
MI201 - Apprentissage Automatique

Résumé Théorique

12 septembre 2024

Guilherme Nunes Trofino
2022-2024

Table des matières

1	Introduction	2
1.1	Information Matier	2
2	Classification	3
2.1	Supervised Learning	3
2.1.1	kNN, k Nearest Neighbours	3
2.1.2	Decision Trees	3
2.1.3	SVM, Support Vector Machines	3
2.1.4	Neural Networks	3
2.2	Semi-Supervised Learning	3
2.2.1	Linear Regression	3
2.3	Unsupervised Learning	3
2.3.1	k-Means Clustering	3
2.4	Reinforcement Learning	3
3	Prédiction	4
3.1	Analyses	4
3.1.1	Confusion Matrix	4
3.1.2	Méthodes Ensemblistes	4
3.1.3	Validation Croisée	5
3.1.4	Compromis Biais-Variance	5
3.1.5	Surapprentissage	5

1. Introduction

Repository Hello! My name is Guilherme Nunes Trofino and this is my LaTeX notebook of MI201 - Apprentissage Automatique that can be found in my GitHub repository : https://github.com/tr0fin0/classes_ensta.

Disclaimer This notebook is made so it may help others in this subject and is not intend to be used to cheat on tests so use it by your on risk.

Suggestions If you may find something on this document that does not seam correct please reach me by e-mail : guitrofino@gmail.com.

1.1. Information Matier

Référence <https://cs231n.github.io/python-numpy-tutorial/>

2. Classification

2.1. Supervised Learning

2.1.1. kNN, k Nearest Neighbours

La choix de k déterminera la qualité de la prédiction. Si on augmente k la qualité de la prediction améliore et le sur-apprentissage diminue. Généralement on aura le comportement suivant :

k ↑	biais ↑	variance ↓
k ↓	biais ↓	variance ↑

TABLE 2.1 : Comportement kNN

2.1.2. Decision Trees

La choix de la profondeur p de l'Arbre de Décision déterminera la qualité de la prédiction. Si on augment p la prediction améliore. Généralement on aura le comportement suivant :

p ↑	biais ↓	variance ↑
p ↓	biais ↑	variance ↓

TABLE 2.2 : Comportement Decision Tree

2.1.3. SVM, Support Vector Machines

Généralement on aura le comportement suivant :

merge rigide	C ↑	biais ↓	variance ↑
merge souple	C ↓	biais ↑	variance ↓

TABLE 2.3 : Comportement SVM

2.1.4. Neural Networks

La choix de n déterminera la qualité de la prédiction. Si on réduire n la qualité de la prediction améliore et le sur-apprentissage diminue. Généralement on aura le comportement suivant :

n ↑	biais ↓	variance ↑
n ↓	biais ↑	variance ↓

TABLE 2.4 : Comportement Neural Network

2.2. Semi-Supervised Learning

2.2.1. Linear Regression

2.3. Unsupervised Learning

2.3.1. k-Means Clustering

2.4. Reinforcement Learning

3. Prédiction

3.1. Analyses

Parmi les possibilités d'évaluation la plus simple c'est le Taux d'Erreur Moyen :

Définition 3.1.

$$e = \frac{\text{nombre d'étiquettes erronées}}{\text{nombre total de données}} \quad (3.1)$$

3.1.1. Confusion Matrix

Util qui permet de visualiser la performance d'un algorithme, généralement supervisé, défini par le suivant :

Définition 3.2. Table qui permet de visualiser les prévisions d'un algorithme :

$$\mathcal{M} = \begin{bmatrix} m_{00} & \cdots & m_{0j} & \cdots & m_{0n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ m_{i0} & \cdots & m_{ij} & \cdots & m_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ m_{n0} & \cdots & m_{nj} & \cdots & m_{nn} \end{bmatrix} \quad (3.2)$$

Où :

1. i : chaque ligne représente des instances de la classe actuelle ;
2. j : chaque colonne représente des instances de la classe de prévision ;

Quand on considère que les données avaient n caractéristiques.

Remarque. Dans un bon résultat :

$$m_{ij} \rightarrow \begin{cases} 0 & \forall i \neq j \\ N \in \mathbb{N} & \forall i = j \end{cases}$$

3.1.2. Méthodes Ensemblistes

À fin de contourner la tendance d'un algorithme au surapprentissage on peut utiliser deux Méthodes Ensemblistes pour réduire la variance d'un model :

Définition 3.3. Méthodes Ensemblistes sont des méthodes agrégeant des ensembles de classifieurs en échantillonnant différemment les données. On utilise :

1. **Bagging** :

Définition 3.4. Développement de plusieurs modèles en parallèle.

2. **Boosting** :

Définition 3.5. Développement de plusieurs modèles en série.

3.1.3. Validation Croisée

Les algorithmes auront des performances différents pour des données distinctes, alors comme les données sont divisées entre apprentissage et validation ira affecter les résultats. À fin de minimiser cette variation on peut utiliser la **Validation Croisée** :

Définition 3.6. Cross Validation is a resampling method that uses different portions of the data, named **Fold**, to test and train a model on different iterations.

In the end a mean of the results is taken.

Phrase. useful to compare the performance of different algorithmes in the same data or to validate a model

3.1.4. Compromis Biais-Variance

Toujours quand on analyse une algorithme de Apprentissage Automatique il faut considère le **Compromis Biais-Variance** :

Définition 3.7. Le problème de minimiser simultanément deux sources d'erreurs qui empêchent les algorithmes d'apprentissage supervisé de généraliser au-delà de leur échantillon d'apprentissage :

1. **biais** :

- (a) l'erreur provenant de la quantité d'hypothèses erronées ;

Remarque. Un biais élevé peut entraîner un **sous-apprentissage**, algorithme qui manque de relations pertinentes entre les données en entrée et les sorties prévues.

On peut minimiser ce problème :

- train more ;
- increase model complexity en ajoutant des features ;

2. **variance** :

- (a) l'erreur provenant de la sensibilité aux petites fluctuations ;

Remarque. Une variance élevée peut entraîner un **sur-apprentissage**, algorithme modélise le bruit aléatoire des données d'apprentissage plutôt que les sorties prévues.

On peut minimiser ce problème :

- introduce more data ;
- use regularization ;

La décomposition biais-variance est une façon d'analyser l'espérance de l'erreur de prédiction d'un algorithme d'apprentissage d'un problème particulier comme une somme de trois termes : le biais, la variance et une quantité, appelée erreur irréductible, résultant du bruit dans le problème lui-même.

3.1.5. Surapprentissage

Quand on balance le Biais et la Variance il faut toujours considère le Surapprentissage, Over-Fitting :

Définition 3.8. Quand le modèle proposé est très spécialisée, apprend pas coeur les données de traînement, et portant est mauvais pour la généralisation des données de validation.

C'est indésirable et donc il faut toujours l'éviter. À ce propos on peut :

1. **Data Augmentation :**

Définition 3.9. Ajouter des données pour l'apprentissage.

2. **Reduced Complexity :**

Définition 3.10. Réduire le nombre de caractéristiques considérées pour l'apprentissage.

On peut le faire en :

- (a) early stopping;