

Syllabus M1 IMA : version préliminaire

Attention ce document résume la structure du futur M1 IMA et détaille les syllabi des différentes Ues. Nous prevenons le lecteur que ces syllabi sont encore susceptibles de changer sur des points mineurs. La version définitive sera publiée sur SGCE et puissur le site de l'université dans les prochains mois. Le tableau ci-dessous résume lesUes au programme au 1er et au deuxième semestre.

Nom de l'UE	UE Obligatoire ou Optionnelle	IP formation dans l'UE	IP totales dans l'UE	Dept.	ECTS										
						H CM	H CTDI	H TD	H TD LV	H TP	H TPDE	H TPT	H projet	H stage	
Mise à niveau	Option	6	76	Inf.	0	0	24	0	0	0	0	0	0	0	
Théorie des Langages (IAFA)	O	4	198	Inf.	6	22		14		18					
Algorithmique avancée (IAFA)	O	4	234	Inf.	6	20		24		10					
Optimisation (MAPI3)	O	4	48	Math.	6	24		24		12					
Probabilités (MAPI3)	O	4	48	Math.	6	24		24		12					
Simulation aléatoire (MAPI3)	O	4	48	Math.	6	18		20		24			0,3		
LANGUE VIVANTE															
LANGUE VIVANTE	O	4	218	LVG	3	0	0	0	24	0	0	0	0	0	
Intelligence Artificielle 1 (IAFA)	O	4	92	Inf.	6	24		24		12					
Traitement de données 1 : image, son et texte (IAFA)	O	4	92	Inf.	6	28		12		20					
Travaux d'Initiation à la recherche	O	4	20	ANITI	3	18							1		
Statistiques (MAPI3)	O	4	53	Math.	6	24		24		12					

Syllabus version française

Nom du module :

- Algorithmique

Objectifs : (800 caractères)

- Donner ici une description textuelle des objectifs de formation du module en une dizaine de lignes.
 - o Ce module est destiné à donner une culture générale sur les algorithmes de résolution de problèmes combinatoires, de décision ou d'optimisation. Cela passe par la connaissance de problèmes génériques et de leur complexité avec leurs représentations et leurs algorithmes associés.
- Identifier la (ou les) compétence(s) attestée(s) après validation du module
 - o Décrire et Implémenter les opérations manipulant des arbres de recherche et des tas, établir leur complexité temporelle et spatiale
 - o Caractériser un problème d'optimisation combinatoire
 - o Formaliser un problème spécifique en langage naturel sous la forme d'un problème de décision (en SAT ou CSP) ou d'optimisation (en PL)
 - o Implémenter et utiliser un algorithme de résolution de Flots, de SAT, de CSP, de PL
 - o Choisir et implémenter un algorithme approprié de recherche locale
- Identifier la (ou les) connaissances acquises(s) après validation du module
 - o Complexité des problèmes, algorithmes DPLL, algorithme Ford-Fulkerson, algorithme du Simplex, notion de recherche locale.

Contenu : (1200 caractères)

- Donner le plan détaillé du cours et le déroulé des travaux dirigés associés
 - o A) Rappels de Complexité des Algorithmes
 - Intérêt, concepts, notations, exemples
 - Exemples complexité algos de recherche ou de tri avec ABR, ARN, AVL, Tas binaire
 - o B) Exemples de problèmes d'Optimisation combinatoire (méthodes complètes)
 - problème sur graphes: polynomial (flots)
 - exemples de problèmes classiques (knapsack, binpacking, TSP, ...) avec algos gloutons non exacts (coefficients d'approximation)
 - intro du pb SAT + algo résol DPLL
 - ex ordonnancement codé en SAT= transformation polynomiale
 - problème sur graphes: NP (coloration ou autres)
 - Complexité des pbs (NP, NP-complétude, réduction)
 - o C) Formalismes génériques
 - pb de décision : SAT + CSP et algos backtrack +FC
 - pb d'optimisation : PL (formalisme, résol graphique, dual) + PLNE
 - pbs de modélisation: méthodes génériques pour trouver la bonne modélisation avec exemples
 - o D) Approximations et algos incomplets
 - classes d'approximation (complexité)
 - algos heuristiques avec un taux d'approximation (spécifiques aux problèmes)
 - méta-heuristiques (algo génériques de recherche locale ou sur population)
- Donner le plan des TPs
 - o implémentation solveur SAT, PLNE, Meta-Heuristiques
- En cas de projet dans le module, décrire le travail attendu pendant le projet

Pré-requis : (240 caractères)

- Décrire les connaissances/compétences requises pour suivre cet enseignement

Graphes, Structures de Données (arbres binaires de recherche, Tas), Complexité

Références bibliographiques : (320 caractères)

- Donner 2 ou 3 références bibliographiques de livres pouvant être utilisé comme support et/ou complément pour le cours. Donner au moins 1 référence en anglais.
 - o [Introduction to Algorithms](#). By Thomas Cormen, Charles Leiserson, Ronald Rivest and Clifford Stein. MIT Press 2009
 - o [Graphes et Algorithmes](#). By Michel Gondran and Michel Minoux. (4e ed.). Lavoisier 2009.
 - o [Combinatorial Optimization](#). By Christos Papadimitriou and Kenneth Steiglitz. Dover Publications Inc. 2013
 - o [Metaheuristics from design to implementation](#). By El-Ghazali Talbi. John Wiley & Sons, Inc. 2009
- Si les enseignement (à travers les TPs par exemple) reposent sur des outils/environnements logiciels particuliers, donner, si pertinent, le lien vers un site web officiel de l'outil/environnement.

