FUZZY HYPER-CLUSTERING FOR PATTERN CLASSIFICATION IN MICROARRAY GENE EXPRESSION DATA ANALYSIS

Jin Liu and Tuan D. Pham

School of Engineering and Information Technology, University of New South Wales, Canberra ACT 2600, Australia

Keywords: Fuzzy *c*-means, Hyperplanes, Microarray gene expression data.

Abstract: Based on the motivation by computational challenges in microarray data analysis, we propose a fuzzy hyper-cluster analysis as a new framework for pattern classification using such type of data. This approach uses hyperplanes to represent the cluster centers in the fuzzy *c*-means algorithm. We present in this position paper the formulation of a hyperplane-based fuzzy objective function and suggest possible solutions. Fuzzy hyper-clustering approach appears to have potential as a novel alternative to analyze microarray gene expression data. Furthermore, the proposed hyper-clustering algorithm is not only confined to microarray data analysis but can be used as a general approach for classifying closely related features.

1 INTRODUCTION

Microarray technology has been developed during last few years and becomes a popular analysis method for studying gene expression. The advantage of microarray analysis lies in that it enables researchers to work on the expression patterns of tens of thousands genes simultaneously. Although microarray technology provides convenient methods to analyze gene expression, the analysis process is complex and difficult. Many computational tools has been applied to microarray gene expression data analysis. According to (Pham et al., 2006), these methods can be classified into two categories, classification-based and clustering-based. Classification-based methods like support vector machines (SVMs) (Statnikov et al., 2005) and *k*-nearest neighborhood (*k*-NN) (Pham, 2005). Clustering-based methods like fuzzy *c*-means (FCM)(Asyali and Alci, 2005) and self-organizing map (SOM) (Dougherty et al., 2002).

In this paper, we propose a fuzzy hyper-clustering approach for pattern classification in microarray gene expression data. The proposed method can be viewed as an extension of the fuzzy *c*-means clustering which uses hyperplanes as cluster centers. We formulate the objective function for the fuzzy hyper-clustering and discuss possible solutions using iterative numerical method or nature-inspired optimization method.

In literature review, there are some methods being similar to the proposed fuzzy hyper-clustering. In (Bradley and Mangasarian, 2000), the authors proposed to use *k* planes as cluster prototypes and adopted eigenvalue decomposition to calculate these fitting planes, the work is followed by a number of research. Some of the following research used parallel or non-parallel fitting planes to perform binary classification (Yang et al., 2009), some methods were extended to multicategory classification by using one-from-rest methods for each class (Jayadeva et al., 2007).

The papers mentioned above are similar but different to the proposed approach. In the proposed method, memberships of data samples assigned to the cluster prototypes are of continuous values which make it suitable for gene expression data in which patterns often overlap, and the proposed clustering method is a kind of unsupervised learning which is also different from the work mentioned above.

The rest of the paper is organized as the follows. In Section 2, we report current challenges in microarray data analysis. Section 3 presents the proposed fuzzy hyper-clustering and possible solutions. Finally, concluding remarks of this position paper is given in Section 4.

2 CHALLENGES IN MICROARRAY DATA ANALYSIS

Although microarray data sets can be in different forms due to various experiments platforms, they impose common challenges in the analysis which we

Liu J. and D. Pham T. (2010).
FUZZY HYPER-CLUSTERING FOR PATTERN CLASSIFICATION IN MICROARRAY GENE EXPRESSION DATA ANALYSIS.
In *Proceedings of the Third International Conference on Bio-inspired Systems and Signal Processing*, pages 415-418
DOI: 10.5220/0002719504150418

will discuss in the next section.

2.1 High-dimension Small-sample

Firstly, microarray data are often in high-dimension and small-sample. Most of the publicly available microarray data sets usually consist of less than 20 samples with more than thousands of gene features (Golub et al., 1999). The high-dimension small-sample problem is partly caused by the imbalanced developing speed between slow sample collection and rapid sequencing technology. As the development of microarray chips also follows the Moore's law from the semiconductor, the high-dimension small-sample problem would probably exist for long time.

