Clustering Microarray Data based on Density and Shared Nearest Neighbor Measure

Ranapratap Syamala, Taufik Abidin, and William Perrizo
Computer Science Department
North Dakota State University
Fargo, ND 58105, USA
+701-231-6403
{pratap.syamala, taufik.abidin, william.perrizo} @ndsu.edu

Abstract
Microarray technology is being used to study several genes at a time generating huge amounts of data. Managing and extracting useful information from such data is a big challenge for the bioinformatics community. Clustering is an important data mining technique that has been proved to be useful for the analysis and understanding of such data. Clustering is used to automatically identify similar groups of objects based on a given measure. In microarray data, the genes that exhibit similar expression profiles are co-expressed and will be grouped into a single cluster. In this paper, we propose a new clustering algorithm based on density and shared nearest neighbor measure to identify clusters of genes exhibiting similar expression profiles. In our algorithm, we used an efficient bitwise vertical data structure called P-tree\(^1\) to decompose the microarray data into separate bit vectors. *Pearson's correlation coefficient* is used as the similarity measure to identify the core points of the clusters by calculating the density of the genes. Also in our algorithm, there is no need to specify the number of clusters ahead. The clusters in the data set are identified automatically based on the core genes. We experimentally show that our algorithm is fast and scalable when applied on Iyer’s microarray data set for cluster analysis.

Keywords: Clustering, P-tree, Microarray data, Gene expression profiles, Co-expressed genes

1 INTRODUCTION

Microarray technology is one of the biggest breakthroughs in the field of genomics that had enabled to perform high throughput experiments for genome-wide monitoring of genes. These experiments generate large amounts of data and analysis of such data is a major challenge in the field of bioinformatics. The gene expression analysis across whole genome is highly effective in identifying and studying co-expressed genes in a given organism.

Co-expressed genes represent genes that exhibit similar expression profiles in a microarray experiment. The common trend exhibited by the co-expressed genes is called *coherent gene expression pattern* (coherent pattern) [10]. Analysis of co-expressed genes and coherent patterns is useful in identifying functional categories of a group of genes characterizing different regulatory mechanisms in the cells and give an indication of gene expression levels in different cells at different stages of cell cycle. They also provide an insight into how genes and gene products interact to form interaction networks. Figure 1 shows an example of profiles of co-expressed genes in a cluster.

![Figure 1 Example: Gene expression profiles or patterns of co-expressed genes](image)

The main objectives of microarray data analysis can be divided into three categories: class discovery, class prediction, and class comparison. The important goal of class discovery is to identify the clusters of genes that have similar gene expression profiles over a time series of experiments. Clustering is the main technique employed in class discovery. Class prediction is assigning an unspecified gene to a class given the expression of other genes with known class labels. Classification is the main technique used in class prediction. Class comparison aims at identifying the genes that differ in expression profiles between different classes of genes.

\(^1\) P-tree technology is patented by NDSU. United States Patent No. 6,941,303.
In this paper, we concentrated on clustering or unsupervised learning of microarray data analysis to address the class discovery problem. Typically in clustering, the objective is to find clusters of objects such that the objects within a cluster are more similar to each other than to other objects in a different cluster. In microarray data analysis, genes that exhibit similar gene expression profile or similar patterns of expression will be clustered together. To calculate the similarity between genes, several statistical measures have been developed. In our algorithm, we used Pearson’s correlation coefficient to calculate the similarity of two genes across different time series as this statistic captures similarity in shape of the expression profile.

The rest of the paper is organized as follows. Section 2 gives a brief overview on some of the existing clustering algorithms that are being used for microarray data analysis. In section 3, we provide definitions and describe the new clustering algorithm. In section 4, we show the empirical results obtained by our algorithm on Iyer’s microarray data set and provide a discussion about the results of our algorithm. Section 5 gives a short conclusion and directions for future work.

