Outlier detection - Détection de fraudes

H. Andres & M. Bouazza & M. Casanave & M. Karpe

École des Ponts ParisTech

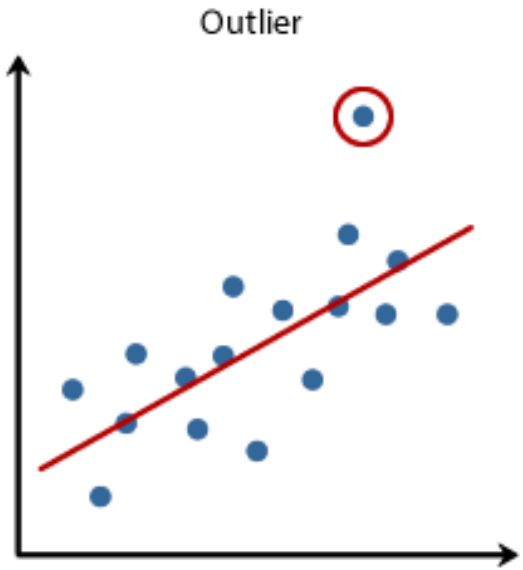
25 mai 2018

Sommaire

Introduction

Comment détecter les données aberrantes ?

Une étude comparative des différents algorithmes d'apprentissage non supervisé



Sommaire

Autoencoder

Principe :

- Codeur : $\psi : x \in \mathbb{R}^d \longmapsto z \in \mathbb{R}^{d'}$ $(d' < d)$
- Decodeur : $\phi : z \in \mathbb{R}^{d'} \longmapsto \tilde{x} \in \mathbb{R}^d$
- $\psi, \phi = \underset{\psi,\phi}{argmin} \|X - \phi \circ \psi(X)\|^2_{\mathbb{R}^d}$ (X étant le jeu de données)

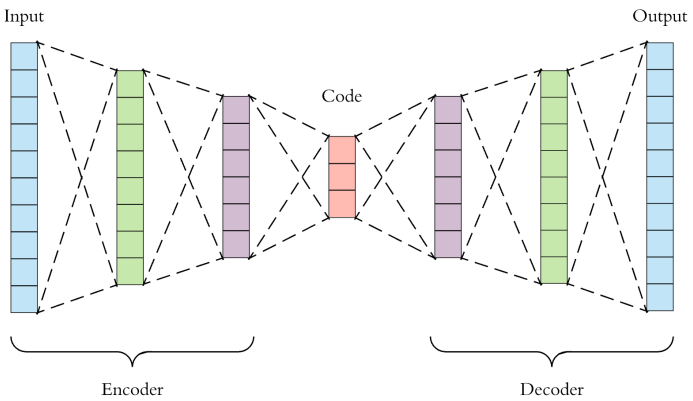


Figure – Illustration du principe de *Autoencoder*.

Autoencoder

- **Idée :** "outlier" moins bien représenté
- **Conséquence :** erreur d'approximation $\|x - \phi \circ \psi(x)\|^2_{\mathbb{R}^d}$ plus grande
- **Détection :** données ayant les plus grandes erreurs

