



Département d'informatique
IFT 603/IFT 712 — Techniques d'apprentissage
Plan d'activité pédagogique
Hiver 2019

Enseignant :

Nom de l'enseignant : Pierre-Marc Jodoin
Courriel : pierre-marc.jodoin@USherbrooke.ca
Local : D4-1016-1
Téléphone : 819-821-8000 (62025)
Site : <http://info.usherbrooke.ca/pmjodoin/cours/IFT603/index.html>
Disponibilité : à déterminer

Responsable(s) : Pierre-Marc Jodoin et Shengrui Wang**Horaire :**

Exposé magistral :	Mardi 10 h 30 à 12 h 20	Local D4-2021
Vendredi	13 h 30 à 14 h 20	Local D4-2021

Description officielle de l'activité pédagogique¹

Cibles de formation	S'initier aux techniques d'apprentissage automatique en intelligence artificielle.
Contenu	Concepts fondamentaux de l'apprentissage automatique, formulation mathématique probabiliste, régression linéaire, classification linéaire, astuce du noyau, machine à vecteurs de support, apprentissage bayésien, modèle de mélange de gaussiennes, algorithme EM, réduction de dimensionnalité et combinaison de modèles.
Crédits	3
Organisation	3 heures d'exposé magistral par semaine 6 heures de travail personnel par semaine
Préalable(s)	IFT 436 et STT 418
Particularités	Aucune

¹ <https://www.usherbrooke.ca/admission/fiches-cours/IFT603/>

1 Présentation

Cette section présente les objectifs spécifiques et le contenu détaillé de l'activité pédagogique. Cette section, non modifiable sans l'approbation d'un comité de programme du Département d'informatique, constitue la version officielle.

1.1 Mise en contexte

L'apprentissage automatique ou l'apprentissage par machine (Machine Learning) s'intéresse à la conception, l'analyse, l'implémentation et l'application de programmes d'ordinateur capables de s'améliorer, au fil du temps, soit sur la base de leur propre expérience, soit à partir des données antérieures fournies par d'autres programmes. De nos jours, l'apprentissage automatique joue un rôle essentiel dans de nombreux domaines d'applications, tels que la vision par ordinateur, le traitement automatique du langage, la reconnaissance vocale, les systèmes tutoriels intelligents, la modélisation de l'utilisateur, la robotique, la bio-informatique, les finances, le marketing, les jeux vidéo, la télédétection, etc. En fait, la plupart des programmes de l'intelligence artificielle contiennent un module d'apprentissage. Presque tous les systèmes de reconnaissances de formes sont basés sur des techniques d'apprentissage.

1.2 Cibles de formation spécifiques

Ce cours vise à introduire les concepts fondamentaux de l'apprentissage automatique et à présenter une vaste gamme d'approches différentes utilisées dans des systèmes d'apprentissage automatique :

1. Comprendre les notions fondamentales de l'apprentissage automatique ;
2. Comprendre la formulation probabiliste de l'apprentissage automatique ;
3. Maîtriser des algorithmes de régression et classification en apprentissage supervisé ;
4. Comprendre comment utiliser l'astuce du noyau ;
5. Maîtriser les fondements des réseaux de neurones et leur apprentissage par rétro propagation ;
6. Maîtriser des algorithmes de combinaison de modèles.

1.3 Contenu détaillé

Thème	Contenu	Heures	Objectifs
1	Concepts fondamentaux : <ul style="list-style-type: none"> - apprentissage supervisé vs non-supervisé - performance d'entraînement vs. généralisation - régularisation - sélection d'hyper-paramètres, validation croisée - exemple de la régression polynomiale 	3	1
2	Formulation probabiliste : <ul style="list-style-type: none"> - théorie des probabilités (cas discret et continu) - théorie de l'information (entropie, information mutuelle, divergence de Kullback-Leibler) - Bayes et probabilités jointes, conditionnelles, a priori - Introduction au maximum de vraisemblance - Introduction au maximum a posteriori 	3	1, 2
3	Régression linéaire : <ul style="list-style-type: none"> - Formulation probabiliste de la régression polynomiale - apprentissage par maximum de vraisemblance - apprentissage par maximum a posteriori 	3	3
4	Classification linéaire : <ul style="list-style-type: none"> - Classification comme un problème de régression - Méthode de Fisher 2 classes 	7	1, 3, 5