2 RELATED WORKS

Several efficient and effective clustering techniques have been developed in statistics, machine learning, and data mining. These include partition-based clustering methods like K-means and K-medoids algorithms, self-organizing maps (SOM) [14], hierarchical clustering like AGNES and DIANA [7], and BIRCH [15], and density-based approaches like DBSCAN [5], OPTICS [1]. However, partition-based clustering methods always require the number of clusters need to be specified and almost always only identify globular clusters. They are not suitable for large data sets having clusters with different shapes, sizes, and high dimensions while in hierarchical clustering methods a decision has to be made regarding selection of merge points or split points which is critical because once a decision is made, it is difficult to undo the step. Also, these hierarchical methods do not scale well for large data sets with high dimensions and the computational complexity is very high. Agglomerative hierarchical clustering technique was implemented by Eisen et. al for cluster analysis of microarray data [4].

Density based approaches are used to identify clusters of different shapes, sizes, and densities in a data space. For each point the density of a data point has to be greater than a give threshold to be included in a cluster. But DBSCAN cannot find clusters with different densities because the core point definition used makes it difficult to identify core points for clusters with varying density [11]. Typically, in microarray data, genes express differently to different treatments and hence to find the genes that have similar expression profiles using partition-based or hierarchical clustering techniques is difficult.

The proposed clustering algorithm is based on density and utilizes a shared nearest neighbor measure to identify the clusters of co-expressed genes. The concept of shared nearest neighbors was proposed by Jarvis and Patrick in [9] and is further studied in ROCK [6]. They define that two points in a cluster are similar when they share the same nearest neighbors. In our approach, we use this property to merge nearest neighbors of two genes with highest density and also while assigning border genes to appropriate clusters. The algorithm is explained in detail in section 3.

3 THE PROPOSED ALGORITHM

3.1 P-tree Overview

In our approach, we employed an efficient and scalable vertical data structure called P-tree [13] that has been proven to be effective in clustering [1] [11]. P-trees are vertical, lossless and data-mining ready data structures. P-trees can be created from relational databases by decomposing each attribute into separate bit vectors, one for each bit position of numeric values in that attribute. The main operations that can be carried out on P-trees are basic logical operations like AND ($\land$), OR ($\lor$), and NOT ($'$). A huge advantage can be gained while performing select operation and other aggregate operations such as root count, max, and min using P-trees.

The root count is the count of the number of 1-bits in a basic P-tree or P-trees resulting from any logical operations. These operations and others are discussed in more detail in [3].

3.2 Density and Shared Nearest Neighbor Measure

Pearson’s correlation coefficient is one of the standard statistics that has been used to calculate the similarity between genes in microarray data analysis. A similarity matrix is built for the whole data set based on the correlation coefficients between the genes across a time series. Density of a gene $g_i$ is defined as the sum of the similarity of its neighbors and can be represented as equation (1).

$$\text{density} (g_i) = \sum_{j=1}^{n} \text{sim} (g_i, g_j)$$ (1)
where \( n \) = number of neighbors of \( g \), \( g \geq \) similarity threshold

We use the shared nearest neighbor measure while processing the neighbors of two most dense genes \((g_i, g_j)\) to a cluster and then assigning the border genes to the appropriate clusters. Specifically, we assign the neighbors of two most dense genes \((g_i, g_j)\) to the same cluster if both the genes share neighbors greater than a given shared nearest neighbor threshold \( (\text{snnThreshold}) \). If \( g_i \) and \( g_j \) are the genes with highest density identified from equation (1), then the number of shared nearest neighbors can be obtained from the following equation:

\[
\text{shared nearest neighbors,} \\
\text{snn} (g_i, g_j) \equiv \text{size} (\text{NN}(g_i) \cap \text{NN}(g_j)) \quad (2)
\]

In the above formula, \( \text{NN} (g_i) \) and \( \text{NN} (g_j) \) represent the nearest neighbor lists of \( g_i \) and \( g_j \) greater than or equal to a given similarity threshold respectively. The use of shared nearest neighbor measure is justified by the fact that the presence of shared neighbors between two dense genes means that the density around the dense genes is similar and hence should be included in the same cluster along with their neighbors.

