Projet d'analyse des données *ASOS* Italie de 2011 à 2014

Claire Guyot - Haifei Zhang
TD2-5-6

1 Introduction

L'objectif de ce compte-rendu de projet est de mettre en évidence les résultats de différentes méthodes de visualisation à un jeu de données réelles : les données ASOS de l'Italie entre 2011 et 2014 (disponibles ici). Il s'agit des données météo récoltées par différentes stations d'observation dispersées au sein de l'Italie. Nous allons entre autres nous intéresser à trois problématiques distinctes.

2 Choix des données

Initialement, nous devions travailler sur les données de l'Italie entre 2002 et 2011. Cependant, après avoir obtenu un aperçu des données, nous nous sommes rendus compte qu'il y avait anormalement peu de données. En effet, aucune donnée n'est disponible avant l'année 2011. Nous avons jugé peu pertinent de réaliser ce projet avec ce jeu de données puisque nous allons nous intéresser entre autres à des saisonnalités et des comparaisons de saisonnalité.

Ainsi, nous avons choisi de travailler sur les données entre 2011 et 2014 pour lesquelles il y a suffisamment de données. Elles ne sont également pas en trop grandes quantités, ce qui permet un traitement suffisamment rapide.

3 Description des données

Avant toute chose, nous avons étudié la structure des données et la signification des différentes variables afin d'en donner une interprétation adéquate *via* des visualisations. L'explication des différentes variables est visible en annexe A, page 6.

Aussi, en étudiant les données, nous remarquons que nous avons 105 stations, ainsi que 3175968 observations pour 31 variables.

Nous avons par ailleurs voulu déterminer la quantité de données disponibles pour chaque variable et chaque année. Nous avons pour cela réalisé un histogramme du nombre de valeurs manquantes (voir Fig. 1). Nous pouvons remarquer qu'il n'y a aucune donnée manquante pour *station*, *lat*, *lon* et *valid*. Il manque également très peu de données pour *tmpf*, *alti*, *dwpf*, *feel*, *relh*, *sknt* et *vsby*.

Grâce à cela, nous avons identifié plusieurs varibles d'intérêts qui seront particulièrement pertinente pour notre analyse. Il s'agit des variables : *station*, *valid*, *lat*, *lon*, *tmpf*, *dwpf*, *relh*, *drct*, *sknt*, *feel* et *vsby*. Celles-ci nous permettent de comparer les variations de température ou bien la direction du vent par année et par station par exemple.

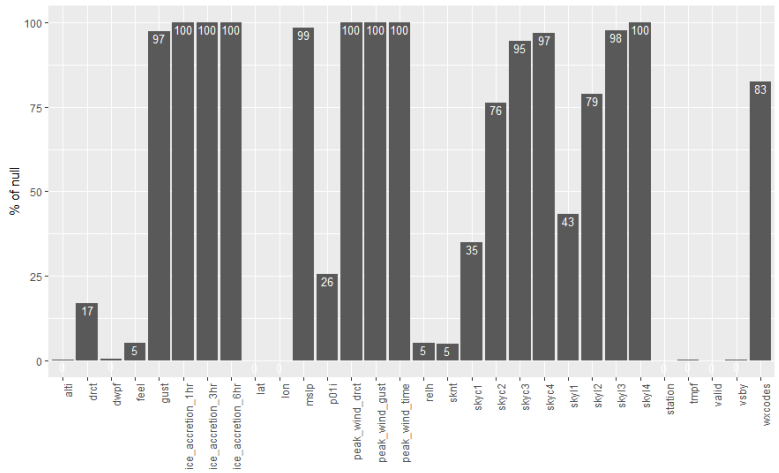

FIGURE 1 – Pourcentage de données manquantes pour chaque variable du jeu de données

4 Stockage des données

Afin d'exploiter les données à notre disposition, nous avons créé des stockages orientés colonne. Pour les construire, nous avons tout d'abord dû établir les clés de partitionnement et de tri.

4.1 Identification des clés

Pour identifier les différentes clés de partitionnement et de tri, nous avons analysé les besoins que les différentes problématiques suggèrent.