	<ul style="list-style-type: none"> - Méthodes génératives <ul style="list-style-type: none"> ▪ Perceptron binaire vs. Multiclasse ▪ Régression logistique binaire vs multiclasse ▪ descente de gradient ▪ Sigmoid vs softmax ▪ Distribution de Bernouilli vs entropie croisée 		
5	Méthode à noyau : <ul style="list-style-type: none"> - Représentation duale et régression à noyau - Construction de noyau - Régression à noyau 	3	4
6	Méthode à noyau parcimonieuse : <ul style="list-style-type: none"> - Classifieur à marge maximale - Machine à vecteurs de support (SVM) 	3	3, 4
7	Réseaux de neurones multi-couches <ul style="list-style-type: none"> - Approche linéaire vs non linéaire - Classification binaire et multiclassés - Fonction d'activation - Couche d'entrée vs couche cachée vs couche de sortie - Rétro propagation - Initialisation d'un réseau - <i>Sanity checks</i> et vectorisation 	6	5
8	Combinaison de modèles : <ul style="list-style-type: none"> - <i>Boostrap</i> - <i>Boosting</i> 	3	6
9	Mélange de gaussiennes : <ul style="list-style-type: none"> - Partitionnement de données (<i>clustering</i>) - Algorithme EM - Fenêtre de Parzen - Spectral Clustering 	5	2
10	Théorie de la décision (si le temps le permet) : <ul style="list-style-type: none"> - Décomposition biais-variance - Théorie de la décision 	3	1, 2, 3

2 Organisation

Cette section propre à l'approche pédagogique de chaque enseignante ou enseignant présente la méthode pédagogique, le calendrier, le barème et la procédure d'évaluation ainsi que l'échéancier des travaux. Cette section doit être cohérente avec le contenu de la section précédente.

2.1 Méthode pédagogique

Une semaine comprend trois heures de cours magistraux. La plupart des présentations en classe se feront à l'aide de diapositives disponibles sur le site du cours au format pdf. Tout au long de la session, l'étudiant devra implémenter les notions vues en classe à travers cinq travaux pratiques. Des instructions particulières seront données pour chacun des travaux et les périodes de consultation seront déterminées en classe. En ce qui concerne les examens, seules des notes manuscrites non photocopiées sont admises, tout manuel et notes de cours imprimées étant interdits. L'examen final est récapitulatif.

2.2 Calendrier du cours approximatif (heures au lab à confirmer en classe)

	Semaine du	Chapitre	Travaux
1	7 janvier	1	
2	14 janvier	2	
3	21 janvier	3	
4	28 janvier	4 (1 h au lab)	Remise tp1
5	4 février	4	
6	11 février	4-5	
7	18 février	5 (1 h au lab + 1 heure de révision)	Remise tp2
8	25 février	Examen périodique (date à déterminer)	
9	4 mars	Relâche des activités pédagogiques	
10	11 mars	5-6	
11	18 mars	7	
12	25 mars	7	Remise tp3
13	1 avril	8	
14	8 avril	9	Remise tp4
	XYZ avril	Examen final (date à déterminer)	

2.3 Évaluation

IFT601

Devoirs (4) :	40 %
Examen intra :	20 %
Examen final :	40 %

IFT712

Devoirs (4) :	30 %
Examen intra :	15 %
Examen final :	30 %
Projet de session :	25 %

Projet de session – IFT712

Les étudiantes et étudiants inscrits au cours IFT 712 sont tenus de faire un projet de session en python seuls.es ou par équipe de deux. Le projet a pour objectif de tester au moins **six méthodes** de classification sur une base de données Kaggle (www.kaggle.com) et avec la bibliothèque Sklearn (<https://scikit-learn.org>). Les équipes sont libres de choisir la base de données de leur choix, mais une option simple est celle du challenge de classification de feuilles d'arbres (www.kaggle.com/c/leaf-classification). Pour ce projet, on s'attend à ce que les bonnes pratiques de cross-validation et de recherche d'hyper-paramètres soient mises de l'avant pour identifier la meilleure solution possible pour résoudre le problème.

Le barème de correction est le suivant :

Qualité du code / commentaires	/10
Choix de design	/10
Gestion de projet (Git - Trello)	/10
Rapport	/70
Démarche scientifique	/50

Analyse des résultats	/20
-----------------------	-----

Qualité du code et commentaires : on vous demande du code rédigé 100% en anglais ou en français, mais pas de franglais! Votre code doit aussi être bien documenté, respecter le standard pep8 et respecter un standard uniforme pour la nomenclature des variables, des noms de fonctions et des noms de classes. Évitez également les variables « hardcodées » empêchant l'utilisation de votre programme sur un autre ordinateur que le vôtre.

