

A review of outliers detection

Titouan Vayer

May 19, 2017

1 Introduction

- Qu'est-ce qu'un outlier ?
- Comment détecter un outlier ?
- Comment traiter un outlier ?

2 Les modèles statistiques

- Boxplot
- Extreme Studentized Deviate
- Mahalanobis distance

3 Les modèles basés sur des distances

- K-NN

- Local outlier factor
- K-means

4 Les modèles paramétriques

- PCA
- Normal mixture models

5 Les modèles semi paramétriques

6 Neural Network

- Self-Organized Map (SOM)
- Replicator Neural Network (RNN)

Les travaux suivants sont basés essentiellement sur le papier de
[Victoria J.Hodge : A Survey of Outlier Detection Methodologies]

Qu'est-ce qu'un outlier ?

Deux définitions possibles :

- (Grubbs, 1969) Un outlier est une observation qui semble dévier fortement par rapport aux autres observations du sample dans lequel il se situe
- (Barnett & Lewis, 1994) Une observation (ou un ensemble d'observations) est un outlier s'il apparaît comme étant contradictoire avec le reste des données.

Comment mettre en place une stratégie de détection ?

- **Unsupervised clustering** : Déterminer l'outlier sans à priori sur les données. Suppose d'avoir un dataset suffisamment fourni (type I)
- **Supervised classification** : Modéliser l'anormalité et la normalité. Requiert d'avoir des données labélisées. type(II)
- **Semi-supervised classification** : Modéliser seulement la normalité ou alors l'anormalité. Utilisable pour des données figées et non figées, apprend et s'améliore au fur et à mesure que les données arrivent. (type III)

Comment traiter un outlier ?

On a deux approches possibles pour traiter un outlier :

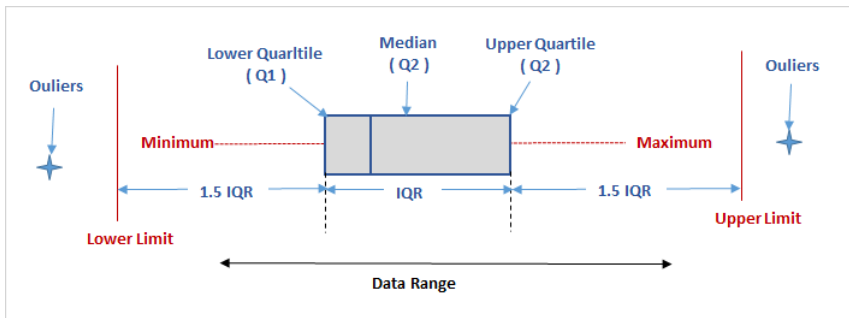
- **Le diagnostique** : on analyse les outliers et on les enlève ou pas
- **L'accomodation** : on garde les outliers quoiqu'il se passe

Les approches statistiques sont les algorithmes les plus anciens pour détecter les outliers.

Ils sont généralement bien dimensionnés pour des données quantitatives et s'intéressent à la distribution des données.

Boxplot

C'est une technique de détection **non supervisée**. L'idée est de produire une représentation graphique qui permet de juger si des points sont outliers ou non.



Laurikkala et al. suggère une distance (heuristique) de 1.5 inter-quartile range beyond entre la limite haute et basse pour détecter les outliers.

ESD test ou test de Grubb

C'est une technique de détection **non supervisée**.

The Extreme Studentized Deviate ou ESD test (Rosner 1983) est utilisé pour détecter un ou plusieurs outliers pour des données univariées qui suivent approximativement une **distribution normale**. Il teste les hypothèses suivantes :

- H_0 : Il n'y a pas d'outlier
- H_a : Il y a au moins un outlier

Le test statistique est le suivant :

$$G = \frac{\max_{i=1,\dots,N} |Y_i - \bar{Y}|}{s}$$

Le test de Grubb correspond à la plus grande déviation par rapport à la moyenne par rapport à la variance.

