0310120 NCTU EE 陳家煒

Data Mining HW2

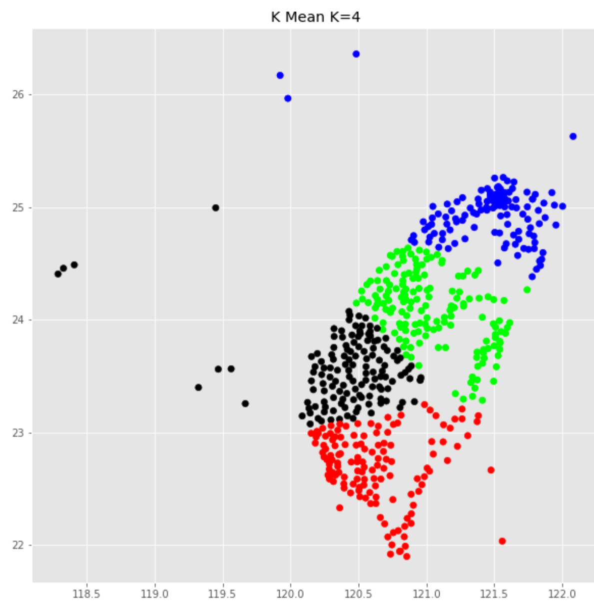
Spatial Clustering:

● use geometric information of locations(氣象觀測站) to do clustering

Brief take a look of the data:

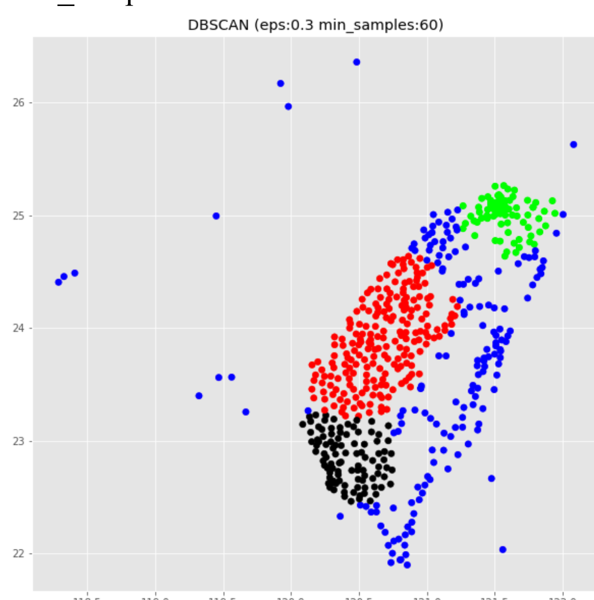
	站名	經度	緯度
0	五分山雷達站	121.7812	25.0712
1	板橋	121.4420	24.9976
2	淡水	121.4489	25.1649
3	鞍部	121.5297	25.1826
4	臺北	121.5149	25.0377

Apply K-Means with K = 4



K=4, K-Means approach separate the data into nearly north, center and south, but cannot generate east part

DBSCAN with eps = 0.3 min_samples=60



DBSCAN is a density-based algorithm, thus it more complicate to fine-tune parameter in order to find desire separated cluster. Note that blue dot stands for noise

Evaluation:

Since these clustering tasks don't have such as ground truth, we can only use internal evaluation to measure how good is our clustering. As a result, I choose silhouette score to evaluate the result.

> The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.
>
> $$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \qquad -1 \le s(i) \le 1$$

K-Means silhouette score: 0.39218
DBSCAN silhouette score: 0.23916

However, this approach is only suitable for distance-based algorithm like K-Means, as to DBSCAN, its density-based property might decrease the score of silhouette.

Temporal Clustering:

- Use temperature and taipower data from 2016/10/01 to 2017/06/30 (272 days).

Brief take a look of the data:

- Temperature of Taipei from 2016/10/01 to 2017/06/30 every day from 9:00 to 18:00
- NorthSupply from 2016/10/01 to 2017/06/30 every day from 9:00 to 18:00

Timestamp	Temp
2016-10-01 09:00:00	30.5
2016-10-01 10:00:00	31.4
2016-10-01 11:00:00	32.2
2016-10-01 12:00:00	32.6
2016-10-01 13:00:00	31.8
2016-10-01 14:00:00	31.9
2016-10-01 15:00:00	31.8
2016-10-01 16:00:00	31.1
2016-10-01 17:00:00	29.9
2016-10-01 18:00:00	28.9
2016-10-02 09:00:00	30.9
2016-10-02 10:00:00	32.1
2016-10-02 11:00:00	33.6
2016-10-02 12:00:00	34.5
2016-10-02 13:00:00	34.7
2016-10-02 14:00:00	34.8