Mots-clés : (160 caractères)

- Liste des mots clés (entre 3 et 6 ?) synthétisant le contenu du cours.
 - o [Tas binaire](#), [Arbre Binaire de Recherche](#), [Flot](#), [Programmation Linéaire](#), [Programmation linéaire en nombres entiers](#), [SAT](#), [CSP](#), [NP-complet](#), [Algorithmes non exacts](#), [Meta-Heuristique](#),

Modalités d'évaluation : (240 caractères)

- Décrire succinctement les modalités de contrôle des connaissances et de suivi de l'acquisition de compétences qui sont envisagées pour ce module.
 - o [20% 1 CCTP](#) correspondant à 3 ou 4 Epreuves de TP avec R16(abs-> 0)
 - o [30% 1 CP écrit](#) : 1^e partie du cours avec R16
 - o [50% 1 CT écrit](#) : reste du cours
 - o [En 2^e session: 1 CT de 100%](#) (pas de report)

Actuellement les CCTPs comptent pour 20% en algo avancée, 15% en optim comb., ça me semble bien de continuer avec ça. Par contre, la note de CC est annulée en 2nde session en algo avancée, avec la règle R11 (abs -> DEF), et conservée en optim comb. avec la règle R16 (abs ->0); comme l'UE sera *a priori* au S7, durant lequel des étudiant-e-s arrivent très en retard, ça me semble mieux de faire comme en AA (pas de report, par contre R16 ?).

Actuellement le CP à un coef de 30% en algo avancée, avec la règle R16, et le CT donc un coef. de 50%. Si on laisse le CP à 30%, d'après la répartition horaire dont nous avons discuté au printemps, ça ferait un CP après les flots et la PL (pour ces parties nous avons prévu 21h, sur 54, ~38%).

Syllabus version anglaise

Module Name :

- Indiquer le nom du module
- **Algorithmic**

Objectives and learning outcomes : (800 caractères)

- Donner ici une description textuelle des objectifs de formation du module en une dizaine de lignes.
- **This module is intended to give a general culture on algorithms for combinatorial problem solving, decision and optimization. It includes the knowledge of generic problems and their complexity with their representations and their associated algorithms.**
- Identifier la (ou les) compétence(s) attestées(s) après validation du module
- **Describe and implement operations that manipulate search trees and heaps, establish their temporal and spatial complexity**
- **Characterize a combinatorial optimization problem**
- **Formalize a specified problem in natural language as a decision problem (in SAT or CSP) or optimization problem (in PL)**
- **Implement and use an algorithm to solve Flow, SAT, CSP, PL problems.**
- **Choose and implement an appropriate local search algorithm**
- Identifier la (ou les) connaissances acquises(s) après validation du module
- **Problem Complexity, DPLL algorithms, Ford-Fulkerson algorithm, Simplex algorithm, Local search notions.**

Course content : (1200 caractères)

- Donner le plan détaillé du cours et le déroulé des travaux dirigés associés
- **A) Algorithm Complexity Reminders**
 - o Interest, concepts, notations, examples
 - o Complexity Examples for search and sorting algos (BST, RBT, AVL, Binary heap)
- **B) Examples of combinatorial optimization problems (complete methods)**
 - o problem on graphs: polynomial (flows)
 - o examples of classical problems (knapsack, binpacking, TSP, ...) with non-exact greedy algorithms (approximation coefficients)
 - o intro of the SAT pb + solver algorithm DPLL
 - o SAT encoded scheduling = polynomial transformation
 - o problem on graphs: NP (coloring or other)
 - o Complexity of pbs (NP, NP-completeness, reduction)
- **C) Generic formalisms**
 - o decision pb: SAT + CSP and backtrack algos + Forward Checking
 - o optimization pb: LP (formalism, graphical solving, duality) + ILP
 - o modeling pbs: generic methods to find the suitable model with examples
- **D) Approximations and Incomplete Algos**
 - o approximation classes (complexity)
 - o heuristic algorithms with an approximation rate (problem specific)
 - o meta-heuristics (generic algo of local or population-based search)
- Donner le plan des TPs
- **solver implementation for SAT, ILP, Meta-Heuristics**
- En cas de projet dans le module, décrire le travail attendu pendant le projet

Prerequisite : (240 caractères)

- Décrire les connaissances/compétences requises pour suivre cet enseignement
- **Graphs, Data Structures (Binary Search Trees, Heaps), Complexity**

Bibliography : (320 caractères)

- Donner 2 ou 3 références bibliographiques de livres pouvant être utilisé comme support et/ou complément pour le cours. Donner au moins 1 référence en anglais.

- Introduction to Algorithms. By Thomas Cormen, Charles Leiserson, Ronald Rivest and Clifford Stein. MIT Press 2009
- Graphes et Algorithmes. By Michel Gondran and Michel Minoux. (4e ed.). Lavoisier 2009.
- Combinatorial Optimization. By Christos Papadimitriou and Kenneth Steiglitz. Dover Publications Inc. 2013
- Metaheuristics from design to implementation. By El-Ghazali Talbi. John Wiley & Sons, Inc. 2009
- Si les enseignement (à travers les TPs par exemple) reposent sur des outils/environnements logiciels particuliers, donner, si pertinent, le lien vers un site web officiel de l'outil/environnement.

