Proceedings of the 5th WSEAS Int. Conf. on SIMULATION, MODELING AND OPTIMIZATION, Corfu, Greece, August 17-19, 2005 (pp225-230)

Biclustering of gene expression data

KRISTA RIZMAN ŽALIK,
Department of Mathematics and Computer Science, PEF
University of Maribor, Koroška cesta 160,
SI-2000 Maribor, Slovenia

Abstract: - Biclustering is an important problem that arises in diverse applications, including the analysis of gene expression and drug interaction data. A large number of clustering approaches have been proposed for gene expression data obtained from microarray experiments. However, the results from the application of standard clustering methods to genes are limited. This limitation is imposed by the existence of a number of experimental conditions or gene samples, where the expression levels of the same genes are uncorrelated. A similar limitation exists when condition-clustering is performed. The goal of biclustering is to find submatrices of genes and conditions, or samples where the genes have nearly the same expression levels for nearly all conditions. Some clustering methods have been adopted or proposed. However, some concerns still remain, such as the robustness of mining methods on the noise and input parameters. In this paper we tackle the problem of effectively clustering gene expression data by proposing an algorithm. We use a density-based approach to identify clusters. Our experimental results show that the algorithm is effective.

Key-Words: -data mining, clustering, biclustering

1 Introduction

DNA microarray technology allows for measuring the expression levels of thousands of genes. An important task is to find genes with similar expression patterns. Co-expressed genes may help to find the regulatory elements for the functional analysis of genes, or discover a disease.

Clustering techniques, which are essential in data mining applications for identifing interesting patterns and to discover groups in a dataset, have proved to be useful in finding co-expressed genes. A clustering problem is a problem of partitioning data into a number of clusters and noise, such that data within the clusters are similar and data in the different clusters or in the noisy partitions are dissimilar. The determination of similarity is hard and it depends on the task and application. Clustering is, therefore the grouping of data that have some quality measure of similarity inside a cluster, and dissimilarity between different clusters. The number of database applications using high-dimensional data sets is increasing and, therefore, the clustering of very large high-dimensional data sets is an important challenge.. Clustering algorithms look for clusters in the whole dimensional space but are incapable of discovering the gene expression patterns in only a subset of experimental conditions. Biclustering, which has been applied intensively in molecular biology research, has recently provided a framework for finding hidden structures in high-dimensional matrices, clusters - both in genes and in samples. It is possible to define these similarities in terms of correlation of gene expression vectors or high density of gene expression features.

Many clustering algorithms originate from non-biological fields and may suffer from some problems when mining gene expression data. Such problems are in the required input parameters as, for example, the number of clusters and the lack of robustness to noise.

In contrast to other data sets such as large amounts of transactional data or multimedia data, gene expression data are often small in size. A microarray experiment usually contains 1.000 to 10.000 genes and the number of samples is less than 100. For many dedicated microarray experiments, only certain useful subset patterns of genes are of interest. Co-expressed genes can be grouped into clusters, based on their expression patterns. The samples can be partitioned into homogenous groups. Each group may correspond to some particular microscopic phenotype, such as clinical syndrome or disease types.

The gene expression data set can be represented by a real-value expression matrix, where each element of the matrix is a real number a_{ij} and represents the expression level of gene g_i in the sample S_j. Row-microarray data are transformed into gene expression matrices in which a row represents a gene and a column represents a sample. The values of a matrix having column samples and genes in rows represent the gene expression levels of each gene in that particular sample.

	sample1	…	sample m
Gene1	a_{11}	…	a_{1m}
…	…	…	…
gene n	a_{2n}	…	a_{nm}

Figure 1: An example of a gene expression matrix

A bicluster is a subset of rows (genes) that exhibit similar behaviour across a subset of columns (samples). The term biclustering was first used by Cheng [2]. We are looking for a set of such biclusters where each bicluster satisfies some specific characteristics of homogenity.