Local Outlier Factor

Idée : Comparaison de la densité avec les k voisins

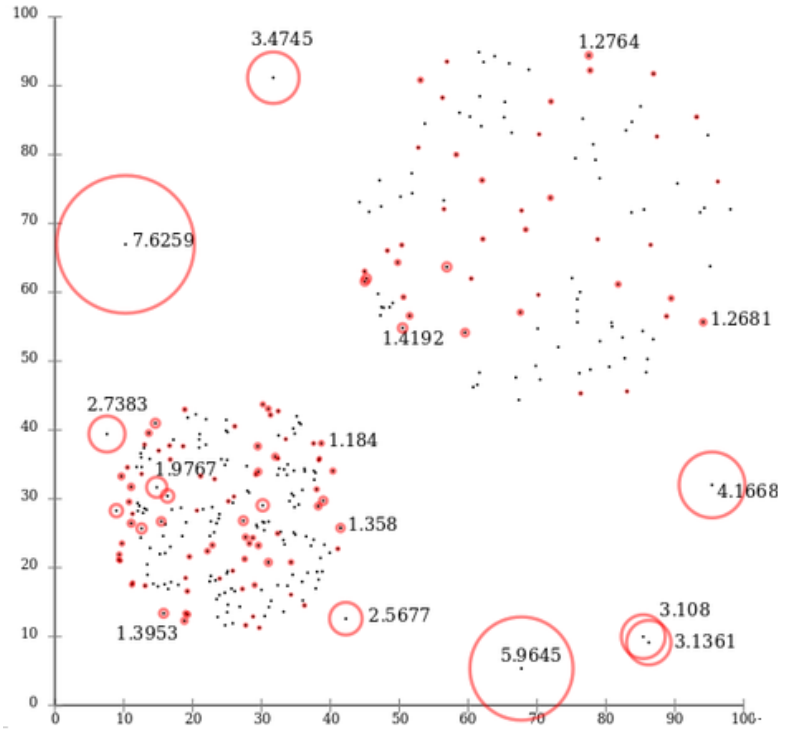


Figure – Illustration de la méthode *Local Outlier Factor*.

Local Outlier Factor

En pratique :

- "Distance d'atteinte" $d_k(x, y)$: maximum de la vraie distance $d(x, y)$ et de la distance entre y et son k^e voisin
- "Densité locale" de x : inverse de la moyenne des distances d'atteinte de x depuis ses k voisins

Robust Estimator of Covariance

Hypothèses : $X_i \sim N_p(\mu_i, \sigma_i)$ données générées à partir d'une densité elliptique

Robustesse : Modèle imposé par **la majorité** des données, donc moins influencé par les outliers

Distance de Mahalanobis : $D_M(x) = \sqrt{(x-\mu)^T \sum^{-1} (x-\mu)}$

x outlier si n'appartient pas à **l'ellipsoïde de tolérance** :

$$\left\{ x \mid D_M(x) \leq \sqrt{\chi^2_{p,0.975}} \right\}$$

Robust Estimator of Covariance

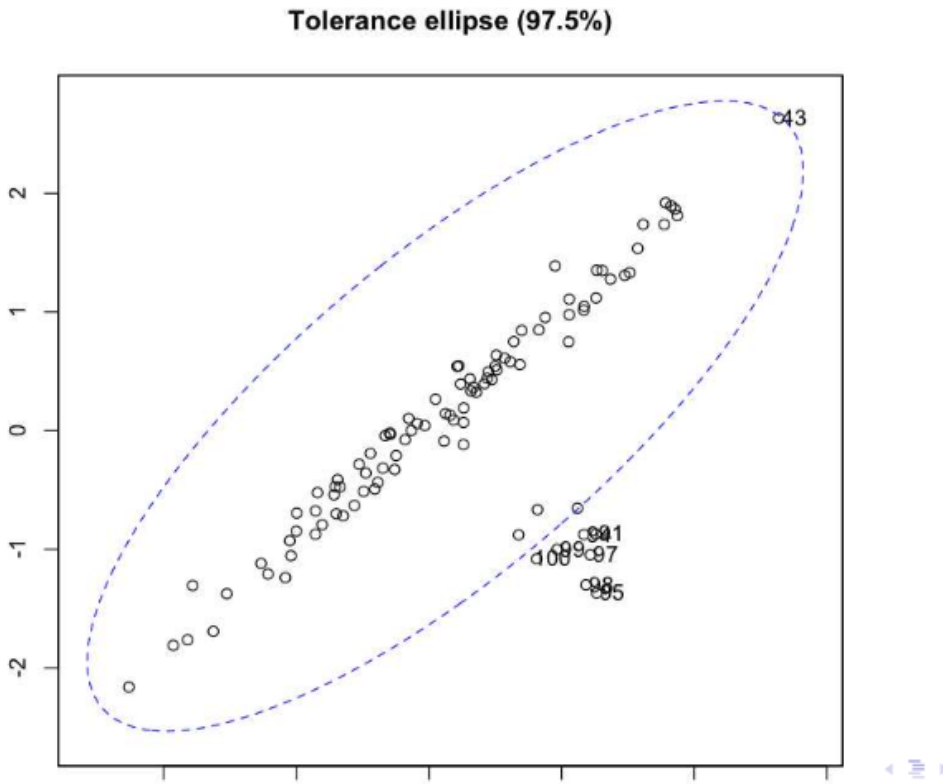
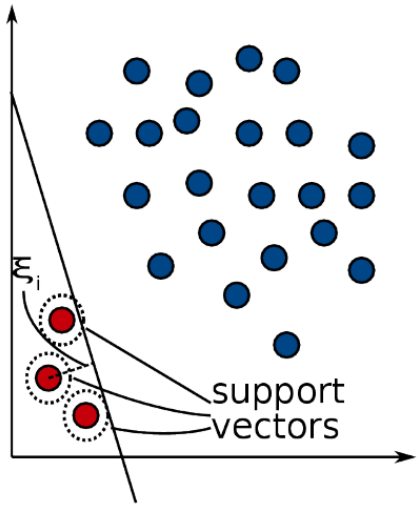


Figure – Illustration d'une ellipsoïde de tolérance.

