

Advanced Machine Learning and Deep Learning

Consignes pour l'examen

Prof. Xavier Siebert et Prof. Sidi Ahmed Mahmoudi

2020-2021

L'évaluation de l'UE « Advanced Machine Learning and Deep Learning » se fera cette année sur base d'un projet en deux parties à réaliser par groupes de 3 étudiants maximum.

1. partie **AML (Advanced Machine Learning)** : un projet à choisir parmi ceux figurant dans ce document.
2. partie **ADL (Advanced Deep Learning)** : un projet à choisir parmi ceux dans le document annexe.

Comme expliqué au cours, il s'agit de combiner un projet **théorique** pour l'une des parties (**AML** ou **ADL**, au choix) avec un projet **pratique** pour l'autre.

- Veuillez nous communiquer par email pour le 18/12/2020
 - les membres du groupe (3 maximum)
 - 2 choix de combinaisons de projets AML-ADL. Nous essayerons de vous donner votre premier choix. Par exemple :
 1. premier choix : le projet 3-théorique pour AML combiné avec le projet 2-pratique pour ADL.
 2. deuxième choix : le projet 1-pratique pour AML combiné avec le projet 4-théorique pour ADL.
- Chaque partie (AML et ADL) fera l'objet d'un rapport de 10 pages maximum. Les codes de la partie pratique devront également être fournis. Le travail final est à remettre pour le **15 janvier 2021** au plus tard sur la boîte à dépôt sur la page **Moodle** du cours **Advanced Machine Learning**.

Les projets proposés pour la partie **Advanced Machine Learning** sont décrits dans ce document. Quelques références sont données, vous en trouverez facilement d'autres. N'oubliez pas de citer les références utilisées dans le rapport.

1 SLT1 - Méthode Adaboost

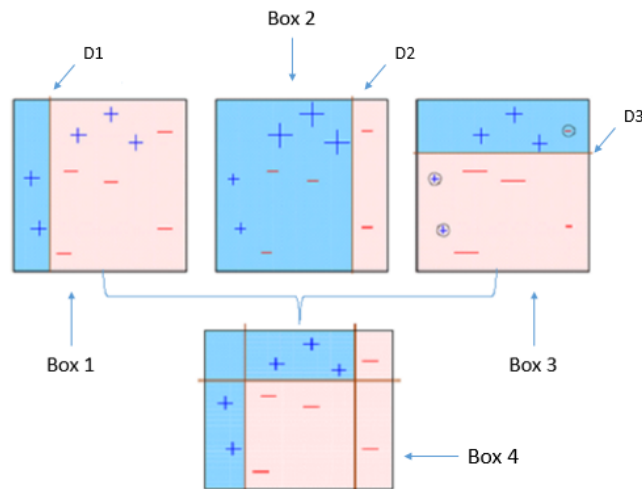


FIGURE 1 – Illustration de l'algorithme Adaboost (source : [6])

1. Partie théorique

- (a) Expliquez le principe du *boosting* et la méthode **Adaboost**.
- (b) Expliquez comment on obtient les bornes sur l'erreur de généralisation pour **Adaboost**

2. Partie pratique

- (a) Pour un jeu de données synthétique et un jeu de données réel, générer les courbes d'apprentissage avec la méthode **Adaboost**, en considérant comme classifieurs faibles
 - les classifieurs linéaires,
 - les arbres de décision.
 - (b) Discuter les résultats :
 - La courbe empirique obtenue est-elle en accord avec la théorie ?
 - Les prédictions théoriques relatives sur les différents classifieurs reflètent-elles l'efficacité d'un classifieur par rapport à un l'autre ?
- Références : Chapitre 10 de [1]
 - Données : [8]

2 SLT2 - Entropie de Vapnik-Chervonenkis

1. Partie Théorique

- (a) Expliquer l'utilisation de l'entropie de Vapnik-Chervonenkis. dans l'obtention de bornes de l'erreur de classification.
- (b) Comparer avec les bornes qui utilisent la dimension de Vapnik-Chervonenkis.

2. Partie Pratique

- (a) Pour un jeu de données synthétique et un jeu de données réel, générer les courbes d'apprentissage pour
 - un classifieur SVM
 - un classifieur k NN
 - (b) Comparer avec les bornes théoriques.
 - avec la dimension de Vapnik-Chervonenkis
 - avec l'entropie de Vapnik-Chervonenkis
 - (c) Discuter les résultats.
 - les prédictions théoriques absolues sont-elles proches des valeurs expérimentales ?
 - les prédictions théoriques relatives sur les différents classifieurs reflètent-elles l'efficacité d'un classifieur par rapport à un autre ?
- Références : [2]
 - Données : [8]

3 SLT3 - Variables de Rademacher

1. Partie Théorique

- (a) Expliquer l'utilisation des variables de Rademacher dans l'obtention de bornes de l'erreur de classification.
- (b) Comparer avec les bornes qui utilisent la dimension de Vapnik-Chervonenkis.