ESD test ou test de Grubb

Il peut être aussi défini en tant que "one-side test", pour tester si la valeur minimale est un outlier avec

$$G = \frac{\bar{Y} - Y_{\min}}{s}$$

ou alors

$$G = \frac{Y_{\max} - \bar{Y}}{s}$$

Pour le cas général, l'hypothèse nulle est rejeté pour un degré α si

$$G \geq \frac{N-1}{\sqrt{N}} \sqrt{\frac{t_{\alpha/(2N), N-2}^2}{N-2} + t_{\alpha/(2N), N-2}^2}$$

avec $t_{\frac{\alpha}{2N}, N-2}^2$ représente la borne supérieure de la t-distribution à $N-2$ degrés de liberté.

Mahalanobis distance

C'est une technique de détection **non supervisée**.

Contrairement à la méthode précédente elle permet de prendre en compte l'aspect multivarié d'un dataset. La méthode se base sur le principe suivant : si $X \sim \mathcal{N}(\mu, \Sigma)$ alors $D^2(X, \mu) \sim \chi_p^2$ où $D^2(X, \mu)$ est la distance de Mahalanobis définie par :

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1}(\vec{x} - \vec{\mu})}$$

avec μ et S la moyenne et la matrice de variance-covariance du dataset.

On peut donc avoir un intervalle de confiance de tel sorte que :

$$\mathbb{P}[D^2(X, \mu) \leq \chi_{p, 1-\alpha}^2] \leq 1 - \alpha$$

La détection par la distance de Mahalanobis a été implémentée sur le github suivant [Titouan Vayer]

<https://github.com/bigtdu53/outlierdetection>

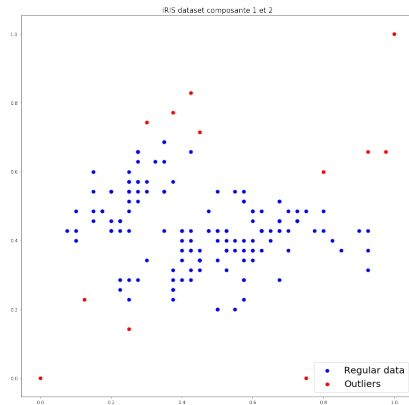
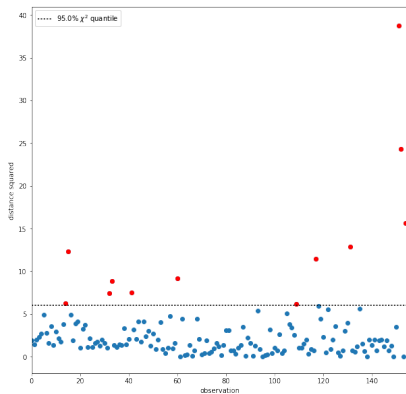
Les données utilisées ont été :

- Iris Dataset
- dtmcross201606.csv

L'algorithme présente deux inconvénients :

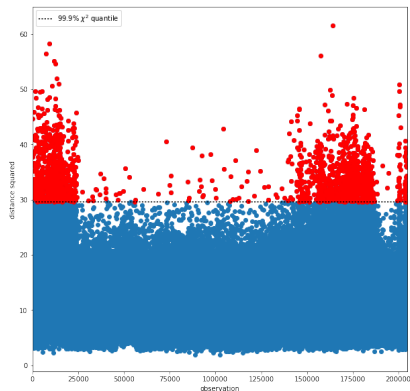
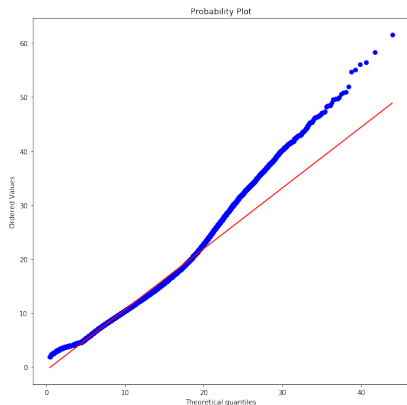
- La normalité des données
- L'inversion de la matrice très coûteuse en grande dimension