One-class SVM

Principe :

- Fonction de décision : hyperplan
- Applique le noyau et sépare les données
- Outliers relâchés par une variable ξ
- ν est choisi par l'utilisateur



Programme de minimisation :

$$\min_{w,\xi_i,\rho} \frac{1}{2}\|w\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho$$
$$\text{s.t. } (w \cdot \phi(x_i)) \geq \rho - \xi_i,\ \xi_i \geq 0,\ \forall i = 1, \ldots, n$$

Fonction à noyau : $K(x, x_i) = \phi(x)^T \phi(x_i)$
Fonction de décision : $f(x) = \text{sgn}((w \cdot \phi(x_i)) - \rho)$

One-class SVM

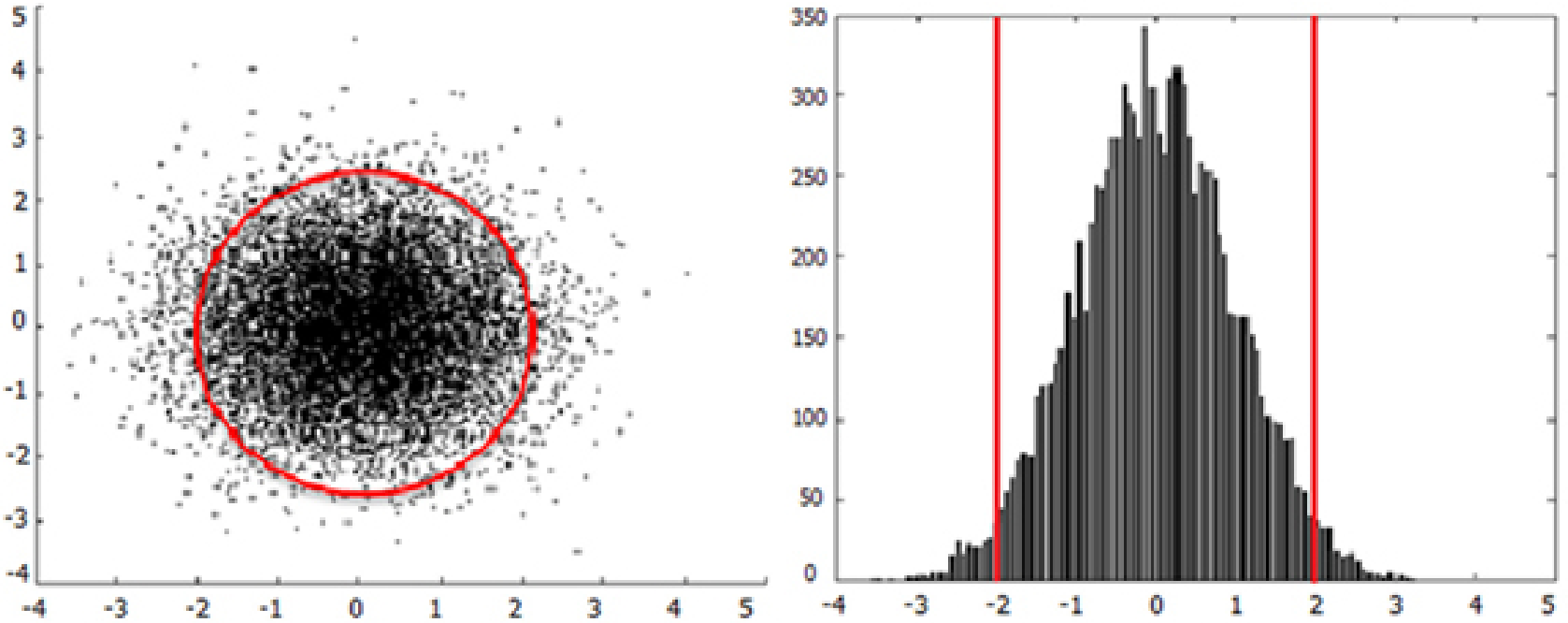


Figure – Illustration de la méthode *One-class SVM*.

