NF26 - Data warehouse et outils décisionnels
Jean-Benoist Leger

Université de Technologie de Compiègne

Projet NF26

Données ASOS en Italie de 2005 à 2014

Rapport

Loïc Adam - Xiang Li
TD2 - Groupe 19-20

Printemps 2019

1 Introduction

Le but de ce projet est d'étudier un jeu de données volumineux, dans notre cas les données ASOS en Italie de 2005 à 2014. Afin d'étudier le jeu de données, nous devions concevoir et alimenter des bases de données. Une fois les données stockées, nous devions réaliser un ensemble de fonctions python afin de répondre à trois questions.

Tout d'abord, la partie 2 de ce rapport introduit notre jeu de données, en mettant en avant les problèmes que nous avons observés. Ensuite, la partie 3 présente les bases de données que nous avons employées. Enfin, la partie 4 détaille les trois questions et les solutions que nous avons proposés, tout en exposant leurs limites.

2 Présentation du jeu de données

Ce produit étudie l'ensemble de données provenant de capteurs automatiques relevant des données météorologiques. Ce système s'appelle *Automated Surface Observing System* (ASOS). Ces données servent principalement en météorologie et en aviation. Les relevés se sont généralement toutes les heures, mais cela peut changer si le temps évolue.

Les variables dans le jeu de données sont les suivantes :

- La station, représentée par quatre caractères en Italie. Le système italien est composé de 104 stations (+1 en Irak pour une raison obscure).
- La date à laquelle l'observation a été réalisée, comprenant l'année, le mois, le jour, l'heure et la minute.
- La longitude et la latitude de la station.
- La température de l'air en Fahrenheit vers deux mètres.
- La température de rosée (température à laquelle l'air devient saturé en vapeur d'eau) en Fahrenheit vers deux mètres.
- L'humidité relative.
- La direction en degrés du vent en se basant sur la direction nord.
- La vitesse du vent en nœuds.
- Les précipitations pendant une heure.
- La pression en pouces et celle au niveau de la mer en millibars.
- La visibilité en milles terrestres.
- La vitesse des rafales de vent en nœuds.

- La couverture du ciel en nuage (la quantité) en quatre niveaux ainsi que la hauteur des nuages en pieds sur les mêmes quatre niveaux.
- Code météo.
- Température ressentie en Fahrenheit.
- Accumulation de givre sur une, trois et six heures en pouces.
- La vitesse (nœuds), la direction (degrés) et l'instant de mesure de la rafale de vent la plus forte.
- Les données non traitées en format METAR.

Le jeu de données comporte 3 175 968 enregistrements et donc 31 variables qui sont pour la quasi-totalité mesurées avec le système impérial. Bien que l'étude soit de 2005 à 2014, il n'y a pas de données avant 2011.

Néanmoins, le jeu de données est assez incomplet. En téléchargeant les données, la plupart des manques sont représentés par « null », mais certains sont représentés par trois espaces. Lors de l'insertion de nos données sur nos bases de données, nous avons tenu compte de cette erreur pour bien enregistrer ces manques en tant que « null » afin de ne pas se retrouver avec des surprises lors de l'exploitation de données.

3 Présentation des bases de données

Afin de répondre aux questions, nous avons construit deux bases de données orientées colonnes avec l'outil Cassandra : AsosItalyStation et AsosItalyTime. Chaque base de données contient toutes les variables, mais la partition et le tri sont différents. De plus, chaque donnée de chaque base est présente au moins deux fois sur le réseau.

3.1 AsosItalyStation

- $C_{partition}$ = station, année.
- C_{tri} = mois, jour, heure, minute.

Cette base permet de répondre à la question 1 en ayant toutes les observations en rapport avec une station.

La clé de partition contient l'année pour éviter que chaque partition grandisse au cours du temps.

3.2 AsosItalyTime

- $C_{partition}$ = année, mois.
- C_{tri} = date d'observation, longitude, latitude.

Cette base permet de répondre aux questions 2 et 3 en ayant toutes les observations par rapport à un instant t ou à une période de temps.

La clé de partition ne contient que l'année et le mois, car ce grain nous paraît suffisant (la journée pourrait être ajoutée à la limite) et le nombre de partitions ($4 \times 12 = 48$) est suffisant par rapport au nombre de serveurs (3) de notre réseau.

Quant à la clé de tri, elle comprend une date (type *timestamp*). Cela nous permet pour la troisième question, où il faut sélectionner des périodes de temps, de prendre les partitions qui nous intéressent (année et mois) tout en vérifiant que les données sont dans l'intervalle.

4 Réponses aux questions

Pour chaque question, l'interface est très minimaliste : l'utilisateur doit saisir une fonction avec les paramètres souhaités.

4.1 Question 1

La première question exige pour un point donné de l'espace de pouvoir avoir un historique du passé, en mettant en avant la saisonnalité et les écarts en lien avec celle-ci. La fonction est la suivante :

drawStation(latitude, longitude, année)

La première chose que le script python fait est de trouver la station la plus proche à partir des coordonnées. Pour cela, la liste des stations avec leurs coordonnées est présente sur un fichier csv crée à partir du jeu de données initial. Le calcul des distances se fait à partir du théorème de Pythagore.

