrôle des études à l'EPFL[9] est abrogée.

8 RS **414.132.3**
9 [RO **2004** 4323, **2006** 4125, **2008** 3721]

Art. 34 Disposition transitoire

Le chapitre 2 de l'ordonnance du 14 juin 2004 sur le contrôle des études à l'EPFL[10] demeure applicable jusqu'au 31 août 2017 aux étudiants répétant le cycle propédeutique durant l'année académique 2016–2017.

Art. 35 Entrée en vigueur

La présente ordonnance entre en vigueur le 1er septembre 2016.

10 RO **2004** 4323, **2006** 4125, **2008** 3721

Plan d'études
Master en Data Science

2 0 20 - 2 0 2 1

arrêté par la direction de l'EPFL le 2 juin 2020

Directeur de la section	**Prof. S. Vaudenay**
Adjointe de la section	**Mme S. Dal Mas**
Conseiller d'études :	
1ère année cycle Master	**Prof. B. Rimoldi**
2ème année cycle Master	**Prof. P. Thiran**
Coordination des stages en industrie	**Mme S. Dal Mas**
Secrétariat de la section	**Mme C. Dauphin**

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

2020-2021 **DATA SCIENCE** **Cycle Master**

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

Stage d'ingénieur :
Voir les modalités dans le règlement d'application

Mineurs :
Le cursus peut être complété par un des mineurs figurant dans l'offre de l'EPFL (renseignements à la page sac.epfl.ch/mineurs),
à l'exclusion des mineurs "Data Science", "Informatique", "Cyber security" et "Systèmes de communication" qui ne peuvent pas être choisis.
Parmi les mineurs offerts par l'EPFL, la section recommande à ses étudiants les mineurs suivants :
- Biocomputing (SIN)
- Computational Science and Engineering (SMA)
- Management de la technologie et entrepreneuriat (SMTE)
- Technologies biomédicales (SMT)
- Technologies spatiales (SEL)

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

2020-2021 DATA SCIENCE - Options — Cycle Master

Code	Matières	Enseignants sous réserve de modification	Sections	Semestres MA1			MA2			Crédits	Période des épreuves	Type examen	Cours biennaux donnés en
				c	e	p	c	e	p				
COM-501	Advanced cryptography	Vaudenay	SC				2	2		4	E	écrit	
COM-417	Advanced probability and applications	Lévêque	SC				3	2		6	E	écrit	
CS-523	Advanced topics on privacy enhancing technologies	Troncoso	IN				3	1	2	7	E	écrit	
MATH-493	Applied biostatistics	Goldstein	MA				2	2		5	sem P		
CS-456	Artificial neural networks	Gerstner	IN				2	2		5	E	écrit	
EE-592	Automatic speech processing	Bourlard	EL	2	1					3	H	écrit	
BIO-465	Biological modeling of neural networks	Gerstner	IN				2	2		4	E	écrit	
MATH-453	Computational linear algebra	Kressner	MA				2	2		5	E	oral	
CS-524	Computational complexity (pas donné en 2020-2021)	Svensson	IN	3	1					4	sem A		2021-2022
CS-413	Computational Photography	Süsstrunk	SC				2		2	5	sem P		
CS-442	Computer vision	Fua	IN				2	1		4	E	écrit	
CS-453	Concurrent algorithms	Guerraoui	SC	3	1	1				5	H	écrit	
COM-401	Cryptography and security	Vaudenay	SC	4	2					7	H	écrit	
COM-480	Data visualization	Vuillon	SC				2		2	4	sem P		
EE-559	Deep learning	Fleuret	EL				2	2		4	E	écrit	
CS-411	Digital education & learning analytics	Dillenbourg/Jermann	IN	2		2				4	H	oral	
CS-451	Distributed algorithms	Guerraoui	SC	3	2	1				6	H	écrit	
CS-423	Distributed information systems	Aberer	SC	2	1					4	H	écrit	
ENG-466	Distributed intelligent systems	Martinoli	SIE				2	3		5	E	écrit	
CS-550	Formal verification	Kuncak	IN	2	2	2				sem A			
MATH-360	Graph Theory	Maffucci	MA	2	2					5	H	écrit	
EE-451	Image analysis and pattern recognition	Thiran J.-P.	EL				2		2	4	sem P		
COM-404	Information theory and coding	Telatar	SC	4	2					7	H	écrit	
CS-430	Intelligent agents	Faltings	IN	3	3					6	H	écrit	
CS-486	Interaction design	Pu	IN				2	1	1	4	sem P		
CS-431	Introduction to natural language processing	Chappelier/Rajman	IN	2	2					4	H	écrit	
COM-490	Lab in data science	Bouillet/Sarni/Verscheure/Delg	SC						4	4	sem P	sans retrait	
CS-526	Learning theory	Macris/Urbanke	SC				2	2		4	E	écrit	
MATH-341	Linear models	Panaretos	MA	2	2					5	H	écrit	
CS-421	Machine learning for behavioral data	Käser	IN				2		2	4	E	écrit	
COM-516	Markov chains and algorithmic applications	Lévêque/Macris	SC	2	2					4	H	écrit	
COM-514	Mathematical foundations of signal processing	Simeoni/Bejar Haro	SC	3	2					6	H	écrit	
EE-556	Mathematics of data: from theory to computation	Cevher	EL	2	2					4	H	écrit	
COM-512	Networks out of control (pas donné en 2020-2021)	Thiran P./Grossglauser	SC				2	1		4	E	écrit	2021-2022
COM-508	Optional project in Data Science	Divers enseignants	SC	◄			2		►	8	sem A ou P		
COM-503	Performance evaluation	Le Boudec	SC				3	1	2	7	E	écrit	2020-2021
MATH-447	Risk, rare events and extremes	Davison	MA	2	2					5	H	écrit	2020-2021
MATH-441	Robust and nonparametric statistics (pas donné en 2020-2021)	Morgenthaler	MA				2	2		5	E	oral	2021-2022
CS-412	Software security	Payer	IN				3	2	1	6	sem P		
MATH-486	Statistical mechanics and Gibbs measures	Friedli	MA				2	2		5	E	oral	
MATH-442	Statistical Theory	Koch	MA	2	2					5	H	écrit	
COM-506	Student seminar: security protocols and applications	Oechslin/Vaudenay	SC				2			3	E	écrit	
CS-448	Sublinear algorithms for big data analysis (pas donné en 2020-2021)	Kapralov	IN				3			4	sem P		2021-2022
CS-410	Technology ventures in IC (pas donné en 2020-2021)	Bugnion	IN				2		2	4	sem P		
MATH-342	Time Series	Olhede	MA				2	2		5	E	écrit	
CS-455	Topics in theoretical computer science	Kapralov	IN	3	1					4	sem A		2020-2021
CS-444	Virtual reality	Boulic	IN				2	1		4	sem P		

2020-2021 **Data Science** **Section de Systèmes de communication**

Mineur disciplinaire **Responsable : Sylviane Dal Mas**

Les enseignants, les crédits et la période des cours sont indiqués sous réserve de modification. **136** **crédits offerts**

Les cours déjà suivis au bachelor ou au master ne peuvent pas être pris également dans un mineur.

Codes	Matières (liste indicative)	Enseignants	Livret des cours	Crédits	Période des cours	
CS-450	Advanced algorithms	Kapralov	IN	7		P
COM-501	Advanced cryptography	Vaudenay	SC	4		P
COM-417	Advanced probability and applications	Lévêque	SC	6		P
CS-250	Algorithms	Svensson	IN	6	A	
MATH-474	Applied biostatistics	Goldstein	MA	5		P
CS-401	Applied data analysis	West	IN	6	A	
MATH-435	Bayesian computation (pas donné en 2020-21)	Dehaene	MA	5		P
CS-442	Computer vision	Fua	IN	4		P
COM-480	Data visualization	Vuillon	SC	4		P
EE-559	Deep learning	Fleuret	EL	4		P
CS-210	Functional programming	Kuncak/Odersky	IN	5	A	
COM-402	Information security and privacy	Hubaux/Oechslin	SC	6	A	
COM-406	Foundations of Data Science	Urbanke	SC	6	A	
CS-430	Intelligent agents	Faltings	IN	6	A	
CS-322	Introduction to database systems	Ailamaki/Koch	IN	4		P
CS-431	Introduction to natural language processing	Chappelier/Rajman	IN	4	A	
CS-433	Machine learning	Jaggi/Flammarion	IN	7	A	
COM-300	Modèles stochastiques pour les communications	Thiran	SC	6	A	
COM-512	Networks out of control (pas donné en 2020-21)	Grossglauser/Thiran	SC	4		P
CS-439	Optimization for machine learning	Jaggi/Flammarion	IN	5		P
COM-508	Optional project in data science*	divers	SC	8	A ou P	
COM-503	Performance evaluation	Le Boudec	SC	7		P
MATH-447	Risk, rare events and extremes	Davison	MA	5	A	
MATH-413	Statistics for Data Science	Olhede	MA	6	A	
CS-449	Systems for Data Science	Kermarrec	IN	6		P

* Inscription sur dossier; seulement pour étudiants en 2ème année de Master; superviser par un professeur en IC

Légende :

A = automne, P = printemps

1 semestre comprend 14 semaines.