Isolation Forest

Hypothèses principales : *outliers* = points en faible nombre et différents des autres

Conséquence : *outliers* faciles à isoler

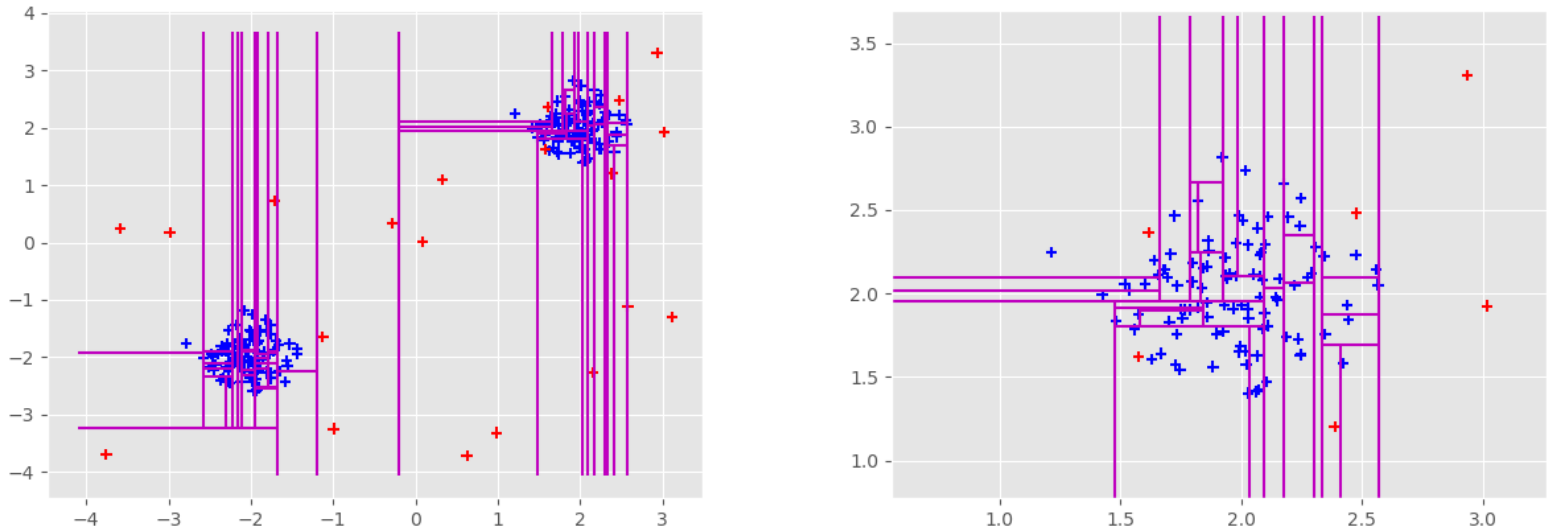
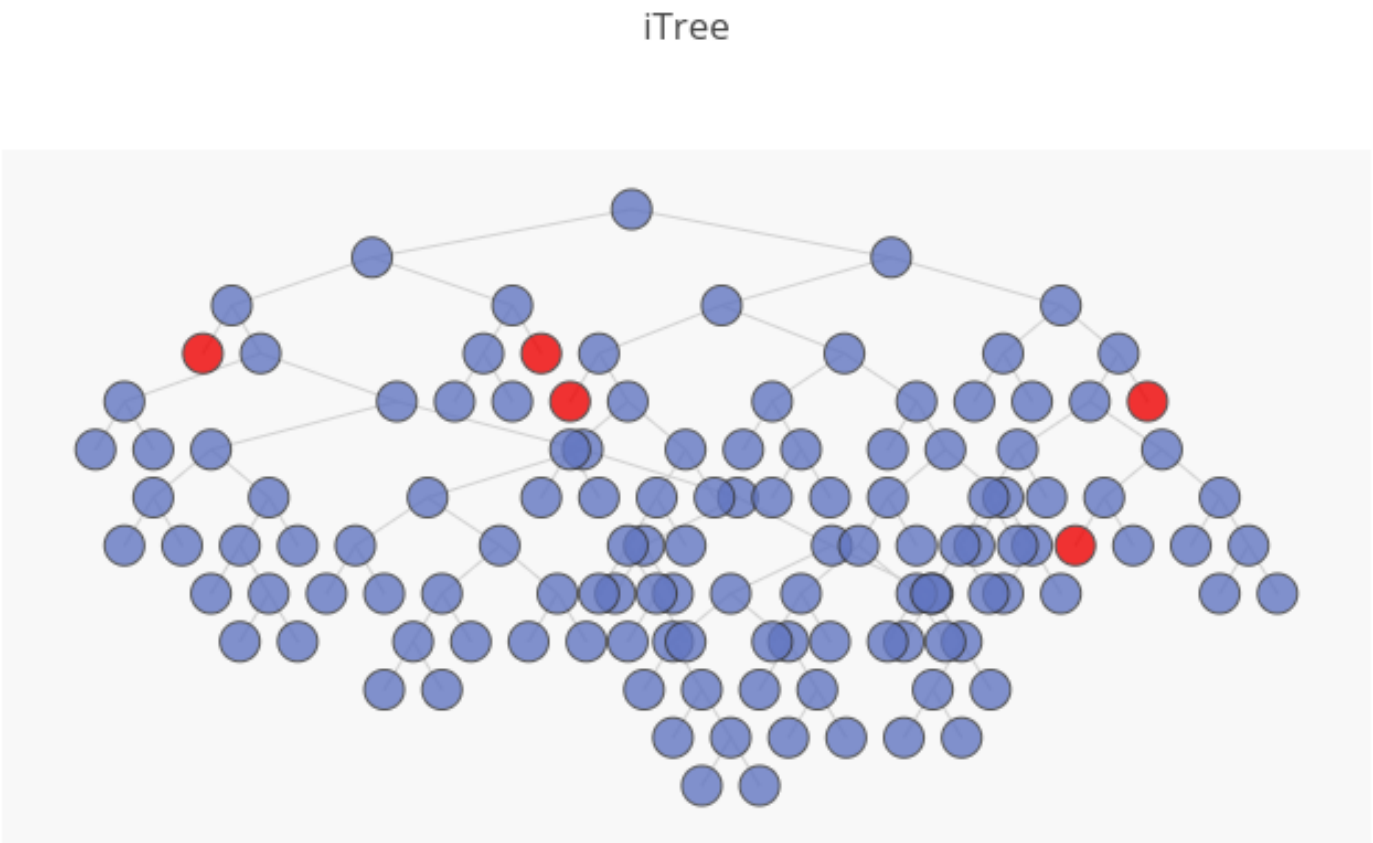