Key words : (160 caractères)

- Liste des mots clés (entre 3 et 6 ?) synthétisant le contenu du cours.
- Binary Heap, Binary Search Tree, Flow, Linear Programming, Integer Linear Programming, SAT, CSP, NP-completeness, Non exact Algorithms, Meta-Heuristics.

Assessment methodology : (240 caractères)

- Décrire succinctement les modalités de contrôle des connaissances et de suivi de l'acquisition de compétences qui sont envisagées pour ce module.
- 1 CCTP corresponding to 3 or 4 Tests
- 1 CP written test : 1st part of the course
- 1 CT written test : rest of the course
- 2nd session: 1 CT of 100%

Nom du module :

Théorie des Langages (TL)

Objectifs :

Cette UE a deux objectifs : (a) comprendre le fonctionnement d'un analyseur de code source et l'obtention de code intermédiaire (*front-end*) et (b) exploiter cette représentation pour générer et optimiser des codes exécutables (*back-end*) tout en mettant en oeuvre des stratégies de vérification afin d'assurer la correction du compilateur.

Les compétences visées incluent :

- mettre en oeuvre un analyseur de code source,
- développer des traducteurs vérifiés appliqués à la représentation intermédiaire du programme,
- savoir optimiser les performances d'un programme traduisant des requêtes relationnelles.

Les connaissances acquises incluent :

- notions de compilation et exécution/interprétation,
- organisation du compilateur,
- notion de grammaire et analyse LL(k) et LR(k),
- représentation intermédiaire (arbre de syntaxe abstraite),
- analyse statique (typage),
- sémantique formelle des langages de programmation,
- optimisation de code et vérification de transformation,
- principe de la génération de code.

Contenu :

Compilation de langage impératif

Exécution et compilation

Analyse syntaxique

Génération de code

Vérification

Preuve assistée

Modélisation d'AST typés

Sémantique de langages

Transformations vérifiées

Compilation d'une requête exprimée dans un langage déclaratif

Introduction et motivations

Optimisation de code

Génération de code

Les TPs (18h) concernent seulement les parties "Compilation en Langage Impératif" et "Vérification". Ils consistent en la mise en oeuvre d'un mini-projet qui consiste à réaliser un mini-compilateur composé d'un analyseur pour un langage de programmation réel exécuté par une machine virtuelle (fournie aux étudiants).

La première partie va consister à réaliser un mini-compilateur de l'analyse du source à la génération du code: analyse lexicale (S1), analyse syntaxique (S2-3), construction des arbres de syntaxe abstraite (S4) et génération de code (S5). Dans une seconde partie, des optimisations vérifiées seront réalisées sur la représentation intermédiaire. L'environnement de preuve interactive utilisé pour cela (Coq) est d'abord présenté avec un rappel des notions sous-jacentes (S6), puis un ensemble d'exemples de sémantiques de langages de programmation est formalisé (S7),

avant de modéliser et vérifier une transformation agissant sur la représentation intermédiaire (S8-9).

Pré-requis :

- bases de la théorie des langages
- automates finis
- programmation fonctionnelle (OCaml)
- introduction à la preuve interactive (Coq)
- bonnes connaissances en bases de données relationnelles

Références bibliographiques :

- A. Aho, M. Lam, R. Sethi, J. Ullman. *Compilateurs : principes, techniques et outils (Dragon Book)*. Pearson Education.
- OCaml (<https://ocaml.org/>)
- Y. Bertot. *Coq in a Hurry*. Types Summer School, Nice, EJCP, 2016. <https://cel.archives-ouvertes.fr/inria-00001173>
- B. C. Pierce et al. *Software foundations*, volume 1. <https://softwarefoundations.cis.upenn.edu/>
- Coq (<https://coq.inria.fr/>)
- M. Bouzeghoub et al. *Systèmes de BD : des techniques d'implantation à la conception de schémas*. Eyrolles.
- C. Delobel et M. Adiba. *BD et systèmes relationnels*. Dunod.

Mots-clés :

Compilation, traduction, optimisation, génération de code, analyse syntaxique, représentation intermédiaire, sémantiques, preuve assistée, transformation vérifiée.

Modalités d'évaluation :

- Contrôle Terminal (60%)
- Contrôle Continu pour la réalisation d'un mini-projet (40%)

Module Name :

Language Theory (LT)

Objectives and learning outcomes :

This course aims: (a) to understand the work of code source parser and the production of intermediate code (front-end) and (b) to exploit this representation to generate and optimize executable code (back-end). During this process, this course shows how to apply verification strategies in order to ensure the soundness of the compiler.

Skills:

- setting up a code source parser,
- developing a verified translator applied on the program intermediate representation,
- generating optimized code translating requests from declarative languages.

Knowledges:

- concepts of compilation and execution/interpretation,
- compiler organization,
- concepts of grammars and LL(k) and LR(k) analyzes,
- intermediate representation (abstract syntax trees),
- static analysis (typing),
- formal semantics of programming languages,
- code optimization and verified transformations,
- code generation principles.

Course content :

Imperative language compilation
Execution and compilation
Syntactic Analysis
Code Generation

Verification
Assisted Proof
Typed AST Modelisation
Language Semantics
Verified Transformations

Compilation of a request expressed in a declarative language
Introduction and motivations
Code optimisation
Code generation

Lab work (18h) concerns only the parts “Imperative Language Compilation” and “Verification”. The lab work is made as a mini-project that consists in developing a mini-compiler including an actual programming language parser that is executed on a virtual machine (provided to students).