In this paper we investigate the problem of effectively clustering gene expression data. Firstly, we analyze and examine the existing clustering algorithms within the contents of gene expression data. Then, we develop a density-based approach, which effectively solves some problems that the majority of distance-based methods cannot handle. The experiments show that it is effective and matches knowledge provided by bioinformatics experts.

2 Related work

Various clustering algorithms have been used for gene expression data.

Partition-based clustering divides the original data set into *k* partitions. Clusters are formed to optimize the distances between objects. Clusters are represented by the mean (k-mean) or by one representative data called k-medoid.

Partitions should divide data into clusters, so that data in each partition are similar to each other. In k-means the gravity centre of the cluster represents each cluster. Once cluster representatives are selected, data points are asigned to them. Hartigan [3] introduced a partition-based algorithm called Block clustering. K-means[9] and SOM (Self Organizing Map) [10] are two typical partition-based clustering algorithms.

Partition-based clustering methods have a similar clustering quality but the major difficulties with these methods include:

- The number of clusters must be known prior to clustering, which requires some domain knowledge not always available.
- It is difficult to identify clusters which vary considerably in size.
- Each object is forced and placed in one of the discovered clusters.

Hierarchical clustering generates a set of nested clusters that can be represented by a tree. A hierarchical clustering algorithm produces a dendogram representing the nested grouping relationships between objects. If the clustering hierarchy is formed from the bottom up, then at the beginning each data element is a cluster. The gene expression matrix is very noisy and there are many genes and a small number of samples. As we may have hundred of samples and thousands of genes, finding the optimal clusters can be very costly. Similar clusters are tied into bigger clusters and at the end, all data forms only one cluster. This form of the hierarchical method is called agglomerative hierarchical clustering. The opposite approach is called divisive hierarchical clustering, where all data are divided up from one to more clusters. The common disadvantage of hierarchical clustering algorithms is the setting of a termination condition for merging or dividing, which requires some domain knowledge. Hierarchical clustering algorithms have high computational complexity.

Density-based clustering methods have the main advantage of discovering clusters with arbitrary shapes and it is unnecessary to define the number of clusters a parameter. Within each cluster we have a typical density of points, which is considerably higher than outside the cluster. Conventional density-based approaches, such as DBSCAN[5], group data into clusters by means of the rule that the density of points around one point in a cluster has to be above a certain threshold and that each cluster must contain at least minimum number of points. Since the noise of the data sets are typically randomly distributed, the density within a cluster should be significantly higher than that of the noise. So the density–based algorithms have the advantage of extracting clusters from a very noisy environment. Such a noisy environment is that of gene expression data. The performance of DBSCAN is sensitive to the parameters of object density i. e. the minimal number of points and the threshold. The time complexity of DBSCAN is O(n *log n). Other newer algorithms such as Optics [1] and Denclude [7] are more robust to parameters. In Denclude the overall density of the data space is calculated as the sum of the influence functions, which are applied to each data point. Denclude uses a hill-climbing algorithm based on the local density function. Density-based clustering has the advantage of extracting clusters from a very noisy environment. The performance and results are quite sensitive to the input parameters such as minimal number of points in a cluster and number of dimensions.

Grid-based algorithms are unrelated with the nearest neighbour problem in dimensional spaces. STING [17] divides the spatial area into rectangular cells using hierarchical structure. It stores all the statistical parameters (such as mean, minimum, maximum) of the objects within cells. STING goes once through the data to compute statistical parameters for cells, so the time complexity is O(n). The hierarchical representation of grid cells provides a response time for a query to be O(k), where k is the number of grid cells at the lowest level of a hierarchy. Grid-based methods divide the input space into hyper-rectangular cells, discarding the low-density cells, and then combines them into high-density cells in order to form clusters. The main advantage is that the grid-based methods are capable of discovering clusters of any shape and are reasonably fast. These methods also work well regarding input spaces with low to moderate numbers of dimensions. With an increase in dimensions, the number of cells grows exponentially and finding adjacent high–density cells from clusters becomes expensive.

