Similarity and cluster analysis algorithms for Microarrays using R* trees

Jiaxiong Pi [1], Yong Shi [1,2] and Zhengxin Chen [1]

[1] *College of Information Science and Technology*
University of Nebraska at Omaha, Omaha, NE 68182
[2] *Chinese Academy of Sciences Research Center on Data Technology & Knowledge Economy,*
Graduate University of the Chinese Academy of Sciences, Beijing 100080, China

{jpi, yshi, zchen}@mail.unomaha.edu, yshi@gscas.ac.cn

Abstract

Similarity and cluster analysis are important aspects for analyzing microarray data. Based on our perspective of viewing microarrays as time series data, both similarity analysis and cluster analysis are carried out through indexing on time series data using R-Trees. We have developed algorithms for similarity and cluster analysis on microarray data, and conducted experimental studies and comparative studies. First, our study shows that principle components analysis (PCA) has superiority over several other methods (such as DFT and PAA) as far as distance conservation is concerned. A similarity analysis tool based on PCA has been developed, which is able to explore less R*-Tree nodes before finding its similar counterparts and returns less false positives than other methods. In addition, we also extend R*-Tree's application to cluster analysis. With the aid of R*-Tree indexing, two clustering algorithms, KMeans-R and Hierarchy-R, are proposed as an improved version of K-Means and hierarchical clustering, respectively. Experiments for similarity search and cluster analysis based on proposed algorithms have been carried out and have shown favorable results. Experiments related to yeast cell cycle dataset are reported in this paper.*

1. Introduction

The problem of efficiently and accurately locating the similar counterparts of a query in a massive time series dataset, in particular, a microarray time series dataset, is important. Among those similarity analysis methods proposed so far, the most promising method is to using R*-Tree to index the data converted from original data through dimensionality reduction method to avoid "dimensionality curse" in R*-Trees. DFT [1] and PAA [2] are two existing time series dimensionality reduction methods. We have proposed to use PCA as another dimensionality reduction method for time series data, and developed a similarity tool to compare its performance against that of DFT and PAA in terms of time efficiency and false positives returned for a similarity search.

When an R*-Tree is used to index time series data, each of its leaf nodes contains a collection of proximate points. An R*-Tree leaf node, however, is not a cluster in general. Nevertheless the centroids of the data stored in each leaf node can be used as the initial centroids of K-Means, and leaf nodes themselves with small capacity can be used as building blocks when hierarchical clustering method is used. Based on these two aspects, we have proposed improved versions of K-Means (KMeans-R) and hierarchical clustering (Hierarchy-R).

2. Similarity analysis

We have applied our similarity analysis tool with each of three dimensionality reduction modules included on an yeast cell cycle dataset. The yeast cell cycle dataset was extracted from a dataset which shows the fluctuation of expression levels of approximately 6000 genes over two cell cycles. Out of those 6000 genes, 420 genes were categorized into five phase of cell cycle. Furthermore 384 genes were classified into only one phase. The data was normalized to have mean 0 and variance 1 with size of 384×17. Two types of queries, namely exactly matching query (Type I) and similar query are constructed (Type II) with 30 queries each, the results are the average over 10 runs. Computation is done on IBM PC running Linux with CPU 3.4GHZ, RAM 1GB and the capacity of hard disk 80 GB

Indexing time efficiency: Figure 1 is the total time involved in the indexing. PAA is slightly faster than PCA, and 0.5 times faster than DFT at the dimension of 8, but the difference among three indexing times shrinks when dimension is further reduced. General speaking, the three total index times are comparable.

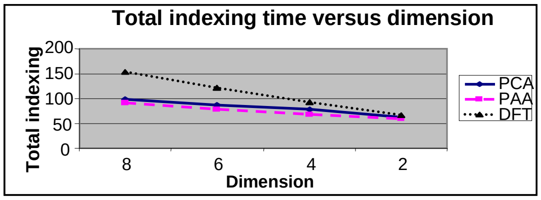

Figure 1. Total indexing time

Query time: Figure 2 shows query in PCA is faster than PAA and DFT, especially at low dimension. The query time of PCA is 47.8% of PAA and 78.8% of DFT at the dimension of 8. It becomes 32.6% of PAA and 41.8% of DFT at the dimension of 2. The faster query time of PCA is due to its better distance conservation property. As a result, less R*-Tree nodes need to be explored to answer a query, and less false positives generated as well (see Table 1). Table 1 shows that PCA works well even when dimension is reduced to 6. For Type II query, query time increases a little, but the pattern of variation of query time is same (not shown).