Les points clés de la première problématique sont mis en évidence ci-après : *Pour un **point de l'espace** donné, je veux pouvoir avoir un **historique** du passé, avec des courbes adaptées. Je veux pouvoir mettre en évidence la **saisonnalité** et les **écarts de saisonnalité**.*

Les points clés de la deuxième problématique sont mis en évidence ci-après : *Pour un **instant** donné, je veux pouvoir obtenir une carte me représentant n'importe quel indicateur d'une **station**.*

Les points clés de la troisième problématique sont mis en évidence ci-après : *Pour une **période** donnée, je veux pouvoir **clusteriser l'espace** et représenter cette clusterisation.*

Nous observons que les problématiques 2 et 3 ne nécessitent pas des clés de partionnement différentes. Nous avons ainsi identifié trois stockages distincts :

- **Stockages géographiques**
  - **Stockage des stations par coordonnées géographiques**
    - *Clé de partitionnement :* latitude, longitude
    - *Clé de tri :* station
  - **Stockage des autres données par coordonnées géographiques**
    - *Clé de partitionnement :* latitude, longitude
    - *Clé de tri :* année, mois, jour, heure, minute
- **Stockage temporel**
  - *Clé de partitionnement :* instant de l'observation
  - *Clé de tri :* station, latitude, longitude

La première problématique utilisera les deux premiers stockages, et les problématiques 2 et 3 utiliseront le dernier.

En effet, l'utilisateur peut vouloir entrer des coordonnées géographiques afin de définir le point de l'espace de la première problématique. Il peut ne pas connaître le nom de la station la plus proche d'un point donné. Ainsi, un premier stockage permet d'accéder au nom de la station en fonction de ses coordonnées géographiques, et un second permet d'accéder aux données présentes aux données géographiques disponibles.

4.2 Création des tables

Pour créer nos trois tables dans *Cassandra* correspondant à nos trois stockages, nous avons utilisé le même *namespace* : `claire_haifei_projet`. La première est nommée `stations`, la deuxième `asos1` et la dernière `asos2`.

4.3 Insertion des données

Pour insérer les données dans nos tables de stockage, nous avons créé un générateur pour chaque table. Ainsi, nous avons pu lire notre fichier `asos.csv` de base contenant l'ensemble de nos données, et insérer les variables d'intérêt pour chaque stockage.

Dans le cas de `asos1`, un de nos stockages géographiques, nous avons séparé la donnée concernant l'instant d'observation en cinq variables distinctes : année, mois, jour, heure, minute afin de disciminer au mieux les partitions.

Par ailleurs, pour `asos1` et `asos2`, nous avons insérer les données correspondant aux variables *tmpf*, *dwpf*, *relh*, *drct*, *sknt*, *alti*, *vsby*, *skyc1*, *wxcodes*, *feel* et *metar*. En effet, celles-ci correspondent globalement aux variables ayant le moins de données manquantes et nous permettent de réaliser différentes analyses.

5 Utilisation des données

5.1 Librairies utilisées

Afin de répondre aux différentes problématiques, nous avons utilisé différentes librairies :

— ***cassandra.cluster*** permettant de se connecter aux *clusters* du serveur ;

— ***findspark*** permettant d'importer *pyspark* en tant que librairie ;

— ***pyspark*** permettant d'analyser un grand nombre de données simplement ;

— ***numpy*** permettant de manipuler des tableaux et matrices simplement ;

— ***matplotlib*** permettant de visualiser les données de différentes manières, notamment grâce à des courbes ;

— ***pandas*** permettant de manipuler des structures particulières de données ;

— ***folium*** permettant de visualiser les données géographiques sur une carte.

5.2 Problématique 1

Pour rappel, la problématique est la suivante : *Pour un point de l'espace donné, je veux pouvoir avoir un historique du passé, avec des courbes adaptées. Je veux pouvoir mettre en évidence la saisonnalité et les écarts de saisonnalité.*

Nous avons ainsi réalisé différentes visualisations donnant un aperçu des différences entre les années pour une station précise.