Model-based methods create a model for each of the clusters and find the best fit of the data to that model [18].

Requirements for algorithms when clustering gene expression data

A clustering must meet the following requirements:

- Results should be easily visualized.
- The number of clusters must be automatically defined.
- The method must be robust to noise and parameters.
- Algorithms should be efficient although data can have a lot of attributes (dimensions) and each attribute can have a large domain of values.
- Algorithms using all dimensions, although some combinations represent noise, can be ineffective.

The methods usually do not cover all these requirements. Most methods only cover some requirements well.

Density-based clustering has the advantage of extracting clusters from very noisy environments. They are appropriate for clustering gene expression data.

However, with the increase in dimensions, searching for high-density cells becomes expensive.

3. Problem Statement

Let *G* be a set of genes, where each gene is associated with a set of conditions – samples *S*. We are interested in the subsets of genes that exhibit coherent values on a subset of samples *S*. The tendency between each pair of conditions can be defined in terms of the relative order of the expression values, and samples representing expression levels of specific genes. The relative order between a pair of conditions can be: equivalent, lower or higher. We create an ordered sequence of sample labels by sorting the values for each row of a matrix i.e. the expression levels for each gene in all samples. We view n rows of the data matrix as n sequences of the sample labels.Example:

	sample1	sample2	sample3	sample 4
gene1	100	120	118	180
gene2	100	160	136	200

Ordered sequences:
gene1: <sample1,sample2,sample3,sample4>
gene2:<sample1,sample3,sample2, sample4>

We use two rules for ordering and grouping samples.

First rule: If the difference between the expression level values of a gene is under two conditions or the samples are insignificant, then we consider the two samples equivalent. They form an equivalent group. No order is placed on such samples. The insignificant difference in values can be defined by the maximum difference allowed within a group. We define it as a percentage of the minimum value of the group.

Suppose we have four samples *a,b,c,d* and the expression levels for the first gene for all four samples are {426, 280, 425, 290}, and for the second gene they are {410, 415 ,420, 210}. If the insignificant difference is 0.1 then order-equivalent groups are: *a,c* and *b,d* for the first gene and for the second gene there is one order-equivalent group *a,b,c.* So we obtain these two equivalent sequences:

gene1: <(a,c),(b,d)>
gene2:<d,(a,b,c)>

Samples a and c in gene1 are equivalent and no order is placed on these two samples. Although the expression level of a is greater than that of c, a is before c in the equivalent group.

In equivalent groups, sequences are ordered by sample number and disregard the expression level value for particular gene.

Our model allows a subset of adjacent column labels in a sequence to be grouped as an equivalent group, if their values are similar. Within the group, no strict order of samples regarding gene expression values is defined.

When more than one condition or samples defines the gene expression values of a sample, where more than one sample corresponds to the same stage of a disease described by the sample, then the order of all those similarly expressed conditions is unimportant.

The second criterion for grouping guarantees that, for each sample, the minimum difference between the expression level and the rest of its order-equivalent group is always smaller than the difference between expression levels outside its group.

Suppose we have four samples *a,b,c,d* and the expression levels for a gene for all four samples are {50, 55, 75, 80}. If the insignificant difference is 0.5 then the closest neighbour of c is d and not b, so instead of grouping (a,b,c) we group (a,b) and (c,d), separately.

We have to discover the patterns of samples. We get patterns of samples for each gene by simply sorting the values in each row and then selecting subsets of continuous values that are valid regarding the threshold δ, denoting an insignificant difference in expression values.

For an expression matrix, we are seeking groups of samples that correspond to the same empirical phenotype structure that can be represented by a common subsequence S of length s shared by k genes. The k genes are co-expressed in the s samples.

We can now define gene expression pattern and maximal pattern-biclusters.

For gene expression matrix *M*, a subset of the set of genes G and a subset of the set of samples S uniquely define a *i x j* submatrix $M_{S,G}$ of matrix M. $M_{S,G}$ as a δ-valid ij-pattern, if each row is tightly clustered in an interval of size up to δ. δ thus denoting the insignificant difference in expression values. It can be defined by the maximum difference of values. We define it as a percentage of the minimum value of the group and describe it with δ.

