Ant-MST: An Ant-Based Minimum Spanning Tree for Gene Expression Data Clustering

Deyu Zhou, Yulan He, Chee Keong Kwoh, and Hao Wang

School of Computer Engineering, Nanyang Technological University
Nanyang Avenue, Singapore 639798
{zhou0063,asylhe,asckkwoh,wang0046}@ntu.edu.sg

Abstract. We have proposed an ant-based clustering algorithm for document clustering based on the travelling salesperson scenario. In this paper, we presented an approach called Ant-MST for gene expression data clustering based on both ant-based clustering and minimum spanning trees (MST). The ant-based clustering algorithm is firstly used to construct a fully connected network of nodes. Each node represents one gene, and every edge is associated with a certain level of pheromone intensity describing the co-expression level between two genes. Then MST is used to break the linkages in order to generate clusters. Comparing to other MST-based clustering approaches, our proposed method uses pheromone intensity to measure the similarity between two genes instead of using Euclidean distance or correlation distance. Pheromone intensities associated with every edge in a fully-connected network records the collective memory of the ants. Self-organizing behavior could be easily discovered through pheromone intensities. Experimental results on three gene expression datasets show that our approach in general outperforms the classical clustering methods such as K-means and agglomerate hierarchical clustering.

Keywords: gene expression data, clustering, ant-based clustering, minimum spanning tree.

1 Introduction

Microarrays enable biologists to study genome-wide patterns of gene expressions in any given cell type, at any given time, and under any given set of condition. Using these arrays can generate large amounts of data, potentially capable of providing fundamental insights into biological processes ranging from gene function to cancer, ageing and pharmacology [1]. Even partial understanding of the available information can provide helpful clues. For example, co-expressions of novel genes may provide leads to the function of many genes for which information is not available currently.

Clustering is a fundamental technique in exploratory data analysis and pattern discovery, aiming at extracting underlying cluster structures. Cluster analysis is concerned with multivariate techniques that can be used to create groups

J.C. Rajapakse, B. Schmidt, and G. Volkert (Eds.): PRIB 2007, LNBI 4774, pp. 198–205, 2007.

amongst the observations, where there is no *a priori* information regarding the underlying group structure. Clustering of the genes on the basis of the tissues can be used to search for groups of gene that might be regulated together. Dozens of clustering algorithm exist in the literature and a number of *ad hoc* clustering procedures have been applied to microarray data. Available methods can be categorized broadly as being hierarchical such as agglomerative hierarchical clustering (AHC) [2, 3] or non-hierarchical such as k-means clustering [4] and clustering through Self-Organizing Maps [5]. A major limitation of hierarchical methods is their inability to determine the number of the clusters. The limitation of k-means methods is their high computational complexity.

The concepts and properties of graph theory make it very convenient to describe clustering problems by means of graphs [6]. Nodes of a weighted graph correspond to data points in the pattern space and edges reflect the proximities between each pair of data points. Approaches based on minimum spanning trees have been proposed for clustering gene expression data [7]. Minimum spanning tree (MST), a concept from the graph theory, is used for representing multidimensional gene expression data. Based on the representation, gene expression data clustering problem is converted to a tree partitioning problem. Advantages of using this method have been described and demonstrated as follows [7]: 1) the simple structure of a tree facilitates efficient implementations of rigorous clustering algorithm; 2) clustering based on MST does not depend on detailed geometric shape of a cluster; 3) inter-data relationship is greatly simplified in MST representation and no essential information for clustering is lost.

We have proposed an ant-based clustering algorithm for document clustering based on the traveling salesperson (TSP) scenario [8]. It not only has the traits of self-organization and robustness, but also can generate optimal number of clusters without incorporating any other algorithms such as K-means or AHC. In [8], to break the linkages of the fully connected network in order to generate clusters, average pheromone strategy is used. The average pheromone of all the edges is computed at first and then edges with pheromone intensity less than the average pheromone will be removed form the network. Nodes will then be separated by their connecting edges to from clusters. In this paper, we investigate using the method based on minimum spanning trees (MST) to break the linkages in order to generate clusters. The reasons behind are: 1) the method based on MST has been proven efficient in the domain of gene expression clustering, 2) and it has strong mathematical foundation.