RÈGLEMENT D'APPLICATION DU CONTRÔLE DES ÉTUDES DE LA SECTION DE SYSTÈMES DE COMMUNICATION POUR LE MASTER EN DATA SCIENCE pour l'année académique 2020-2021
du 2 juin 2020

La direction de l'École polytechnique fédérale de Lausanne

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

arrête:

Article premier - Champ d'application

Le présent règlement fixe les règles d'application du contrôle des études de master de la section de systèmes de communication pour le master en Data Science qui se rapportent à l'année académique 2020-2021.

Art. 2 – Étapes de formation

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

Art 3 – Sessions d'examen

1. Les branches d'examen sont examinées par écrit ou par oral pendant les sessions d'hiver ou d'été. Elles sont mentionnées dans le plan d'études avec la mention H ou E.

2. Les branches de semestre sont examinées pendant le semestre d'automne ou le semestre de printemps. Elles sont mentionnées dans le plan d'études avec la mention sem A ou sem P.

3. Une branche annuelle, c'est à dire dont l'intitulé tient sur une seule ligne dans le plan d'étude, est examinée globalement pendant la session d'été (E).

4 Pour les branches de session, la forme écrite ou orale de l'examen indiquée pour la session peut être complétée par des contrôles de connaissances écrits ou oraux durant le semestre, selon indications de l'enseignant.

Art. 3 – Prérequis

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

Art. 4 – Conditions d'admission

1. Les étudiants issus du Bachelor en Informatique ou en Systèmes de communications sont admis automatiquement.

2. Pour les autres étudiants, l'admission s'effectue sur dossier.

Art. 5 - Organisation

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

2. Le bloc « Projets et SHS » est composé d'un projet de 12 crédits et de l'enseignement SHS.

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

4. Le groupe 2 « Options » est composé
   - des cours de la liste du groupe 2 « options » du plan d'études dans la rubrique « Master » ;
   - des crédits surnuméraires obtenus dans le groupe 1 « Core courses » ;
   - d'un projet optionnel de 8 crédits ;
   - de cours hors plan d'études suivant l'alinéa 6.

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

6. Des cours, comptant pour un maximum de 15 crédits au total, peuvent être choisis en dehors de la liste des cours du plan d'études dans la rubrique « Master ». Le choix de ces cours doit être accepté préalablement par le directeur de la section qui peut augmenter le maximum de 15 crédits si la demande est justifiée.

Art. 6 - Examen du cycle master

1. Le bloc « Projets et SHS » est réussi lorsque **18 crédits** sont obtenus.

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

3. Le groupe 1 « Core courses » est réussi lorsqu'**au moins 30 crédits** sont obtenus.

Art. 7 - Enseignement SHS

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

Art. 8 – Mineurs

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

2. Le choix des cours qui composent un mineur se fait avec la section de systèmes de communication et avec le responsable du mineur. Les mineurs « Data Science » « Informatique », « Cyber Security » et « Systèmes de Communication » ne peuvent pas être choisis.

3. L'étudiant annonce le choix d'un mineur à sa section au plus tard à la fin du premier semestre des études de master.

4. Un mineur est réussi quand 30 crédits au minimum sont obtenus parmi les branches avalisées.

Art. 8 – Stage d'ingénieur

1. Les étudiants commençant leur cycle master doivent effectuer un stage d'ingénieur durant leur master :
- soit un stage d'été de minimum 8 semaines
- soit un stage de minimum 6 mois en entreprise (en statut stage durant un semestre). Durant la période du COVID-19, la durée du stage peut être adaptée.
- soit un Projet de Master de 25 semaines en entreprise (valide le stage et le Projet de Master)

2. Le stage peut être effectué dès le 2ème semestre du cycle master, mais avant le projet de master.

3 L'étudiant ne peut pas faire de cours/projet en parallèle à son stage.

4. Le responsable du stage de la section évalue le stage, par l'appréciation « réussi » ou « non réussi ». Sa réussite est une condition pour l'admission au projet de master. En cas de non réussite, il peut être répété une fois, en règle générale dans une autre entreprise.

5. Il est validé avec les 30 crédits du projet de master.

6. Les modalités d'organisation et les critères de validation du stage font l'objet d'une directive interne à la section.

Art. 9 – Spécialisation Enseignement

1. Les étudiants en Master Data Science ont la possibilité de suivre une spécialisation en informatique pour l'enseignement.

2. L'étudiant admis à cette spécialisation ne peut pas suivre de mineur. Le plan d'études est modifié comme suit : (i) Un nouveau groupe de 30 ECTS de cours à la HEP Vaud est rajouté et le nombre de ECTS du Cycle Master passe de 60 à 30 ECTS ; (ii) les cours SHS sont remplacés par un cours à la HEP Vaud ; (iii) le Projet de Master peut s'étaler sur deux semestres et commencer après que l'étudiant a complété le bloc « Projets et SHS » et le groupe « Core courses » ; (iv) la durée maximale des études ne peut pas dépasser 8 semestres.

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

Art. 10 – Procédure d'admission

1. L'admission à cette spécialisation n'est pas automatique. Pour être admis à la spécialisation, le candidat doit être inscrit au Master en Data Science de l'EPFL et répondre aux conditions pour l'admission au Diplôme d'enseignement pour le degré secondaire II fixées par le Règlement d'application de la loi sur la HEP du 3 juin 2009 (RLHEP).

2. L'étudiant s'inscrit auprès de la HEP Vaud selon les conditions et délais de la candidature en ligne et transmet les pièces requises par le RLHEP ainsi qu'une attestation d'immatriculation à l'EPFL.

Au nom de la direction de l'EPFL

Le président, M. Vetterli
Le vice-président pour l'éducation, P. Vandergheynst

Lausanne, le 2 juin 2020

DATA SCIENCE

Cycle

Master

2020 / 2021

CS-450 **Advanced algorithms**

Kapralov Mikhail

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

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

Summary

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

Content

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

Keywords

See content.

Learning Prerequisites

Required courses

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

Recommended courses

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

Important concepts to start the course

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

Learning Outcomes

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

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

Teaching methods

Ex cathedra lecture, reading

Assessment methods

Supervision

Others	For details, see the course web page.

Resources

Bibliography

See web page for the course.

Ressources en bibliothèque

- Randomized Algorithms / Motwani
- Approximation Algorithms / Vazirani
- Computational Complexity / Papadimitrou
- Algebraic Complexity Theory / Buergisser
- Quantum Computation and Quantum Information / Nielsen

Notes/Handbook

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

Advanced cryptography

COM-501

Vaudenay Serge

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

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

Summary

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

Content

1. **Cryptographic security models:** security notions for encryption and authentication, game reduction techniques
2.
Public-key cryptography: Factoring, RSA problem, discrete logarithm problem, attacks based on subgroups
3. **Interactive proofs:** NP-completeness, interactive systems, zero-knowledge
4. **Conventional cryptography:** differential and linear cryptanalysis, hypothesis testing, decorrelation
5. **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

Others	Lecturers and assistants are available upon appointment.

Resources

Bibliography

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

Ressources en bibliothèque

- Algorithmic cryptanalysis / Joux
- A computational introduction to number theory and algebra / Shoup
- Communication security / Vaudenay

COM-417

Advanced probability and applications

Lévêque Olivier

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

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

Summary

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

Content

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

Keywords

probability theory, measure theory, martingales, convergence theorems

Learning Prerequisites

Required courses

Basic probability course
Calculus courses

Recommended courses

complex analysis

Important concepts to start the course

This course is NOT an introductory course on probability: the students should have a good understanding and practice of basic probability concepts such as: distribution, expectation, variance, independence, conditional probability.
The students should also be at ease with calculus. Complex 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 lectures + exercise sessions

Expected student activities

active participation to exercise sessions

Assessment methods

Midterm 20%, graded homeworks 20%, exam 60%

Resources

Bibliography

Sheldon M. Ross, Erol A. Pekoz, A Second Course in Probability,1st edition, www.ProbabilityBookstore.com, 2007.
Jeffrey S. Rosenthal, A First Look at Rigorous Probability Theory,2nd edition, World Scientific, 2006.
Geoffrey R. Grimmett, David R. Stirzaker, Probability and Random Processes,3rd edition, Oxford University Press, 2001.
Richard Durrett, Probability: Theory and Examples, 4th edition, Cambridge University Press, 2010.
Patrick Billingsley, Probability and Measure, 3rd edition, Wiley, 1995.