2.2 High Redundancy

Another character of microarray gene-expression data is that the data is highly redundant. Thus, the algorithm which is to be utilized has to be able to discover expression patterns in a large amount of irrelevant genes. In this case, feature selection becomes important to improve the performance of pattern classification. Unfortunately, most of the conventional feature selection algorithms may not work well in high-dimension and small-sample microarray data sets (Ding and Peng, 2003). The requirements for feature selection in microarray include that the algorithm should be computationally efficient, and it should work well with small-sample data.

2.3 Inherent Noise

The effect of noise introduced from the manufacturing process of microarray chips could not be ignored. The noise could be introduced from different stages and by different reasons. The inherent noise makes the analysis difficult for some computational tools like *c*-means clustering and hierarchical clustering as these methods are sensitive to incomplete or inaccurate information. Although the missed feature value can be imputed through some kind of estimation, the estimated value could be even more unreliable, and may adversely affect the analysis results (Suzuki et al., 2000). To produce reliable analysis results, the algorithms which are to be applied should be robust to noise and reduce the negative effects.

2.4 Gene Expression Overlapping

Another problem in microarray data analysis is that the clusters usually overlap (Baken et al., 2008). This is because each gene can have more than one function. Popular computational tools like *c*-means and SOM assign crisp membership to genes, which would distort the clustering shape in overlapped gene expression analysis and could not identify co-expression with different groups of genes. As an alternative, fuzzy clustering, which assigns continuous membership grades, can be used to analyze overlapped information about gene multi-functionality and to reveal the relative likelihood of each gene belonging to each cluster.

To get a reliable results and meaningful explanations from microarray gene expression data, the computational tools have to be capable for analyzing data sets with characters listed above. Facing challenges for microarray data analysis, there is always a need for new analysis method that could bear better performance.

3 A FUZZY HYPER-CLUSTERING TECHNIQUE

Being motivated by the useful concepts of coupling fuzzy *c*-means clustering with hyperplane mapping, we aim to solve the expression-overlapping, high-dimension, and small-sample problems in microarray gene expression data analysis by proposing a fuzzy hyper-clustering technique. We discuss the proposed method in subsequent sections.

3.1 Hyperplane-based Fuzzy Clustering

Being different from most current clustering techniques such as *c*-means and fuzzy *c*-means clustering, which represent the cluster centers using *p*-dimension mean vectors of the data samples, the proposed fuzzy hyper-clustering adopts the geometry of hyperplanes, which was employed to develop support vector machines and other kernel-based methods (Cristianini and Shawe-Taylor, 2000), to represent its cluster centers $\mathbf{h}_j = (\mathbf{w}_j, v_j)$, $j = 1, \ldots, c$, where c is the number of clusters. From now on, we will refer $\mathbf{h}_j$ as hypercluster, an example of a two dimensional hypercluster in a single class is shown in Figure 1.

In the proposed clustering technique, sample points are assigned fuzzy memberships to each hypercluster according to its distances to the hyperclusters. The aim of the fuzzy hyper-clustering is to find a fuzzy partition matrix $\mathbf{U} = [u_{ij}]$, $i = 1, \ldots, n$, n is the number of samples; and hyperclusters $\mathbf{h}_j$ that mini-

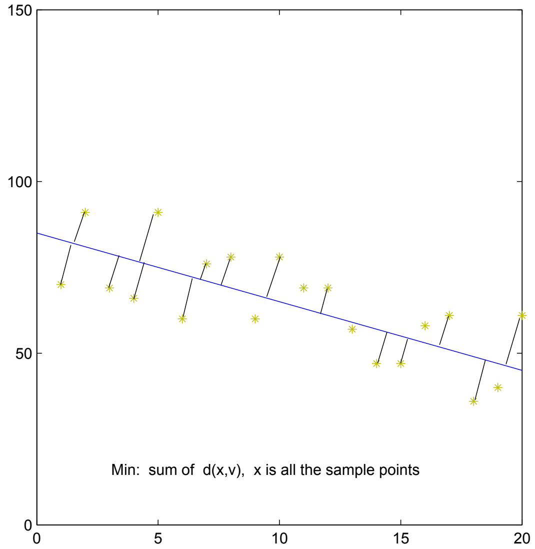

Figure 1: Two-dimensional fuzzy hyper-clustering for a single class.