Choix de design : vous devez organiser votre code de façon professionnelle. Pour ce faire, on s'attend à une hiérarchie de classes cohérente, pas seulement panoplie de fonctions disparates. Aussi, du code dans un script « qui fait tout » se verra automatiquement attribuer la note de zéro. Bien que non requis, on vous encourage à faire un design de classes avant de commencer à coder. Aussi, le code, les données et la documentation doivent être organisés suivant une bonne structure de répertoires. Pour vous aider, vous pouvez utiliser le projet « cookiecutter » (<https://github.com/audreyr/cookiecutter>). La solution proposée doit aussi être facile à utiliser.

Gestion de projet : comme tout projet qui se respecte, vous devez utiliser un gestionnaire de version de code. On vous demande d'utiliser « git » via la plateforme « gitHub » ou « bitBucket ». On s'attend également à ce que vous fassiez une bonne utilisation de git. Par exemple : évitez de « pousser » du code dans le master sans *merge*, éviter les « méga » commits, etc. On s'attend à ce que vous utilisiez Trello pour gérer votre projet à haut niveau.

Démarche scientifique : pour ce volet, vous devez vous poser les questions suivantes : avez-vous bien « cross-validé » vos méthodes? Avez-vous bien fait votre recherche d'hyper-paramètres? Avez-vous entraîné et testé vos méthodes sur les mêmes données? Est-ce que cela transparaît dans le rapport? Avez-vous uniquement utilisé les données brutes ou vous avez essayé de les réorganiser pour améliorer vos résultats? Etc.

Analyse des résultats : que vos résultats soient bons ou non, vous devez en produire une analyse cohérente. Vous devez également analyser les résultats de test tel qu'obtenus par le système en ligne.

NOTE IMPORTANTE : Veuillez noter que jusqu'à 30% peuvent être retirés pour toute faute aberrante. Par exemple, une équipe qui se refuserait d'utiliser git pourrait perdre jusqu'à 40 points (10 pour la gestion de projet et 30 pour faute aberrante).

Conformément au règlement facultaire d'évaluation des apprentissages², l'enseignant peut retourner à l'étudiante ou à l'étudiant tout travail non conforme aux exigences quant à la qualité de la langue et aux normes de présentation.

Le plagiat consiste à utiliser des résultats obtenus par d'autres personnes afin de les faire passer pour sien et dans le dessein de tromper l'enseignant. Si une preuve de plagiat est attestée, elle sera traitée en conformité, entre autres, avec l'article 9.4.1 du Règlement des études³ de l'Université de Sherbrooke. L'étudiant ou l'étudiante peut s'exposer à de graves sanctions, dont automatiquement une note de zéro (0) au devoir ou à l'examen en question.

Ceci n'indique pas que vous n'avez pas le droit de coopérer entre deux équipes tant que la rédaction finale des documents et la création du programme restent le fait de votre équipe. En cas de doute de plagiat, l'enseignant peut demander à l'équipe d'expliquer les notions ou le fonctionnement du code qu'il considère comme étant plagié. En cas de doute, ne pas hésiter à demander conseil et assistance à l'enseignant afin d'éviter toute situation délicate par la suite.

2.4 Échéancier des travaux

TP	Thème	Pondération	Remise
1	Régression linéaire (ch. 3)	10 % (7.5%)	1 ^{er} février
2	Classification linéaire (ch.4)	10 % (7.5%)	22 février
3	Méthodes à noyau – classification non-linéaire (ch.5)	10 % (7.5%)	29 mars
4	Réseaux de neurones (ch.7)	10 % (7.5%)	12 avril
	Projet IFT712	25 %	12 avril

²https://www.usherbrooke.ca/sciences/fileadmin/sites/sciences/Etudiants_actuels/Informations_academiques_et_reglements/2017-10-27_Reglement_facultaire_-_evaluation_des_apprentissages.pdf

³<https://www.usherbrooke.ca/registraire/droits-et-responsabilites/reglement-des-etudes/>

Directives particulières

- Les devoirs doivent être effectués **par équipe de 2 ou de 3**;
- L'implémentation d'algorithmes dans le cadre des devoirs doit se faire dans le langage de programmation Python. Le code soumis doit être compatible avec (c'est-à-dire exécutable sous) la version de Python installée dans les laboratoires, sous Ubuntu;
- La qualité du français et de la présentation peut être considérée lors de l'évaluation des travaux;
- Toute soumission en retard implique une perte de 10 points par jour (incluant samedi et dimanche), sauf celles motivées par des raisons valables et conformes au règlement des études (par exemple, maladie avec attestation d'un médecin).