Ressources en bibliothèque

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

Notes/Handbook

available on the course website

Websites

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

Prerequisite for

Advanced classes requiring a good knowledge of probability

CS-523 **Advanced topics on privacy enhancing technologies**

Troncoso Carmela

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

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

Summary

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

Content

The course will cover the following topics:

- Privacy definitions and concepts.
- Privacy-preserving cryptographic solutions: anonymous credentials, zero-knowledge proofs, secure multi-party computation, homomorphic encryption, Private information retrieval (PIR), Oblivious RAM (ORAM)
- Anonymization and data hiding: generalization, differential privacy, etc
- Machine learning and privacy
- Protection of metadata: anonymous communications systems, location privacy, 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-402 Information Security and Privacy
COM-301 Computer Security

Recommended courses
COM-401 Cryptography

Important concepts to start the course
Basic programming skills; basics of probabilities and statistics; basics of cryptography

Learning Outcomes

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

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

Teaching methods

Lectures and written exercises to deepen understanding of concepts

Programming-oriented assignments to practice use of privacy technologies

Expected student activities

Participation in the lectures. Active participation is encouraged.
Participation in exercise session and complete the exercises regularly
Completion of programming assignments

Assessment methods

Final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

MATH-493

Applied biostatistics

Goldstein Darlene

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

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

Summary

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

Content

- Types of studies
- Design and analysis of studies
- R statistical software
- Reproducible research techniques and tools
- Report writing
- Exploratory data analysis
- 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

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 proactical showing how to implement the methods using R. In each practical, students use R to carry out analyses of the relevant data type for that week.

Expected student activities

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

Assessment methods

Evaluation is based on written reports of projects analyzing biostatistical data.
Dans le cas de l’art. 3 al. 5 du Règlement de section, l’enseignant décide de la forme de l’examen qu’il communique aux étudiants concernés.

EPFL

CS-401 **Applied data analysis**

West Robert

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

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

Summary

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

Content

Thanks to a new breed of software tools that allows 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
- Schema alignment, data reconciliation
- Data quality testing with crowdsourcing

Data Interpretation

- Stats in practice (distribution fitting, statistical significance, etc.)
- Working with "found data" (design of observational studies, regression analysis)
- Machine learning in practice (supervised and unsupervised, feature engineering, more data vs. advanced algorithms, curse of dimensionality, etc.)

• Text mining: vector space model, topic models, word embedding
• Social network analysis (influencers, community detection, etc.)

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

Reporting
• Results reporting, infographics
• How to publish reproducible results
• Anonymiziation, ethical concerns

The students will learn the techniques during the ex-cathedra lectures and will be introduced, in the lab sessions, to the software tools required to complete the homework assignments and the in-class quizzes.
In parallel, the students will embark on a semester-long project, split in agile teams of 3-4 students. The project consists of two parts: (1) replication of a data analysis pipeline from a published scientific paper, (2) a "free-style" component where students propose and execute their own extension of part 1. The outcome of this team effort will be a project portfolio that will be made public (and available as open source).
At the end of the semester, students will also take a 3-hour final exam in a classroom with their own computer, where they will be asked to complete a data analysis pipeline (both with code and extensive comments) on a dataset they have never worked with before.

Keywords

data science, data analysis, data mining, machine learning

Learning Prerequisites

Required courses

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

Recommended courses

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

Important concepts to start the course

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

Learning Outcomes

By the end of the course, the student must be able to:
• Construct a coherent understanding of the techniques and software tools required to perform the fundamental steps of the Data Science pipeline
• Perform data acquisition (data formats, dataset fusion, Web scrapers, REST APIs, open data, big data platforms, etc.)
• Perform data wrangling (fixing missing and incorrect data, data reconciliation, data quality assessments, etc.)

- Perform data interpretation (statistics, knowledge extraction, critical thinking, team discussions, ad-hoc visualizations, etc.)
- Perform result dissemination (reporting, visualizations, publishing reproducible results, ethical concerns, etc.)

Transversal skills

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

Teaching methods

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

Expected student activities

Students are expected to:

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

Assessment methods

- 33% continuous assessment during the semester (homework and in-class quizzes)
- 33% final exam, data analysis task on a computer (3 hours)
- 33% final project, done in groups of 3

Supervision

Others http://ada.epfl.ch

CS-456 Artificial neural networks

Gerstner Wulfram

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

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

Summary

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

Content

- *Simple perceptrons for classification*

- *Reinforcement Learning 1: Bellman equation and SARSA*
- *Reinforcement Learning 2: variants of SARSA, Q-learning, n-step-TD learning*
- *Reinforcement Learning 3: Policy gradient*

- *Deep Networks 1: BackProp and Multilayer Perceptrons*
- *Deep Networks 2: Regularization and Tricks of the Trade in deep learning*
- *Deep Networks 3. Error landscape and optimization methods for deep networks*
- *Deep Networks 4. Statistical Classification by deep networks*
- *Application 1: Convolutional networks*
- *Application 2: Sequence prediction and recurrent networks*
- *Deep reinforcement learning: Actor-Critic networks*
- *Deep reinforcement learning: applications*

Keywords

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

Learning Prerequisites

Required courses

CS 433 Machine Learning (or equivalent)
Calculus, Linear Algebra (at the level equivalent to first 2 years of EPFL in STI or IC, such as Computer Science, Physics or Electrical Engineering)

Recommended courses

stochastic processes
optimization

Important concepts to start the course

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

Learning Outcomes

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

Transversal skills

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

Teaching methods

ex cathedra lectures and 2 miniprojects. Every week the ex cathedra lectures are interrupted for a short in-class exercise which is then discussed in classroom before the lecture continues. Additional exercises are given as homework. For the exercise sessions two time slots of 45 minutes will be offered, and students will sign up for one of the two.

Expected student activities

work on miniproject
solve all exercises
attend all lectures and take notes during lecture, participate in quizzes.
If you cannot attend a lecture, then you must read the recommended book chapters

Assessment methods

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

Supervision

Office hours	Yes

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

Resources

Bibliography

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

Pdfs of the preprint version for both books are availble online

Ressources en bibliothèque

- Reinforcement Learning by Sutton and Barto
- Deep Learning by Goodfellow, Bengio, Courville

EE-554 **Automatic speech processing**

Bourlard Hervé

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

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

Summary

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

Content

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

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

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

4. Speech Synthesis: Morpho-syntactic analysis, phonetic transcription, prosody, speech synthesis models.

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

6. Speaker recognition and speaker verification: Formalism, hypothesis testing, HMM based speaker verification.

7. Linguistic Engineering: state-of-the-art and typical applications

Keywords

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

Learning Prerequisites

Required courses

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

Important concepts to start the course

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

Learning Outcomes

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

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

Transversal skills

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

Teaching methods

Lecture + lab exercises

Expected student activities

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

Assessment methods

Written exam without notes

Resources

Ressources en bibliothèque

• Traitement de la parole / Boite

BIO-465 **Biological modeling of neural networks**

Gerstner Wulfram

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

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

Summary

In this course we study mathematical models of neurons and neuronal networks in the context of biology and establish links to models of cognition.

Content

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

Keywords

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

Learning Prerequisites

Required courses

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

Recommended courses

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

Important concepts to start the course

Differential equations, stochastic processes,

Learning Outcomes

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

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

Transversal skills

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

Teaching methods

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

Expected student activities

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

Assessment methods

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

Supervision

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

Resources

Bibliography

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

Ressources en bibliothèque

- Neuronal Dynamics / Gerstner

Notes/Handbook

The textbook is online at: http://neuronaldynamics.epfl.ch/

Videos

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

MATH-453 **Computational linear algebra**

Kressner Daniel

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

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

Summary

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

Content

Introduction
Sources of large-scale linear algebra problems. Recap of required linear algebra concepts.
Eigenvalue problems
Krylov subspace methods. Singular value problems. Preconditioned iterative methods.
Linear systems
Direct sparse factorizations. Krylov subspace methods and preconditioners.
Matrix functions
Theory and algorithms.

Keywords

linear systems, eigenvalue problems, matrix functions

Learning Prerequisites

Required courses

Linear Algebra, Numerical Analysis

Learning Outcomes

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

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

Teaching methods

Ex cathedra lecture, exercises in the classroom and with computer

Expected student activities

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

Assessment methods

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

Resources

Bibliography

Lecture notes will be provided by the instructor. Complimentary reading:
H. Elman, D. J. Silvester, and A. J. Wathen. Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics. Oxford University Press, 2005.
G. H. Golub and C. Van Loan. Matrix computations. Johns Hopkins University Press, 1996.
Y. Saad. Iterative methods for sparse linear systems. Second edition. SIAM, 2003.