Outlier detection using Mahalanobis distance on Iris Dataset



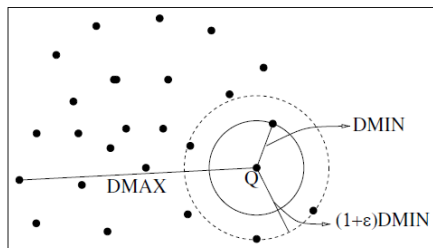
Outlier detection using Mahalanobis distance on dtmcross

Dans le cas de dtmcross on a une très grand dataset (environ 1000 variables). Aussi pour pouvoir appliquer la méthode on passe d'abord par une étape (PCA) de réduction de dimension.



Limitation : Malédiction de la dimension

Plus la dimension augmente, plus les points sont regroupés dans un volume plus grand et qui devient de moins en moins dense.([Kevin Beyer])



Les modèles statistiques utilisent différentes approches pour s'affranchir du problème de la malédiction de la dimension, toutes ces techniques entraînent un grand coût de processing. Une autre alternative est de réduire la dimension de l'espace.

Le théorème de Beyer

On considère $X_1, \dots, X_d \in \mathbb{R}^n$ de telle sorte que $\forall m \in [1, \dots, d], X_{m,1}, \dots, X_{m,n}$ sont n points indépendants tirés selon une distribution \mathcal{F}_m .

On choisit un point Q_m indépendamment des $X_{m,i}$. On note :

$d_{\min}(m) = \min\{d(X_{m,i}, Q_m) | 1 \leq i \leq n\}$ et

$d_{\max}(m) = \max\{d(X_{m,i}, Q_m) | 1 \leq i \leq n\}$

Theorem

Si $\lim_{d \rightarrow +\infty} \text{var}\left(\frac{d(X_{m,1}, Q_m)^p}{\mathbb{E}[d(X_{m,1}, Q_m)^p]}\right) = 0$ alors

$$\forall \epsilon \geq 0, \mathbb{P}(d_{\max}(m) \leq (1 + \epsilon)d_{\min}(m)) = 1$$

Ce théorème est valable par exemple lorsque les données sont i.i.d dans chaque dimension, que les moments sont finis et que les query point sont choisis indépendamment vis à vis des données

Les techniques de "Proximity-based" ou basées sur des distances, sont simples à implémenter et ne font pas d'à priori sur le modèle de distribution des données. Elles sont valables pour les type I et II de détection d'outliers.

Cependant elles souffrent généralement d'un grand coût en terme de complexité étant donné qu'elles sont basées sur le calcul de toutes les distances entre les points. La complexité est directement proportionnelle à la dimension d des données et au nombre d'éléments n .

Ramaswamy et al. (Ramaswamy et al., 2000) ont mis en place un algorithme kNN optimisé qui produit une liste triée d'outliers potentiels.

Soit n le nombre de points dans le dataset. On note $D_m(x)$ la distance de x à son m -ième voisin. Alors un point p est considéré comme étant un outlier si il y a moins de $n - 1$ points qui ont un D_m plus grand.

Knorr & Ng (Knorr and Ng, 1998) ont mis en place une autre approche : si m des k nearest neighbours ($m \leq k$) sont regroupés dans un cercle de rayon d (avec d spécifié) alors le point est considéré comme normal car il est dans une région suffisamment dense en points.

D'autres approches de kNN permettent de faire de la détection d'outliers dans le cas du type II. On peut citer l'approche du vote à la majorité de Wettschereck en 1994 qui requiert un jeu de données labellé. Pour chaque point à tester un kNN est computed et le nouvel exemplaire est classé selon un vote à la majorité.

Une extension consiste à avoir un vote pondéré selon la distance à l'item à classer. (le poids du vote diminue quand la distance augmente)

Using K-Means

C'est une technique de détection **semi supervisée**.