2. Partie Pratique

- (a) Pour un jeu de données synthétique et un jeu de données réel, générer les courbes d'apprentissage pour
 - un classifieur SVM
 - un classifieur k NN
 - (b) Comparer avec les bornes théoriques.
 - avec la dimension de Vapnik-Chervonenkis
 - obtenues grâce aux variables de Rademacher
 - (c) Discuter les résultats.
 - les prédictions théoriques absolues sont-elles proches des valeurs expérimentales ?
 - les prédictions théoriques relatives sur les différents classifieurs reflètent-elles l'efficacité d'un classifieur par rapport à un autre ?
- Références : [2, 7]
 - Données : [8]

4 SLT4 - Inégalités de Hoeffding et de Bernstein

1. Partie théorique

- (a) Démontrer en détails l'inégalité de Hoeffding pour les sommes de variables aléatoires bornées sur $[a, b]$
- (b) Enoncer l'inégalité de Bernstein. Quel est son intérêt par rapport à l'inégalité de Hoeffding ?
- (c) Donner un exemple d'application de l'inégalité de Bernstein.

2. Partie pratique : pas de partie pratique pour ce sujet

- Références : [2]

5 AL1 - Algorithme A^2

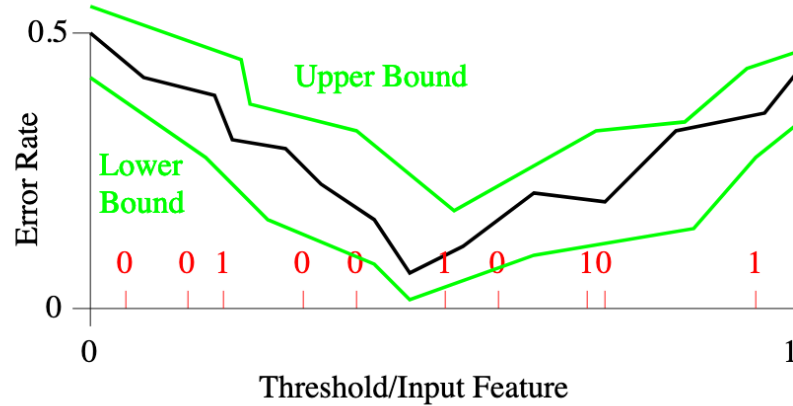


FIGURE 2 – Illustration de l'algorithme A^2 (source : [4])

1. Partie Théorique :

- Expliquer le fonctionnement de l'algorithme A^2
- Quelles sont les garanties théoriques sur le fonctionnement de cet algorithme et comment les obtient-on ?

2. Partie Pratique :

- Implémenter l'algorithme A^2 sur un jeu de données synthétique et un jeu de données réel
- Commenter les résultats.
 - l'algorithme est-il conforme aux prédictions théoriques ?
 - l'apprentissage actif avec A^2 est-il avantageux par rapport à un apprentissage « passif » ?
 - Quelles sont les possibles limitations de l'algorithme ?

- Références : [4]
- Données : [8]

6 AL2 - Algorithme IWAL

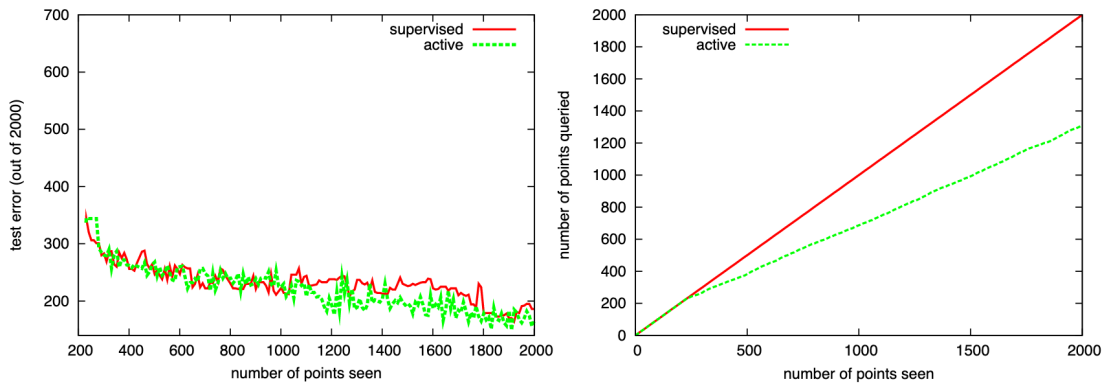


FIGURE 3 – Courbes d'apprentissage (gauche) et nombre de points choisis (droite) pour l'algorithme IWAL (source : [3])