Our proposed approach called Ant-MST consists of two steps. First, a fully connected network of nodes is generated using the ant-based clustering method. Then the linkages is broken based on MST in order to generate clusters. It uses pheromone intensity to measure the similarity between two genes instead of using Euclidean distance or correlation distance. Pheromone intensities associated with every edge in a fully-connected network records the collective memory of the ants. Self-organizing behavior could be easily discovered through pheromone intensities.

The rest of the paper is organized as follows. Section 2 presents the Ant-MST approach. Experimental results on three gene expression datasets are discussed in section 3. Finally, section 4 concludes the paper and outlines the future work.

2 Ant-MST: An Ant-Based Minimum Spanning Tree

2.1 Ant-Based Clustering

The Ant Colony Optimization (ACO) algorithm belongs to the natural class of problem solving techniques which is initially inspired by the efficiency of real ants as they find their fastest path back to their nest when sourcing for food. An ant is able to find this path back due to the presence of pheromone deposited along the trail by either itself or other ants. An open loop feedback exists in this process as the chances of an ant taking a path increases with the amount of pheromone built up by other ants.

Early approaches in applying ACO to clustering are to first partition the search area into grids. A population of ant-like agents then move around this 2D grid and carry or drop objects based on certain probabilities so as to categorize the objects. However, this may result in too many clusters as there might be objects left alone in the 2D grid and objects still carried by the ants when the algorithm stops. Therefore, Some other algorithms such as k-means are normally combined with ACO to minimize categorization errors. More recently, variants of ant-based clustering have been proposed, such as using inhomogeneous population of ants which allow to skip several grid cells in one step, representing ants as data objects and allowing them to enter either the active state or the sleeping state on a 2D grid. Existing approaches are all based on the same scenario that ants move around in a 2D grid and carry or drop objects to perform categorization.

We have proposed an ant-based clustering algorithm for document clustering based on the travelling salesperson (TSP) scenario [8]. The advantages of our ant-based clustering approach are: 1) It does no rely on a 2D grid structure. 2) It can generate optimal number of clusters without incorporating any other algorithms such as k-means or AHC. 3) When compared with both the classical document clustering algorithms such as K-means and AHC and the Artificial Immune Network (aiNet) based method, it shows improved performance when tested on the subsets of 20 Newsgroup data[1]. Here, we investigate the ant-based clustering algorithm for gene expression data analysis.

2.2 Minimum Spanning Trees

The concept of minimum spanning trees (MSTs) is from graph theory. For a connected and undirected graph G, a spanning tree of the graph G, T is a subgraph which is a tree and connects all the vertices together. A single graph can have many different spanning trees. If we assign a weight to each edge,

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/

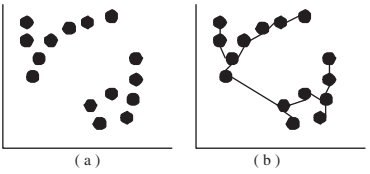

Fig. 1. 2D representation of a set of gene expression data (a) and its corresponding MST (b)

Table 1. Three objective functions and their corresponding clustering algorithms

Method	**Objective Function**	**Procedure**
Removing longest edges (MST-R)	Partition an MST into K subtrees so that the total edge-distance of all the K subtrees is minimized	Find the $K-1$ longest MST-edges, cut them and get a K-clustering achieving the global optimality of the objective function.
Iterative clustering (MST-I)	Partition an MST T into K subtrees $\{T_i\}_{i=1}^{K}$ to optimize: $\sum_{i=1}^{K}\sum_{d\in T_i} Dist(d, \text{center}(T_i))$ where d is the data point in the T_i and center(T_i) is dependent on the distance measure.	Start with an arbitrary K-partitioning of the tree and iteratively do the following until converging. For each pair of adjacent clusters, go through all tree edges within the merged cluster to find an edge which globally optimizes the 2-partitioning of the merged cluster and then cut the edge.
Global optimization (MST-G)	Partition the tree T into K subtrees and select K representatives $d_1, \ldots, d_K \in D$ to optimize $\sum_{i=1}^{K}\sum_{d\in T_i} Dist(d, d_i)$	Use dynamic programming to find the K representatives

and use this to assign a weight to a spanning tree by computing the sum of the weights of the edges in that spanning tree, a minimum spanning tree or minimum weight spanning tree is then a spanning tree with weight less than or equal to the weight of every other spanning tree.