Figure – Illustration de la méthode *Isolation Forest*.

Isolation Forest



Sommaire

Coûts et profits pour la banque

		PREDICTED	
		"P"	"N"
ACTUAL	P	TRUE POSITIVE Outlier detected as outlier Profit: €160	FALSE NEGATIVE Outlier detected as inlier Cost: €160
	N	FALSE POSITIVE Inlier detected as outlier Cost: €1	TRUE NEGATIVE Inlier detected as inlier Profit: €0

Figure – Illustration des différents coûts intervenant dans la modélisation.

Stratégie de maximisation

$$\pi = 160(TPR - FNR) - FPR \quad \text{avec}$$

$$TPR = \frac{\#\text{ Vrais positifs}}{\#\text{ Positifs}}$$

$$FNR = \frac{\#\text{ Faux négatifs}}{\#\text{ Positifs}}$$

$$FPR = \frac{\#\text{ Faux positifs}}{\#\text{ Négatifs}}$$

Sommaire

Jeux de données

	Données	Dimensions	Outliers
Synthetic	1000	100	13,6 %
Creditcard	284807	29	0,17 %
KDD10	494021	38	1,77 %
SA10	100655	38	3,36 %
SF10	73237	3	4,50 %
HTTP10	58725	3	3,76 %
SMTP10	9571	3	0,03 %

Figure – Description des jeux de données utilisés dans le cadre du projet.