The neighbors of genes can be identified by building a nearest neighbor mask for the dense genes using our vertical approach. Nearest neighbor mask of a gene \( g \) is a bit pattern of 1’s and 0’s with 1 if a gene is neighbor to \( g \), and 0 otherwise. Consider that the neighbors of genes \((g_i, g_j)\) are identified, then the shared nearest neighbors between them is nothing but the root count of ANDing the nearest neighbor masks of the two genes \( g_i \) and \( g_j \) and can be computed vertically using the following equation (3).

\[
\text{shared nearest neighbors in P-tree format,} \\
\text{psnn}(g_i, g_j) = \text{rootCount}(\text{NNm}(g_i) \land \text{NNm}(g_j)) \quad (3)
\]

where \( \text{NNm} (g_i) \) and \( \text{NNm} (g_j) \) are the nearest neighbor masks of genes \( g_i \) and \( g_j \) respectively. In the case of traditional algorithms, to identify the nearest neighbors of the dense genes and check to see whether the neighbors of \( g_i \) are present in the neighborhood of \( g_j \), scanning of neighborhoods of both the genes is required. This step can be computationally expensive when the neighborhoods of genes are large. Further explanation on how we obtain the nearest neighbor mask is explained in section 3.3. Calculating the density from the similarity measure using P-trees is more efficient in order to identify the nearest neighbors of the two densest genes.

### 3.3 The Clustering Algorithm

In our approach, we represent the experiments on the x-axis and the genes on y-axis for a given microarray data set. If the data is not present in this format, we transpose the data to the above format. Similarity matrix is built using Pearson’s correlation coefficient, and P-trees are generated from the matrix. Larger correlation coefficient represents genes that are more similar. Always, two genes with highest density are considered in our algorithm unless the unprocessed genes vector contains only one gene in which case it will be processed as a core gene, a border gene, or a noise gene.

The pseudo-code for the clustering algorithm is given in Figure. 2. The clustering procedure is initiated by identifying the two most dense genes \( g_i \) and \( g_j \) from the unprocessed genes. First we check to see if these genes have any neighbors. The neighbors of the dense genes are obtained using the P-tree neighbor mask which is a basic bit pattern that has a 1-bit if a neighbor is present and 0-bit if not. This process eliminates the scanning process which most of the clustering algorithms use and is also a computationally expensive if the data set is quite large. If there are no neighbors, then we label both genes as noise genes, else we check if the neighbors of \( g_i \) and \( g_j \) share neighbors greater than the given \( \text{snnThreshold} \). If they share neighbors, then we merge the neighbors of both genes and assign them to a cluster. Using P-trees, the merge operation is nothing but \( \text{ORing} \) (union) of neighbor mask P-trees of the two genes. If genes \( g_i \) and \( g_j \) do not share any neighbors, then each dense gene is processed by checking whether it is a core gene or noise gene. If it is a core gene, then its neighbors are processed and clustered together, else it is labeled as a border gene. If a gene \( g \) has neighbors and if one of its neighbors has higher density than \( g \), then we define gene \( g \) as border gene. The processed genes and the unprocessed genes are updated and the process is repeated until all genes are processed.

While assigning the border genes to clusters, we look whether the border gene share any neighbors or not and is not dependent on the \( \text{snnThreshold} \). The fact that the border gene shares neighbors with any of the clusters suggests that the border has similar local density with that of the
cluster to which it is being assigned. The following two cases will be considered:

**Case I:** If the border gene share neighbors with any cluster: Find the border gene with highest density and get all its neighbors greater than or equal to the given similarity threshold. Then find the cluster with which the neighbors of the border gene share highest number of neighbors and assign the border gene to that cluster.

**Case II:** If the border gene does not share any neighbors with any of the clusters: Find the most similar gene to the current border gene and assign the border gene to the cluster to which the most similar gene belongs. In case of ties, assign the border genes based on its similarity to the core gene of the cluster combined with the above criteria.