Par exemple, il est possible d'observer la variation de la température maximale par jour pour une année et une station particulière, ou encore les maximums et minimums de température par mois sur l'ensemble des années pour une station (voir Fig. 2).

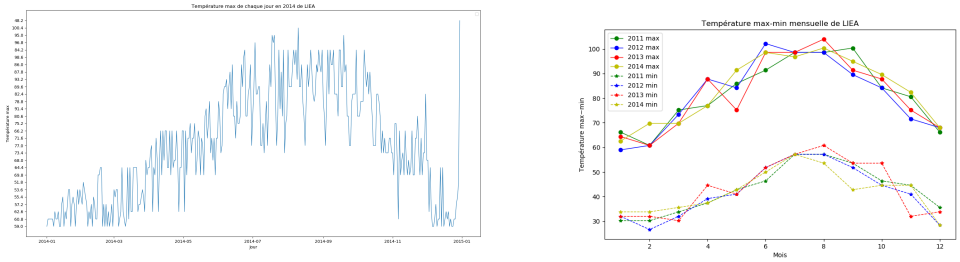

FIGURE 2 – Température maximale par jour en 2014 pour *LIEA* (à gauche), Comparaison des minimums et maximums de temprérature entre 2011 et 2014 pour *LIEA* (à droite)

Aussi, il est par exemple possible de visualiser les moyennes de température trimestrielles sur l'ensemble des années pour une station donnée, ou encore les directions du vent par trimestre pour une année et une station particulière (voir Fig. 3).

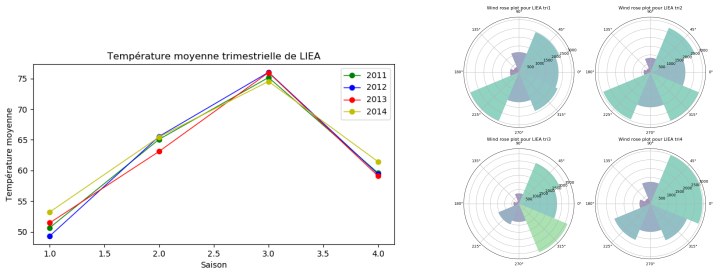

FIGURE 3 – Moyennes des températures trimestrielles entre 2011 et 2014 pour *LIEA* (à gauche), Direction du vent pour chaque trimestre de 2014 pour *LIEA* (à droite)

5.3 Problématique 2

Pour rappel, la problématiques est la suivante : *Pour un instant donné, je veux pouvoir obtenir une carte me représentant n'importe quel indicateur d'une station.*

Pour répondre à cette problématique, nous avons permis d'observer les différentes stations à un instant données avec pour chacune, les observations réalisées à cet instant précis. Aussi, il est possible de visualiser les différences de température par exemple à ce même instant entre les différentes stations en une *heat map* (voir Fig. 4).

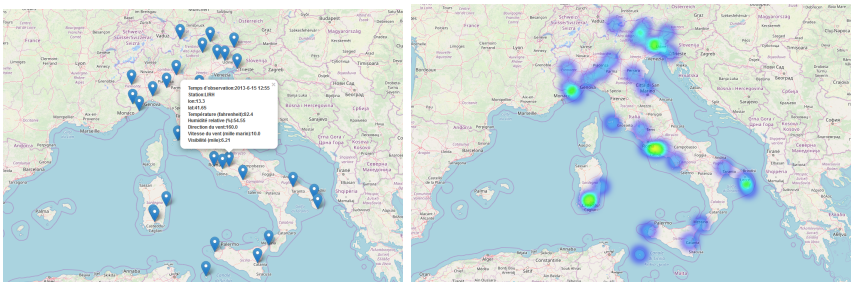

FIGURE 4 – Informations de chaque station le 15/06/2014 à 12h55 (à gauche), Température observée à chaque station le 15/06/2014 à 12h55 (à droite)

5.4 Problématique 3

Pour rappel, la problématique est la suivante : *Pour une période donnée, je veux pouvoir clusteriser l'espace et représenter cette clusterisation.*

Enfin, pour cette dernière problématique, nous avons réalisé une *clusterisation* de l'espace qui peut être visualisée à l'aide de points (représentant les stations) de couleurs différentes (chacune correspondant à une classe) sur la carte de l'Italie. Les *clusters* sont calculés à l'aide de l'algorithme des *k-means* (voir annexe B, page 6).