Optimisation des hyperparamètres

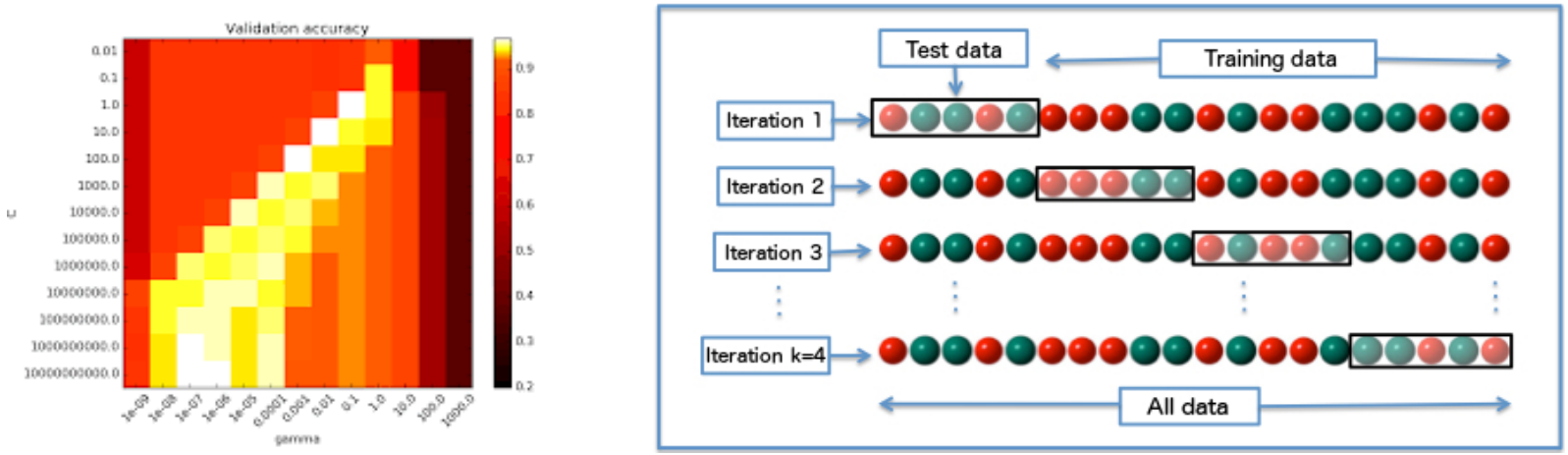


Figure – Illustration des méthodes de *Grid Search* (à gauche) et de *validation croisée* (à droite).