Ressources en bibliothèque

- Finite elements and fast iterative solvers / Elman
- Iterative methods for sparse linear systems / Saad
- Matrix computations / Golub

CS-413 **Computational photography**

Süsstrunk Sabine

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

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

Summary

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

Content

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

Basics: Human vision system, Light and illumination, Geometric optics, Color science, Sensors, Digital camera systems.
Generalized illumination: Structured light, High dynamic range (HDR) imaging, Time-of-flight.
Generalized optics: Coded Image Sensing, Coded aperture, Focal stacks.
Generalized sensing: Low light imaging, Depth imaging, Plenoptic imaging, Light field cameras.
Generalized processing: Super-resolution, In-painting, Compositing, Photomontages, Panoramas, HDR imaging, Multi-wavelength imaging, Dynamic imaging.
Generalized display: Stereoscopic displays, HDR displays, 3D displays, Mobile displays.

Keywords

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

Learning Prerequisites

Required courses

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

Recommended courses

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

Important concepts to start the course

- Basic signal processing.
- Basic computer vision.
- Basic programming (iOS, Android, Matlab).

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.

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.

CS-442

Computer vision

Fua Pascal

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

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

Summary

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

Content

Introduction

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

Extracting 2D Features

- Contours
- Texture
- Regions

3D Shape Recovery

- From one single image
- From multiple images

Learning Outcomes

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

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

Teaching methods

Ex cathedra lectures and programming exercises using matlab.

Assessment methods

With continuous control

Resources

Bibliography

- R. Szeliski, Computer Vision: 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

CS-453

Concurrent algorithms

Guerraoui Rachid

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

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

Summary

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

Content

Model of a parallel system
A multicore architect
Processes and objects
Safety and liveliness

Parallel programming
Automatic parallelism
Mutual exclusion and locks
Non-blocking data structures

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

Hierarchy of objects
The FLP impossibility
The consensus number
Universal constructions
Transactional memories
Transactional algorithms
Opacity and obstruction-freedom

Keywords

Concurrency, parallelism, algorithms, data structures

Learning Prerequisites

Required courses

ICC, Operatings systems

Recommended courses

This course is complementary to the Distributed Algorithms course.

Important concepts to start the course

Processes, threads, 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

Midterm and final exam
Project

Assessment methods

With continuous control, midterm final exams and project

Supervision

Office hours	Yes
Assistants	Yes
Forum	No

Resources

Notes/Handbook

Concurrent Algorithms, R. Guerraoui and P. Kouznetsov

Websites

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

COM-401 **Cryptography and security**

Vaudenay Serge

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

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

Summary

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

Content

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

Keywords

cryptography, encryption, secure communication

Learning Prerequisites

Required courses

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

Recommended courses

- Computer security (COM-301)

Important concepts to start the course

- Mathematical reasoning

- Probabilities
- Algebra, arithmetics
- Algorithmics

Learning Outcomes

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

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

Teaching methods

ex-cathedra

Expected student activities

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

Assessment methods

Mandatory continuous evaluation:

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

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

Supervision

Others	Lecturers and assistants are available upon appointment.

Resources

Bibliography

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

Ressources en bibliothèque

- A computational introduction to number theory and algebra / Shoup
- Communication security / Vaudenay

Prerequisite for

- Advanced cryptography (COM-401)
- Algorithms in public-key cryptography (COM-408)

EPFL

COM-480 Data visualization

Vuillon Laurent Gilles Marie

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

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

Summary

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

Content

Tentative course schedule
Week 1: Introduction to Data visualization Web development
Week 2: Javascript
Week 3: More Javascript
Week 4: Data Data driven documents (D3.js)
Week 5: Interaction, filtering, aggregation (UI /UX). Advanced D3 / javascript libs
Week 6: Perception, cognition, color Marks and channels
Week 7: Designing visualizations (UI/UX) Project introduction Dos and don’ts for data-viz
Week 8: Maps (theory) Maps (practice)
Week 9: Text visualization
Week 10: Graphs
Week 11: Tabular data viz Music viz
Week 12: Introduction to scientific visualisation
Week 13: Storytelling with data / data journalism Creative coding
Week 14: Wrap-Up

Keywords

Data viz, visualization, data science

Learning Prerequisites

Required courses

CS-305 Software engineering (BA)
CS-250 Algorithms (BA)
CS-401 Applied data analysis (MA)

Recommended courses

EE-558 A Network Tour of Data Science (MA)
CS-486 Human computer interaction (MA)
CS-210 Functional programming (BA)

Important concepts to start the course

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

Supervision

Office hours	No
Assistants	No
Forum	No

Resources

Bibliography

Visualization Analysis and Design by Tamara Munzner, CRC Press (2014). Fee online version at EPFL.
Interactive Data Visualization for the Web by Scott Murray O'Reilly (2013) - D3 - Free online version.

Ressources en bibliothèque

- Visualization Analysis and Design / Munzner
- Interactive Data Visualization for the Web / Murray

Notes/Handbook

Lecture notes

Websites

- https://www.kirellbenzi.com

Moodle Link

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

EPFL

EE-559

Deep learning

Fleuret François

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

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	**342**

Summary

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

Content

The course aims at teaching the required skills to use deep learning methods on applied problems.
It will show how to design and train a deep neural network for a given task, and the sufficient theoretical basis to go beyond the topics directly seen in the course.
The planned content of the course:

- What is deep learning, introduction to tensors.

- Basic machine-learning, empirical risk minimization, simple embeddings.

- Linear separability, multi-layer perceptrons, back-prop.

- Generalized networks, autograd, batch processing, convolutional networks.

- Initialization, optimization, and regularization. Drop-out, activation normalization, skip connections.

- Deep models for Computer Vision.

- Analysis of deep models.

- Auto-encoders, embeddings, and generative models.

- Recurrent models and Natural Language Processing.
- pytorch tensors, deep learning modules, and internals.

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

Keywords

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

Learning Prerequisites

Required courses

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

Recommended courses

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

Teaching methods

Ex-cathedra with exercise sessions and mini-projects. Invited speakers from the industry will present how deep learning is used in practice for their applications.

Assessment methods

Two mini-projects by groups of three students, and one final written exam.

Resources

Notes/Handbook

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

CS-411

Digital education & learning analytics

Dillenbourg Pierre, Jermann Patrick

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

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

Summary

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

Content

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

Keywords

learning, pedagogy, teaching, online education, MOOCs

Learning Prerequisites

Recommended courses

One of these courses is recommended:
- Machine Learning (Jaggi / Urbanke)
- Applied Data Analysis (West)

Learning Outcomes

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

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

Transversal skills

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

Teaching methods

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

Expected student activities

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

Assessment methods

- Project + exam
- 50 / 50

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Moodle Link

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

EPFL

CS-451

Distributed algorithms

Guerraoui Rachid

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

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

Summary

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

Content

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

Keywords

Distributed algorithms, checkpointing, replication, consensus, atomic broadcast, 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

Midterm and final exams
Project

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Ressources en bibliothèque

- Introduction to reliable and secure distributed programming / Cachin

Notes/Handbook

Reliable and Secure Distributed Programming
Springer Verlag
C. Cachin, R. Guerraoui, L. Rodrigues

Websites

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

CS-423

Distributed information systems

Aberer Karl

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

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

Summary

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

Content

Information Retrieval

1. Information Retrieval - Introduction
2. Text-Based Information Retrieval
3. Vector Space Retrieval
4. Inverted Files
5. Distributed Retrieval
6. Probabilistic Information Retrieval
7. Query Expansion
8. Latent Semantic Indexing
9. Word Embeddings
10. Link-Based Ranking

Data Mining

1. Data Mining – Introduction
2. Association Rule Mining
3. Clustering
4. Classification
5. Classification Methodology
6. Document Classification
7. Recommender Systems
8. Mining Social Graphs

Knowledge Bases

1. Semi-structured data
2. Semantic Web
3. RDF Resource Description Framework
4. Semantic Web Resources
5. Keyphrase extraction
6. Named entity recognition
7. Information extraction
8. Taxonomy Induction
9. Entity Disambiguation
10. Label Propagation

11. Link Prediction
12. Data Integration

Learning Prerequisites

Recommended courses

Introduction to Database Systems

Learning Outcomes

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

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

Teaching methods

Ex cathedra + programming exercises (Python)

Assessment methods

25% Continuous evaluations with bonus system during the semester
75% Final written exam (180 min) during exam session

ENG-466

Distributed intelligent systems

Martinoli Alcherio

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

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

Summary

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

Content

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

Keywords

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

Learning Prerequisites

Required courses

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

Recommended courses

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

Learning Outcomes

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

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

Transversal skills

- Demonstrate a capacity for creativity.
- Access and evaluate appropriate sources of information.
- Collect data.
- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Make an oral presentation.
- 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 a course project involving teamwork

Expected student activities

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

Assessment methods

Continuous control (40%) with final written exam (60%).