Nous avons choisi la température et l'humidité relative comme indicateurs du clustering. Après avoir sélectionné toutes les données pour une période donnée, nous calculons les valeur moyenne, écart-type, maximum et minimum de chaque indicateur pour chaque station par le biais d'un *map reducing*. Nous obtenons ainsi à partir de $(station, instant, tmp, relh)$:

$(station, tmp\_moy, tmp\_std, tmp\_min, tmp\_max, relh\_moy, relh\_std, relh\_min, relh\_max)$

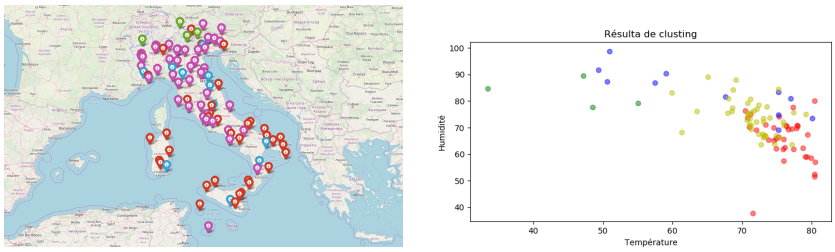

FIGURE 5 – Stations réparties en *clusters* selon la température et l'humidité relative observées

6 Conclusion

Pour conclure, nous avons réussi à comprendre les différentes données et à en donner une visualisation adéquate. De plus, nous avons réalisé un *clustering* des différentes stations en fonction de certains indicateurs, qui semble pertinent.

En revanche, nous aurions plusieurs points à améliorer. Tout d'abord, nous avons remarqué tardivement qu'il y avait une station *QLQ* qui ne se trouvait pas en Italie et qui a potentiellement faussé nos analyses. Ensuite, le *clustering* n'est réalisé qu'avec la température et l'humidité relative pour l'instant. Enfin, le *map reducing* pour le *clustering* n'est pas optimal puisqu'il ne prend pas en compte la temporalité. Nous n'avons en effet pas réussi à réaliser un *map reducing* multi-dimensionnel.

A Aperçu des données

TABLE 1 – Variables du jeu de données ASOS

station	id de la station
valid	timestamp de l'observation
tmpf	température de l'air (F)
dwpf	température au point de rosée (F)
relh	humidité relative (%)
drct	direction du vent (deg. du Nord)
sknt	vitesse du vent (noeud)
p01i	heure de précipitation
alti	altimètre de pression (pouce)
mslp	pression au niveau de la mer (mbar)
vsby	visibilité (mile)
gust	rafale de vent (noeud)
skycx	couverture du niveau x du ciel (x = 1,2,3,4)
skylx	altitude du niveau x du ciel (pied) (x = 1,2,3,4)
wxcodes	codes du temps actuel
feel	température ressentie (F)
ice_accretion_xhr	accumulation de glace en x heures (pouce) (x = 1,3,6)
peak_wind_gust	rafale de vent maximale (noeud)
peak_wind_drct	direction de la rafale de vent maximale (deg. du Nord)
peak_wind_time	instant de la rafale de vent maximale
metar	autres observations non traitées (format metar)

B Algorithme des *k-means*

Input : data, K
Output : (station, classe)

1. Sélectionner K points aléatoires à partir de *data* comme points centraux initiaux de chaque classe
2. **Do :**
   (a) Calculer les distances entre chaque station et chaque centre
   (b) Renvoyer la classe avec la plus petite distance en tant que classe de la station
   (c) Calculer les nouveaux points centraux en fonction des résultats du *clustering*
   While : diff(Vieux centres, nouveaux centres) $> \epsilon$
3. Calculer le résultat final du *clustering*