Fuzzy mixed-prototype clustering algorithm for microarray data analysis

Jin Liu, Tuan Pham, Hong Yan and Zhizhen Liang

Fuzzy Mix-Prototype Clustering Algorithm for Microarray Data Analysis

Jin Liu[a], Tuan D. Pham[b], Hong Yan [c], Zhizhen Liang[a]

[a] *School of Computer Science*
China University of Mining and Technology
Xuzhou, Jiangshu 221008, China
[b] *Department of Biomedical Engineering*
Linkoping University
581 83 Linkoping, Sweden
[c] *Department of Electronic Engineering,*
City University of Hong Kong, Kowloon, Hong Kong

Abstract

Being motivated by combining the advantages of hyperplane-based pattern analysis and fuzzy clustering techniques, we present in this paper a fuzzy mix-prototype (FMP) clustering for microarray data analysis. By integrating spherical and hyper-planar cluster prototypes, the FMP is capable of capturing latent data models with both spherical and non-spherical geometric structures. Our contributions of the paper can be summarized into three folds: First, the objective function of the FMP is formulated. Second, an iterative solution which minimizes the objective function under given constraints is derived. Third, the effectiveness of the proposed FMP is demonstrated through experiments on yeast and Leukemia data sets.

Keywords: FMP, microarray data analysis, fuzzy clustering

1. Introduction

Despite the fact that the fuzzy c-means(FCM) algorithm has been applied in different areas successfully, it has been known that the FCM may perform well only when the data set is of spherical or hyperspherical structure. However, in real world applications, there may be many other different types of

Email address: liujincumt@gmail.com (Jin Liu[a])

data structures in which most of the current clustering algorithms may fail to perform well[1], for instance, linear or hyperplane shaped data clusters. Some techniques are good at linear or non-linear cluster structures detection, i.e., graph-theoretic methods, but there are no explicit prototypes for the clusters, hence it is difficult to further explain the clustering results and perform classification. Furthermore, in some certain research areas, such as image processing and computer vision, clustering algorithms need to consider not only the cluster prototypes but also the geometry of clusters to perform structure segmentation. Last but not least, data samples in real world applications often overlap with each other, i.e., microarray gene expression data [2, 3, 4]. For any clustering algorithms, how to take both properties of overlapping and the linear subspace structure of the data samples into consideration is worth investigation.

Since the proposal of Support Vector Machines(SVMs), hyperplanes-based pattern analysis is attracting more and more attention from research community as a result of that the technique provides researchers with great power to handle many different pattern classification problems [5, 6, 7, 8]. As one of the most successful classification methods, the SVMs aim to find an optimal separating hyperplane between two different categories of data to perform data classification. Through taking advantage of the kernel trick method, the SVMs are capable of differentiating linearly inseparable data set. The technique is famous for their excellent performance in pattern classification and have been used widely and successfully. Nevertheless, the SVMs are also known for their computation cost during their training process. Estimation of an optimal separating hyperplane is achieved by solving a quadratic programming problem which involves kernel matrix invertion. The training process of SVMs is of complexity on the order of $O(n^3)$, where n is the number of samples in the training set. Recently, many efforts have been devoted to relieve the computational burden of the SVMs while withholding the classification accuracy through adopting the hyperplanes-based approximation [9, 10, 11, 12, 13]. Being different from the original SVMs, hyperplanes in these works are adopted to approximate different types of data rather than to split them from each other. The optimal

hyperplane minimizes the sum of squared Euclidean distances from one cluster and maximize the sum of squared Euclidean distances from the other cluster. The objective functions are in the form of Rayleigh quotient and the solution can be achieved by generalized eigenvalue decomposition. By this means, the efficiency of these algorithms and the accuracy of classification were reported.

For unsupervised pattern recognition techniques, hyperplane-based clustering algorithms are also attracting research attention widely. A k-planes clustering technique was put forward in [14], where hyperplanes were adopted to represent cluster centers. The objective of the clustering is to minimize the sum of the squared Euclidean distances between data and their projections on their representative hyperplane. The k-planes clustering algorithm iteratively updates the partition matrix and clustering hyperplanes until convergence reached. The k-bottleneck hyperplane clustering (k-bHPC), which is another hyperplane-based clustering technique, was put forward in [6]. The objective function of k-bHPC is the minimum of maximum distance from the data samples to their belonging hyperplane. The clustering algorithm aims to find a group of hyperplanes and a partition matrix which can minimize the given objective function.

There are also some other related works have been put forward recently [15, 16, 17, 18, 19, 20]. An Extreme Learning Mchine (ELM)-based method for heat load prediction in district heating system was presented in [18]. Nine different ELM predictive models were developed for time horizon from 1 to 24h ahead. Experiments results were compared with that of genetic programming (GP) and artificial neural networks (ANNs) models. Improvements in predictive accuracy and capability of generalization were demonstrated. In [17], an Expert Multi Agent System(E-MAS) based Support Vector Regression(SVR) was proposed to determine collar dimensions around bridge pier. In [19], a fuzzy clustering approach based on fuzzy distance measurement was presented, and multi-objective mathematical programming was then adopted for further optimization. In [16], a novel density-based fuzzy clustering algorithm based on Active Learning Method (ALM) was presented. In [20], a collaborative clustering framework which combines fuzzy c-means (FCM) and mixture mode

was presented for mixed data which contains both numerical and categorical attributes.

Being motivated by the useful concepts of combining hyperplane-based data proximation with fuzzy clustering techniques, we presented herein a fuzzy mix-prototype clustering technique in which hyperplanes and hyperspheres are used to form cluster prototypes. The objective function of the proposed clustering technique is the sum of the distances from all of the data points to the clustering hyperplanes, weighted by the degree of the point belonging to the corresponding clusters, and penalized by distances of data samples to cluster mass centers. The proposed fuzzy mix-prototype clustering aims to find a solution to minimize the fuzzy objective function under given constraints. The clustering problem can then be considered as a constrained optimization problem and an iterative solution can then be obtained by using the Lagrangian multiplier method. The solutions are the resulting clusters that minimize the objective function.