Les clusters trouvés grâce au K-means minimise la variance intra-class c'est à dire l'équation suivante :

$$\sum_{j=1}^K \sum_{n \in S_j} \|x^n - \mu_j\|^2$$

L'idée est qu'après la construction de ces clusters, dans chaque partition il existe un rayon maximum, qui est en fait la distance entre le centre du cluster et le point le plus éloigné dans le cluster. Ce rayon définit une frontière de normalité et est *local à chaque cluster* au lieu d'être une distance globale comme dans plusieurs approches ((Knorr and Ng, 1998), (Ramaswamy et al., 2000) and (Byers and Raftery, 1998) avec leurs approches basées sur des K-NN).

Using K-Means

Ainsi l'algorithme est le suivant :

Semi supervised outlier detection using K-means

- Choisir un jeu de données où chaque donnée est **normale**
- Choisir K le nombre de cluster à former et computer K-means sur ces données
- Pour chaque cluster trouver la distance maximale entre le centre du cluster et le point le plus éloigné du cluster.
- Pour chaque nouveau point à tester :
 - Si pour chaque cluster, le point n'est pas dans le cercle de centre le centre du cluster et de rayon la distance maximale précédente alors le point est un outlier
 - Sinon, le point est normal

L'avantage est que l'algorithme, une fois les frontières de normalité mises en place, peut être exécuté en online.

Une variante de l'algorithme existe en utilisant les K-medoids. L'algorithme est aussi un algorithme de clusterisation et est dans la même veine que les Kmeans. Tandis que K-means minimise la variance intra class K medoids lui minimise la somme des dissimilarités entre les points d'un cluster vis à vis d'un point désigné comme étant le centre de ce cluster. La différence réside donc dans le fait que **le centre est un point du cluster et non le barycentre**

Il a l'avantage d'être plus robuste aux outliers comparé au K-means car il minimise les distances pairwise.

La détection par K-means a été implémentée sur le github suivant [Titouan Vayer] <https://github.com/bigtd53/outlierdetection>

Les données utilisées ont été :

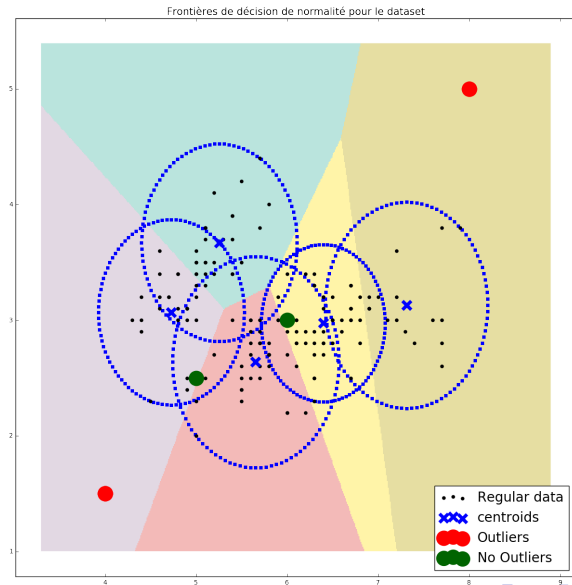
- Iris Dataset
- dtmcross201606.csv

L'algorithme présente deux inconvénients :

- Nécessité de connaître suffisamment de données "normales"
- Coût du K-means

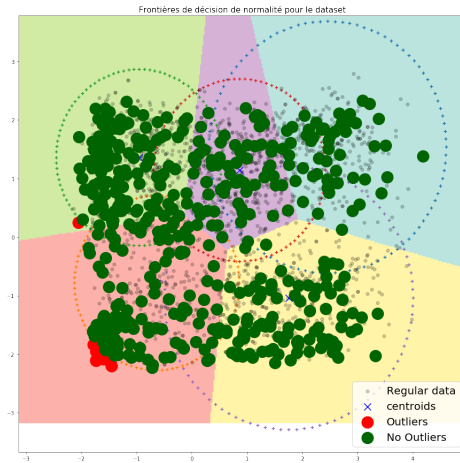
Une version parallélisée a été mise en place, ainsi qu'une version utilisant les K-medoids

Outlier detection using K-means distance on Iris Dataset



Outlier detection using Kmeans distance on dtmcross

Dans le cas de dtmcross on a un très grand dataset (environ 1000 variables). Pour éviter la malédiction de la dimension on fait une PCA d'abord. A des fins de visualisation on ne garde que deux composantes.