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

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

Websites

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

Moodle Link

• https://moodle.epfl.ch/course/view.php?id=15472

Prerequisite for

R&D activities in engineering

CS-550 **Formal verification**

Kuncak Viktor

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

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

Summary

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

Content

Topics may include among the 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
- Modeling Hardware: Verilog to Sequential Circuits
- Linear Temporal Logic. System Verilog Assertions. Monitors
- SAT Solvers and Bounded Model Checking
- Model Checking using Binary Decision Diagrams
- Loop Invariants. Hoare Logic. Statically Checked Function Contracts. Relational Semantics and Fixed-Point Semantics
- Symbolic Execution. Satisfiability Modulo Theories
- Abstract Interpretation and Predicate Abstraction
- Information Flow and Taint Analysis
- Verification of Security Protocols
- Dependent and Refinement Types

Learning Prerequisites

Recommended courses

Computer Language Processing / Compilers

Important concepts to start the course

Learning Outcomes

By the end of the course, the student must be able to:
- Formalize specifications
- Synthesize loop invariants
- Specify software functionality
- Generalize inductive hypothesis
- Critique current software development practices

Teaching methods

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

Expected student activities

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

Assessment methods

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

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

- 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
- Aaron Bradley and Zohar Manna: The Calculus of Computation - Decision Procedures with Applications to Verification, Springer 2007.
- Nielson, Flemming, Nielson, Hanne R., Hankin, Chris: Principles of Program Analysis. ISBN 978-3-662-03811-6. Springer 1999.
- Peter B. Andrews: An Introduction to Mathematical Logic and Type Theory (To Truth Through Proof), Springer 2002.
- http://logitext.mit.edu/tutorial

Ressources en bibliothèque

- Handbook of model checking
- Introduction to mathematical logic and type theory
- Handbook of Model Checking

- Tobias Nipkow, Gerwin Klein: Concrete Semantics with Isabelle/HOL
- Michael Huth and Mark Rayan: Logic in Computer Science - Modelling and Reasoning about Systems
- Peter B. Andrews: An Introduction to Mathematical Logic and Type Theory
- Nielson, Flemming, Nielson, Hanne R., Hankin, Chris: Principles of Program Analysis
- Aaron Bradley and Zohar Manna: The Calculus of Computation - Decision Procedures with Applications to Verification

Websites

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

Moodle Link

- https://moodle.epfl.ch/course/view.php?id=13051

Videos

- https://youtu.be/mm6CCGSDmOw?t=39
- https://www.youtube.com/watch?v=oLS_y842fMc
- https://www.youtube.com/channel/UCP2eLEql4tROYmIYm5mA27A

COM-406

Foundations of Data Science

Urbanke Rüdiger

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

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

Summary

Information Theory and Signal Processing are key underpinnings of Data Science. They provide frameworks for signal representation and for fundamental performance bounds.

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. Signal Representations
2. Measures of Information
3. Compression and Quantization
4. Sparsity
5. Exponential Families, Maximum Entropy
6. Detection and Estimation Theory

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, 90% final exam.

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

MATH-360 **Graph theory**

Maffucci Riccardo Walter

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

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

Summary

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

Content

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

Keywords

Graphs, isomorphism, complements, complete, bipartite, products, graphic sequences, connected, paths, circuits, cycles, Eulerian, Hamiltonian, trees, spanning trees, planar, maximal planar, polyhedra, colourings, forbidden graphs, extremal graphs.

Learning Prerequisites

Recommended courses

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

Learning Outcomes

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

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

Assessment methods

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

Resources

Bibliography

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

Ressources en bibliothèque

• Graph Theory / Diestel
• Modern Graph Theory / Bollobas
• Graph Theory / Harary
• (electronic version)
• Combinatorics and Graph Theory / Harris, Hirst & Mossinghoff
• (electronic version)

EE-451

Image analysis and pattern recognition

Thiran Jean-Philippe

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

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

Summary

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

Content

Introduction
Digital image acquisition and properties.
Pre-processing: geometric transforms, linear filtering, image restoration.
Introduction to Mathematical Morphology
Examples and applications
Segmentation and object extraction
Thresholding, edge detection, region detection.
Segmentation by active contours. Applications in medical image segmentation.
Shape representation and description
Contour-based representation, region-based representation. Morphological skeletons
Shape recognition
Statistical shape recognition, Bayesian classification, linear and non-linear classifiers, perceptrons, neural networks and unsupervised classifiers.
Applications.
Practical works on computers

Learning Prerequisites

Recommended courses

Introduction to signal processing, Image processing

Learning Outcomes

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

Transversal skills

• Assess one's own level of skill acquisition, and plan their on-going learning goals.
• 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.
• Make an oral presentation.
• Summarize an article or a technical report.

Teaching methods

Ex cathedra and practical work and oral presentation by the students

Assessment methods

Continuous control

Resources

Bibliography

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

Ressources en bibliothèque

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

Prerequisite for

Semester project, Master project, doctoral thesis

COM-402

Information security and privacy

Hubaux Jean-Pierre, Oechslin Philippe

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

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

Summary

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

Content

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

Keywords

security, privacy, protection, intrusion, anonymization, cryptography

Learning Prerequisites

Required courses

Basic Python programming or better
Basec networking knowledge

Learning Outcomes

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

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

Expected student activities

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

Assessment methods

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

EPFL

COM-404 **Information theory and coding**

Telatar Emre

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

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

Summary

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

Content

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

Learning Outcomes

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

- Formulate the 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

Ressources en bibliothèque

- Elements of Information Theory / Cover

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.
Energy Management and Sustainability	MA1, MA3	Opt.
Financial engineering minor	H	Opt.
Financial engineering	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
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

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

Resources

Bibliography

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

Ressources en bibliothèque

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

CS-486 **Interaction design**

Pu Pearl

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

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

Summary

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

Content

Design methods for HCI
What is HCI: its aims and goals
Design thinking
Goal-directed Design
Mental model and different types of users
Qualitative research and user interviews
User modeling: persona and empathy diagram
Scenarios, requirements and framework design
Visual design
Information Visualization design
Basic prototyping methods for HCI
Storyboarding
Context scenario
Interactive prototype
Video prototype
Human computer interaction evaluation methods
Cognitive walkthrough
Heuristic evaluation
Evaluation with users

Keywords

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

Learning Prerequisites

Required courses

Introduction to Visual Computing

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

Important concepts to start the course
Goal-direction design

Learning Outcomes

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

- Interview users and elicit their needs using the goal-directed design method
- Design and implement interfaces and intearctions
- 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, exercises, hands-on practice, design review

Expected student activities

Lectures, readings, design project, quiz

Assessment methods

Group project, presentation, mid-term exam

Resources

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

Ressources en bibliothèque

- About Face 3

CS-431 **Introduction to natural language processing**

Chappelier Jean-Cédric, Rajman Martin

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

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

Summary

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

Content

Several models and algorithms for automated textual data processing will be described: (1) morpho-lexical level: electronic lexica, spelling checkers, ...; (2) syntactic level: regular, context-free, stochastic grammars, parsing algorithms, ...; (3) semantic level: models and formalisms for the representation of meaning, ...
Several application domains will be presented: Linguistic engineering, Information Retrieval, Text mining (automated knowledge extraction), Textual Data Analysis (automated document classification, visualization of textual data).

Keywords

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

Learning Outcomes

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

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

Teaching methods

Ex cathedra ; practical work on computer

Expected student activities

attend lectures and practical sessions, answer quizzes.

Assessment methods

4 quiz during semester 25%, final exam 75%

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

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

- http://coling.epfl.ch

COM-490 **Lab in data science**

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

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

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

Summary

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

Content

1. Crash-course in Python for data scientists

- Main Python libraries for data scientists
- Interactive data science with web-based notebooks
- Reusable compute environments for reproducible science

- **Homework**: Curating data from a network of CO2 sensors

2. Distributed data wrangling at scale

- Understand the main constituents of an Apache Hadoop distribution
- Put Map-Reduce into practice
- Focus on HDFS, Hive and HBase and associated data storage formats

- **Homework**: Big data wrangling with massive travel data from SBB/CFF

3. Distributed processing with Apache Spark

- RDDs and best practices for order of operations, data partitioning, caching
- Data science packages in Spark: GraphX, MLlib, etc.