The rest of the paper is organized as the follows. Section 2 gives a summary of some related works. Section 3 describes the proposed fuzzy mix-prototype clustering in detail, including formulation of the fuzzy objective function, derivation of an solution and description of the resulting algorithm. In Section 4, we report the experimental results of the proposed method and compare these results with those obtained from some existing methods. Concluding remarks of the proposed approach are addressed in Section 5.

2. Related Work

Some methods which are closely related to the proposed fuzzy mix-prototype clustering are briefly discussed in the following subsections.

2.1. Fuzzy c-means clustering

Fuzzy c-means clustering is a kind of soft clustering which allows a data point to belong to more than one cluster [1]. The membership u_{ij} is an continuous value which denotes the degree of data point $\mathbf{x}_i$ belong to cluster j, and it is the entry of ith row and jth column of membership matrix U.

FCM uses Euclidean distance to represent the dissimilarity between vectors and the algorithm is derived according to minimization of the following objective function

$$J_{FCM} = \sum_{i=1}^{n}\sum_{j=1}^{c}(u_{ij})^m||\mathbf{x}_i - \mathbf{v}_j||_2^2 \tag{1}$$

where $m \in [1, +\infty)$ denotes the fuzziness degree, n and c denote the number of vectors and cluster centers respectively, $\mathbf{v}_j$ denotes the j-th cluster center, and $||\mathbf{x}||_2^2$ represents the squared norm 2 of vector $\mathbf{x}$.

The minimization of J_{FCM} subjects to the following constraints

$$u_{ij} \in [0,1], i = 1, ..., n, j = 1, ..., c \tag{2}$$

$$\sum_{j=1}^{c} u_{ij} = 1 \tag{3}$$

$$0 < \sum_{i=1}^{n} u_{ij} < n, j = 1, ..., c \tag{4}$$

By using the Lagrangian multiplier method, necessary conditions for minimizing J_{FCM} under the given constraints can be derived and the cluster centers and partition matrix can be updated according to

$$\mathbf{v}_j = \sum_{i=1}^{n}(u_{ij})^m\mathbf{x}_i / \sum_{i=1}^{n}(u_{ij})^m \tag{5}$$

$$u_{ij} = \sum_{k=1}^{c}\left(\frac{||\mathbf{x}_i - \mathbf{v}_j||_2^2}{||\mathbf{x}_i - \mathbf{v}_k||_2^2}\right)^{\frac{1}{1-m}} \tag{6}$$

And the algorithm is summarized as

1. Randomly initialize the membership $u_{ij}, i = 1, ..., n; j = 1, ..., c$;
2. Given termination criterion $\varepsilon \in (0, 1)$;
3. Set t=0, iterate:
   (a) update cluster center according to Eq. 5;
   (b) compute $||\mathbf{x}_i - \mathbf{v}_j||$;

(c) update membership u_{ij} according to Eq. 6;

(d) if $||U^{(t+1)} - U^{(t)}|| < \varepsilon$ then stop, otherwise continue.

where the fuzzy weighting exponent m is usually chosen as 2.

2.2. Kernel FCM

Kernel FCM (KFCM) is an variant of FCM which extends fuzzy clustering into kernel space [21]. The clustering method make use of kernel transformations to map vectors from the original p-dimensional feature space to a kernel space which is of higher dimensionality. Through this mapping, problems that are linearly non-separable in the original feature space become linearly separable in the kernel space, and then fuzzy clustering algorithms can be used to perform data analysis.

Kernel FCM takes advantage of the 'kernel trick' to perform data analysis. The 'kernel trick' is achieved by using a continuous, symmetric, positive semi-definite function which is known as 'kernel function'. By using this kernel function, the inner product between two vectors in the kernel space can be directly computed, without knowing the explicit form of the vectors in the kernel space.

For example, kernel function $K(\mathbf{x},\mathbf{y})$ where

$$K(\mathbf{x},\mathbf{y}) = \phi(\mathbf{x})^T \phi(\mathbf{y}), \tag{7}$$

represents the inner product between two vectors $\mathbf{x}, \mathbf{y}$ in the kernel space, and $\mathbf{x}, \mathbf{y} \in \mathbf{R}^p$ are p-dimension vectors, function $\phi(\mathbf{x})$ denotes the transformation of vector $\mathbf{x}$ from the p-dimensional feature space to kernel space. By this means, many pattern classification algorithms using inner products between vectors can be extended to their kernel versions.

Some of the commonly used kernel functions including Gaussian kernel, Polynomial kernel, Hyper-tangle kernel, etc. which are shown as in Table 1.

Table 1: Commonly used kernel functions

Type of kernel functions	Expression
Gaussian kernel	$e^{-\|\|\mathbf{x}-\mathbf{y}\|\|^2/\delta^2}, \delta^2 > 0$
Polynomial kernel	$(\mathbf{x}^t\mathbf{y} + \theta)^p, \theta \geq 0, p \in N$
Hyper-tangent kernel	$\tanh(\mathbf{x}^t\mathbf{y}) + \theta, \theta \geq 0$

For KFCM, the algorithm can be further categorized into two subtypes according to whether the prototypes are located in feature space (KFCMf), or kernel space (KFCMk).

The KFCMf tries to minimize the following objective function

$$J_{KFCMf} = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^m ||\phi(\mathbf{x}_i) - \phi(\mathbf{v}_j))||_2^2 \tag{8}$$

while KFCMk tires to minimize

$$J_{KFCMk} = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^m ||\phi(\mathbf{x}_i) - \mathbf{v}_j||_2^2 \tag{9}$$

The minimizations of J_{KFCMf} and J_{KFCMk} are under the same constrains of FCM algorithm. Lagrangians can be written and algorithms can be derived by iterating between the necessary conditions that minimizes the Lagrangians.