The number *j* denotes the number of samples belonging to the *ij*-pattern. The number *i* denotes the number of genes for which the samples belonging to the ij-pattern have approximately the same expression level.

We are looking for maximal patterns and largest submatrices called biclusters. The δ−valid ij-pattern is the maximal pattern and it is a solution bicluster if the following conditions are true:

- Maximal pattern cannot be extended into δ−valid ij'-pattern, where j'>j, by adding samples to a subset of samples.
- Maximal pattern cannot be extended into δ−valid i'j-pattern, where i'>I, by adding genes to a subset of genes.

The discovered maximal patterns are biclusters of informative genes defining samples of the same phenotype or disease structure.

Example:

	sample1	sample2	sample3	sample4
gene1	100	120	118	180
gene2	100	160	136	200
gene3	160	130	116	200

For gene subsets gene1 and gene2 and the sample subsets sample1 and sample2, $M_{S,G}$ is

$$M_{S,G} = \begin{bmatrix} 100 & 120 \\ 100 & 160 \end{bmatrix}$$

$M_{S,G}$ is not 0.05−valid ij-pattern, because the values in the second row are spread over an interval greater than 0.05. $M_{S,G}$ is a valid 0.08-pattern. It is not maximal, because adding gene2 and sample3 produces a pattern, which is still 0.05-valid. It is maximal because adding any other sample makes submatrices that are no longer 0.08-valid.

For a database D with gene set G and set of conditions in samples S and a given threshold δ denoting insignificant differences in values, it is difficukt problem to find all maximal biclusters C containing subsets of genes and subsets of conditions, according to the definition of maximal pattern- bicluster.

The problem of finding maximal patterns- biclusters can be transformed into the following problem:

1. Presenting each row of the gene expression matrix by ordered sequence of samples.
2. Selecting each subset of continuous values that are δ−valid and form equivalent groups.
3. Mining the subsets of rows and identifying the longest common sample subsequences - equivalent groups for any subset of at least n rows (genes), and forming maximal patterns-biclusters.

4. Forming Sequences

Each row in a database, representing the expression level of a gene in different samples, is converted into an ordered sequence of columns-samples. The ordered sequences will be generated by the following approach. First, the minimum value and the maximum value of expression level for a row of matrix is searched and δ is defined, so that data are classified into ten or less equivalent groups. Then samples are classified into equivalent groups: $m..m*(1+\delta)$, $m*m*(1+\delta),..m*(1+\delta)*(1+\delta),\ldots$, where m is the minimal expression level.

In the next scan samples can be regrouped in the sense of the second rule of grouping: the difference in expression levels inside a group is smaller than with the expression levels outside the group.

In the example in Figure 3, after the first scan four pairs are grouped together for δ=100%.

Gene expression values for samples 0..34:

```
3105   1118   4543   5467   3469   3309   3936
4745   4081   1658   2853   551    4746   1534
5311   4326   7155   1178   3427   6870   9177
4836   3085   5815   1872   3265   2877   794
1312   485    408    1047   335    680
```

1st scan –samples are denoted with numbers 0..34:

(11 29 30 32)
(1 17 27 28 31 33)
(9 13 24)
(0 2 4 5 6 7 8 10 12 14 15 18 21 22 25 26)
(3 16 19 20 23)

Figure 3: An example of identifying equivalent groups

Each row in a matrix has been converted into a sequence of samples.

The goal is to find frequent approximate subsequences. The general idea is not to find exact patterns, but identify patterns shared by many sequences.