- **Homework**: Uncovering world events using Twitter hashtags

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

- Window-based processing of unbounded data

- **Homework**: Geospatial analysis and visualization of real-time train geolocation data from the Netherlands

5. Final project - Summing it all up

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

Keywords

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

Learning Prerequisites

Required courses

Students must have prior experience with Python

Recommended courses

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

Important concepts to start the course

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

Learning Outcomes

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

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

Transversal skills

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

Teaching methods

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

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

Expected student activities

- STUDY : Attend the lab sessions
- WORK : Complete homework assignments
- ENGAGE : Contribute to the 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 in small groups

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

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

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

Ressources en bibliothèque

• Python data science handbook

Websites

• https://dslab2020.github.io

CS-526

Learning theory

Macris Nicolas, Urbanke Rüdiger

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

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

Summary

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

Content

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

Learning Prerequisites

Recommended courses

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

Learning Outcomes

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

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

Teaching methods

- Lectures
- Exercises

Expected student activities

- Attend lectures
- Attend exercises sessions and do the homework

Assessment methods

Final exam and graded homeworks

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes
Others	Course website

MATH-341

Linear models

Panaretos Victor

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

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

Summary

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

Content

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

Learning Prerequisites

Recommended courses

Analysis, Linear Algebra, Probability, Statistics

Learning Outcomes

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

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

• Prove basic results related to the statistical theory of linear models

Teaching methods

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

Assessment methods

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

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Ressources en bibliothèque

- Statistical Models / Davison
- Methods and Applications of Linear Models / Hocking
- Applied Regression Analysis / Draper

CS-433

Machine learning

Flammarion Nicolas, Jaggi Martin

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

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

Summary

Machine learning and data analysis 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, and k-NN.*
2. *Fundamental concepts: cost-functions and optimization, cross-validation and bias-variance trade-off, curse of dimensionality.*
3. *Unsupervised learning: k-Means Clustering, Gaussian mixture models and the EM algorithm.*
4. *Dimensionality reduction: PCA and matrix factorization, word embeddings*
5. *Advanced methods: generalized linear models, SVMs and Kernel methods, Neural networks and deep learning*

Keywords

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

Learning Prerequisites

Required courses

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

Recommended courses

- *Introduction to differentiable optimization (MATH-265)*
- *Linear Models (MATH-341)*

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

Teaching methods

- Lectures
- Lab sessions
- Course Projects

Expected student activities

Students are expected to:

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

Assessment methods

• Written final exam
• Continuous control (Course projects)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

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

Ressources en bibliothèque

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

Notes/Handbook

https://github.com/epfml/ML_course

Websites

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

EPFL

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.
SC master EPFL	MA2, MA4	Opt.

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

Summary

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

Content

The users of computer environments such as intelligent tutoring systems, interactive games, and web services are often very heterogeneous and therefore it is important to adapt to their specific needs and preferences.
This course will cover the core methods of adaptation and personalization, with a focus on educational data. Specifically we will discuss approaches to the task of accurately modeling and predicting human behavior within a computer environment. Furthermore, we 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-233a / CS-233b Introduction to machine learning

Important concepts to start the course

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

Learning Outcomes

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

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

Teaching methods

- Lectures
- Weekly lab sessions
- Course project

Expected student activities

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

Assessment methods

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

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

COM-516

Markov chains and algorithmic applications

Lévêque Olivier, Macris Nicolas

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

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

Summary

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

Content

Part 1: Markov chains (~6 weeks):
- basic properties: irreducibility, periodicity, recurrence/transience, stationary and limiting distributions,
- ergodic theorem: coupling method
- detailed balance
- convergence rate to the equilibrium, spectral gap, mixing times
- cutoff phenomenon

Part 2: Sampling (~6 weeks)
- classical methods, importance and rejection sampling
- Markov Chain Monte Carlo methods, Metropolis-Hastings algorithm, Glauber dynamics, Gibbs sampling
- applications: function minimization, coloring problem, satisfiability problems, Ising models
- coupling from the past and exact simulation

Keywords

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

Learning Prerequisites

Required courses

Basic probability course
Basic linear algebra and calculus courses

Recommended courses

Stochastic Models for Communications (COM-300)

Important concepts to start the course

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

Learning Outcomes

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

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

Teaching methods

ex-cathedra course

Expected student activities

active participation to exercise sessions and implementation of a sampling algorithm

Assessment methods

graded homeworks (20%), mini-project (20%), final exam (60%)

Resources

Bibliography

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

Ressources en bibliothèque

- Probability and random processes / Grimmett

Notes/Handbook

Lecture notes will be provided

Websites

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

Prerequisite for

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

COM-514 Mathematical foundations of signal processing

Bejar Haro Benjamin, Simeoni Matthieu

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

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

Summary

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

Content

Sequences, Discrete-Time Systems, Functions and Continuous-Time Systems (review of discrete-time Fourier transform; DFT; Fourier transform and Fourier series).
From Euclid to Hilbert: Linear Algebra Fundamentals for Representation Theory (vector spaces; Hilbert spaces; approximations, projections and decompositions; bases and frames; linear operators; adjoint; generalized inverses; matrix representations; computational aspects)
Sampling and Interpolation (sampling and interpolation with normal and non orthogonal vectors, sequences and functions; sampling and interpolation of bandlimited sequences and functions)
Polynomial and Spline Approximation (Legendre and Chebyshev polynomials; Lagrange interpolation; minimax approximation; Taylor expansions; B-splines)
Regularized Inverse Problems (regularized convex optimisation; Tikhonov regularisation; penalised basis pursuit; proximal algorithms; pseudo-differential operators and L-splines; representer theorems for continuous inverse problems with Tikhonov penalties)
Computerized Tomography (line integrals and projections, Radon transform, Fourier projection/slice theorem, filtered backprojection algorithm, algebraic reconstruction techniques).
Finite Rate of Innovation: Sampling Non Bandlimited Signals (overview and definitions, reconstruction methods and applications)
Adaptive Filtering (Wiener filtering, matrix inversion lemma, RLS, LMS, beamforming)

Learning Prerequisites

Required courses

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

Recommended courses

Signals and Systems

Important concepts to start the course

Good knowledge of linear algebra concepts. Basics of Fourier analysis and signal processing.

Learning Outcomes

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

- Master the right tools to tackle advanced signal and data processing problems
- Develop an intuitive understanding of signal processing through a geometrical approach
- Get to know the applications that are of interest today
- Learn about topics that are at the forefront of signal processing research

Teaching methods

Ex cathedra with exercises and homeworks.

Expected student activities

Attending lectures, completing exercises

Assessment methods

mini project 30%, final exam (written) 70%

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

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

Ressources en bibliothèque

- Signal Processing: Foundations / Vetterli

EE-556 **Mathematics of data: from theory to computation**

Cevher Volkan

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

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

Content

The course consists of the following topics
Lecture 1: Introduction. The role of models and data. Maximum-likelihood formulations. Sample complexity bounds for estimation and prediction.
Lecture 2: The role of computation. Challenges to optimization algorithms. Optimality measures. Structures in optimization. Gradient descent. Convergence rate of gradient descent.
Lecture 3: Optimality of convergence rates. Accelerated gradient descent. Concept of total complexity. Stochastic gradient descent.
Lecture 4: Concise signal models. Compressive sensing. Sample complexity bounds for estimation and prediction. Challenges to optimization algorithms for non-smooth optimization.
Lecture 5: Introduction to proximal-operators. Proximal gradient methods. Linear minimization oracles. Conditional gradient method for constrained optimization.
Lecture 6: Time-data trade-offs. Variance reduction for improving trade-offs.
Lecture 7: A mathematical introduction to deep learning. Double descent curves and over-parameterization. Implicit regularization.
Lecture 8: Structures in non-convex optimization. Optimality measures. Escaping saddle points. Adaptive gradient methods.
Lecture 9: Adversarial machine learning and generative adversarial networks (GANs). Wasserstein GAN. Difficulty of minimax optimization.
Lecture 10: Primal-dual optimization-I: Fundamentals of minimax problems. Pittfalls of gradient descent-ascent approach.
Lecture 11: Primal-dual optimization-II: Extra gradient method. Chambolle-Pock algorithm. Stochastic primal-dual methods.
Lecture 12: Primal-dual III: Lagrangian gradient methods. Lagrangian conditional gradient methods.
Recitation 1: Generalized linear models. Logistic regression.
Recitation 2: Computation of Gradients. Reading convergence plots. Helpful definitions on linear algebra.
Recitation 3: Activation functions in neural networks. Backpropogation. Introduction to pytorch.