Classical PCA

$$\begin{array}{ll} \text{minimize} & \|M - L\| \\ \text{subject to} & \text{rank}(L) \leq k \end{array}$$

Souffre souvent lorsqu'il y a beaucoup de bruit ou d'outliers.

On considère n points dans un espace de dimension p . On suppose qu'une fraction $1 - \gamma$ des points font partie d'une sous espace de faible dimension r de \mathbb{R}^p , tandis que le reste des $n\gamma$ points sont arbitrairement localisés : ces points sont considérés comme étant des outliers. Le but est d'apprendre le sous espace de dimension r et d'identifier les outliers.

On considère alors la matrice des observations M de dimension $p \times n$. On peut alors la décomposer en :

$$M = L_0 + C_0$$

avec C_0 est la column-sparse matrice avec $(1 - \gamma)n$ colonnes nulles (corrupted measures, supposé sparse) et $\text{rank}(L_0) = r$.

Si on considère alors la SVD suivante :

$$L_0 = U_0 \Sigma_0 V_0^T$$

Alors les colonnes de U_0 forment une base orthonormée de note espace de dimension r recherché et C_0 est la matrice corespondant aux outliers ; avec le nombre de non-zero column (γn) qui est l'ensemble des outliers.

Robust PCA via outlier pursuit

Pour trouver L_0 et C_0 ce problème l'algorithme suivant est mis en place :

Outlier Pursuit

Find (L^*, C^*) the optimum of the following convex optimization program :

$$\begin{aligned} & \text{minimize} && \|L\|_* + \lambda \|C\|_{1,2} \\ & \text{subject to} && M = L + C \end{aligned}$$

Compute SVD $L^* = U_1 \Sigma_1 V_1^T$ and output $U^* = U_1$ Output the set of non-zero columns of C^*

[Emmanuel J. Candes, John Wright][Huan Xu]

Outlier detection using Robust PCA

Robust PCA est un domaine de recherche très actif. L'algorithme suivant (décrit ici [Huan Xu]) est un algorithme de de détection **non supervisée**.

Robust PCA for outlier detection

- Compute PCA on your data
- Project your data on to the top d principal components
- Throw away "at random" one of the data points whose projection is "too large"
- Repeat this "a few" times

L'idée sous-jacente est que les vecteurs dont la projection après la PCA est grande ont affecté l'estimation trop largement, et il faut ainsi les enlever.

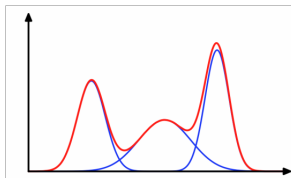
Outliers detection using Normal Mixture Models

Les données sont supposées comme étant réparties suivant une mixtute de gaussiennes, c'est à dire que la densité des données est :

$$f(x|\alpha_1, \dots, \alpha_K, \theta_1, \dots, \theta_K) = \sum_{k=1}^K \alpha_k f_k(x|\theta_k), \sum_{k=1}^K \alpha_k = 1$$

avec pour $\theta_k = (\mu_k, \sigma_k)$:

$$f_k(x|\theta_k) = \frac{\exp\left(\frac{-(x-\mu_k)^2}{2\sigma_k^2}\right)}{\sqrt{2\pi\sigma_k^2}}$$



Outliers detection using Normal Mixture Models

[Katie Evans] L'idée est de calculer la variance dans chaque cluster avec et sans les points à tester et regarder si elle varie beaucoup :

"Based on our experience that the variance is likely to be the parameter most impacted by the presence of an outlier, we have developed a method for outlier detection based on leave-one-out re-estimation of cluster parameters. If an observation is near the center of the cluster, the cluster variance estimates will be very similar when calculated with or without this observation. However, if an observation lies much further from the center of the cluster to which it was assigned, the variance estimate of that cluster will be inflated when the observation is included. For implementation, all cluster memberships are held constant and, for each cluster, the selected variance structure is estimated without each cluster member."