In the first part, the mini-compiler made of a source code parser and a code generator is implemented: lexical analysis (S1), syntactic analysis (S2-3), abstract syntactic tree building (S4) and code generation (S5). In the second part, verified optimizations are written on the intermediate representation : first, the interactive proof environment used to this aim (Coq) is presented, recalling the underlying notions (S6), then a set of examples of programming language semantics is

formalised (S7), before modeling and verifying a transformation that operates on the intermediate representation (S8-9).

Prerequisite :

- basics of language theory
- finite automata
- functional programming (OCaml)
- introduction to interactive theorem proving (Coq)
- good knowledge in relational databases

Bibliography :

- A. Aho, M. Lam, R. Sethi, J. Ullman. *Compilers: principles, techniques and tools (Dragon Book)*. Pearson Education.
- OCaml (<https://ocaml.org/>)
- Y. Bertot. *Coq in a Hurry*. Types Summer School, Nice, EJCP, 2016. <https://cel.archives-ouvertes.fr/inria-00001173>
- B. C. Pierce et al. *Software foundations*, volume 1. <https://softwarefoundations.cis.upenn.edu/>
- Coq (<https://coq.inria.fr/>)
- M. Bouzeghoub et al. *Systèmes de BD : des techniques d'implantation à la conception de schémas*. Eyrolles.
- C. Delobel et M. Adiba. *BD et systèmes relationnels*. Dunod.

Key words :

Compilation, translation, optimization, code generation, syntactic analysis, intermediate representation, semantics, assisted proof, verified transformation.

Assessment methodology :

- Terminal Exam (60%).
- Lab work assessment resulting for the achievement of a mini-project (40%).

Syllabus Probabilité version française

Nom du module :

- Probabilités

Objectifs :

L'étude de phénomènes complexes issus de la biologie, l'économie, l'industrie, la physique requiert de plus en plus souvent l'utilisation de familles de variables aléatoires dépendant du temps, ou plus exactement de processus stochastiques.

Cet enseignement a pour objectif principal d'introduire et d'étudier, à partir d'exemples, les processus stochastiques les plus classiques :

- les martingales, i.e. la dynamique de base dans le cadre dépendant ;
- les chaînes de Markov à espace d'états fini ou dénombrable, i.e. la dynamique markovienne de base, qui servira dans divers algorithmes et modèles ;
- le processus de Poisson, i.e. la dynamique à temps continu de base.
- Différents modèles et processus présentés seront mis en œuvre lors de TP sur Scilab, Matlab, R ou Python.

Contenu :

6 ECTS, 24h de CM, 24h de TD, 12h de TP

- Conditionnement
- Espérance conditionnelle dans L2.
- Chaînes de Markov
Propriété de Markov. Classification des états. Mesures invariantes pour espace d'états finis, convergence. Ergodicité (LGN et TLC) admise.
Exemples : modèle de Wright-Fisher, marches aléatoires sur \mathbb{Z} , \mathbb{Z}^2 , processus de Galton-Watson...
- Processus de Poisson
- Processus de Bernoulli. Processus de Poisson homogène sur \mathbb{R} .
- Introduction aux Martingales
- Surmartingales, sousmartingales. Théorème d'arrêt. Exemple ruine du joueur.

Pré-requis :

- Notions de base en théorie de la mesure, intégration, probabilités, théorèmes-limites.

Références bibliographiques :

- P. Barbe, M. Ledoux Probabilité, EDP sciences
- B. Bercu, D. Chafaï Modélisation stochastique et simulation : cours et applications, Dunod
- D. Foata, A. Fuchs Processus stochastiques, Processus de Poisson, chaînes de Markov et martingales, Dunod
- <https://www.python.org/>

Mots-clés :

- Processus stochastiques, martingales, chaînes de Markov

Modalités d'évaluation :

- CCTP+CP+CT

Syllabus version anglaise

Module Name :

- Probability

Objectives and learning outcomes :

The study of complex phenomenon from biology, economics, industry, and physics require more and more frequently the use of a family of random variables indexed by the time, or more precisely, stochastic processes.

The principal objective of this class is to introduce and to study, from examples, stochastic processes :

- martingales, i.e. the basic dynamic in dependent case ;
- Markov chains with finite or countable state space, i.e. the usual Markov dynamic, which will be useful in several models and algorithms ;
- Poisson process, i.e. the Markov dynamic in continuous time.
- Numerous models and processes will be implemented during practical sessions (Travaux Pratiques) on Scilab, Matlab, R or Python.

Course content :

6 ECTS, 24h CM, 24h TD, 12h TP

- Conditioning
- Conditional expectation in L2.
- Markov chains
Markov property. Classification of states. Invariant measures for finite state space, convergences. Ergodicity (LGN and TLS) without proofs.
Examples : Wright-Fisher model, random walks on \mathbb{Z} , \mathbb{Z}^2 , Galton-Watson process...
- Poisson process, Bernoulli process. Homogeneous Poisson process.
- Introduction to Martingales
- Surmartingales, sousmartingales. Stopping theorems. Example gambler's ruine.

Prerequisite :

- Basic knowledge in measure theory, integration, probability theory, limit theorems.

