

Interactions de l'Informatique et des Mathématiques pour l'IA (M1 IMA)

Nom de l'UE	UE Obligatoire ou Optionnelle	Dept.	ECTS							
				H CM	H CTDI	H TD	H TD	LV	H TP	H projet
SEMESTRE IMPAIR										
Mise à niveau	Option	Inf.	0	0	24	0	0	0	0	0
Théorie des Langages (IAFA)	O	Inf.	6	22		14		18		
Algorithmique (IAFA)	O	Inf.	6	20		24		10		
Optimisation (MAPI3)	O	Math.	6							
Probabilités (MAPI3)	O	Math.	6							
Simulation aléatoire (MAPI3)	O	Math.	6							
SEMESTRE PAIR										
LANGUE VIVANTE	O	LVG	3	0	0	0	24	0	0	0
Intelligence Artificielle 1 (IAFA)	O	Inf.	6	24		24		12		
Traitement de données 1 (IAFA)	O	Inf.	6	28		12		20		
Initiation à la recherche	O	ANITI	3	18					1	
Statistiques (MAPI3)	O	Math.	6							
Image-Signal-Simulation (MAPI3)	O	Math.	6							

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-compileur 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-compileur 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, EJCPC, 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%)

Nom du module :

- Algorithmique

Objectifs :

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.

Les compétences visées incluent :

- Décrire et Implémenter les opérations manipulant des arbres de recherche et des tas, établir leur complexité temporelle et spatiale
- Caractériser un problème d'optimisation combinatoire
- Formaliser un problème spécifié en langage naturel sous la forme d'un problème de décision (en SAT ou CSP) ou d'optimisation (en PL)
- Implémenter et utiliser un algorithme de résolution de Flots, de SAT, de CSP, de PL
- Choisir et implémenter un algorithme approprié de recherche locale

Les compétences visées incluent :

- Complexité des problèmes, algorithmes DPLL, algorithme Ford-Fulkerson, algorithme du Simplex, notion de recherche locale.

Contenu :

A) Rappels de Complexité des Algorithmes

- Interet, concepts, notations, exemples
- Exemples complexité algos de recherche ou de tri avec ABR, ARN, AVL, Tas binaire

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)

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

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)

Plan des Tps : implémentation solveur SAT, PLNE, Meta-Heuristiques

Pré-requis :

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

Références bibliographiques :

- 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

Mots-clés :

- 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 :

- 20% 1 CCTP correspondant à 3 ou 4 Epreuves de TP avec R16(abs-> 0)
- 30% 1 CP écrit : 1^e partie du cours avec R16
- 50% 1 CT écrit : reste du cours
- En 2^e session: 1 CT de 100% (pas de report)

Nom du module :

- Optimisation

Objectifs :

L'optimisation joue un rôle fondamental dans la conception, la production et la gestion des biens et des services. Les domaines d'application de l'optimisation sont extrêmement variés. On peut citer comme exemples : la forme d'un objet, le rendement d'un appareil, le fonctionnement d'un moteur, le contrôle de l'espace aérien, le choix des investissements économiques, etc. Cet enseignement a pour objectif d'introduire les principaux algorithmes d'optimisation (sans contraintes ou avec contraintes) en se basant sur l'étude mathématique des conditions d'optimalité. Les méthodes décrites en cours seront illustrées par des TP en Python.

Contenu :

Introduction et définitions: Structure générale d'un problème d'optimisation. Exemples. Contraintes fonctionnelles de type égalités et inégalités. Classification des problèmes d'optimisation. Définition de minimum local, global. Structure générale des algorithmes d'optimisation.

Analyse convexe: Ensembles convexes et fonctions, intérieur relatif, théorème de séparation, domaine et épigraphe, dérivée directionnelle, semicontinuité inférieure, sous-gradient et sous différentielle, fonction conjuguée, Dualité de Fenchel.

Conditions d'optimalité: Résultats d'existence, extrema avec et sans contrainte, condition nécessaire d'optimalité, suffisantes, règle de Fermat, conditions de KKT, rôle de la convexité, points selles et dualité Lagrangienne.

Algorithmes:

- Unconstrained optimization: descent algorithms, gradient method, line search, Wolfe conditions, the Newton algorithm, quasi-Newton algorithm (in particular BFGS algorithm).
- optimisation sans contraintes : méthodes de descente, méthodes de gradient, méthodes de recherche linéaire, critères de Wolfe, d'Armijo, méthode de Newton, méthodes de quasi-Newton (BFGS).
- Constrained optimization: projected gradient, dual methods, the Uzawa algorithm, augmented Lagrangians.
- Algorithmes d'optimisation avec contraintes : méthode du gradient projeté, méthodes duales , méthode d'Uzawa, méthode SQP, Lagrangien augmenté.

Pré-requis :

Calcul différentiel: gradients, matrice Hessienne, développement de Taylor.

Références bibliographiques :

- |M. Bierlaire, Introduction a l'optimisation dierentiable.
- |J. Nocedal & S. J. Wright, Numerical optimization.
- |D. G. Luenberger & Y. Ye, Linear and nonlinear programming.