C'est une technique de détection **non supervisée** [Simon Hawkins].

Le RNN est un multi layer perceptron avec trois couches cachées. Le but du RNN est de *reproduire le pattern des données en entrée dans la couche de sortie*. Le nombre de noeuds dans les trois couches est choisi expérimentalement pour minimiser l'erreur moyenne de reconstruction.

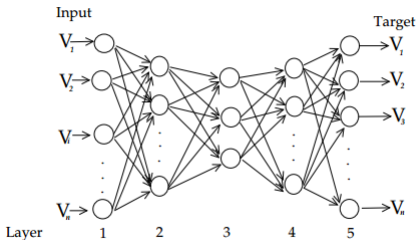


Figure: Schéma du RNN

La sortie du noeud i de la couche k est calculée par la fonction d'activation $S_k(I_{ki})$ où I_{ki} est la somme pondérée des couches d'entrée du noeud défini par :

$$I_{ki} = \sum_{j=0}^{L_{k-1}} \omega_{kij} Z_{(k-1)j}$$

avec Z_{kj} la sortie du noeud j de la couche k .

Les couches d'entrée et de sortie les fonctions d'activation sont :

$$S_k(\theta) = \tanh(a_k \theta); k = 2, 4$$

Pour la couche intermédiaire la fonction d'activation est une fonction étagée :

$$S_3(\theta) = \frac{1}{2} + \frac{1}{2(k-1)} \sum_{j=1}^{N-1} \tanh(a_3(\theta - \frac{j}{N}))$$

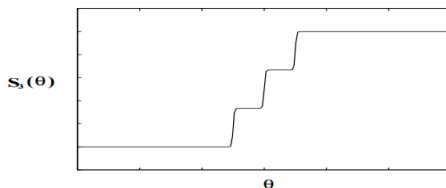


Figure: Fonction d'activation de la couche intermédiaire

Elle permet de faire un mapping à des catégories discrètes, d'en quelque sorte de clusteriser les données.

Pour mesurer le potentiel d'outlier le papier définit le *Outlier Factor* pour un record i qui est en fait l'erreur de reconstruction moyenne sur tous les features :

$$OF_i = \frac{1}{n} \sum_{j=1}^n (x_{ij} - o_{ij})^2$$

References



Victoria J.Hodge (2017)

A Survey of Outlier Detection Methodologies

http://book.itep.ru/depository/security/anomaly/Hodge+Austin_OutlierDetection_AIRE381.pdf



Titouan Vayer Github (2017)

GitHub repository

<https://github.com/bigtd53/outlierdetection>



Kevin Beyer (1998)

When is Nearest Neighbor Meaningful ?

[https:](https://members.loria.fr/MOBerger/Enseignement/Master2/Exposes/beyer.pdf)

[//members.loria.fr/MOBerger/Enseignement/Master2/Exposes/beyer.pdf](https://members.loria.fr/MOBerger/Enseignement/Master2/Exposes/beyer.pdf)




Katie Evans (2015)

Outlier Identification in Model-Based Cluster Analysis (2015)

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4720172/>

 Simon Hawkins (2002)
Outlier Detection Using Replicator Neural Networks
<https://togaware.com/papers/dawak02.pdf>

 Huan Xu (2013)
Outlier-Robust PCA: The High Dimensional Case
http://users.ece.utexas.edu/~cmccaram/pubs/HRPCA_Journal.final.pdf

 Emmanuel J. Candes, John Wright (2009)
Robust Principal Component Analysis?
<https://statweb.stanford.edu/~candes/papers/RobustPCA.pdf>



Huan Xu (2009)

Robust PCA via Outlier Pursuit (2010)

<https://arxiv.org/pdf/1010.4237.pdf>