2.3. Gustafson-Kessel clustering

Being different from FCM and KFCM which use Euclidean distance to measure the similarities, the objective function of Gustafson-Kessel (GK) clustering algorithm is in the form [22]

$$J_{GK} = \sum_{k=1}^{n}\sum_{i=1}^{c} (u_{i,j})^m d^2(\mathbf{x}_i, \mathbf{v}_j) \tag{10}$$

where $d^2(\mathbf{x}_i, \mathbf{v}_j)$ denotes the squared distance between vector $\mathbf{x}_i$ and $\mathbf{v}_j$, and the distance is defined as

$$d^2(\mathbf{x}_i, \mathbf{v}_j) = (\mathbf{x}_i - \mathbf{v}_j)^t \mathbf{A}_j (\mathbf{x}_i - \mathbf{v}_j) \tag{11}$$

where $\mathbf{A}_j$ is a positive define matrix and is calculated according to

$$\mathbf{A}_j = (\rho_j |\mathbf{C}_j|^{1/d} \mathbf{C}_j)^{-1} \tag{12}$$

where ρ_j is the j-th positive scaling parameter, $|\mathbf{C}_j|$ and $\mathbf{C}_j^{-1}$ denote the determinant and inverse of fuzzy covariance matrix $\mathbf{C}_j$, respectively.

The J_{GK} is to be minimized under same constrains with those of FCM, and Lagrangian can be written and the algorithm can be derived by updating between necessary conditions that minimize the Lagrangian, and the clustering centers and partition matrix in GK algorithm are updated according to the follows

$$\mathbf{v}_j = \frac{\sum_{i=1}^{N} u_{ij}^m \mathbf{x}_i}{\sum_{i=1}^{N} u_{ij}} \tag{13}$$

$$\mathbf{C}_j = \frac{\sum_{i=1}^{N} u_{ij}^m (\mathbf{x}_i - \mathbf{v}_j)(\mathbf{x}_i - \mathbf{v}_j)^t}{\sum_{i=1}^{N} u_{ij}} \tag{14}$$

$$u_{ij} = \sum_{k=1}^{c} \left(\frac{(\mathbf{x}_i - \mathbf{v}_j)^t \mathbf{A}_j (\mathbf{x}_i - \mathbf{v}_j)}{(\mathbf{x}_- \mathbf{v}_k)^t \mathbf{A}_k (\mathbf{x}_i - \mathbf{v}_k)} \right)^{\frac{1}{1-m}} \tag{15}$$

2.4. Hyperplane-based clustering and classification

In previous works, some researchers used hyperplanes to perform pattern analysis [6]. A hyperplane based clustering which adopted hyperplane as clustering centroid was presented in [14]. The objective function is formulated as the sum of distance of data samples to their belonging hyperplanes. By adopting constrained optimization method, necessary conditions for minimizing the objective function can be obtained. The k-planes clustering algorithm iteratively updates the membership matrix by assigning data sample to its closest hyperplane and updates the hyperplanes by eigenvalue decomposition. The algorithm iterates till the convergence criterion is satisfied. In [6], the authors presented a hyperplane based clustering method which is called k-bottleneck hyperplane clustering (k-bHPC). The aim of k-bHPC is to partition data into several groups, and to find a hyperplane for each group that minimizes the maximum distance between data points to their projections on the hyperplane. The

objective function k-bHPC can be written as

$$argmin_{(w_j,v_j)} \max \frac{|\mathbf{w}_j^t \mathbf{x}_i - v_j|}{||\mathbf{w}_j||_2^2} \tag{16}$$

where the $\{\mathbf{w}_j, v_j\}$ is the hyperplane of the j-th cluster and $\mathbf{x}_i$ is the i-th sample.

Many researchers also use hyperplanes to perform classification. The support vector machines (SVMs) [23] is a well known classification method, which aims to find an optimal separating hyperplane that maximizes the margin between different types of data. Quadratic programming is used to calculate the optimal hyperplane for linearly separable data sets, and for data sets that are linearly unseparable, kernel trick is utilized to map the data into a higher dimension feature space. Although SVMs can give good classification performance, the training process of optimal hyperplane is computationally expensive. Some extensions of SVMs made efforts to reduce the computational burden while maintain the predictive accuracy. In these work, hyperplane is used to approximate each type of data rather than to separate them. The optimal hyperplane minimizes the sum of squared Euclidean distance of one type data and meanwhile maximizes the sum of squared Euclidean distance of the other type data. The objective function can be written in the form of Rayleigh quotient and the solution can be obtained through generalized eigenvalue decomposition. Some of the approximating hyperplanes are parallel to each other [5, 24], some are extended to be non-parallel [25, 12, 13], and others were extended to perform multi-category classification by using the one-from-the-rest approach[24].

3. The Fuzzy Mix-Prototype Clustering

3.1. The FMP objective function

The traditional clustering techniques such as the c-means and the fuzzy c-means clustering adopt p-dimensional data vector to represent the underlying data structures. Being different, the proposed fuzzy mix-prototype clustering uses the geometrical hyperplanes $\mathbf{h}_j = (\mathbf{w}_j, v_j)$, $j = 1, ..., c$ which maximizes the cluster variances, where c is the number of clusters. In the following parts of the

paper, the $\mathbf{h}_j$ will be quoted as a hypercluster. In addition, the mass center for each hypercluster is calculated, the memberships of samples to hyperclusters are penalized if they are far away from the mass center, by which indefinite clusters can be avoided. In the rest parts of the paper, all the vectors are column vectors by default and written in bold. Transpose of a vector or matrix is written with superscript t. For instance, $\mathbf{w}_j$ represents a column vector, which is the normal vector of the j-th hyperplane while $\mathbf{w}_j^t$ represents its transpose.

In the proposed clustering method, the sample points are assigned continuous memberships values to each hypercluster based on its distances to the hyperclusters and the mass centers. The FMP aims to find a fuzzy partition matrix $\mathbf{U} = [u_{ij}]$, $i = 1, ..., n$, where n is the number of vectors; hyperclusters $\mathbf{h}_j$, $j = 1, ..., c$, and mass centers $\mathbf{g}_j$, $j = 1, ..., c$, that minimize the sum of the distances from all points to all hyperclusters and mass centers. The term u_{ij} is the membership of the i-th object data vector assigned to the j-th hypercluster.