Datetime	NorthSupply
2016-10-01 09:00:00	813.9
2016-10-01 10:00:00	885.9
2016-10-01 11:00:00	943.0
2016-10-01 12:00:00	907.5
2016-10-01 13:00:00	901.7
2016-10-01 14:00:00	920.8
2016-10-01 15:00:00	919.1
2016-10-01 16:00:00	882.2
2016-10-01 17:00:00	873.4
2016-10-01 18:00:00	876.7
2016-10-02 09:00:00	683.0
2016-10-02 10:00:00	710.2
2016-10-02 11:00:00	740.3
2016-10-02 12:00:00	766.2
2016-10-02 13:00:00	779.2
2016-10-02 14:00:00	777.5

Merge into 10-dimension data:

- **Temperature**

```
[[ 30.5   31.4   32.2 ...,   31.1   29.9   28.9]
 [ 30.9   32.1   33.6 ...,   31.6   30.2   28.6]
 [ 30.1   31.2   32.2 ...,   31.2   29.8   29. ]
 ...,
 [ 30.2   32.9   33.2 ...,   32.6   31.6   30.8]
 [ 31.1   32.2   32.  ...,   31.9   31.4   31.4]
 [ 32.3   34.2   34.9 ...,   27.    27.4   27.4]]
```

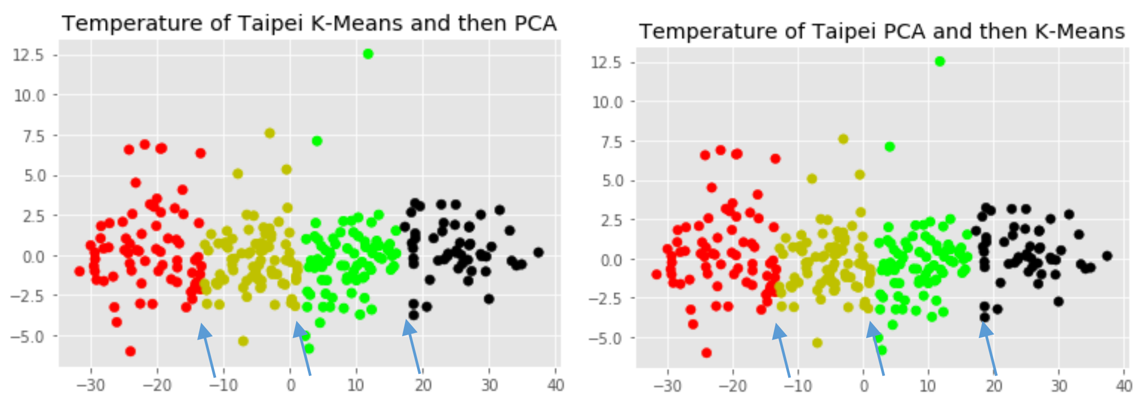
- **NorthSupply**

```
[[  813.9,   885.9,   943. , ...,   882.2,   873.4,   876.7],
 [  683. ,   710.2,   740.3, ...,   806.8,   784.1,   827.9],
 [  827.9,   827.9,   827.9, ...,  1114.3,  1094.9,  1107.7],
 ...,
 [ 1084.3,  1102.8,  1194. , ...,  1195.9,  1180.1,  1170.6],
 [ 1110.4,  1152.1,  1187. , ...,  1172.5,  1095.1,  1076.4],
 [ 1058.2,  1092.1,  1149.6, ...,  1169.5,  1174.7,  1166.1]])
```

- **Temperature**

In order to visualize the high dimension clustering result, I apply Principal Component Analysis (PCA) to reduce dimension from 10-dim to 2-dim. Also, I try to change the order of applying clustering and PCA. Surprisingly, the results of two different order are identical.

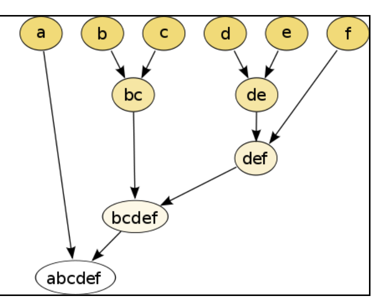
Apply K-Means with K = 4



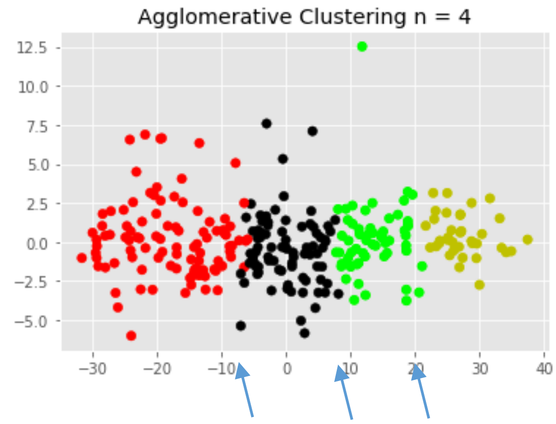
When applying DBSCAN algorithm on this dataset, however, it cannot separate the data as good as K-Means, so I turn to use another clustering algorithm — Agglomerative Clustering, which is a type of hierarchical clustering.

In general, the merges and splits are determined in a greedy manner. The results of hierarchical clustering are usually presented in a dendrogram.

Agglomerative: This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.



Apply Agglomerative Clustering with cluster number = 4

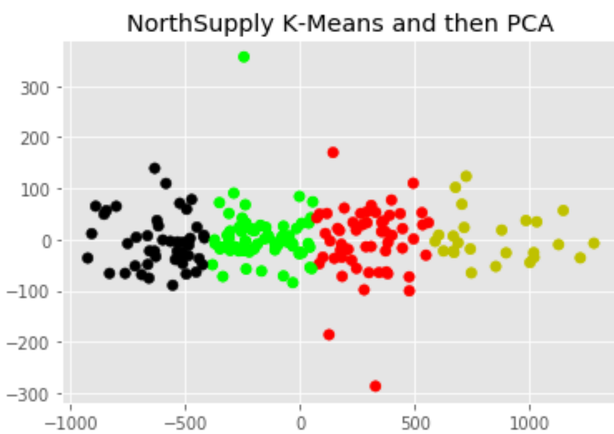


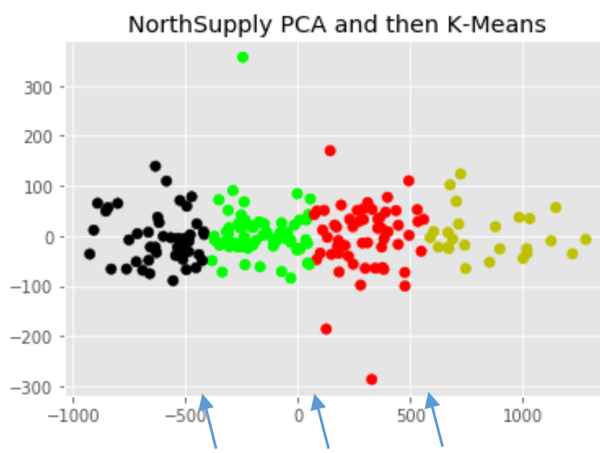
Evaluation:
K-Means silhouette score: 0.47881
Agglomerative Clustering silhouette score: 0.45043

Observation:
It's obvious that using K-Means clustering result in a more uniform distribution of labels in the graph, as to Agglomerative Clustering, the red label's area seems larger than others quite a lot.

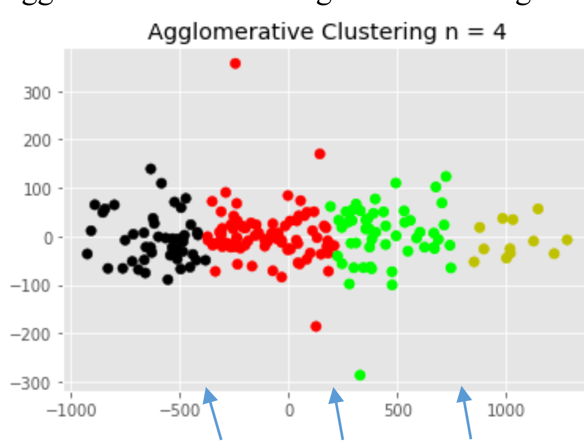
- **NorthSupply**

  Apply K-Means with K = 4





Apply Agglomerative Clustering with clustering number = 4



Evaluation:

K-Means silhouette score: 0.45560
Agglomerative Clustering silhouette score: 0.44315

Observation:

It's obvious that using K-Means clustering result in a more uniform distribution of labels in the graph, as to Agglomerative Clustering, the cyan label's area seems smaller than others quite a lot. Also the split values of each labels are variable between K-Means and Agglomerative approach.

Reference:

https://en.wikipedia.org/wiki/Silhouette_(clustering)
https://en.wikipedia.org/wiki/Hierarchical_clustering
http://scikit-learn.org/stable/modules/clustering.html
http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html