A review of outliers detection

Titouan Vayer

May 17, 2017

Sommaire

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 il apparaît comme étant contradictoire avec le reste des données.

Les trois grandes approches pour la détection d'outlier

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 la normalité 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 :

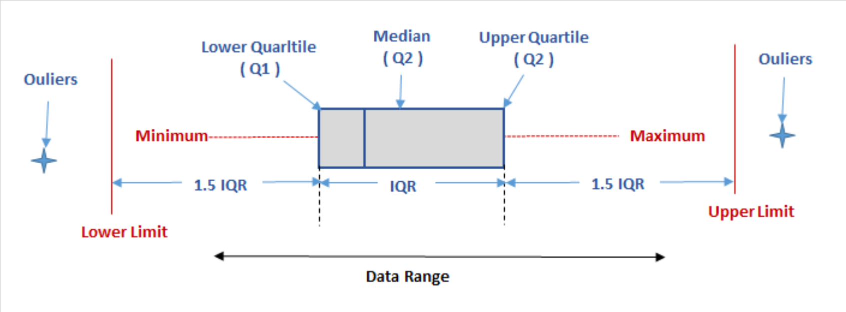
- **Le diagnostique** : on analyse les outliers et on les enlève ou pas
- **L'accomodation** : on garde les outliers quoiqu'il se passe

Introduction

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 :

- H0: Il n'y a pas d'outlier
- Ha: Il y a au moins un outlier

Le test statistique est le suivant :

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

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

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

ou alors

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

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

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

avec $t^2_{\frac{\alpha}{2N},N-2}$ 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 $X \sim \mathbb{N}(\mu, \Sigma)$ 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 :

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

Implémentation

La détection par la distance de Mahanalobis a été implémentée sur le github suivant [Titouan Vayer]
`https://github.com/bigtdu53/outlierdetection`

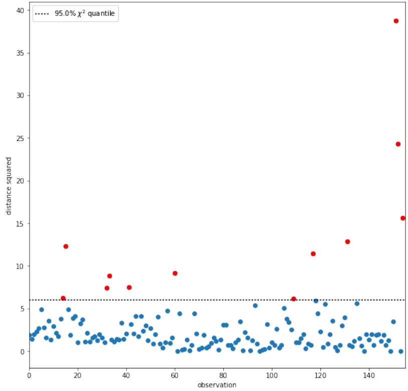
Les données utilisées ont été :

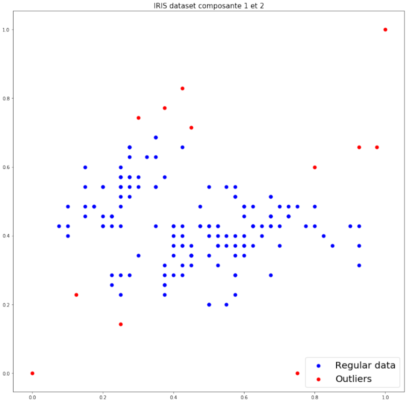
- Iris Dataset
- dtmcross201606.csv

L'algorithme présente deux incovénients :

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

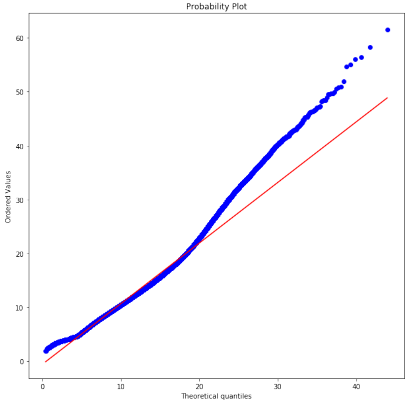
Outlier detection using Mahanalobis distance on Iris Dataset

