TP3 Valorisation des données
M1 Miage 2018-2019

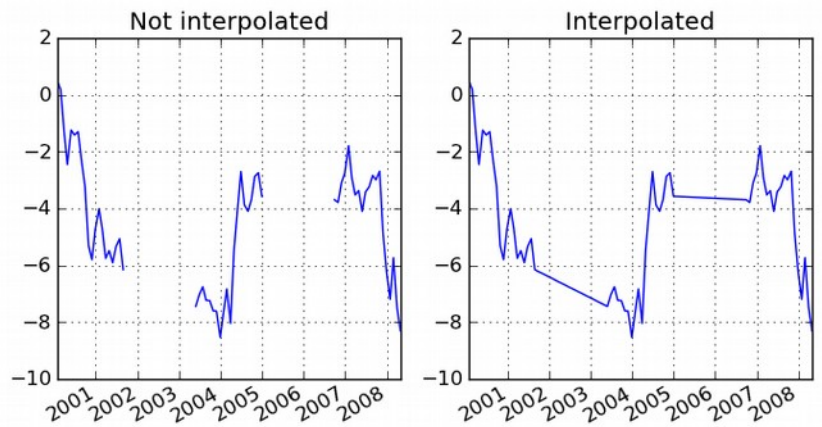
L'objectif de ce TP est de prétraiter le jeu de données **Beijing PM2.5** téléchargeable ici : https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data afin de l'utiliser pour faire du data mining. Ce fichier contient des données météorologiques mesurées toutes les heures entre 2010 et 2014 dans la ville de Beijing (voir description sur le lien). L'attribut à prédire est l'attribut *pm2.5* qui correspond au taux de particules fines mesuré dans l'air. Il s'agit d'un attribut numérique.
Le capteur de mesure des particules fines étant parfois indisponible, il existe des « trous » dans les données, c'est à dire des ensembles de valeurs contigues non renseignées. Nous allons donc dans un premier temps chercher à combler ces trous lorsque cela est possible. Puis nous allons transformer ces données afin de pouvoir faire de l'autoregression.

1) Remplacement des données manquantes.

Etant donné qu'il s'agit de données de type « série temporelle » (évolution de valeurs numériques au fil du temps), nous allons utiliser une stratégie d'interpolation linéaire décrite plus bas.

a) Ecrire un programme (C, java, python, ...) qui parcours les données en recherchant les trous, sachant qu'une donnée manquante est matérialisée par « NA ».
Pour chaque trou, on affichera :
- son numéro (ordre d'apparition dans le fichier 1, 2, 3, ...)
- le nombre de valeurs manquantes dont il est composé

b) Lorsqu'un trou est composé de moins de 15 valeurs manquantes, modifiez votre programme de façon à le combler à l'aide de la méthode de l'interpolation linéaire. Il s'agit de tracer une ligne entre la dernière valeur avant le trou et la première valeur après le trou.
Exemple :



La méthode à suivre est la suivante :
i) Calculer l'incrément i en divisant l'intervalle manquant par le nombre de valeurs manquantes +1.
ii) Remplacer chaque valeur manquante en commancant par la première et en rajoutant à chaque fois l'incrément.
Lorsqu'un trou est composé de plus de 15 valeurs, remplacez « NA » par « ? » de façon à pouvoir utiliser le jeu dans Weka plus tard.

c) Vérifiez qu'il ne reste plus de trous de plus de 15 valeurs manquantes dans le jeu de données en affichant à nouveau l'ensemble des trous présents et leur taille a)

2) Regression et transformation

Dans l'onglet classify de Weka, lorsque l'attribut de classe choisi est de type numérique, les algorithmes choisis effectueront de la régression (prévision de valeurs numériques). Lorsque cela n'est pas possible, ils seront grisés.

a) Lancez les algorithmes de regression suivants: Multilayer Perceptron, RandomTree et observez leur performances Mean Absolute Error (erreur moyenne absolue), Root Mean Square Error (racine carrée de l'erreur quadratique moyenne). Plus ces erreurs sont faibles, plus les performances sont bonnes.
Dans les paramètres, on choisira « percentage split %66 » et « preserve order for%spli t » (dans more option) afin d'apprendre le modèle sur les 66 % au début et tester sur le reste.

b) On cherche maintenant à faire de l'autoregression c'est à dire prédire la valeur d'un attribut à partir de ses anciennes valeurs. On considère donc que l'on veut prédire la valeur de pm2.5 qui sera mesurée dans 3 heures en tenant compte des 3 valeurs les plus récentes de pm2.5.
Pour cela construisez à l'aide d'un programme (C, Java, ...) un nouveau jeu en rajoutant à chaque ligne les attributs suivants :

pm2.5 5 heures plus tot (5 lignes avant)
pm2.5 4 heures plus tot (4 lignes avant)
pm2.5 3 heures plus tot (3 lignes avant)

Testez à nouveau les performances des mêmes algorithmes. Les performances sont-elles meilleures ?

c) Effectuez une sélection d'attributs dans Weka avec une méthode de votre choix et vérifiez si ces nouveaux attributs sont retenus. Relancez les algorithmes de regression avec les attributs sélectionnés et observez les performances.

d) Gardez uniquement les trois nouveaux attributs et supprimez tout les autres (sauf pm2.5) et observez à nouveau les performances.