Université Grenoble Alpes

Master SSD, BDA et IC
Enseignant : Jonathan El Methni
Apprentissage Statistique

Projet n°7 : détection d'anomalies

Détecter la présence éventuelles de données aberrantes ou atypiques est souvent une étape clé lors de la mise en forme d'un jeu de données. L'objectif de ce projet sera d'évaluer l'intérêt des méthodes de Local Outlier Factor et Isolation Forest dans ce contexte.

En visant à définir un critère numérique quantifiant le caractère atypique d'une observation vis à vis de son voisinage, ces approches "par observation" offrent une alternative intéressante aux approches plus classiques qui consistent à estimer, de manière paramétrique ou non-paramétrique, le support global de la distribution sous-jacente. Ces approches mettent néanmoins en jeu des hyperparamètres qui sont difficiles de régler a priori.

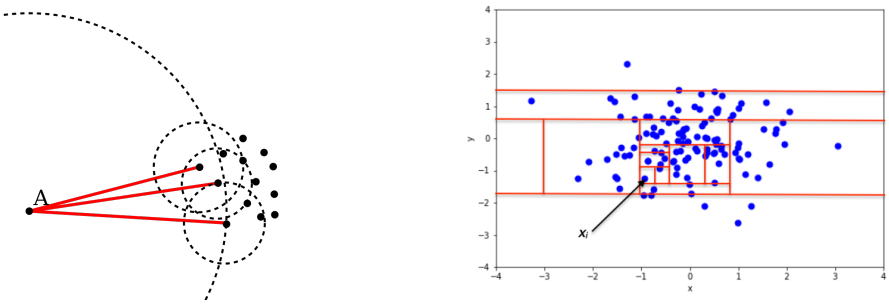

Figure 1 – Illustration des principes des algorithmes de "Local Outlier Factor" (gauche) et "Isolation Forest" pour l'identification de données atypiques - images tirées de `Wikipedia`.

L'objectif de ce projet sera d'apprendre à manipuler ces algorithmes et de comparer leurs performances quand on fait varier leurs hyperparamètres. Vous travaillerez pour cela sur un jeu de données dans lequel les données atypiques sont déjà identifiées, et vous évaluerez dans quelle mesure ces deux algorithmes sont capables de les retrouver.

Le jeu de données `outlier-dataset.Rdata` mettra en jeu $n = 6916$ observations vivant dans un espace de $p = 21$ dimensions, au sein desquelles 250 sont atypiques. Il prendra la forme d'une matrice `X` de dimension 6916×21 et d'un vecteur `y` indiquant si chacune de ces observations est atypique ou non (catégories `"normal"` et `"outlier"`).

Objectifs

L'objectif sera donc en premier lieu d'évaluer les performances des algorithmes Local Outlier Factor et Isolation Forest quand on fait varier leurs hyperparamètres :

— la valeur du paramètre k qui contrôle la taille du voisinage dans le Local Outlier Factor
— le nombre d'arbres à inclure dans la forêt dans l'approche Isolation Forest.

Ces deux méthodes fournissant en sortie un critère quantitatif reflétant le caractère atypique d'une observation, on mesurera la performances de ces méthodes en terme de courbe ROC, et donc de compromis entre vrais positifs (un point atypique détecté en tant que tel) et faux positifs (un point "normal" catégorisé comme atypique).

On évaluera également l'impact et l'intérêt d'appliquer au préalable deux transformations classiques des données : standardisation des variables et réduction de dimension par ACP.

Enfin, pour aller plus loin, on pourra consolider cette analyse en comparant les résultats obtenus par ces deux approches à ceux obtenus par des approches alternatives telles que les "One-Class SVMs" et une modélisation par Gaussienne multivariée.

Le minimum

Voici les consignes qui empêcheraient d'avoir la moyenne si elles ne sont pas respectées :

— Le compte-rendu ne doit pas faire plus de 6 pages (sans compter les éventuelles annexes).
— Il doit contenir le nom des auteurs, un titre explicite, une introduction et une conclusion.
— Il doit également contenir une brève description des algorithmes Local Outlier Factor et Isolation Forest, et du rôle joué par les deux hyperparamètres évoqués précédemment.
— Pour chaque méthode, un graphique illustrant l'impact de son hyperparamètre en terme de courbes ROC doit être présenté, et le compromis obtenu entre vrais positifs et faux positifs doit être commenté.
— L'impact du pré-traitement des données (standardisation et/ou réduction de dimension par ACP) doit être illustré et commenté.
— Enfin, on cherchera éventuellement à comparer ces résultats avec les résultats obtenus par One-Class SVMs et/ou l'utilisation de Gaussiennes multivariées.
— Le code mis en oeuvre doit apparaître, commenté un minimum, en annexe (et uniquement en annexe).

Bibliographie

— Pour la mise en oeuvre du Local Outlier Factor, on pourra par exemple utiliser l'implémentation disponible dans le package `dbscan` :

https://rdrr.io/cran/dbscan/man/lof.html

— Le package `isotree` propose une implémentation de l'algorithme des Isolation Forest :

https://rdrr.io/cran/isotree/