Edit distance is used as a distance measure of sample sequences. It is defined as the minimum cost of editing (i.e. insertions and deletions) required to change one sequence to another. Operational insert and delete are represented by INS_DEL. To make edit distance comparable among sequences with different lengths, we normalize edit distance with the length of the longest sequence. Edit distance between sequences S1 and S2:

$$(3.1) dist(S1, S2) = \frac{\min number\ of\ INS\_DEL}{\max\{length(S1), length(S2)\ \}}$$

Using edit distance (Equation 3.1), we can apply a density-based clustering algorithm to cluster sequences. A sequence is dense if there are many sample sequences for different genes similar to it.

5. Alignment of Sequences and Pattern Generation

Sequences of samples in equivalent groups for different genes are similar to each other. Now comes the problem of how to summarize the general pattern of samples in more biclusters.

For storing, the alignment results of sample sequences and equivalent groups effectively, we can use weighted sequences.

A weighted sequence of two equivalent groups of samples WS = (S_1:n_1, S_2:n_2, ., S_4:n_4):m_1(gene set_1) (S_5:n_1, S_6:n_2, ., S_7:n_n):m_2 (gene set_2) carries the following information:

- The equivalent group occurs in m_i number of gene sequences, precisely defined by gene set_i. Number m_i is called a global weight of the weighted sequence.
- Sequence S_i appears in equivalent group in n_i gene sequences equivalent group.

Each equivalent group of aligning row is aligned with an equivalent group of weighted sequence with the smallest edit distance.

Suppose we have two sequences of two genes each having five equivalent groups:

0 gen:
11 29 30 32
1 17 27 28 31 33
9 13 24
0 2 4 5 6 7 8 10 12 14 15 18 21 22 25 26
3 16 19 20 23

1 gen:
2 21 33
11 17 27 28 31 32
0 7 13 16 22 26 29 30
1 3 4 5 6 9 15 20 23 24 25
2 8 10 12 14 18 19

The resulting weighting sequence of equivalent groups:
WS1:
(11:1 29:1 30:1 32:1):1(gene 0,1)
(11:1 17:2 27:2 28:2 31:2 32:1 33:1):2 (gene 0,1)
(9:1 13:1 24:1): 1(gene 0)
(0:2 1:1 2:2 3:1 4:2 5:2 6:2 7:2 8:2 9:1 10:2 12:2 13:1 14:2 15:2 16:1 18:2 19:1 20:1 21:1 22:1 23:1 24:1 25:2 26:2 29:1 30:1):2 (gene 0,1)
(3:1 16:1 19:1 20:1 21:123:1):1 (gene 0)

After the aligning of sample sequences for the first gene, we align other sequences for other genes with the current weighting sequence. The alignment result for all sequences is summarized in the last weighting sequence.

The weighted sequence explicitly keeps information about various item-sets (genes and samples) in the sequences. This information is summarized into the item weights in the weighted sequence. Aligning the sequences in different order may result in slightly different weighted sequence.

6. Generating biclusters:

Biclusters can be generated by picking equivalent groups of samples shared by most sequences for genes. The strength of each sample in an equivalent group is defined as n_i/m 100%, where n_i is the strength for a particular sample and m is the global weight of an equivalent group. The strength of the equivalent group is n_i/n*100, where n_i is strength of the equivalent group and n is the total number of genes. More equivalent groups in sequences with more genes share a sample with higher strength value. Biclusters can be formed for different strength thresholds T. A user can also specify strength threshold T, 0 <= T <=1. A bicluster is a maximal pattern of sequences for equivalent groups, operating on the same gene sets with a global weight greater than threshold T * n, where n is the number of genes and by the removal of items in the sequence with strength values lower than the threshold T.

7. Case study

We have tested an algorithm on a real data set containing samples corresponding to acute lymphoblast leukaemia (ALL) and acute myeloid leukaemia (AML samples). The leukemia data set is based on a collection of leukemia patient samples reported in [4][6], with selected features. The experiments show that it is effective and matches the knowledge about known data given by the bioinformatics experts.

Why is an algorithm effective and efficient?

By measuring the density of samples for a particular gene and gene expression levels, an algorithm captures the natural distribution of data. Compared to other algorithms like K-means and SOM, the input parameter-number of clusters is unnecessary, and does not affect the resulting biclusters. By locating the dense areas in the object space, the algorithm automatically detects the number of clusters.