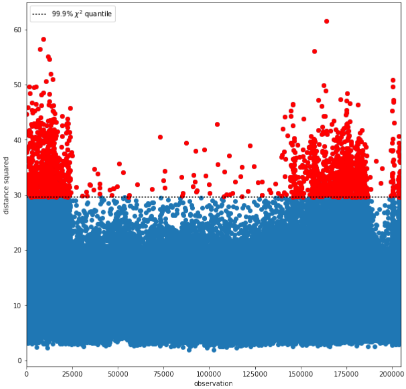




Outlier detection using Mahalanobis distance on dtmcross

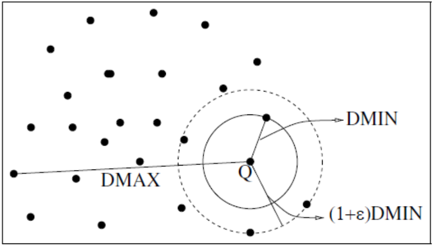
Dans le cas de dtmcross on a une très grand dataset (environ 1000 variables). Aussi pour pouvoir appliquer la méthode on passe d'abord par une étape (PCA) de réduction de dimension.





Limitation : Malédiction de la dimension

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



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

Le théorème de Beyer

On considère $X_1, ..., X_d \in \mathbb{R}^n$ de telle sorte que $\forall m \in [1, ..., d], X_{m,1}, ..., X_{m,n}$ 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

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

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

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

Proximity based techniques

Les techniques de "Porximity-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 compléxité 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.

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 défini 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 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 c'est algortihme, une fois les frontières de normalité mises en place, peut être exécuté en online.

Variante K-medoids

Une variante de l'algorithme existe en utilisant les K-medoids. L'algorithme est aussi un algorithme de clusterisation et est assez dans la même veine que les Kmeans, mais 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.

Implementation

La détection par K-meanss a été implémentée sur le github suivant [Titouan Vayer] `https://github.com/bigtdu53/outlierdetection`

Les données utilisées ont été :

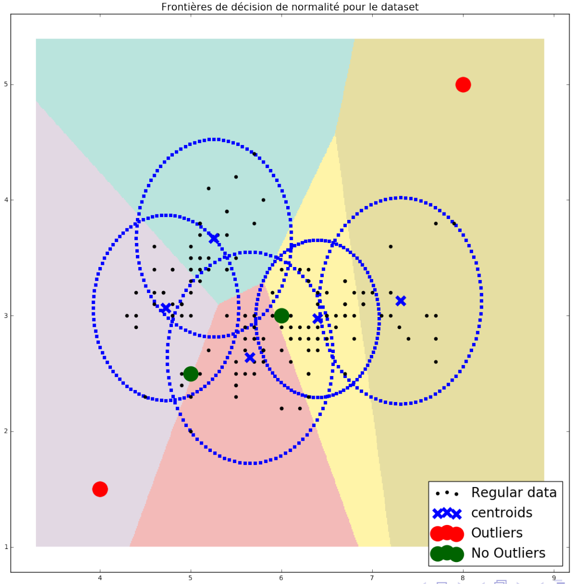
- Iris Dataset
- dtmcross201606.csv

L'algorithme présente deux incovénients :

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

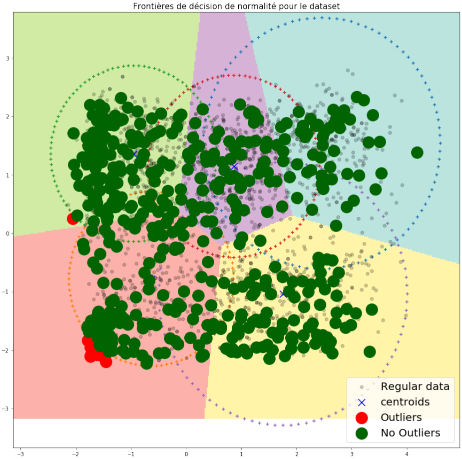
Une version parallélisée a été mise en place, ainsi qu'une version utilisant les K-medoids

Outlier detection using K-means distance on Iris Dataset



Outlier detection using Kmeans distance on dtmcross

Dans le cas de dtmcross on a une 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.



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/bigtdu53/outlierdetection

When is Nearest Neighbor Meaningful ? (1998)
When is Nearest Neighbor Meaningful ?
https://members.loria.fr/MOBerger/Enseignement/Master2/Exposes/beyer.pdf