The resulting partition matrix, hyperclusters and mass centers are designed to minimize the following objective function

$$J_{FMP} = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^{m}\Big(\gamma \cdot d^2(\mathbf{x}_i, \mathbf{h}_j) + (1-\gamma)\cdot d^2(\mathbf{x}_i, \mathbf{g}_j)\Big) \tag{17}$$

where $\gamma \in (0, 1)$ is a tradeoff parameter between the weight of hypercluster and mass center, $\mathbf{h}_j$ represents the j-th hypercluster $(\mathbf{w}_j, v_j)$, $\mathbf{g}_j$ represents the j-th mass center

$$\mathbf{h}_j = \{w_{1j}, w_{2j}, w_{3j}, ..., w_{pj}, v_j\} \tag{18}$$

$$\mathbf{g}_j = \{g_{1j}, g_{2j}, g_{3j}, ..., g_{pj}\} \tag{19}$$

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

$$d(\mathbf{x}_i, \mathbf{h}_j) = \frac{|\mathbf{w}_j^t \cdot \mathbf{x}_i - v_j|}{||\mathbf{w}_j||^2} \tag{20}$$

$$d(\mathbf{x}_i, \mathbf{g}_j) = ||\mathbf{x}_i - \mathbf{g}_j||_2 \tag{21}$$

$$||\mathbf{w}_j|| = 1; j = 1, ..., c; \exists w_{ij} \neq 0 \tag{22}$$

$$\sum_{j=1}^{c} u_{ij} = 1, i = 1, ..., n, u_{ij} \in [0, 1] \tag{23}$$

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

Minimizing the J_{FMP} under the given constraints can be formulated into a constrained optimization problem. By using the Lagrangian multiplier method, the problem can be solved and necessary conditions which minimizes the original objective function can be obtained. A numerical model can then be formulated by iteratively updating the fuzzy partition matrix, hyperclusters and mass centers until the convergence criterion is satisfied. The derivation for the iterative numerical solution will be given in the following subsection in detail.

3.2. An iterative solution to FMP

The objective function of the FMP is given to minimize:

$$J_{FMP} = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m \left(\gamma \cdot \frac{|\mathbf{w}_j^t \cdot \mathbf{x}_i - v_j|^2}{||\mathbf{w}_j||^2} + (1 - \gamma) \cdot ||\mathbf{x}_i - \mathbf{g}_j||_2^2 \right) \tag{24}$$

which subjects to the constraints expressed in Eq. (22) and Eq. (23).

The Lagrangian function of the J_{FMP} can be formulated as

$$L = J_{FMP} - \sum_{j=1}^{c} \lambda_j (\mathbf{w}_j^t \cdot \mathbf{w}_j - 1) - \sum_{i=1}^{n} \alpha_i (\sum_{j=1}^{c} u_{ij} - 1) \tag{25}$$

Taking the first partial derivatives of L with respective to u_{ij},

$$\begin{aligned} \frac{\partial}{\partial u_{ij}}(L) &= \frac{\partial}{\partial u_{ij}}(J_{FMP}) - 0 - \frac{\partial}{\partial u_{ij}} \left(\sum_{i=1}^{n} \alpha_i (\sum_{j=1}^{c} u_{ij} - 1) \right) \\ &= \frac{\partial}{\partial u_{ij}} \left[\sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m \left(\gamma \cdot d^2(\mathbf{x}_i, \mathbf{h}_j) + (1 - \gamma) \cdot d^2(\mathbf{x}_i, \mathbf{g}_j) \right) \right] - \alpha_i \\ &= m u_{ij}^{m-1} \left(\gamma \cdot d^2(\mathbf{x}_i, \mathbf{h}_j) + (1 - \gamma) \cdot d^2(\mathbf{x}_i, \mathbf{g}_j) \right) - \alpha_i \end{aligned} \tag{26}$$

by setting Eq. (26) equal to 0 and taking consideration of constraint Eq. (22),

$$mu_{ij}^{m-1}\Big(\gamma \cdot d^2(\mathbf{x}_i, \mathbf{h}_j) + (1-\gamma) \cdot d^2(\mathbf{x}_i, \mathbf{g}_j)\Big) = \alpha_i \tag{27}$$

from which the optimal u_{ij}^* which minimizes the Lagrangian L can be obtained

$$\begin{aligned} u_{ij}^* &= \left[\frac{\alpha_i}{m} \cdot \frac{1}{\gamma \cdot d^2(\mathbf{x}_i, \mathbf{h}_j) + (1-\gamma) \cdot d^2(\mathbf{x}_i, \mathbf{g}_j)}\right]^{\frac{1}{m-1}} \\ &= \left[\frac{\alpha_i}{m}\right]^{\frac{1}{m-1}} \left[\frac{1}{\gamma \cdot d^2(\mathbf{x}_i, \mathbf{h}_j) + (1-\gamma) \cdot d^2(\mathbf{x}_i, \mathbf{g}_j)}\right]^{\frac{1}{m-1}} \end{aligned} \tag{28}$$

Substituting Eq. (28) into constraint expressed in Eq. (23), giving

$$\sum_{j=1}^{c} \left[\frac{\alpha_i}{m} \cdot \frac{1}{\gamma \cdot d^2(\mathbf{x}_i, \mathbf{h}_j) + (1-\gamma) \cdot d^2(\mathbf{x}_i, \mathbf{g}_j)}\right]^{\frac{1}{m-1}} = 1 \tag{29}$$

then

$$\left[\frac{\alpha_i}{m}\right]^{\frac{1}{m-1}} = \left[\sum_{j=1}^{c} \left(\frac{1}{\gamma \cdot d^2(\mathbf{x}_i, \mathbf{h}_j) + (1-\gamma) \cdot d^2(\mathbf{x}_i, \mathbf{g}_j)}\right)^{\frac{1}{m-1}}\right]^{-1} \tag{30}$$

Substituting Eq. (30) into Eq. (28), update of membership can be obtained

$$u_{ij}^* = \frac{\left(\frac{1}{\gamma \cdot d^2(\mathbf{x}_i, \mathbf{h}_j) + (1-\gamma) \cdot d^2(\mathbf{x}_i, \mathbf{g}_j)}\right)^{\frac{1}{m-1}}}{\sum_{k=1}^{c} \left(\frac{1}{\gamma \cdot d^2(\mathbf{x}_i, \mathbf{h}_k) + (1-\gamma) \cdot d^2(\mathbf{x}_i, \mathbf{g}_k)}\right)^{\frac{1}{m-1}}} \tag{31}$$