Identification of interesting patterns from noisy data

As we have already mentioned, a good clustering algorithm for gene expression data must be robust to noise. It must automatically detect interesting patterns from noisy data sets. Distance-based algorithms may collapse with noise. The suggested algorithm captures the core area of a cluster that has significantly higher density than noise. The experimental results suggest that the density structure remains robust and the algorithm still identifies the corresponding patterns, even in a noisy environment. We have measured the patterns identified by the Leukemia-G1 data set and have added one fold, three and eight fold noises (permutation of real expression data) to the dataset. All known biclusters have been discovered.

Time complexity

In density-based methods, with the increase of dimensions, search for high-density cells becomes expensive. Our algorithm is density-based and consists of three main steps: forming sequences, aligning sequences, and generating biclusters. Forming sequences and equivalent groups requires no sorting, because samples in equivalent groups are not sorted in regard to the value of expression level thus causeing time complexity O(N), where N is number of all elements. N is m * n, m is number of samples and n is number of genes.Aligning sequences requires time complexity O(c*m*n), where c is the number of equivalent groups and is less than 10. Usually c<<n and then time complexity is O(m*n) = O(N). Number of samples m << n, then the expected time complexity is O(n). Generating biclusters requires time complexityO(m). The total time complexity is O(m*n) and for m<<n it is O(n).

8. **Conclusions**

Clustering gene expression data is an important task in bio-medical applications. Although methods have been adopted from other applications and new methods have been suggested, there are still some current problems, such as robustness of mining results with the noise and parameters.

We propose an algorithm that organizes gene expression data matrix into sample sequences and searches maximal sequence patterns representing the density-based connectivity of samples, and particular expression levels of particular genes. Then dense areas of samples are identified and biclusters are obtained. This algorithm is robust in terms of handling noise, structures of clusters and parameters. This is required for bioinformatic data analysis.

References:

[1] Cho, R et al., *A genome-wide transcriptional analysis of the mitotic cell cycle*. Molecular Cell 2:65-73, 1998.

[2] Cheng Y., M.Church, *Biclustering of Expression Data*, Proc Eight Int'l conf. Intelligent System for Mulecular Biology, 75-85, 2000.

[3] Hartigan, J.A. *Direct Clustering of a Data Matrix*, J. Am. Statistical Assoc. vol. 67, no 337, pp. 123/129, 1972.

[4] Eisen, M. B. *Cluster analysis and display of genome-wide expression patterns*, Proc. Natl. Acad. Sci. USA, Vol95, pp.14868, December 1988.

[5] Ester M., H.-P.Kriegel, J.Sander, X.Xu, *A density based algorithm for discovering clusters in large spatial database.* In Proc.1996 Int.Conf. Knowledge Discovery and data mining. 226-231, 1996.

[6] Golub T. R., SLONIM D.K. et al Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science, Vol. 286(15):531-537, October 1999.

[7] Hinneburg , A., Keim D. A., *An efficient approach to clustering in multimedia databses with noise*. In Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining, 58/65, 1998.

[8] Hoffmann and J. Buhman, *Active data clustering.* In Advances in Neural Information Processing Systems, 528-534,1997.

[9] Tavazoie,S et al. *Systematic determination of genetic network architecture*, Nature Genet, 281-285,1999.

[10] Tomayo P. et. al.: *Interpreting patterns of gene expression with self-organizing maps: Methods and applications to hematopoietic differentiation.* Proc. Natl. Acad.Sci, USA, Vol.96(6): 2907-2912 March 1999.

[11] Wang, W., J. Yang, M. Muntz, STING : *A statistical information grid approach to spatial data mining.* In Proc on Very Large Data Bases, 186-195, 1997.

[12]Yeung, K.Y., Fraley, C., Murua, Rafterz, Ruzzo, *Model-based clustering and data transformations for gene expression data*, Bioinformatics, 17:977-987, 2001