Keywords

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

Learning Prerequisites

Required courses

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

Learning Outcomes

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

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

CS-439

Optimization for machine learning

Flammarion Nicolas, Jaggi Martin

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

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

Summary

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

Content

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

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

Advanced Contents:

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

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

Keywords

Optimization, Machine learning

Learning Prerequisites

Recommended courses

- CS-433 Machine Learning

Important concepts to start the course

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

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

Learning Outcomes

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

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

Transversal skills

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

Teaching methods

• Lectures
• Exercises with Theory and Implementation Assignments

Expected student activities

Students are expected to:

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

Assessment methods

• Continuous control (course project)
• Final Exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Websites

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

EPFL

COM-508

Optional project in data science

Profs divers *

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

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

Summary

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

Content

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

Learning Outcomes

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

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

Transversal skills

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

Teaching methods

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

Assessment methods

Oral presentation and written report.

Resources

Websites

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

COM-503

Performance evaluation

Le Boudec Jean-Yves

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

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

Remarque

Cours biennal donné les années paires

Summary

In this course you will learn the methods and techniques that are used to perform a good performance evaluation during a research or development project.

Content

MethodologyA Performance Evaluation Methodology. The scientific method. Dijkstra and Occam's principle.
Statistics and Modeling.
Statistics and modeling, why and how. Comparing systems using sampled data. Regression models. Factorial analysis. Stochastic load and system models. Load forecasting. The Box-Jenkins method.
Practicals.
Using a statistics package (Matlab). Measurements. Discrete event simulation. Stationarity and Steady State. Analysis of simulation results. Perfect Simulations.
Elements of a Theory of Performance. Performance of systems with waiting times. Utilization versus waiting times. Operational laws. Little's formula. Forced flows.law. Stochastic modeling revisited. The importance of the viewpoint. Palm calculus. Application to Simulation Performance patterns in complex systems. Bottlenecks. Congestion phenomenon. Performance paradoxes.

Mini-Project proposed by student.

Learning Prerequisites

Required courses

A first course on probability
A first course on programming

Learning Outcomes

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

- Estimate confidence intervals
- Design a simulation method
- Critique performance metrics and factors
- Organize a performance evaluation study
- Quantify performance

- Conduct a performance analysis
- Synthesize performance results
- Systematize factors and metrics
- Present results of a performance analysis

Transversal skills

- Use a work methodology appropriate to the task.
- Demonstrate the capacity for critical thinking

Teaching methods

Lectures + pencil and paper exercises + labs + miniproject

Expected student activities

Lectures
Paper and pencil exercises
Labs
Miniproject (last 4 weeks)
Online quizzes.

Assessment methods

E = grade at final exam (during exam session)
L = average of labs
M = miniproject grade
Final grade = 1/3 (E+L+M), rounded to the nearest half integer.
All grades except the final grade are not rounded.

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

- Performance Evaluation of Computer and Communication Systems, Le Boudec Jean-Yves, EPFL Press 2010
- also freely available online at perfeval.epfl.ch

Ressources en bibliothèque

- Performance evaluation of computer and communication systems / Le Boudec

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=14395

MATH-447 **Risk, rare events and extremes**

Davison Anthony

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

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

Remarque

Cours donné en alternance sur deux ans

Summary

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

Content

• **Mathematical bases**: behaviour of maxima and threshold exceedances in large samples, both for independent and dependent data. Poisson process modelling.

• **Statistical methods**: modelling using the GEV and GP distributions, for independent and dependent data. Likelihood and Bayesian inference. Non-stationarity. Extremal coefficients. Multivariate extreme-value distributions. Max-stable processes.

• **Applications**: Environmental, financial, and engineering applications. Use of R for extremal modelling.

Learning Prerequisites

Important concepts to start the course

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

Learning Outcomes

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

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

Teaching methods

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

Assessment methods

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

Resources

Bibliography

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

COM-412

Semester project in Data Science

Profs divers *

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

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

Summary

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

Content

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

Learning Outcomes

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

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

Transversal skills

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

Assessment methods

Written report and oral presentation

Resources

Websites

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

CS-412 **Software security**

Payer Mathias

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

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

Summary

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

Content

This course focuses on software security fundamentals, secure coding guidelines and principles, and advanced software security concepts. Students will learn to assess and understand threats, learn how to design and implement secure software systems, and get hands-on experience with common security pitfalls.
Software running on current systems is exploited by attackers despite many deployed defence mechanisms and best practices for developing new software. In this course students will learn about current security threats, attack vectors, and defence mechanisms on current systems. The students will work with real world problems and technical challenges of security mechanisms (both in the design and implementation of programming languages, compilers, and runtime systems).

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

Focus topic : Mobile security

Keywords

Software security, mitigation, software testing, sanitization, fuzzing

Learning Prerequisites

Required courses

- COM-402 Information security and privacy

Important concepts to start the course

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

Learning Outcomes

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

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

Transversal skills

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

Teaching methods

The lectures are denser early in the semester, then tapering off before the end. They may be peppered with occasional short surprise quizzes that are not mandatory but may earn points for successful participants. They are backed up by PDF files of all the lecture material, as well as a few textbook recommendations.
The exercises sessions start slowly early in the semester but pick up and occupy all time towards the end. They consist mostly of paper questions involving the analysis, critical review, and occasional correction of software. They include a reading, writing, and presentation assignment.

Expected student activities

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

Assessment methods

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

Supervision

Office hours	Yes
Assistants	Yes
Forum	No

Resources

Notes/Handbook

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

EPFL

MATH-486 **Statistical mechanics and Gibbs measures**

Friedli Sacha

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

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

Summary

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

Content

The goals of this course are to present

- the probabilistic description of large systems with interacting components,
- the mathematical description of phase transitions occuring in certain discrete models (Curie-Weiss, Ising model, Gaussian Free Field, systemswith continuous symmetries, 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

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

Websites

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

MATH-442 **Statistical theory**

Koch Erwan

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

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

Summary

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

Content

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

Learning Prerequisites

Recommended courses

Real Analysis, Linear Algebra, Probability, Statistics.

Learning Outcomes

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

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

Teaching methods

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

Assessment methods

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

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Ressources en bibliothèque

- Mathematical Statistics (e-book)
- Mathematical Statistics / Knight

Notes/Handbook

The slides will be available on Moodle.

MATH-413

Statistics for data science

Olhede Sofia Charlotta

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

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

Summary

Statistics lies at the foundation of data science, providing a unifying theoretical and methodological backbone for the diverse tasks 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

Moodle Link

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

COM-506 Student seminar: security protocols and applications

Oechslin Philippe, Vaudenay Serge

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

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

Summary

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

Content

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

Keywords

network security, security protocols, cryptography

Learning Prerequisites

Required courses

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

Learning Outcomes

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

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

Transversal skills

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

Expected student activities

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

Assessment methods

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

Supervision

Others	Lecturers and assistants are available upon appointment.

CS-449 **Systems for data science**

Kermarrec Anne-Marie

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

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

Summary

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

Content

Big data systems design and implementation :

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

Large-scale storage systems :

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

Large-scale processing :

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

Keywords

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

Learning Prerequisites

Required courses

CS-322: Introduction to database systems

Recommended courses

CS-323: Introduction to operating systems

CS-206 Parallelism and concurrency

Important concepts to start the course

• Algorithms and data structures – sorting algorithms, balanced trees, graph traversals.

• The Scala programming language will be used throughout the course. Programming experience in this language is strongly recommended.

• Basic knowledge or computer networking and distributed systems

Learning Outcomes

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

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

Teaching methods

Ex cathedra; including exercises in class, practice with pen and paper or with a computer, and a project

Expected student activities

During the semester, the students are expected to:

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

Assessment methods

Homeworks, written examinations, project. Continuous control

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes
Others	Office ours by appointment

Resources

Bibliography

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

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.
Mineur STAS Russie	E	Opt.

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

Summary

A first course in statistical time series analysis and applications.

Content

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

Learning Prerequisites

Required courses

Probability and Statistics

Recommended courses

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

Important concepts to start the course

The material from first courses in probability and statistics.