1. Partie Théorique :

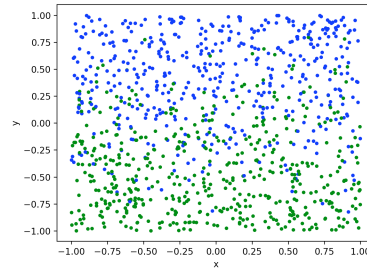
- Expliquer le fonctionnement de l'algorithme IWAL.
- Quelles sont les garanties théoriques sur le fonctionnement de cet algorithme et comment les obtient-on ?

2. Partie Pratique :

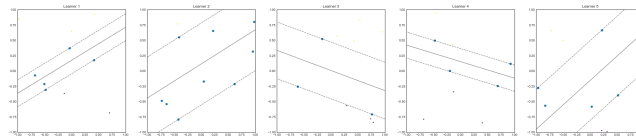
- Implémenter l'algorithme IWAL sur un jeu de données synthétique et un jeu de données réel
 - Commenter les résultats.
 - l'algorithme est-il conforme aux prédictions théoriques ?
 - l'apprentissage actif avec A^2 est-il avantageux par rapport à un apprentissage « passif » ?
 - Quelles sont les possibles limitations de l'algorithme ?
- Références : [3]
 - Données : [8]

7 AL3 - Apprentissage actif par comité avec abstention

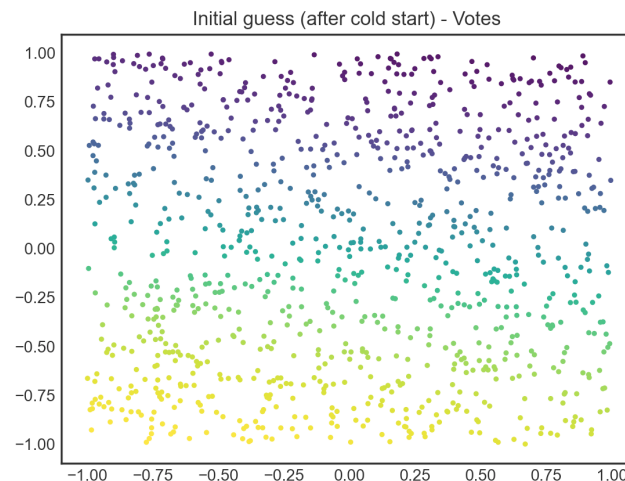
1. Partie Théorique : pas de partie théorique pour ce sujet
2. Partie Pratique : Implémenter l'algorithme QBC (vu au cours) pour une classification binaire, en y incorporant une possibilité pour l'oracle de s'abstenir si l'incertitude sur le label des points est trop grande, autrement dit :
 - (a) Générer deux groupes de 500 points chacun, de sorte que chaque groupe corresponde à une classe, et que les classes se chevauchent



- (b) Construire un comité de 5 classifieurs linéaires, entraînés chacun sur 10 points aléatoires.



- (c) Faire voter ces 5 classifieurs sur les labels de tous les points, pour déterminer les points les plus incertains



- (d) Implémenter deux scénarios et comparer les courbes d'apprentissage :
 - en apprentissage QBC classique, les points les plus incertains (probabilité d'appartenir à une classe $\approx 0,5$) seraient considérés comme les plus informatifs et on demande leur label à l'oracle, puis on itère
 - en apprentissage QBC avec abstention, si ces points ont une probabilité d'appartenir à une classe trop proche de 0,5 (disons $0,5 \pm \lambda$ avec λ), l'oracle s'abstient. Essayer avec différentes valeurs de λ .

8 B1 - Bandits adversariels



FIGURE 4 – Bandits (illustration de [5])

1. Partie Théorique :

- (a) Expliquer ce qu'est un problème de Bandits adversariels, par opposition au problème de bandit stochastique vu en classe
- (b) Expliquer le fonctionnement de l'algorithme **Exp3**.