Mots-clés :

- Condition d'optimalité, méthode du gradient, algorithme d'optimisation, analyse de convergence, dualité de Fenchel et dualité Lagrangienne.

Modalités d'évaluation :

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 Z , Z^2 , processus de Galton-Watson...
- Processus de Poisson
- Processus de Bernoulli. Processus de Poisson homogène sur 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

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 :

Nom du module :

Intelligence Artificielle 1 :

Partie Apprentissage automatique**Objectifs :**

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 :

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 :

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

Références bibliographiques :

- 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 bibliothèques standard d'apprentissage profond seront utilisées pour les travaux pratiques.

Mots-clés :

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

Modalités d'évaluation :

Les étudiants seront

- CT : 70%
- CCTP : 30%, note reportée à la session 2

Partie Raisonnement sur des connaissances hiérarchiques ou imparfaites**Objectifs :**

Pour modéliser des systèmes complexes capables de raisonner et de décider, nous devons disposer des méthodes et outils adéquats. L'objectif est donc de présenter des méthodes de représentation et de traitement de connaissances certaines et organisées, ou imprécises et/ou incertaines et/ou inconsistantes, à savoir le formalisme de la logique de description, celui de la logique possibiliste et les mécanismes de gestion de l'inconsistance.

- Identifier la (ou les) compétence(s) attestée(s) après validation du module
- Mettre en relation une catégorie de connaissances (parfaite ou imparfaite) avec le formalisme adapté
- Mettre en œuvre un raisonnement avec des connaissances ontologiques, incertaines ou inconsistantes
- Identifier la (ou les) connaissances acquises(s) après validation du module
- Bases de la logique des description
- Mesures de Nécessité, de Possibilité, notion d'inconsistance
- Mécanismes d'inférence pour les différents formalismes

Contenu :

Partie I. Représenter et raisonner sur des connaissances certaines

2h C, 8h TD, 0 TP

- Cas d'usage adaptés à chaque mineure, lien entre logiques de description et ontologies/graphes conceptuels (pourquoi et comment on applique ces modes de représentation aux pbs des ontologies) (2h),
- présentation du langage (4h)
- Les pbs d'efficacité de raisonnement (complexité) suite à la logique de description utilisée. Inférence avec les logiques de description (4h)

Partie II. Représenter et raisonner sur les connaissances incertaines/inconsistantes

8h C, 8 TD, 2h TP

- Introduction (imprécision, vague, mesures incertain, rappels proba, fonctions de croyance) 2h C 2h TD (rappels probabilités).
- Possibilités: raisonnement en logique possibiliste 4hC + 4hTD + 2h TP
- Gestion de l'inconsistance dans les bases de connaissances/bases de données 2hC + 2hTD

TP de 2h sur logique possibiliste avec Sat4J.

Pré-requis :

Formalismes logiques (logique des propositions et logique des prédicats).

Théorie des graphes. Notions élémentaires en théorie des probabilités

Références bibliographiques :

- Panorama de l'Intelligence Artificielle, vol 1. P Marquis, O Papini, H Prade. 2014. Cépaduès
- A Guided Tour of Artificial Intelligence Research. Vol I: Knowledge Representation, Reasoning and Learning. P Marquis, O Papini, H Prade (Eds.). 2020. Springer.

<https://www.sat4j.org/>

Mots-clés :

Représentation de connaissances, Logique de description, Incertitude, Logique possibiliste, Inconsistance, Intelligence Artificielle.

Modalités d'évaluation :

CCC : 30%, R16, pas de report

CT : 70%, écrit 2h

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
 - Analyses temporelle et fréquentielle.
- Application au traitement d'image
 - 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.

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

Nom du module :

- Image, signal, simulations.

Objectifs :

Le cours présente un panorama des principales problématiques du traitement d'images : la restauration, la segmentation, le recallage. Nous verrons aussi les principaux modèles visant à les résoudre, ainsi que les stratégies numériques permettant d'en calculer une solution. Les méthodes décrites en cours seront illustrées par des TP en Python.

Contenu :

Introduction - Rappels

- Rappels sur l'imagerie : convolution, fenêtrage, échantillonnage.
Rappels d'optimisation en imagerie : algorithme du gradient et du gradient proximal.
- Problèmes inverses.

La restauration d'images

- Introduction à la restauration d'images : loi de Bayes et estimateurs MAP en imagerie.
- La variation totale dans le domaine continu et discret.

Le débruitage d'images, l'inversion d'opérateurs (déconvolution, désoccultation, zoom), la déquantification d'images, la restauration d'images compressées.

Les représentations parcimonieuses dans un dictionnaire d'atomes

- La minimisation de la "norme" L_0 , approximation non-linéaire, l'échantillonnage compressé (cas de la minimisation L_0), résolution numérique dans le cas orthonormée.
- La minimisation de la norme L_1 , l'échantillonnage compressé (cas de la minimisation L_1), résolution numérique par l'algorithme du gradient proximal.
- Exemple d'algorithme glouton : OMP.

La segmentation d'images

- Modèle stochastique de forme : périmètre d'une forme discrète.
- Modèles d'objets : estimation des lois des couleurs ou de caractéristiques d'un objet.
Les modèles de Mumford-Shah, Chan-Vese et Boykov-Jolly.

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

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