Description et analyse des résultats

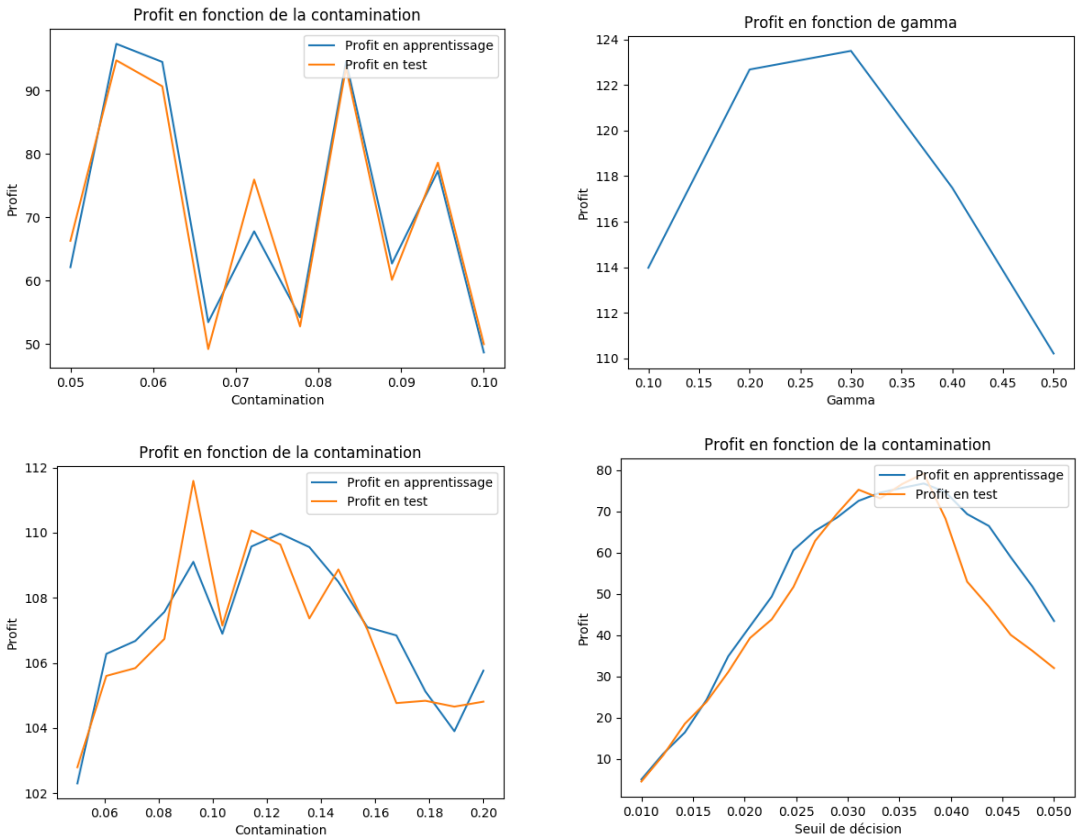


Figure – Evolution de la fonction de profit en fonction de la valeur de l'hyperparamètre caractéristique de l'algorithme.

Combinaison des algorithmes

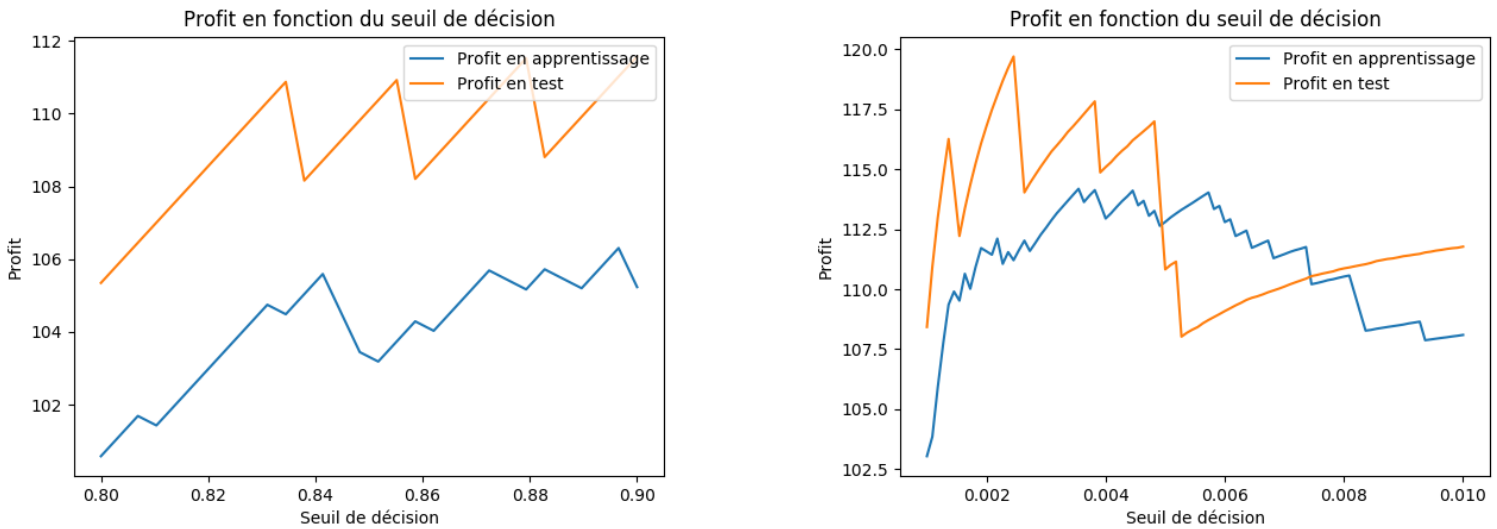


Figure – Evolution de la fonction de profit en fonction du seuil de décision pour la combinaison des algorithmes par *rank averaging* (à gauche) et par *bagging* (à droite).

Sommaire

Synthèse et perspectives

Constats majeurs :

- One-class SVM semble être le plus efficace
- Combinaison des algorithmes peu efficace
- Importance de la fonction de coût / profit

Limites du projet :

- Taille des bases de données
- Temps d'exécution des algorithmes

Principales références I

Observatoire de la sécurité des moyens de paiement - Banque de France

Rapport annuel de l'Observatoire de la sécurité des moyens de paiement

https://www.banque-france.fr/sites/default/files/medias/documents/osmp2016_web.pdf

2016

Journal of Machine Learning Research

Scikit-learn : Machine Learning in Python

http://scikit-learn.org/stable/

2011