Bibliography :

- P. Barbe, M. Ledoux Probabilité, EDP sciences
- B. Bercu, D. Chafaï Modélisation stochastique et simulation : cours et applications, Dunod
- D. Foata, A. Fuchs Processus stochastiques, Processus de Poisson, chaînes de Markov et martingales, Dunod
- <https://www.python.org/>

Key words :

- Stochastic processes, martingales, Markov chains

Assessment methodology :

- CCTP+CP+CT

Syllabus Simulations aléatoires version française

Nom du module :

- Simulations aléatoires

Objectifs :

L'objectif principal est d'étudier quelques méthodes de simulation de variables aléatoires, et d'illustrer certains modèles et résultats de probabilités et de statistiques par des simulations.

Les différentes modélisations et méthodes présentées seront mises en oeuvre lors de TP sur R, Scilab, Matlab ou Python.

Contenu : 6 ECTS, 18 h CM, 20h TD, 24h TP, 12.5h projet

- Simulations de variables aléatoires réelles et méthodes de Monte Carlo
Méthode de la fonction de répartition. Méthode du rejet.
Méthode de Monte Carlo brute. Théorème de la limite centrale. Analyse de l'erreur.
(facultatif : quantification d'une probabilité sur R)
- Méthodes de rééchantillonnage
Jackknife, bootstrap, validation croisée
Méthode d'échantillonnage préférentiel par vraisemblance
- Inégalités de concentration et/ou grandes déviations (facultatif)
Utilisation de Chebyshev, Cramer-Chernoff, Bahadur-Rao, ou encore Hoeffding
- Simulation et prévision dans le modèle gaussien
Vecteur gaussien. Simulations d'un vecteur gaussien (méthode de Box-Muller, transformation linéaire). Conditionnement gaussien.
Prévision en régression linéaire gaussienne. Filtrage de Kalman
Théorème central limite multidimensionnel.
- Chaînes de Markov (théorie faite dans le cours de probabilités en parallèle)
Simulations de chaînes de Markov à espace d'états fini et continu
Estimation des probabilités de transition
Illustration des convergences
- Introduction aux MCMC
Méthode de Metropolis
Méthode de Monte Carlo utilisant une chaîne de Markov. Recuit simulé
Quelques algorithmes stochastiques : moyenne, médiane, minimisation d'une fonction

Pré-requis :

- Notions de base en intégration et probabilités, algèbre linéaire.

Références bibliographiques :

- Barbe, Ledoux (1998) « Probabilité »
- Bercu, Chafai (2007) Modélisation stochastique et simulation
- Barbe, Bertail (1995) « The weighted bootstrap ».
- <https://www.python.org/>

Mots-clés :

- simulations, Monte-Carlo, algorithmes stochastiques

Modalités d'évaluation :

- CCTP+CP+CT

Syllabus version anglaise

Module Name :

- Simulation of randomness

Objectives and learning outcomes :

The principal objective is to study some methods of simulation of random variables, and to illustrate by simulations some models and results of probability and statistics. The various modelisations and methods presented will be illustrated during practical sessions (Travaux Pratiques) on R, Scilab, Matlab ou Python.

Course content : 6 ECTS, 18 h CM, 20h TD, 24h TP, 12.5h projet

- Simulation of real random variables and Monte-Carlo method
Method of the cumulative distribution function. Rejection sampling.
Monte Carlo methods. Central limit theorem. Error analysis.
(optional : quantification of a probability on R)
- Bootstrap
Jackknife, bootstrap, cross-validation
Method of importance sampling
- Concentration inequality and/or large deviations (optional)
Use of Chebyshev, Cramer-Chernoff, Bahadur-Rao, or Hoeffding
- Simulation and forecasting in a Gaussian model
Gaussian vector. Simulations of a Gaussian vector (Box-Muller, linear transform). Gaussian conditioning.
Forecasting in Gaussian linear regression. Kalman filter
Multidimensional central limit theorem.
- Markov chain (theory done during the course of probability, same semester)
Simulation of Markov chain (finite or continuous state space)
Estimation of transition probabilities
Illustration of convergences
- Introduction to MCMC
Metropolis method
Monte Carlo method using Markov chain. Simulated annealing
Some stochastic algorithms : mean, median, minimisation of a cost function

Prerequisite :

- Basic knowledge in integration and probability, linear algebra.

Bibliography :

- Barbe, Ledoux (1998) « Probabilité »
- Bercu, Chafai (2007) Modélisation stochastique et simulation
- Barbe, Bertail (1995) « The weighted bootstrap ».
- <https://www.python.org/>

Key words :

- Simulations, Monte-Carlo, stochastic algorithms

Assessment methodology :

- CCTP+CP+CT

Syllabus version française

Nom du module : Intelligence Artificielle 1 : Apprentissage automatique 1

Objectifs : (800 caractères)

Cette matière présente de manière détaillée les approches en apprentissage profond (deep learning), plus précisément les réseaux de neurones profonds. Ce paradigme est à l'origine de grandes avancées dans beaucoup de domaines d'application : reconnaissance d'objets dans les images, super-résolution en imagerie médicale, reconnaissance automatique de la parole, génération de textes, etc.

Nous aborderons les techniques d'apprentissage de ces réseaux, avec l'algorithme de rétro-propagation (back-propagation) et la différentiation automatique, au cœur des frameworks de deep learning actuels.