It is obvious that the Lagrangian function expressed in Eq. (25) will reach its minimum at the same $(\mathbf{w}_j^*, v_j^*)$ with

$$\begin{aligned} L' =& \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^m \left(\gamma \cdot \frac{(\mathbf{w}_j^t \cdot \mathbf{x}_i - v_j)^2}{||\mathbf{w}_j||^2} + (1-\gamma) \cdot ||\mathbf{x}_i - \mathbf{g}_j||_2^2\right) \\ &- \sum_{j=1}^{c} \lambda_j (\mathbf{w}_j^t \cdot \mathbf{w}_j - 1) - \sum_{i=1}^{n} \alpha_i \Big(\sum_{j=1}^{c} u_{ij} - 1\Big) \end{aligned} \tag{32}$$

Similarly, by taking the first derivatives of L' with respective to v_j

$$\begin{aligned} \frac{\partial}{\partial v_j}(L') = \frac{\partial}{\partial v_j}\Bigg(& \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^m \Bigg(\gamma \cdot \frac{(\mathbf{w}_j^t \cdot \mathbf{x}_i - v_j)^2}{||\mathbf{w}_j||^2} \\ &+ (1-\gamma) \cdot ||\mathbf{x}_i - \mathbf{g}_j||_2^2\Bigg)\Bigg) - 0 - 0 \end{aligned} \tag{33}$$

by considering constraint Eq. (22),

$$\begin{aligned}\frac{\partial}{\partial v_j}(L') &= 2\sum_{i=1}^{n} u_{ij}^m \cdot \gamma \cdot (\mathbf{w}_j^t \cdot \mathbf{x}_i - v_j)\left(\frac{\partial}{\partial v_j}(\mathbf{w}_j^t \cdot \mathbf{x}_i - v_j)\right) \\ &= -2\sum_{i=1}^{n} u_{ij}^m \cdot \gamma \cdot (\mathbf{w}_j^t \mathbf{x}_i - v_j) \\ &= 2\gamma \cdot \left(\sum_{i=1}^{n} u_{ij}^m v_j - \sum_{i=1}^{n} u_{ij}^m \mathbf{w}_j^t \mathbf{x}_i\right) \end{aligned} \tag{34}$$

setting Eq. (34) equal to 0,

$$\sum_{i=1}^{n} u_{ij}^m v_j = \sum_{i=1}^{n} u_{ij}^m \mathbf{w}_j^t \mathbf{x}_i \tag{35}$$

and update of v_j can be obtained

$$v_j^* = \frac{\sum_{i=1}^{n} u_{ij}^m \mathbf{w}_j^t \mathbf{x}_i}{\sum_{i=1}^{n} u_{ij}^m} \tag{36}$$

which can be written into matrix form

$$v_j^* = \frac{\mathbf{w}_j^t \mathbf{X} \mathbf{u}_j^m}{\mathbf{e}^t \mathbf{u}_j^m} \tag{37}$$

where $\mathbf{e}$ is a n-dimensional column vector with all of its elements equal to one, $\mathbf{u}_j^m$ is the m-th power to $\mathbf{u}_j$ and $\mathbf{X}$ is the p by n data matrix.

Similarly, by taking the first derivative of L' with respective to $\mathbf{w}_j$,

$$\begin{aligned}\frac{\partial}{\partial \mathbf{w}_j}(L') = &\frac{\partial}{\partial \mathbf{w}_j}\left(\sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^m \left(\gamma \cdot \frac{(\mathbf{w}_j^t \cdot \mathbf{x}_i - v_j)^2}{||\mathbf{w}_j||^2} + (1-\gamma) \cdot ||\mathbf{x}_i - \mathbf{g}_j||_2^2\right)\right) \\ &- \frac{\partial}{\partial \mathbf{w}_j}\left(\sum_{j=1}^{c} \lambda_j (\mathbf{w}_j^t \cdot \mathbf{w}_j - 1)\right) - 0 \end{aligned} \tag{38}$$

consider constraints Eq. (22), the above equation can be written into

$$\begin{aligned}\frac{\partial}{\partial \mathbf{w}_j}(L') =& 2\gamma \cdot \sum_{i=1}^{n} u_{ij}^m (\mathbf{w}_j^t \cdot \mathbf{x}_i - v_j)\left(\frac{\partial}{\partial \mathbf{w}_j}(\mathbf{w}_j^t \cdot \mathbf{x}_i)\right) \\ &- \left(\frac{\partial}{\partial \mathbf{w}_j}(\sum_{j=1}^{c} \lambda_j \mathbf{w}_j^t \cdot \mathbf{w}_j)\right) \end{aligned} \tag{39}$$

Based on the theories of matrix calculus [26], it can be known $\partial(\mathbf{w}^t \cdot \mathbf{w})/\partial \mathbf{w} =$

$2\mathbf{w}$ and $\partial(\mathbf{w}^t \cdot \mathbf{x})/\partial\mathbf{w} = \mathbf{x}$, therefore $\partial L'/\partial\mathbf{w}_j$ can be written as

$$\begin{aligned}\frac{\partial}{\partial\mathbf{w}_j}(L') &= 2\gamma \cdot \sum_{i=1}^{n} u_{ij}^m(\mathbf{w}_j^t\mathbf{x}_i - v_j)\mathbf{x}_i - \lambda_j(2\mathbf{w}_j) \\ &= 2\bigg(\gamma \sum_{i=1}^{n} u_{ij}^m(\mathbf{x}_i^t\mathbf{w}_j - v_j)\mathbf{x}_i - \lambda_j\mathbf{w}_j\bigg) \end{aligned} \tag{40}$$