We can use an MST to represent a set of gene expression data and their significant inter-data relationship. An example of a set of expression data and its corresponding MST is given in 1. In this example, the weight between two node is calculated using Euclidean distance. There are also other ways to measure the distance between two gene expression profiles such as correlational distance and mahalanobis distance. An MST of a weighted graph can be found by a greedy method, such as the classical Kruskal's algorithm [9].

After finding an MST T for a weighted graph, we can partition T into K subtrees, for some specified integer $K > 0$. These K subtrees correspond to K clusters. Since different clustering problems need different objective functions to achieve best performance, three objective functions and their corresponding procedures [7] are presented in Table 1.

2.3 Gene Expression Clustering Based on Ant-MST

We propose Ant-MST, an ant-based minimum spanning tree, for gene expression clustering. Given N genes $g_i, i = 1, \ldots, N$ and their expression profile $E_i = \langle a_{i1}, a_{i2}, \ldots, a_{im} \rangle, i = 1, \ldots, N$, we want to cluster these genes into several categories based on similarities between their expression profiles. Figure 2 describes our algorithm in details.

1. **Initialization**.
   N genes corresponds to N points in the graph. N genes are connected by $\frac{1}{2}N \times (N-1)$ edges.
   For every edge (i, j), set an initial value $\tau_{ij}(t)$ for pheromone intensity.
   Place m ants randomly on the N points.
2. **Construct a fully connected network of nodes** G
   The fully connected network of nodes is built using the ant-based clustering algorithm. Details can be found in [8]. Each edge is associated with a pheromone intensity τ.
3. **Build an MST** T **for the connected graph** G
   Initially, set T contain an edge with the smallest
   pheromone intensity in the G, remove the edge from G.
   Do the following iteratively Until all vertices are
   connected by the selected edges:
     add the edge with the smallest pheromone intensity
     in the G
     make sure that no cycle is formed.
   EndLoop
4. **Partition** T **into** K **subtrees**
   There are three methods to perform the partition which have been presented in Table 1.

Fig. 2. Gene expression clustering algorithm based on Ant-MST

3 Experimental Results

3.1 Setup

After the investigation of the suitability of various datasets in Stanford Genomic Resource Database[2], three datasets were chosen to evaluate the performance of our algorithms.

The dataset I is a subset of gene expression data in the yeast Saccharomyces cerevisiae (SGD)[3], which is commonly known as baker's or budding yeast. A set of 68 genes with each gene having 79 data points is chosen.

The dataset II is a temporal gene expression dataset in response of human fibroblasts to serum[4]. It consists of 517 genes and each gene has 18 data points.

[2] http://genome-www.stanford.edu/
[3] http://www.yeastgenome.org/
[4] http://genome-www.stanford.edu/serum/

In this dataset, genes are listed according to their cluster order along with their Gene bank Accession number and Clone IDs. Gene names with the SID prefix are not sequence verified. 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 dataset III is the rat central nervous system development dataset[5]. It is obtained by researchers using the method of reverse transcription-coupled PCR to study the expression levels during rat central nervous system development.

3.2 Results

Rand index [10] is used to evaluate the performance of the clustering algorithm. It is a metric to measure the similarity between two clusters which contain exactly the same data objects. In our experiments, rand index is used to measure the number of pair-wise agreements of resultant clusters from our algorithms and the "expert" classes, normalized by the total number of pair-wise combinations.

The expression of Rand Index is as following:

$$R(M, N) = \frac{a + d}{a + b + c + d} \tag{1}$$

Where M is the number of "expert" classes, N is the number of clusters to be evaluated, a is "true positive pairs", it is the number of pairs with same class label of "expert class" that are assigned into the same cluster, d is "true negative pairs", it is the number of pairs with different class label that are assigned into different cluster, b is "false negative pairs", it is the number of pairs of the same "expert" class label that are assigned to different clusters, c is "false positive pairs", it is number of pairs of different "expert" class label that are assigned to the same clusters. Rand index lies between 0 and 1; a high value indicates high the degree of agreements of resultant clusters and "expert" classes. Table 2 shows the detailed "expert" information of these datasets.