2.5 Utilisation d'appareils électroniques et du courriel

Selon le règlement complémentaire des études, section 4.2.3⁴, l'utilisation d'ordinateurs, de cellulaires ou de tablettes pendant une prestation est interdite à condition que leur usage soit explicitement permis dans le plan de cours.

Dans ce cours le règlement 4.2.3 s'applique à moins d'avoir obtenu personnellement l'autorisation du professeur. Cette permission peut être retirée en tout temps, si l'appareil n'est pas utilisé uniquement à des fins d'apprentissage.

Comme indiqué dans le règlement universitaire des études, section 4.2.3⁵, toute utilisation d'appareils de captation de la voix ou de l'image exige la permission du professeur.

NOTE : Je ne répond pas aux questions par courriel. **Pour toute question, veuillez passer à mon bureau.**

3 Matériel pour le cours

3.1 Manuel (fortement suggéré, mais non obligatoire pour réussir le cours)

- Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

Disponible en ligne sur Amazon, à la bibliothèque et en format pdf sur le site du cours.

3.2 Ressources en ligne

- Plan de cours
- Diapositives (PDF)

⁴

https://www.usherbrooke.ca/sciences/fileadmin/sites/sciences/documents/Intranet/Informations_academiques/Sciences_Reglement_complementaire_2017-05-09.pdf

⁵<https://www.usherbrooke.ca/registraire/droits-et-responsabilites/reglement-des-etudes/>



L'intégrité intellectuelle passe, notamment, par la reconnaissance des sources utilisées. À l'Université de Sherbrooke, on y veille !

Extrait du Règlement des études (Règlement 2575-009)

9.4.1 DÉLITS RELATIFS AUX ÉTUDES

Un délit relatif aux études désigne tout acte trompeur ou toute tentative de commettre un tel acte, quant au rendement scolaire ou une exigence relative à une activité pédagogique, à un programme ou à un parcours libre.

Sont notamment considérés comme un délit relatif aux études les faits suivants :

- a) commettre un plagiat, soit faire passer ou tenter de faire passer pour sien, dans une production évaluée, le travail d'une autre personne ou des passages ou des idées tirés de l'œuvre d'autrui (ce qui inclut notamment le fait de ne pas indiquer la source d'une production, d'un passage ou d'une idée tirée de l'œuvre d'autrui) ;
- b) commettre un autoplagiat, soit soumettre, sans autorisation préalable, une même production, en tout ou en partie, à plus d'une activité pédagogique ou dans une même activité pédagogique (notamment en cas de reprise) ;
- c) usurper l'identité d'une autre personne ou procéder à une substitution de personne lors d'une production évaluée ou de toute autre prestation obligatoire ;
- d) fournir ou obtenir toute aide non autorisée, qu'elle soit collective ou individuelle, pour une production faisant l'objet d'une évaluation ;
- e) obtenir par vol ou toute autre manœuvre frauduleuse, posséder ou utiliser du matériel de toute forme (incluant le numérique) non autorisé avant ou pendant une production faisant l'objet d'une évaluation ;
- f) copier, contrefaire ou falsifier un document pour l'évaluation d'une activité pédagogique ;

[...]

Par plagiat, on entend notamment :

- Copier intégralement une phrase ou un passage d'un livre, d'un article de journal ou de revue, d'une page Web ou de tout autre document en omettant d'en mentionner la source ou de le mettre entre guillemets ;
- reproduire des présentations, des dessins, des photographies, des graphiques, des données... sans en préciser la provenance et, dans certains cas, sans en avoir obtenu la permission de reproduire ;
- utiliser, en tout ou en partie, du matériel sonore, graphique ou visuel, des pages Internet, du code de programme informatique ou des éléments de logiciel, des données ou résultats d'expérimentation ou toute autre information en provenance d'autrui en le faisant passer pour sien ou sans en citer les sources ;
- résumer ou paraphraser l'idée d'un auteur sans en indiquer la source ;
- traduire en partie ou en totalité un texte en omettant d'en mentionner la source ou de le mettre entre guillemets ;
- utiliser le travail d'un autre et le présenter comme sien (et ce, même si cette personne a donné son accord) ;
- acheter un travail sur le Web ou ailleurs et le faire passer pour sien ;
- utiliser sans autorisation le même travail pour deux activités différentes (autoplagiat).

Autrement dit : mentionnez vos sources