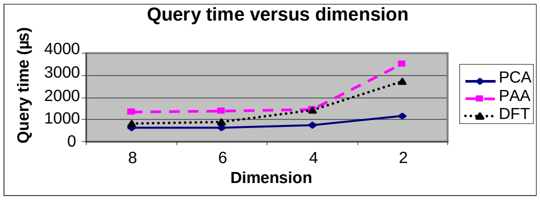

Figure 2. Query time in yeast cell cycle dataset

Table 1. The false positives returned

Dimension	8	6	4	2
PCA	11	12	21	54
PAA	23	23	62	200
DFT	22	30	55	283

3. Clustering for yeast cell cycle dataset

We have used the same yeast cell cycle data for cluster analysis as well. The dataset is made up of five clusters and element numbers in each cluster are known and used as prior knowledge (see Table 2).

Table 2. Clusters and element numbers

Clusters	C_1	C_2	C_3	C_4	C_5
Number	67	135	75	52	55

We applied KMeans-R and Hierarchy-R along with K-Means and KMeans-S on this dataset. Table 3 contains the contingency tables generated by using four mentioned clustering methods, where C_1', C_2', C_3', C_4' and C_5' represent newly formed clusters through each clustering method.

Table 3. Contingency tables

	K-Means					**KMeans-S**				
	C_1	C_2	C_3	C_4	C_5	C_1	C_2	C_3	C_4	C_5
C_1'	39	20	11	3	2	49	30	3	0	1
C_2'	5	70	26	10	7	5	72	29	10	0
C_3'	8	43	29	9	3	0	33	39	19	1
C_4'	2	0	4	19	22	0	0	4	21	28
C_5'	13	2	5	11	21	13	0	0	2	25

	KMeans-R					**Hierarchy-R**				
	C_1	C_2	C_3	C_4	C_5	C_1	C_2	C_3	C_4	C_5
C_1'	49	20	3	0	1	55	23	3	1	6
C_2'	5	111	26	0	0	9	104	23	5	1
C_3'	0	4	31	6	0	0	6	46	0	1
C_4'	0	0	15	35	2	0	1	2	32	0
C_5'	13	0	0	11	52	3	1	1	14	47

The contingency table shows intuitively that KMeans-R and Hierarchy-R are better than K-Means and KMeans-S. To evaluate clustering results quantitatively, we used Rand index (RI), adjusted Rand Index (ARI) and Information Gain (IG) criteria. All three indexes (see Table 4) show KMeans-R is the best in terms of clustering quality, then Hierarchy-R, KMeans-S and KMeans in order.

Table 4. Calculated evaluation indexes

	K-Means	KMeans-S	KMeans-R	Hierarchy-R
RI	0.711	0.739	0.816	0.802
ARI	0.173	0.255	0.486	0.453
IG	0.473	0.809	1.17	1.062

4. Conclusion

For the yeast cell cycle dataset, (1) indexing time of PCA is slightly larger than PAA, but overall slightly better than DFT; (2) PCA is faster than DFT and PAA when answering a query and moreover less false positives were returned; (3) proposed KMeans-R tree and Hierarchy-R show superiority over K-Means and KMeans-S in terms of clustering quality.

References

[1] R. Agrawal, C., Faloutsos and A. Swami, “Efficient similarity search in sequence databases,” *Proc. of the 4th Conference on Foundations of Data Organization and Algorithms*, 1993.

[2] E. Keogh, K. Chakrabarti, M. Pazzani and S. Mehrotra, “Dimensionality reduction for fast similarity search in large time series databases,” *Knowledge and information Systems* 3(3), 2000.