mizes the sum of the distances from all sample points to all hyperclusters.

$$U_{n\times c} = \begin{bmatrix} u_{11} & u_{12} & . & \dots & u_{1c} \\ u_{21} & u_{22} & . & \dots & u_{2c} \\ . & . & . & \dots & . \\ \dots & . & . & u_{ij} & \dots \\ u_{n1} & . & . & \dots & u_{nc} \end{bmatrix} \tag{1}$$

$$\sum_{j=1}^{c} u_{ij} = 1; i = 1,\dots,n; j = 1,\dots,c; \tag{2}$$

$$u_{ij} \in [0,1] \tag{3}$$

where u_{ij} is the fuzzy membership of i-th object data vector to j-th hypercluster. The resulting partition matrix and hyperclusters would minimize the following objective function:

$$J = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^{m} d(\mathbf{x}_i, \mathbf{h}_j) \tag{4}$$

where $\mathbf{h}_j$ is the j th hypercluster $(\mathbf{w}_j, v_j)$,

$$\mathbf{h}_j = \{w_{1j}, w_{2j}, w_{3j}, \dots, w_{pj}, v_j\}$$

and $\mathbf{w}_j = \{w_{1j}, w_{2j}, w_{3j}, \dots, w_{pj}\}$ is a p-dimensional normal vector to the j-th hypercluster. The distance from a data point to the hypercluster is:

$$d(\mathbf{x}_i, \mathbf{h}_j) = \frac{|\mathbf{w}_j \cdot \mathbf{x}_i - v_j|}{||\mathbf{w}_j||^2} \tag{5}$$

$$||\mathbf{w}_j|| = 1; j = 1,\dots,c; \exists w_{ij} \neq 0 \tag{6}$$

where $\mathbf{w}_j \cdot \mathbf{x}_i$ is the dot product between vector $\mathbf{w}_j$ and vector $\mathbf{x}_i$.

3.2 Proposed Solutions

To find a solution that minimizes the above objective function J, we adopt an iterative numerical model which updates the fuzzy partition process until a convergence of the solution is reached. This process is analogous to the fuzzy c-means clustering algorithm. In addition, we propose to use nature-inspired optimization based methods as alternative solutions.

3.2.1 Iterative Numerical Method

For an iterative numerical method, by taking the first derivatives of objective function J with respective to the variants and setting them to zero, we can get the necessary conditions for the objective function to reach a minimum. The parameters can be updated according to the following steps:

1. Initialize partition matrix $\mathbf{U}$ and hyperclusters $\mathbf{h}_j, j = 1,\dots,c$.
2. Calculate the new hyperclusters which minimize the objective function J under the current partition matrix $\mathbf{U}^t$, where t is the iteration count, then we get the updated $\mathbf{h}_j^{t+1}$.
3. Following the above computation, we obtain the newly updated fuzzy hyperclusters $\mathbf{h}_j^{t+1}$, then calculate the new fuzzy partition matrix, the updated fuzzy partition matrix $\mathbf{U}$ that minimizes the objective function J under the current fuzzy hyperclusters $\mathbf{h}_j^{t+1}$.
4. If the algorithm converges, then the computation stops. Otherwise, go to Step 2.

We consider the algorithm converges if the maximum change in the partition matrix between iterations is less than a preset positive small number ε. The resulting partition matrix $\mathbf{U}^*$ and hyperclusters $\mathbf{h}_j^*, j = 1,\dots,c$, satisfy the solution which minimizes the objective function J.

3.2.2 Particle Swarm Optimization

Particle swarm optimization (PSO) is a kind of evolutionary computation which has been used in many areas including clustering (Feng et al., 2006). When applying PSO into the proposed fuzzy hyper-clustering, we need to identify the positions of particles and the fitness function. An intuitive choice is to use the c hyperclusters $\mathbf{h}_j, j = 1,\dots,c$, to represent the position of a particle, and the objective function J to be the fitness function.