Une fois la station trouvée, le script réalise quatre graphiques qui nous paraissent pertinents :

- La température par saison pour une année donnée sous forme de boîte à moustache (permet de voir les valeurs extrêmes).
- Le minimum, la moyenne et le maximum de la température par mois pour une année sur une courbe.
- La température minimale, moyenne, maximale et médiane par saison pour une année sur une courbe.

- La température moyenne par mois pour chaque année de notre jeu de données sur une courbe.

Dès lors, on récupère les partitions correspondant à l'année (ou les années pour l'historique) et à la station choisie. Pour chaque requête, on utilise un MapReduce pour les calculs parallèles. On utilise ainsi cette stratégie pour obtenir les saisons, le minimum, le maximum, la moyenne ainsi que la médiane. Les MapReduce utilisés sont vus en cours et en TD.

Bien que la température permet de voir une certaine saisonnalité, il serait pertinent de voir beaucoup plus de variables, par exemple l'humidité, les précipitations (quand elles sont disponibles), la vitesse du vent... Néanmoins, soit nous devions préparer plus de cartes à l'avance, soit il faut laisser à l'utilisateur la possibilité de rentrer la variable tout en vérifiant que la variable choisie est bien quantitative. Par conséquent, la réponse apportée n'est pas assez complète pour être pertinente dans un contexte professionnel.

4.2 Question 2

La deuxième question est de pouvoir pour un instant t donné d'obtenir une carte représentant n'importe quel indicateur. La fonction est la suivante :

drawmap(année, mois, jour, heure, minute, variable)

La variable à indiquer correspond au nom des variables dans la base de données. Ainsi, la base de données est questionnée afin de récupérer tous les enregistrements qui correspondent à l'instant précis, en ne récupérant que la longitude, la latitude et la variable concernée.

Une fois que les données sont récupérées dans un générateur, l'intégrité des données est vérifié : à chaque fois que la variable que l'on chercher à représenter est « null », l'enregistrement n'est pas inséré dans les listes représentants la longitude, la latitude et la valeur. À ce stade, les listes font chacune au maximum 80 données, ce qui ne pose pas de problème pour la mémoire.

Si aucun enregistrement n'a une valeur non « null », aucune carte n'est dessinée et l'utilisateur est averti. Dans le cas contraire, une carte représentant les données est créée grâce au module Basemap.

Une limite à cette question : la carte n'est pas parfaite et il est difficile de voir certaines variables (comme les données METAR) ou même certains points (à cause des frontières). De plus, sachant que les stations ne fonctionnent pas toujours à intervalle régulier, nous n'autorisons pas de visualiser pour une heure précise ou pour une journée précise.

4.3 Question 3

L'objectif de la question 3 est de réaliser pour une période de temps donnée un partitionnement de l'espace et qu'il soit représenté. La fonction python est la suivante :

partitionMap(date de départ, date de fin, K)

Le script se base sur une année et un mois de départ sous la forme 'année - mois', sur une date de fin sous le même format, ainsi que sur le nombre de partitions voulus K. Ce grain de date est peut-être trop fin, mais permet de faire déjà de bonnes partitions. Les données sont récupérées en fonction de leur périodicité. Les calculs sont faits en parallèle avec calcul de la moyenne, du minimum, du maximum et de la variance.

L'algorithme des k-moyennes, que nous avons réalisé nous-même, se base sur cinq paramètres que nous avons choisis au préalable : la température, la température de rosée, l'humidité relative, la vitesse du vent et la pression. Ces variables sont plutôt complètes (peu de « null », contrairement à la précipitation...) et sont en général un bon indicateur de la météo.

Les résultats sont présentés sur une carte au format html, où chaque station est représentée par un symbole colorié en fonction de la classe. Il est possible de cliquer sur le point pour voir chaque paramètre.

Notre algorithme ne prend en compte que 5 variables, ce qui n'est pas idéal. Notre choix a pour but de régler le problème des variables presque vides, ainsi que le problème des variables qualitatives (non-utilisables telles quel). Pour les variables vides, on aurait pu faire la moyenne de la colonne ou déterminer les valeurs manquantes par des règles de calcul. Pour les variables qualitatives, il faudrait des encodages particuliers (one-hot par exemple). De plus, il faudrait calculer le K optimal pour représenter les données.

5 Conclusion

Bien que notre jeu soit très volumineux, avec beaucoup de variables et de manques, nous avons pu créer deux bases de données nous permettant de répondre correctement aux trois questions qui nous sont posées. Nos solutions se basent beaucoup sur le calcul en parallèle et évitent de gaspiller la mémoire grâce aux générateurs (pas de chargement des données brutes en mémoire).

Nous avons aussi mis en avant des limites sur chacune de nos réponses. Il est possible d'améliorer notre travail en offrant une réelle interface, en exploitant plus de données (plus de cartes dans la question 1, plus de paramètres pour la question 3) et en corrigeant les données (les « null », station en Irak).