Data Mining HW4 Report

0256513梁志彬

1. Problem definition:

給定product的rating matrix以及一些有關product的相關檔案，詢問在rating matrix中打問號的entry值，最後根據RMSE去作performance evaluation。

1. Related works:

Berkhin, Pavel. "A survey of clustering data mining techniques." *Grouping multidimensional data*. Springer Berlin Heidelberg, 2006. 25-71.

介紹各種分群演算法的survey paper

Guha, Sudipto, Rajeev Rastogi, and Kyuseok Shim. "CURE: an efficient clustering algorithm for large databases." *ACM SIGMOD Record*. Vol. 27. No. 2. ACM, 1998.

提出cure這個分群演算法的paper，裡面有詳細介紹該演算法、實作的細節，以及加速這個演算法的方法。

Qian, Yun-Tao, Qing-Song Shi, and Qi Wang. "Cure-ns: A hierarchical clustering algorithm with new shrinking scheme." *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*. Vol. 2. IEEE, 2002.

這篇paper改進了cure演算法，cure有一個步驟為merge兩群cluster，並且對各個代表點作shrink，而此篇paper針對shrinking scheme做了改善。

1. Proposed methods:

CURE Algorithm:

我使用CURE這個分群演算法對各個user作分群，而兩個user間的距離定義為:

1. 當它們沒有共同對某產品評分時 - > 無限大

2. 如果有共同對某些產品評分時 -> 每個產品的評分差之平均

整個CURE algorithm的overview如下(假設想要分成k 群):

1. 一開始先將每個點當作一群，並計算每一群中，離自己最距離最近的群是誰、以及距離。

2. 在所有群中，把距離最短的那兩群merge起來，並對該群內的代表點作shrink的動作，接著update所有群之間的最短距離。

-> 其中在update 所有群之間距離是比較特別的部份，假如要merge cluster w 以及 cluster v，而我現在想更新cluster x的最近群距離以及對像:

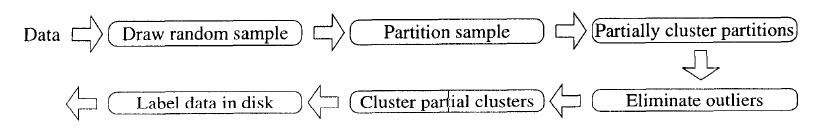
1. 如果x這一群，在merge前的最近群是u / v 則:

(1) 暴力法找x最近群 (枚舉所有代表點的最近點)

(2) 如果dist(x, w) <= dist(x, u/ v) : x.closet = w (因沒人比u / v近了)

2. 已經得知除了w外的最近群是誰，只需比較dist(x, x.closet) , dist(x, w)即可

3. 重複去作2的動作，直到剩下k群。



Overview of CURE Algorithm

Predict Answer:

最後是針對rating metrix上問號的部份做出解答，如果該問號為產品P，而且被user A所評分，則:

1. 去查看和該user A同群的其他user，他們對該產品P評分為何?

最後將這些分數平均即為所求，

2. 如果不幸同群的人都沒有對該產品打分數，則取所有其他人對該產品打的分數平均。

3. 如果更不幸的根本沒人對該產品打分數，則取全部有打分數商品的平均。

1. Implementation:

使用C++去實作，我把要作的事情封裝成三個Class，分別為:

1. Cure : 顧名思義，就是Cure演算法

2. ReadWrite\_data ­: 處理讀檔、寫檔、計算RMSE

3. Misc : 一些無法歸類的雜項，如Show\_result

而一些抽象化的資料型態則統一定義在Definition.h中。

程式主要流程如下:

1. Read input data

2. Run CURE Algorithm

3. Fill question\_mark and Write predict\_answer to file

4. Read question mark's answer and Calculate RMSE

5. Show Result

其中在CURE Algorithm的部份，我並不像paper裡用KD-tree去存每個cluster，而是直接用map去存自己抽象化過的CLUSTER資料，而任兩群的Pair id和距離則放在MinHeap中，是直接使用STL Algorithm中的make\_heap, pop\_heap, push\_heap, sort\_heap去對vector作操作，來實作MinHeap。

而在shrink代表點時，我採取的作法是直接把代表的user評分作平移。

1. Experiments:

RMSE = 1.4

1. Discussion:

這次遇到的問題在於理解Paper上的CURE演算法，以及實做演算法的部份。

而最後跑出來的結果顯示，大部份的user會被分到同一群裡面，而不是每一群的user數量都差不多，這樣其實有點失去分群的意義。我想可能是我在定user之間的similarity還考慮不夠周詳，當發生兩個user間都沒有共同對產品評分時，它們的similarity會非常小，這些user很容易被分到同一群。或許可以嘗試訂立成，兩個user的similarity為對”相同類別”產品的評分差之平均來作定義。

而實作方面，因資料被抽象化很多層，操作比較不直覺，而class有些method和member其實沒有封裝的很好，class之間的耦合性太高，是想要改進的部份。