Setting Eq. (40) equal to 0 and substituting Eq. (36) into Eq. (40), yielding

$$\gamma \sum_{i=1}^{n} u_{ij}^m(\mathbf{x}_i^t\mathbf{w}_j - \frac{\sum_{i=1}^{n} u_{ij}^m\mathbf{w}_j^t\mathbf{x}_i}{\sum_{i=1}^{n} u_{ij}^m})\mathbf{x}_i - \lambda_j\mathbf{w}_j = 0 \tag{41}$$

as $\mathbf{x}_i^t\mathbf{w}_j - \frac{\sum_{i=1}^{n} u_{ij}^m\mathbf{w}_j^t\mathbf{x}_i}{\sum_{i=1}^{n} u_{ij}^m}$ is a scalar value, hence $(\mathbf{x}_i^t\mathbf{w}_j - \frac{\sum_{i=1}^{n} u_{ij}^m\mathbf{w}_j^t\mathbf{x}_i}{\sum_{i=1}^{n} u_{ij}^m}) \cdot x_i = x_i \cdot (\mathbf{x}_i^t\mathbf{w}_j - \frac{\sum_{i=1}^{n} u_{ij}^m\mathbf{w}_j^t\mathbf{x}_i}{\sum_{i=1}^{n} u_{ij}^m})$, after some rearrangements, the above equation is equal to the following

$$\sum_{i=1}^{n} u_{ij}^m\mathbf{x}_i(\mathbf{x}_i^t - \frac{\sum_{i=1}^{n} u_{ij}^m\mathbf{x}_i^t}{\sum_{i=1}^{n} u_{ij}^m})\mathbf{w}_j - \frac{\lambda_j}{\gamma}\mathbf{w}_j = 0 \tag{42}$$

from where it can be found that $\mathbf{w}_j$ is the eigenvector corresponding to eigenvalue $\frac{\lambda_j}{\gamma}$ of matrix $\mathbf{M}_j$

$$\mathbf{M}_j = \sum_{i=1}^{n} u_{ij}^m\mathbf{x}_i(\mathbf{x}_i^t - \frac{\sum_{i=1}^{n} u_{ij}^m\mathbf{x}_i^t}{\sum_{i=1}^{n} u_{ij}^m}) \tag{43}$$

In fact, under condition that $\mathbf{x}_i, \mathbf{g}_i, v_i$ are given, as other terms in L' are fixed, hence minimizing L' is equivalent to minimizing the following function J', which equals to the sum of eigenvalues

$$\begin{aligned} J^{`} &= \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^m\frac{(\mathbf{w}_j^t \cdot \mathbf{x}_i - v_j)^2}{||\mathbf{w}_j||^2} \\ &= \sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}^m\bigg(\mathbf{w}_j^t(\mathbf{w}_j^t \cdot \mathbf{x}_i) \cdot \mathbf{x}_i - 2\frac{\mathbf{w}_j^t \cdot \mathbf{x}_i\mathbf{w}_j^t\mathbf{X}\mathbf{u}_j^m}{\mathbf{e}^t\mathbf{u}_j^m} + \frac{\mathbf{w}_j^t\mathbf{X}\mathbf{u}_j^m\mathbf{w}_j^t\mathbf{X}\mathbf{u}_j^m}{\mathbf{e}^t\mathbf{u}_j^m\mathbf{e}^t\mathbf{u}_j^m}\bigg) \\ &= \sum_{j=1}^{c}\mathbf{w}_j^t\sum_{i=1}^{n} u_{ij}^m\bigg((\mathbf{w}_j^t \cdot \mathbf{x}_i - v_j)\mathbf{x}_i\bigg) \\ &= \sum_{j=1}^{c}\mathbf{w}_j^t\frac{\lambda_j}{\gamma}\mathbf{w}_j \\ &= \sum_{j=1}^{c}\frac{\lambda_j}{\gamma} \end{aligned} \tag{44}$$

Hence, the c eigenvectors corresponding to the smallest c eigenvalues of matrix $\sum_{i=1}^{n} u_{ij}^{m}\mathbf{x}_i(\mathbf{x}_i^t - \frac{\sum_{i=1}^{n} u_{ij}^{m}\mathbf{x}_i^t}{\sum_{i=1}^{n} u_{ij}^{m}})$ are the optimal $\mathbf{w}_j^*$ that minimize the objective function $J^{'}$. As objective function in J_{FMP} will reach its minimum at the same optimal $\mathbf{w}_j^*$, hence $\mathbf{w}_j^*$ also minimizes objective function J_{FMP}.

Similarly, by taking the first derivative of L' with respective to $\mathbf{g}_j$,

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{g}_j}(L') &= \frac{\partial}{\partial \mathbf{g}_j}\Bigg(\sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^{m}\Bigg(\gamma \cdot \frac{(\mathbf{w}_j^t \cdot \mathbf{x}_i - v_j)^2}{||\mathbf{w}_j||^2} + (1-\gamma)\cdot||\mathbf{x}_i - \mathbf{g}_j||_2^2\Bigg)\Bigg) - 0 - 0 \\
&= \frac{\partial}{\partial \mathbf{g}_j}\Bigg(\sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^{m}(1-\gamma)\cdot||\mathbf{x}_i - \mathbf{g}_j||_2^2\Bigg) \\
&= -2(1-\gamma)\sum_{i=1}^{n} u_{ij}^{m}(\mathbf{x}_i - \mathbf{g}_j) \qquad (45)
\end{aligned}$$

let Eq. (45) equal to zero,

$$\sum_{i=1}^{n} u_{ij}^{m}\mathbf{g}_j^* = \sum_{i=1}^{n} u_{ij}^{m}\mathbf{x}_i \qquad (46)$$

from which the update of $\mathbf{g}_j$ can be obtained

$$\mathbf{g}_j^* = \frac{\sum_{i=1}^{n} u_{ij}^{m}\mathbf{x}_i}{\sum_{i=1}^{n} u_{ij}^{m}} \qquad (47)$$

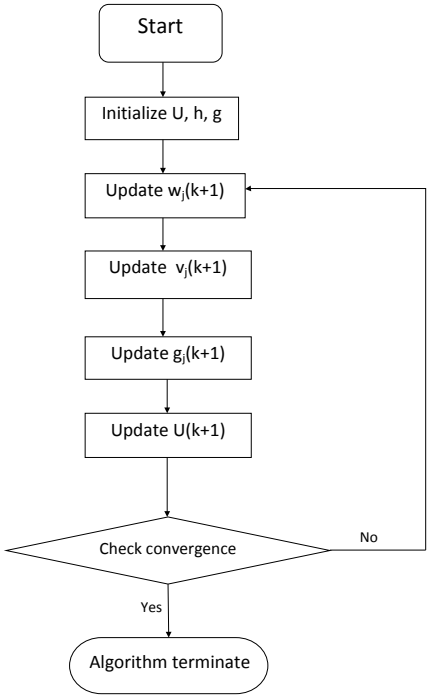

Figure 1: Flowchart of the FMP algorithm.