Nous aborderons les architectures standards utiles en fonction de différentes applications : les réseaux fully-connected, convolutifs, récurrents. Les mécanismes d'attention seront abordés, ceux-ci permettent à un modèle d'apprendre à pondérer lui-même les informations les plus pertinentes dans les données. Nous aborderons la question de l'explicabilité des réseaux profonds, qui sont souvent vus comme des "boîtes noires" au fonctionnement opaque. Nous montrerons par exemple des techniques permettant de visualiser des cartes de saillance sur les données en entrée d'un réseau.

Les compétences acquises permettront d'identifier les problèmes tombant dans le périmètre du deep learning, de choisir et mettre en œuvre une architecture adéquate.

Contenu : (1200 caractères)

Le cours présente les concepts et les bases théoriques nécessaires à la compréhension de l'apprentissage profond, du point de vue algorithmique. En particulier, l'algorithme de rétropropagation du gradient sera central. La mise en œuvre pratique sur des applications et cas d'usage réalistes jouera également un rôle important.

Le cours s'articulera autour des points suivants :

Introduction à l'apprentissage profond : historique, exemples d'application, les différents types de réseaux de neurones.

Algorithme de rétro-propagation et différentiation automatique, règles d'actualisation des poids d'un réseau. Calcul "manuel" de l'apprentissage d'un réseau jouet, vectorisation des calculs. Calcul de gradients par différentiation "automatique".

Les réseaux convolutifs (CNN). Rappels sur les convolutions, couches de convolutions, convolutions à trous.

La modélisation de séquences avec les réseaux de neurones récurrents : RNN, LSTM, GRU. L'algorithme de rétropropagation à travers le temps.

Introduction aux réseaux de neurones sequence-to-sequence et aux mécanismes d'attention.

Introduction à l'explicabilité, techniques de détection de saillance. Limites et perspectives.

Pré-requis : (240 caractères)

Avoir suivi le cours "Apprentissage Automatique" du tronc commun. Connaissances en analyse (calcul de dérivées), algèbre linéaire (produit matriciel), probabilité.

Références bibliographiques : (320 caractères)

Neural Networks and Deep Learning By Michael Nielsen Online book, 2016
<http://neuralnetworksanddeeplearning.com/>
Programming PyTorch for Deep Learning: Creating and Deploying Deep Learning Applications, Ian Pointer, O'REILLY
L'apprentissage profond, [Ian Goodfellow](#), Yoshua Bengio, Aaron Courville, Quantmetry.

Des librairies standard d'apprentissage profond seront utilisées pour les travaux pratiques.

Mots-clés : (160 caractères)

Introduction à l'apprentissage profond. Algorithme de rétropropagation du gradient. Réseaux de neurones convolutifs. Réseaux de neurones récurrents.

Modalités d'évaluation : (240 caractères)

Les étudiants seront

CT : 70%

CCTP : 30%, note reportée à la session 2

Règle d'absence : règle 16, ABI -> note 0, ABJ -> coef 0

Syllabus version anglaise

Module Name : Artificial Intelligence 1 : Machine learning 1

Objectives and learning outcomes : (800 caractères)

This course provides an in-depth presentation of deep learning approaches, more specifically of deep neural networks. This paradigm is at the origin of major advances in many application domains: object recognition in images, super-resolution in medical imaging, automatic speech recognition, text generation, etc.

We will discuss the techniques for training these networks, with the back-prop algorithm and automatic differentiation, at the heart of current deep learning frameworks.

We will discuss standard architectures useful for different applications: fully-connected, convolutional, recurrent networks. Attention mechanisms will be discussed, which allow a model to learn how to weigh the most relevant information in the input data. We will address the question of the explicability of deep networks, which are often seen as "black boxes" with opaque functioning. For example, we will show techniques to visualize saliency maps on the input data given to a network.

The acquired skills will allow the students to identify problems falling within the scope of deep learning, to choose and implement adequate architectures.

Course content : (1200 caractères)

The course presents the concepts and theoretical foundations necessary to understand deep learning from an algorithmic perspective. In particular, the gradient backpropagation algorithm will be central. Practical implementation on realistic applications and use cases will also play an important role.

The course will be structured around the following points:

Introduction to deep learning: history, application examples, the different types of neural networks.

Back-propagation algorithm and automatic differentiation, update rules. Manual calculations for a toy network, vectorization. Calculation of gradients by means of "automatic" differentiation.

Convolutional networks (CNN). Convolution reminders, convolution layers, dilated convolutions.

Sequence modeling with recurrent neural networks: vanilla RNN, LSTM, GRU. The backpropagation through time algorithm.

Introduction to sequence-to-sequence neural networks and attention mechanisms.

Introduction to explicability, techniques of surveillance detection. Limits and perspectives of deep learning.

Prerequisite : (240 caractères)

Basic knowledge in linear algebra, probability and statistics, and machine learning.

Bibliography : (320 caractères)

- Neural Networks and Deep Learning By Michael Nielsen Online book, 2016

<http://neuralnetworksanddeeplearning.com/>

- Programming PyTorch for Deep Learning: Creating and Deploying Deep Learning Applications, Ian Pointer, O'REILLY
- L'apprentissage profond, [Ian Goodfellow](#), Yoshua Bengio, Aaron Courville, Quantmetry.

A standard deep learning library will be used in the practicals.