Learning Outcomes

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

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

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

Teaching methods

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

Assessment methods

final exam

Supervision

Office hours	No
Assistants	Yes
Forum	No

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

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

Ressources en bibliothèque

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

Notes/Handbook

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

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

EPFL

CS-455

Topics in theoretical computer science

Kapralov Mikhail

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

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

Remarque

Cours biennal

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

Others	Electronique forum : Yes

Resources

Bibliography

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

Ressources en bibliothèque

- Randomized Algorithms / Motwani

CS-444 **Virtual reality**

Boulic Ronan

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

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

Summary

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

Content

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

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

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

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

Keywords

3D interaction, display, sensors, immersion, presence

Learning Prerequisites

Required courses

(CS 341) Introduction to Computer Graphics

Recommended courses

(CS 211) Introduction to Visual Computing

Important concepts to start the course

from Computer Graphics:
- perspective transformations
- representation of orientation

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

Learning Outcomes

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

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

Transversal skills

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

Teaching methods

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

Expected student activities

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

Assessment methods

Throughout semester: 4-5 Hand-on sessions (5%), 2 Quizzes (10%), 1 paper citation study (20%), 1 mini-project (40%), 1 oral (25%)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

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

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

Ressources en bibliothèque

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

Notes/Handbook

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

INDEX PAR COURS

	Cours	Enseignant	Semestre	Page
A	Advanced algorithms	Kapralov M.	M2, M4	23-24
	Advanced cryptography	Vaudenay S.	M2, M4	25-26
	Advanced probability and applications	Lévêque O.	M2, M4	27-28
	Advanced topics on privacy enhancing technologies	Troncoso C.	M2, M4	29-30
	Applied biostatistics	Goldstein D.	M2, M4	31-32
	Applied data analysis	West R.	M1, M3	33 à 35
	Artificial neural networks	Gerstner W.	M2, M4	36 à 38
	Automatic speech processing	Bourlard H.	M1, M3	39-40
B	Biological modeling of neural network	Gerstner W.	M2, M4	41 à 43
C	Computational linear algebra	Kressner D.	M2, M4	44-45
	Computational photography	Süsstrunk S.	M2, M4	46-47
	Computer vision	Fua P.	M2, M4	48-49
	Concurrent algorithms	Guerraoui R.	M1, M3	50-51
	Cryptography and security	Vaudenay S.	M1, M3	52 à 54
	Data visualization	Vuillon L.	M2, M4	55 à 57
D	Deep learning	Fleuret F.	M2, M4	58-59
	Digital education & learning analytics	Dillenbourg P. / Jermann P.	M1, M3	60-61
	Distributed algorithms	Guerraoui R.	M1, M3	62-63
	Distributed information systems	Aberer K.	M1, M3	64-65
	Distributed intelligent systems	Martinoli A.	M2, M4	66 à 68
F	Formal verification	Kuncak V.	M1, M3	69 à 71
	Foundations of Data Science	Urbanke R.	M1, M3	72-73
G	Graph theory	Maffucci R.	M1, M3	74-75
I	Image analysis and pattern recognition	Thiran J.-Ph.	M2, M4	76-77
	Information security and privacy	Hubaux J.-P./Oechslin Ph.	M1, M3	78-79
	Information theory and coding	Telatar E.	M1, M3	80
	Intelligent agents	Faltings B.	M1, M3	81-82
	Interaction design	Pu Faltings P.	M2, M4	83-84
	Introduction to natural language processing	Chappelier J.-C. / Rajman M.	M1, M3	85-86
L	Lab in Data Science	Bouillet E./Delgado P./Sarni S./Verscheure O.	M2, M4	87 à 89
	Learning theory	Macris N./Urbanke R.	M2, M4	90-91
	Linear models	Panaretos V.	M1, M3	92-93
M	Machine learning	Flammarion N. / Jaggi M.	M1, M3	94 à 96
	Machine learning for behavioral data	Käser T.	M2, M4	97-98
	Markov chains and algorithmic applications	Lévêque O. / Macris N.	M1, M3	99-100
	Mathematical foundations of signal processing	Simeoni M. / Bejar Haro B.	M1, M3	101-102
	Mathematics of data from theory to computation	Cevher V.	M1, M3	103-104
O	Optimization for machine learning	Flammarion N. / Jaggi M.	M2, M4	105-106
	Optional project in Data science	Divers enseignants	M1,M2,M3,M4	107
P	Performance evaluation	Le Boudec J.-Y.	M2, M4	108-109
R	Risk, rare events and extremes.	Davison A.	M1, M3	110-111
S	Semester project in Data Science	Divers enseignants	M1,M2,M3,M4	112
	Software security	Payer M.	M2, M4	113 à 115
	Statistical mechanics and Gibbs measures	Friedli S.	M2, M4	116-117
	Statistical theory	Koch E.	M1, M3	118-119
	Statistics for data science	Olhede S.	M1, M3	120-121
	Student seminar : security protocols and applications.	Oechslin Ph. / Vaudenay S.	M2, M4	122-123
	Systems for data science	Kermarrec A.-M.	M2, M4	124-125
T	Time series	Olhede S.	M2, M4	126-127
	Topics in theoretical computer science	Kapralov M.	M1, M3	128-129
V	Virtual reality	Boulic R.	M2, M4	130 à 132

INDEX PAR ENSEIGNANTS

	Enseignant	Cours	Semestre	Page
A	Aberer K.	Distributed information systems	M1, M3	64-65
B	Bejar Haro B.	Mathematical foundations of signal processing	M1, M3	101-102
	Boulic R.	Virtual reality.	M2, M4	130 à 132
	Bourlard H.	Automatic speech processing	M1, M3	39-40
C	Cevher V.	Mathematics of data from theory to computation	M2, M4	103-104
	Chappelier J.-C.	Introduction to natural language processing	M1, M3	85-86
D	Davison A.	Risk, rare events and extremes	M1, M3	110-111
	Dillenbourg P.	Digital education & learning analytics	M1, M3	60-61
	Divers enseignants	Optional project in data science	M1,M2,M3,M4	107
		Semester project in data science	M1,M2,M3,M4	112
F	Faltings B.	Intelligent agents	M1, M3	81-82
	Flammarion N.	Machine learning.	M1, M3.	94 à 96
		Optimization for machine learning.	M2, M4.	105-106
	Fleuret F.	Deep learning	M2, M4	58-59
	Friedli S.	Statistical mechanics and Gibbs measures	M2, M4.	116-117
	Fua P.	Computer vision	M2, M4	48-49
G	Gerstner W.	Artificial neural networks	M2, M4	36 à 38
		Biological modelling of neural network	M2, M4	41 à 43
	Goldstein D.	Applied biostatistics	M2, M4	31-32
	Guerraoui R.	Concurrent algorithms.	M1, M3.	50-51
		Distributed algorithms	M1, M3.	62-63
H	Hubaux J.-P.	Information security and privacy	M1, M2	78-79
J	Jaggi M.	Machine learning	M1, M3	94 à 96
		Optimization for machine learning	M2, M4	105-106
	Jermann P.	Digital education & learning analytics	M1, M3	60-61
K	Käser T.	Machine learning for behavioral data.	M2, M4.	97-98
	Kapralov M.	Advanced algorithms.	M2, M4.	23-24
		Topics in theoretical computer science	M1, M3	128-129
	Kermarrec A.-M.	Systems for Data Science.	M2, M4.	124-125
	Koch E.	Statistical theory	M1, M3	118-119
	Kressner D.	Computational linear algebra	M2, M4	44-45
	Kuncak V.	Formal verification	M1, M3	69 à 71
L	Le Boudec J.-Y.	Performance evaluation	M2, M4	108-109
	Lévêque O.	Advanced probability and applications	M2, M4	27-28
		Markov chains and algorithmic applications	M1, M3	99-100
M	Macris N.	Learning theory	M2, M4	90-91
		Markov chains and algorithmic applications	M1, M3	99-100
	Maffucci R.	Graph theory	M1, M3	74-75
	Martinoli A.	Distributed intelligent systems	M2, M4	66 à 68
O	Oechslin Ph.	Information security and privacy	M1, M3	78-79
		Student seminar : Security protocols and applications.	M2, M4	122-123
	Olhede S.	Statistics for data science	M1, M3	120-121
		Time series	M2, M4	126-127
P	Panaretos V.	Linear models	M1, M3	92-93
	Payer M.	Software security	M2, M4	113 à 115
	Pu Faltings P.	Interaction design	M2, M4	83-84
R	Rajman M.	Introduction to natural language processing	M1, M3	85-86
	Simeoni M.	Mathematical foundations of signal processin	M1, M3	101-102
S	Süsstrunk S.	Computational photography	M2, M4	46-47
T	Telatar E.	Information theory and coding	M1, M3	80
	Thiran J.-Ph.	Image analysis and pattern recognition	M2, M4	76-77
	Troncoso C.	Advanced topics on privacy enhancing technologies	M2, M4	29-30
U	Urbanke R.	Foundations of Data Science	M1, M3	72-73
		Learning theory	M2, M4	90-91
V	Vaudenay S.	Advanced cryptography	M2, M4	25-26
		Cryptography and security	M1, M3	52 à 54
		Student seminar : Security protocols and applications	M2, M4	122-123
	Verscheure O.	Lab in Data Science	M2, M4	87 à 89
	Vuillon L.	Data visualization	M2, M4	55 à 57
W	West R.	Applied data analysis	M1, M3	33 à 35