3.3. The FMP algorithm

Algorithm 1 The FMP algorithm.

Require:

The fuzziness parameter, m; the cluster number, c; the penalize parameter, γ

Set iteration count k=0;

Set ε to be a positive small number;

Ensure:

The resulting partition matrix, $\mathbf{U}^*$; the hyperclusters, $\mathbf{h}_j^*, j = 1, ..., c$; the mass centers, $\mathbf{g}_j^*, j = 1, ..., c$;

1: Initialize partition matrix $\mathbf{U}(k)$, hyperclusters $\mathbf{h}_j(k), j = 1, ..., c$ and mass centers $\mathbf{g}_j(k)$;

2: Update $\mathbf{w}_j(k+1)$ through eigenvalue decomposition of matrix $\mathbf{M}_j$ expressed in Eq. (43) by selecting the eigenvector corresponding to the smallest eigenvalue

3: Update $v_j(k+1)$ according to Eq. (37) under the current partition matrix $\mathbf{U}(k)$ and normal vector $\mathbf{w}_j(k)$, then update $\mathbf{h}_j(k+1)$;

4: Update $\mathbf{g}_j(k+1)$ according to Eq. (47) based on the current partition matrix $\mathbf{U}(k)$ and all of the data samples;

5: Based on the newly updated fuzzy hyperclusters $\mathbf{h}_j(k+1)$ and mass center $\mathbf{g}_j(k+1)$, update the fuzzy partition matrix $\mathbf{U}$(k+1) according to Eq. (31), where the distances are calculated according to Eq. (20) and Eq. (21);

6: Check if the algorithm converges, then terminate the iteration. Go to 2 otherwise.;

7: **return** $\mathbf{U}^*$,$\mathbf{h}_j^*, j = 1, ..., c$ and $\mathbf{g}_j^*, j = 1, ..., c$;

The FMP shares a similar computational procedure with that of the FCM, which iteratively updates between $\mathbf{U}$, $\mathbf{h}$ and $\mathbf{g}$ until the criterion of convergence is satisfied. Figure 1 shows the flowchart of the proposed FMP, from which the update steps of the FMP can be summarized as Algorithm 1.

The FMP is considered to be converged if the maximum change in the partition matrix between two successive iterations is less than a preset positive small number ε. Thus, the resulting partition matrix $\mathbf{U}^*$, hyperclusters $\mathbf{h}_j^*, j = 1, ..., c$ and mass centers $\mathbf{g}_j^*, j = 1, ..., c$ are considered to satisfy the solution which minimizes the objective function J_{FMP}.

4. Experiments

To validate the proposed FMP clustering, experiments were conducted on the yeast gene expression data set and the Mixed-Lineage Leukemia (MLL) data set. The results obtained from the FMP were compared with those obtained from some related methods including the FCM and the kernel FCM with Gaussian kernel.

The computer on which the experiments were conducted is of 2 GB memories, Intel dual core processor with each at 1.86 GHz. The software platform is consisted of Microsoft Windows XP operating system version 2002 service pack 3, and Matlab version 2009. The FMP algorithm was implemented in Matlab, other existing algorithms were implemented in Matlab for comparison purpose including the FCM and the KFCM, where the FCM algorithm was based on the functions provided by Matlab, the KFCM algorithm were downloaded from the Matlab website and modified according to the data analysis problems.

4.1. Yeast data

Budding yeast expression data set of *Saccharomyces cerevisiae* contains 6400 distinct DNA sequences measured in 7 time points (0, 9.5, 11.5, 13.5, 15.5, 18.5, and 20.5h) [27]. The microarrays were printed on glass slides and samples were harvested every 2 hours after 9 hours growth. The relative expression ratios were then log-transformed and more than 43,000 expression-ratio were measured. By clipping and data preprocessing, samples with missing values and samples with low variance in the observations were filtered out. The resulting data set rendered for analysis consisting of 614 genes.

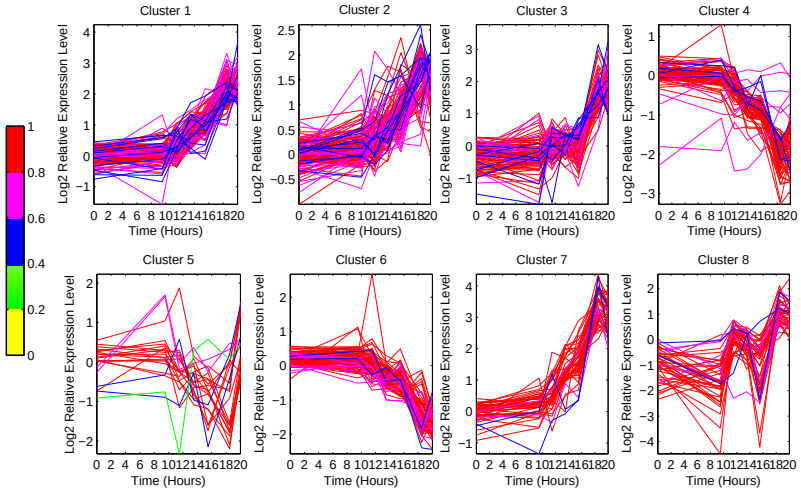

Figure 2: Gene expression trajectory of FCM clustering on yeast data set

Table 2: Maximum change of membership of different clustering algorithms on yeast data set

Iteration \ Algorithms	KFCM	FCM	FMP
20	0.0949	0.1354	**0.2023**
40	0.0312	0.0677	**0.2594**
60	0.0770	0.1203	**0.0495**
80	0.0162	0.2730	**0.1027**
100	0.0785	0.0004	**0.0194**
120	0.0047	0.0000	**0.0000**
140	0.0000	0.0000	**0.0000**