Keywords : (160 caractères)

Introduction to deep learning, backpropagation algorithm, convolutional neural networks, recurrent neural networks

Assessment methodology : (240 caractères)

Students will be evaluated with a written exam (CT) and a practical assignment (CCTP): 70% CT, 30% CCTP

0Syllabus version française

Nom du module :

Introduction au traitement du signal, aux signaux sonores et aux images (ITS)

Objectifs :

Ce cours doit permettre aux étudiants de découvrir le traitement du signal. Il sert de base pour appréhender la formation et l'acquisition des images, des signaux en général, de la parole et de la musique en particulier. Plusieurs notions de traitement de ces signaux sont également introduites.

Compétences

- Appréhender la formation et la représentation des signaux visuels et sonores.
- Comprendre les notions de signal numérique et analogique ainsi que les fréquences.
- Connaître les outils de représentation temporelle, spatiale et fréquentielle des images et du son.
- Savoir analyser un signal numérique par extraction de fréquences et interpréter en fonction du contenu (son et image).
- Concevoir un filtre répondant à une spécification donnée.
- Connaître les bases nécessaires à la manipulation élémentaire des images.
- Connaître les bases nécessaires pour analyser un contenu audio (parole, musique).

Contenu :

Cours-TD

Commentaire à enlever dans la version finale : Outils de traitement du signal

- Notions introductives sur le son et les images.
- Processus de numérisation sonore et visuel : échantillonnage, quantification, les notions d'échantillon sonore et de pixel.
- Notion de fréquence, transformée de Fourier (1D, 2D), illustrations sur les signaux mono-dimensionnels, le son et les images.
- Convolution discrète et filtrage linéaire. Isoler un son ou du bruit, filtrer une image.
- Ondelettes discrètes et décomposition dyadique. Notion de compression d'une image.
- Le bruit et son traitement.

Commentaire à enlever dans la version finale : Mise en œuvre des outils

- Application au traitement du son
 - o Analyses temporelle et fréquentielle.
- Application au traitement d'image
 - o Représentation des images et manipulations élémentaires des pixels.

Aspect pratique

- Découverte de la parole et de la musique à travers des traitements élémentaires.
- Découverte du traitement des images à travers les techniques d'amélioration.

Pré-requis :

Aucun

Références bibliographiques :

- Bellanger, M., "Traitement numérique du signal", 8ème éd., Dunod, 2006.
- R.C. Gonzales, E.R. Woods, Digital Image Processing, Prentice Hall, 2007.
- Rabiner, L.R.; Schafer, R.W. Theory and applications of digital speech processing. Prentice Hall, 2010.

Mots-clés :

Traitement du signal, son, images numériques, transformée de Fourier, fréquence, filtrage.

Modalités d'évaluation :

Contrôle terminal écrit de la partie théorique et contrôle continu des travaux pratiques.

Syllabus version anglaise

Module Name:

- Introduction to signal and image processing: application to sound signals

Objectives and learning outcomes:

The objective of this course is to initiate students to signal processing, and in particular to signal and image acquisition, sampling and basic processing. Applications to speech and music signals will be used to support the theoretical aspects.

Course content :

- Basics on sound signals and images.
- Sampling and quantification of signals and images.
- Frequency, 1D and 2D Fourier transforms.
- 1D and 2D convolution, linear filtering.
- Discrete wavelet transform, basics on images compression.
- Basics on noise affecting signals and images.

Prerequisite :

None

Bibliography :

- Bellanger, M., "Traitement numérique du signal", 8ème éd., Dunod, 2006.
- R.C. Gonzales, E.R. Woods, Digital Image Processing, Prentice Hall, 2007.
- Rabiner, L.R.; Schafer, R.W. Theory and applications of digital speech processing. Prentice Hall, 2010.

Key words :

- signal processing, sound, image, digital images, Fourier transform, filtering.

Assessment methodology :

Syllabus Statistiques version française

Nom du module :

- Statistiques

Objectifs :

L'objectif principal est d'acquérir les notions et propriétés essentielles dans les modèles déjà anciens de la statistique paramétrique ou non paramétrique.

Les incontournables sont :

- le modèle linéaire, le modèle de base pour la dépendance multivariée, qui sera généralisé en Master 2 par les méthodes récentes par pénalisation d'une classe de fonctions.
- les tests non-paramétriques basés sur les rangs, qui ne nécessitent que des hypothèses minimales sur la loi des données.
- l'étude des séries chronologiques, le modèle de base dans la recherche de tendances temporelles, avec un aperçu des modèles non stationnaires.
- Les différentes modélisations et méthodes présentées seront mises en oeuvre lors de TP sur R, Scilab, Matlab ou Python.

Contenu : 6 ECTS, 24h de CM, 24h de TD, 12h de TP

- Modèle linéaire, généralisations - 5 semaines.
Vecteurs gaussiens, transformations linéaires, théorème de Cochran. Modèle linéaire gaussien, estimation par moindres carrés, par maximum de vraisemblance. Coefficient de corrélations multiples. Tests de significativité (Student), test de comparaison de modèles emboîtés (Fisher), intervalles de prédiction. Analyse de la variance à un ou deux facteurs.
- Tests non-paramétriques - 3 semaines.
Rappels de dénombrement et probabilités discrètes, loi des rangs d'un échantillon, test de la somme des rangs de Wilcoxon.
- Séries temporelles - 4 semaines.
Tendance et composantes saisonnières d'une série chronologique. Stationnarité, estimation de la fonction d'auto-corrélation, tests du Portemanteau. Bruit blanc, élimination de la tendance, détection de ruptures. Séries stationnaires, auto-corrélation partielle (AR, ARMA).

Pré-requis :

- Notions de base en statistique mathématique (estimation et test), cours d'intégration et probabilités, algèbre linéaire.