The PSO-based fuzzy hyperclustering can be updated according to the following steps:

1. Encode the position for each particle as c hyperclusters $\mathbf{h}_j, j = 1, ..., c$, then initialize partition matrix $\mathbf{U}$ and first generation of particles.
2. Start the evolution under the current partition matrix $\mathbf{U}^t$ and fitness function J. The PSO updates position and velocity for each particle. Evolution continues until an optimal particle that minimizes the objective function J under the current partition matrix $\mathbf{U}^t$ is found. Then we get the c hyperclusters $\mathbf{h}_j^{t+1}, j = 1, ..., c$.
3. After we get the c hyperclusters $\mathbf{h}_j^{t+1}$, we then calculate the new fuzzy partition matrix for each data point, the updated fuzzy partition matrix $\mathbf{U}^{t+1}$ would minimize the objective function J under the current fuzzy hyperclusters.
4. If the algorithm converges or reaches the maximum iteration numbers, the computation stops. Otherwise, go to Step 2.

The convergence condition is similar to that of the iterative numerical solution.

4 CONCLUSIONS

We have presented a proposed fuzzy hyper-clustering algorithm for pattern classification in microarray gene expression data. We formulated the objective function for the proposed hyper-clustering and discussed possible solutions using numerical and nature-inspired optimization methods. The proposed clustering method can be: 1) suitable for overlapping data samples as fuzzy membership is utilized; 2) computationally efficient as the calculation for hyperclusters may use generalized eigenvalue decomposition which is simpler than that in standard SVMs; 3) potential to handle nonlinear data as a kernelized version of the proposed method can take advantage of the kernel trick for nonlinear data analysis; 4) suitable for high dimensions small sample sizes data sets as the supervised version of the proposed method can be viewed as a variant of SVMs which currently is known as the best high-dimension small-sample problem solver. Furthermore, the proposed approach can be applied to many other different areas, not only confined to microarray gene expression analysis.

REFERENCES

Asyali, M. H. and Alci, M. (2005). Reliability analysis of microarray data using fuzzy c-means and normal mixture modeling based classification methods. *Bioinformatics*, 21:644–649.

Baken, K. A., Pennings, J. L., Jonker, M. J., Schaap, M. M., de Vries, A., van Steeg, H., Breit, T. M., and van Loveren, H. (2008). Overlapping gene expression profiles of model compounds provide opportunities for immunotoxicity screening. *Toxicology and Applied Pharmacology*, 226:46–59.

Bradley, P. S. and Mangasarian, O. L. (2000). k-plane clustering. *J. Global Optimization*, 16:23–32.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge.

Ding, C. and Peng, H. (2003). Minimum redundancy feature selection from microarray gene expression data. In *Proc. 2003 IEEE Computer Society Bioinformatics Conference*, pages 523–529.

Dougherty, E. R., Barrera, J., Brun, M., Kim, S., Cesar, R. M., Chen, Y., Bittner, M., and Trent, J. M. (2002). Inference from clustering with application to gene-expression microarrays. *J. Computational Biology*, 9:105–126.

Feng, H. M., Chen, C. Y., and Ye, F. (2006). Adaptive hyper-fuzzy partition particle swarm optimization clustering algorithm. *Cybernetics and Systems*, 37:463–479.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537.

Jayadeva, Khemchandaniand, R., and Chandra, S. (2007). Fuzzy multi-category proximal support vector classification via generalized eigenvalues. *Soft Computing*, 11:679–685.

Pham, T. D. (2005). An optimally weighted fuzzy k-NN algorithm. In *Proc. 2005 Int. Conf. Advances in Pattern Recognition*, pages 239–247.

Pham, T. D., Wells, C., and Crane, D. I. (2006). Analysis of microarray gene expression data. *Current Bioinformatics*, 1:37–53.

Statnikov, A., Aliferis, C. F., Tsamardinos, I., Hardin, D., and Levy, S. (2005). A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21:631–643.

Suzuki, T., Hashimoto, S.-i., Toyoda, N., Nagai, S., Yamazaki, N., Dong, H.-Y., Sakai, J., Yamashita, T., Nukiwa, T., and Matsushima, K. (2000). Comprehensive gene expression profile of LPS-stimulated human monocytes by SAGE. *Blood*, 96:2584–2591.

Yang, X., Chen, S., Chen, B., and Pan, Z. (2009). Proximal support vector machine using local information. *Neurocomputing*, in-print.