2. Partie Pratique :

- (a) Implémenter l'algorithme **Exp3** sur un problème au choix, par exemple les données boursières [13] ou les données de trafic de sites web [14].
 - (b) Commenter les résultats
- Références : [5, 12]
 - Données : [13, 14]

9 B2 - Monte-Carlo Tree Search

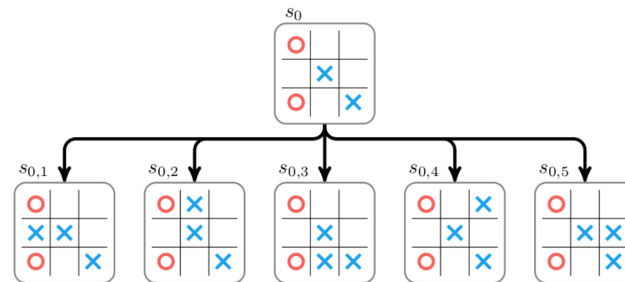


FIGURE 5 – Monte-Carlo Tree Search pour le jeu de Tic-Tac-Toe

1. Partie Théorique :

- Expliquer l'algorithme **Monte-Carlo Tree Search** et son intérêt pratique
- Expliquer la variante « Upper Confidence bound applied to Trees » (UCT), en faisant le lien avec l'algorithme UCB vu au cours.

2. Partie Pratique :

- Implémenter l'algorithme **Monte-Carlo Tree Search** sur un problème au choix, par exemple sur un jeu simple comme Tic-Tac-Toe, que vous pouvez simuler en vous inspirant de [16]
- Commenter les résultats

- Références : [5, 15]
- Données : [16, 17]

10 B3 - Bandits de Thompson



FIGURE 6 – Divers films du système de recommandation.

1. Partie Théorique :

- (a) Expliquer le principe du *Thompson Sampling*
- (b) Montrer comment on obtient des bornes sur le regret pour l'algorithme **Exp3**

2. Partie Pratique :

Cette partie concerne la recommandation de films, proposé sous forme de concours par *Netflix* il ya quelques années. On ne vous demande pas ici de développer un système de recommandation complet, mais d'identifier le film le plus populaire, en supposant qu'au départ aucun film n'a été visionné.

- (a) Il faut donc développer une stratégie pour demander à des utilisateurs de visionner des films et de les noter. Pour ne pas fatiguer les utilisateurs, il faut parvenir le plus rapidement possible à identifier le meilleur film. Formalisez ce problème sous forme de problème de bandits.
- (b) Implémenter l'algorithme **Thompson Sampling** pour résoudre ce problème.
- (c) Commenter les résultats.

- Références : [5, 9, 12]
- Données : Netflix [10], MovieLens [11] ou autre...

Références

Articles et livres

- [1] Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning : From theory to algorithms. Cambridge University Press.
- [2] *Introduction to Statistical Learning Theory*. O. Bousquet, S. Boucheron, and G. Lugosi, Lecture Notes in Artificial Intelligence, 3176 ; pp 169–207 (2004).
- [3] *Importance weighted active learning*. Beygelzimer, A., Dasgupta, S., and Langford, J.. In Proceedings of the 26th annual international conference on machine learning, pp. 49-56, (2009)
- [4] *Agnostic active learning* ; Balcan, M. F., Beygelzimer, A., and Langford, J. Journal of Computer and System Sciences, 75(1), 78-89. (2009)
- [5] *Bandit algorithms*. Lattimore, Tor, and Csaba Szepesvári. Cambridge University Press, (2020).

Les ouvrages cités ci-dessus sont disponibles en format pdf sur Moodle.

Sites web

- [6] <https://towardsdatascience.com/understanding-adaboost-2f94f22d5bfe>
- [7] <https://www.youtube.com/watch?v=gR9Q8pS03ZE>
- [8] UCI Machine Learning Repository (classification datasets)
<https://archive.ics.uci.edu/ml/datasets.php?format=&task=cla&att=&area=&numAtt=&numIns=&type=&sort=nameUp&view=table>
- [9] <https://blog.insightdatascience.com/multi-armed-bandits-for-dynamic-movie-recommendation/>
- [10] www.kaggle.com/laowingkin/netflix-movie-recommendation/data
- [11] <https://www.kaggle.com/grouplens/movielens-20m-dataset>
- [12] <https://jeremykun.com/2013/11/08/adversarial-bandits-and-the-exp3-algorithm>
- [13] <https://jeremykun.com/2013/12/09/bandits-and-stocks>
- [14] <https://sourceforge.net/projects/bandit/files/Datasets>
- [15] <http://tim.hibal.org/blog/alpha-zero-how-and-why-it-works/>
- [16] <https://medium.com/swlh/tic-tac-toe-at-the-monte-carlo-a5e0394c7bc2>
- [17] <https://www.baeldung.com/java-monte-carlo-tree-search>