Table 2. Statistics on experimental data

	Gene Cluster					
Dataset	A	B	C	D	E	F
I	28	17	15	8	-	-
II	305	43	7	162	-	-
III	27	20	21	17	21	6

Table 3 lists the experimental results based on the different objective functions, MST-R, MST-I and MST-G as shown in Table 1, and on different datasets. Results using the classical clustering algorithms such as Agglomerative Hierarchical Clustering (AHC) and K-means are also presented.

[5] http://www.arclab.org/node_pages/265.html

Table 3. Comparison of experimental results on different algorithms

	Rand Index		
Methods	**Dataset I**	**Dataset II**	**Dataset III**
MST-R	0.910	0.541	0.293
MST-I	**0.936**	0.682	0.582
MST-G	0.923	**0.811**	0.568
AHC	0.803	0.628	0.575
K-means	0.701	0.565	**0.676**

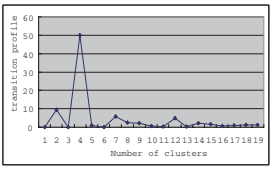

Fig. 3. Transition profile diagram of dataset I

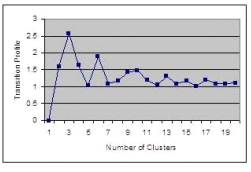

Fig. 4. Transition profile diagram of dataset II

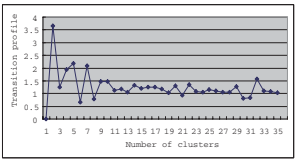

Fig. 5. Transition profile diagram of dataset III

It can be observed from Table 3 that the performance of clustering algorithm based on MST is better than that of AHC and K-means on dataset I and II. The rand index value achieved is 93.6% by MST-I on dataset I and 81.1% by MST-G on dataset II. However, the rand index values obtained using the MST-based methods on dataset III are lower than that of K-means with MST-I slightly outperforming AHC. The probably reason of better performance of K-means on dataset III is that the exact cluster number 6 was preset by the user while in practice it is hard to predict the correct cluster number.

The MST-based methods are able to calculate the optimal number of clusters automatically based on the transition profile values. Figure 3, 4, 5 are the transition profile diagrams for dataset I, II and III respectively. In the transition profile diagram, the x-axis represents the number of cluster, while the y-axis represents transition profile values. The highest transition profile value indicates the optimal number of clusters. It can be observed from Figure 3 that the optimal number of clusters in dataset I is 4, which is same as the actual number of clusters as can be found in Table 2. While for dataset II, the optimal number of clusters is 3 as shown in Figure 4. This is slightly different from the actual cluster number 4. Figure 5 reveals that the optimal number of clusters in dataset III is 3 which is different from the actual cluster number 6. This also explains the worse performance of MST-based methods in dataset III.

4 Conclusions and Future Work

In this paper, we have presented a clustering algorithm Ant-MST for gene expression data clustering. It consists of two stages. First construct a fully connected

network of nodes using the ant-based clustering algorithm and then build an MST from the fully connected graph and partition it into K clusters. Experimental results on three different datasets have been presented to illustrate its feasibility and efficiency. In future work we will continue on the enhancement of the gene expression data clustering component and conduct a large scale of experiments to evaluate the system performance.

References

1. Baldi, P., Brunak, S.: Bioninformatics: The machine learning approach (2001)
2. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences of the United States of America 95(14), 14863–14868 (1998)
3. Wen, X., Fuhrman, S., Michaels, G.S., Carr, D.B.: Large-scale temporal gene expression mapping of central nervous system development. Proceedings of the National Academy of Sciences of the United States of America 95(1), 334–339 (1998)
4. Herwig, R., Poustka, A.J., Mller, C., Bull, C.: Large-scale clustering of cdna-fingerprinting data. Genome Research 9(11), 1093–1105 (1999)
5. Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S.: Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. Proceedings of the National Academy of Sciences of the United States of America 96(6), 2907–2912 (1999)
6. Xu, R., Wunsch II, D.: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)
7. Xu, Y., Olman, V., Xu, D.: Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees. Bioinformatics 18(4), 536–545 (2002)
8. He, Y., Hui, S.C., Sim, Y.: A Novel Ant-Based Clustering Approach for Document Clustering. In: Asia Information Retrieval symposium, pp. 537–544. Springer, Heidelberg (2006)
9. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The design and analysis of computer algorithms (1974)
10. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66, 622–626 (1971)