Références bibliographiques :

- "Probabilités, analyse des données et statistique", Gilbert Saporta.
- "Le modèle linéaire par l'exemple", Jean-Marc Azais et Jean-Marc Bardet.
- <https://www.python.org/>

Mots-clés :

- Modèle linéaire, tests non-paramétriques, séries temporelles

Modalités d'évaluation :

- CCTP+CP+CT

Syllabus version anglaise

Module Name :

- Statistics (second semester)

Objectives and learning outcomes :

The principal objective is to acquire notions and crucial properties of popular models parametric and non-parametric statistics :

- The general linear model, the basic model for multivariate dependencies, which will be generalized in Master 2 by recent method about models penalized by a class of functions.
- non-parametric signed-rank tests, with minimal hypotheses on the distribution of the data.
- study of time series, the basic model for trend researchs, with a glimpse on models which are not stationnary.
- The various modelisations and methods presented will be illustrated during practical sessions (Travaux Pratiques) on R, Scilab, Matlab ou Python.

Course content : 6 ECTS, 24h CM, 24h TD, 12h TP

- The general linear model, and generalisations - 5 weeks.
Gaussians vectors, linear transforms, Cochran's theorem. General linear model, method of least squares, method of Maximum likelihood. regression coefficients in multiple regression. Significance testing (Student), testing nested models (Fisher), Prediction interval. Variance analysis with one or two factors.
- Non-parametric tests - 3 semaines.
Recall of enumerative combinatorics and discrete probability, distribution of order statistics, Wilcoxon rank-sum test.
- Time series - 4 semaines.
Trends and seasonal patterns of a time series. Stationnarity, estimation of the autocorrelation function, Portemanteau test. White noise, removal of an underlying trend, step detection. Stationnary series, partial autocorrelation (AR, ARMA).

Prerequisite :

- Basic knowledge in statistics (estimation and test), integration and probability, linear algebra.

Bibliography :

- "Probabilités, analyse des données et statistique", Gilbert Saporta.
- "Le modèle linéaire par l'exemple", Jean-Marc Azais et Jean-Marc Bardet.
- <https://www.python.org/>

Key words :

- Linear model, tests, time series

Assessment methodology :

- CCTP+CP+CT

IMAGE, SIGNAL, SIMULATIONS

LEARNING OBJECTIVES

The course presents an overview of the main image processing problems: restoration, segmentation, repackaging. We will also see the main models aiming at solving them, as well as the numerical strategies allowing to calculate a solution. The methods presented in the course will be illustrated by tutorials in Python.

COURSE CONTENT

Introduction and reminders

- Reminders on imaging: convolution, windowing, sampling.
- Optimization tools for imaging: gradient and proximal gradient algorithm.
- Inverse problems and ill-posedness.

Image restoration

- Introduction to image restoration: Bayes law and MAP estimators in imaging.
- The total variation in the continuous and discrete domains.
- Image denoising, operator inversion (deconvolution, inpainting, zoom), dequantification of images, restoration of compressed images.

Sparse representations in a dictionary of atoms

- Minimization of the L₀-norm, non-linear approximation, compressed sampling (case of the L₀-minimization), numerical resolution in the orthonormal case.
- Minimization of the L₁-norm, the compressed sampling (case of the L₁-minimization), numerical resolution by the proximal gradient algorithm.
- Example of a greedy algorithm: OMP.
-

Image segmentation

- Stochastic model of a form: perimeter of a discrete form.
- Object models: estimation of the laws of colors or characteristics of an object.
- The models of Mumford-Shah, Chan-Vese and Boykov-Jolly.

PREREQUISITES

Basic Fourier analysis, optimization, Bayes law, MAP and MMSE estimators, density estimation.

KEYWORDS

Imaging, deblurring, ill-posed problems, L₁-minimization, total variation, compressive sensing, image segmentation, shape optimization.

BIBLIOGRAPHY

O. Scherzer, M. Grasmair, H Grossauer, M. Haltmeier & F. Lenzen, *Variational Methods in Imaging*. Vol. 167. Springer Science & Business Media, 2008.

Syllabus English version anglaise

Name of module :

- English

Objectives and learning outcomes :

Level C1 of the CEFRL (Common European Framework of Reference for Languages)

- Develop the skills necessary to students to succeed in their future professional lives in a variety of cultural contexts.
- Acquire the necessary linguistic autonomy and improve the specialized language allowing professional integration and communication of scientific expertise in an international context.

Competences

Speak fluently in front of an audience, using registers adapted to different contexts and different interlocutors.

Use a foreign language other than French easily: written and oral comprehension and expression:

- Understand a scientific or professional article written in English on a subject related to their field.
- Interact orally in English in order to succeed in formal and informal exchanges during lectures, meetings or professional interviews.
- Defend your application in writing (CV) or orally (job interview) in English

Course Content

- Develop skills related to the understanding of scientific or professional publications written in English as well as skills necessary to understand oral scientific communications.
- Acquire expression tools in order to master an oral and/or written presentation and to tackle a critical discussion in the scientific field, (e.g. rhetoric, linguistic elements, pronunciation...) .
- Acquire the linguistic and cultural elements of professional communication (CV, cover letter, interview)

Prerequisite : Level B2 CEFRL

Key words

Project - Scientific English - Writing - Publishing - Communications - interculturality - professionalisation

Assessment methodology : Continuous assessment: