**MDA\_HW3**

102062130 劉鳳軒

1. **Overview架構：**  
   (1) 先把centeroid資料的檔案在本地用filesystem吃進來(一開始是c1.txt 或c2.txt，之後會用新的centroid-r-00000)， 之後用conf.setDouble的方式將10個centroid的存進conf中(其中name property 為 C\_i\_j ，i 為centroid 編號、j 為該centroid 的feature編號)  
   (2) 進入Map Reduce 產生出新的centroid 檔案(centroid-r-00000)，同時產生一個cost檔案紀錄每個cluster的cost(cost-r-00000)  
   (3) 把cost-r-00000中每個cost加起來即是該次的總cost，並記錄在本地的cost.csv中(一次一筆，最後會有20筆)  
   (4) 回到1並做20次

檔案說明:

Kmeans.java: kmeans code

Cost\_c1\_E.xlsx: 用c1 做Euclidean distance 20次的結果(附圖)

Cost\_c2\_E.xlsx: 用c2做Euclidean distance 20次的結果(附圖)

Cost\_c1\_M.xlsx: 用c1 做Manhattan distance 20次的結果(附圖)

Cost\_c2\_M.xlsx: 用c2 做Manhattan distance 20次的結果(附圖)

1. **Mapper**Input: LongWritable, Text

Output: IntWritable, Text  
let point a has features a1, a2, …, a58

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Key | | | Value | |
| **Input** | Type | Format | | Type | Format |
| LongWritable | |  | IntWritable | a1, a2, …., a58 |
| **Output** | IntWritable | | Format | IntWritable | Format |
| IntWritable | | centroidID | Text | a1, a2, …., a58, distance(centroid, a) |

將一個個point 讀進來後，對每個centroid 做距離計算(用Euclidean 或Manhattan)，並把該point分到最近的centroid的cluster中，並以centorid id 為key, 該point 的features加上最近distance為value 輸出。

1. **Reducer**  
   Input: IntWritable, Iterable< Text >

Output: Null, Text

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Key | | | Value | |
| **Input** | Type | Format | | Type | Format |
| IntWritable | | centroidID | Text | **Iterable**<( a1, a2, …., a58, distance(centroid, a))> |
| **Output1:**  New Centroids | Type | | Format | Type | Format |
| NullWritable | | Null | Text | 58 float values (new centroid) |
| **Output2:**  Costs per Cluster | Type | | Format | Type | Format |
| NullWritable | | Null | Text | New Cost |

用cluster中的每個點算出新的centroid(取平均)並把每個點的distance加起來成為cost，最後把centroid輸出一個檔案，  
cost輸出到另一個檔案

1. 問題討論
   * Euclidean 中用c1跟c2差別  
     c1:

C2:   
  
可以看到c2的曲線更平滑地收斂，且下降的比率也高很多。  
不過最後是c1的cost較低，代表平均而言cluster內的相似度更高(歸類更精確)，所以較佳。

* + Manhattan 中用c1跟c2差別  
    c1:   
      
    c2:  
      
    同樣地，可以看到c2的曲線更平滑地收斂，且下降的比率也高很多。  
    不過最後是c1的cost較低，代表平均而言cluster內的相似度更高(歸類更精確)，所以較佳。