### 3.4 Parameterization:

**Similarity threshold:** In our clustering algorithm, pairwise similarity is used to determine the similarity between two genes across a time series. The higher the similarity, the tighter are the clusters obtained and the genes in such clusters will be highly similar either in their function or their localization in the cell. Hence, setting the similarity threshold is easy based on the levels of similarity a user desires. Typically, in microarray experiments, it is desirable to have genes that give as much information as possible and hence clusters with highly co-expressed genes are desired.

**Shared nearest neighbor threshold:** This parameter is used to determine whether the most dense genes being processed should be clustered together or not. If \( \text{snnThreshold} \) is too large, then our algorithm will find few, well-separated clusters that have more genes in clusters when the dense genes share neighbors. On the other hand, if the \( \text{snnThreshold} \) is too small, then there is a chance that cluster with uniform density could be broken into several small tight clusters. Hence this parameter determines cluster size based on the user input and domain knowledge is highly useful while assigning this parameter.

### 4 RESULTS AND DISCUSSION

The performance of the clustering algorithm was tested on an Intel Pentium 4, 2.6 GHz processor with 3.8GB RAM, running Red Hat Linux 2.4.20-8smp. The algorithm is written in C++ programming language.

To show the practical application of our clustering algorithm, we applied our algorithm on Iyer's microarray data set [8]. The results are shown in Figure 3. The data set contains the response of human fibroblasts to serum on cDNA microarrays.
This data set contains the expression profiles of 517 human genes corresponding to changes in mRNA levels at 12 times, ranging from 15 min to 24 hours after serum stimulation. The expression changes are given as the ratio of the expression level at the given time-point to the expression level in serum-starved fibroblasts. The algorithm groups the genes with similar expression profiles into clusters. Figure 3 shows some of the clusters identified by our algorithm with similarity threshold 0.90. The algorithm was able to obtain the same clustering results with snnThreshold of 15 and 20.

Figure 3. Clusters obtained by the clustering algorithm. X-axis is time point, Y-axis is the expression levels of the genes with similarity threshold = 0.90, snnThreshold = 15

The gene expression profiles that show odd peaks in the cluster are the border genes and noise genes, if noise genes are considered in the clustering process. The expression profiles of these genes give additional information. They have similar pattern with respect to the other genes in the cluster except at certain time points in the treatment or cell cycle. Observing these genes closely might give an important insight about the behavior of that particular gene at certain time point in the cell cycle. If the noise genes are excluded, clusters having highly similar expression profiles i.e., the highly co-expressed genes will be obtained. The computation time for processing the clusters once the similarity matrix is built takes only 5.70 seconds for 517 genes. This shows that our algorithm is extremely fast.

In the case of partition-based clustering algorithms, a priori knowledge about the number of clusters is required where as in our algorithm we do not require to specify the number of clusters as the number of clusters will be equal to the number of core genes identified during the clustering process. When compared to the hierarchical clustering algorithms, we do not need any cut off point to identify the clusters. With respect to the density-based algorithms like DBSCAN, where the number of neighborhood points within a given radius is used to identify the core points, our algorithm uses all the neighbors of a given point greater than or equal to a given similarity threshold to identify the core points based on density. Also shared nearest neighbor measure reflects the local density of the points in the data space and hence is relatively stable to high dimensional data while assigning border points to the clusters.

5 CONCLUSION

In this paper, we presented a new clustering algorithm based on density and shared nearest neighbor measure and applied to microarray data to group genes with similar gene expression profiles into a cluster. The density-based approach to identify the core points is useful in finding clusters with different shapes, and the use of shared nearest neighbor measure eliminates the problems with varying densities in the data space. Our experimental results on Iyer’s microarray data set show that our algorithm is fast and scalable. The number of clusters is not needed a priori as the number of clusters is determined automatically based on the density. This kind of cluster analysis on microarray data is extremely useful in the field of genomics as the function of genes can be attributed on a genomic scale. In the future, we would like to include visualization for the results obtained by our algorithm, expand our work to more real world data sets, and compare the results with state-of-the-art algorithms. Also it is reasonable to explore sub-clustering on the main clusters by changing the snnThreshold parameter so that compact small clusters can be obtained.

6 REFERENCES