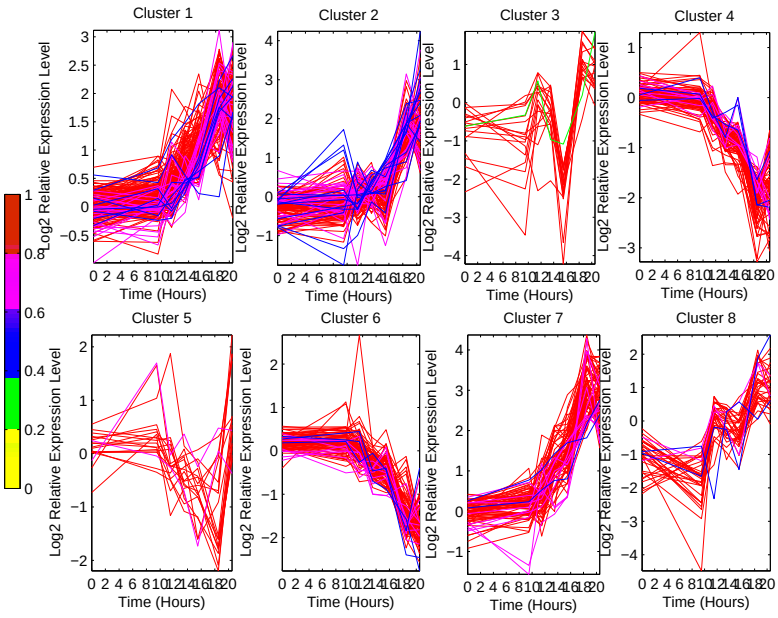

Figure 3: Gene expression trajectory of KFCM clustering with Gaussian kernel on yeast data set

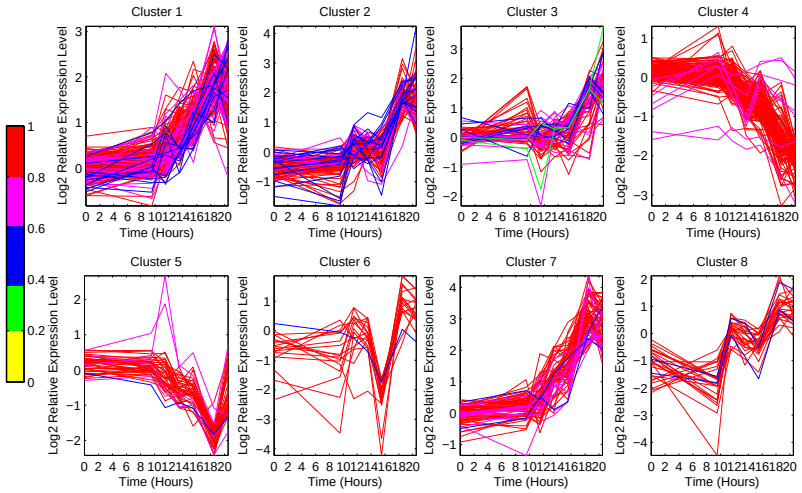

Figure 4: Gene expression trajectory of FMP clustering on yeast data set

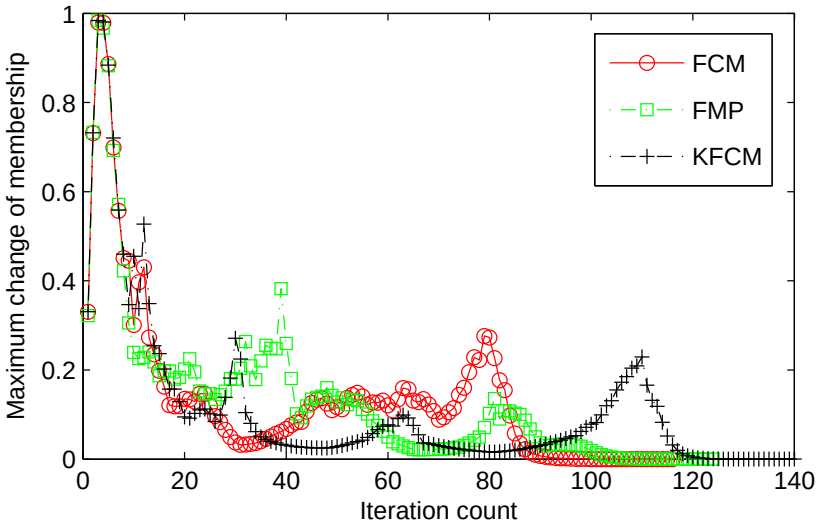

Figure 5: Rate of convergence of different clustering algorithms on Yeast data set

In these validations, based on cluster validity and parameter estimation conducted before [28], the fuzziness parameters in all the clustering techniques were

chosen as 1.2, the weighting parameter γ was set to 0.8 and the cluster number was chosen as 8. The Gaussian kernel was chosen as the kernel function in the KFCM. Figures 2-4 show the corresponding clustering results on a yeast gene profile by using the FCM, KFCM and FMP. Genes were partitioned to groups in which they produced the largest membership values. The expression trajectories in each group are plotted in different colors, corresponding to the different values of the membership, where the color was red if the membership of the gene to the cluster was in the range of $(0.8, 1]$, purple if the membership was in the range of $(0.6, 0.8]$, blue if the membership was in the range of $(0.4, 0.6]$, green if the membership was in the range of $(0.2, 0.4]$ and yellow if the membership was in the range of $(0, 0.2]$.

From Figures 2-4, it can be seen that genes with highly similar expression patterns were grouped into the same clusters by all clustering techniques. But the resulting clusters produced by different clustering techniques were not exactly same with each other: some genes were assigned to different clusters by different methods. In Figures 2 and 4, it can be seen that the majority of genes were partitioned into clusters with high membership values, where most membership values were higher than 0.6, and only a few of them were categorized with low membership values in the range of $(0.2, 0.4]$. Furthermore, the FMP produced different clustering results from those obtained by the FCM and KFCM. The FMP reflected the genes expression patterns in different ways and provided different interpretations for gene functionality.

The rates of convergence of these clustering algorithms on the Yeast data were also studied. The results have been presented in Figure 5 and Table 2. From Figure 5, it can be seen that the maximum change of the membership of all of the three algorithms decreased drastically during the first 20 iterations. The values went down with fluctuations as the algorithms carry on. The trend of maximum change in membership can also be observed from Table 2. From the Table, we can see that the FCM reached convergence after about 100 iterations, the FMP reached convergence after about120 iterations and the KFCM reached convergence after about 140 iterations which is the slowest.

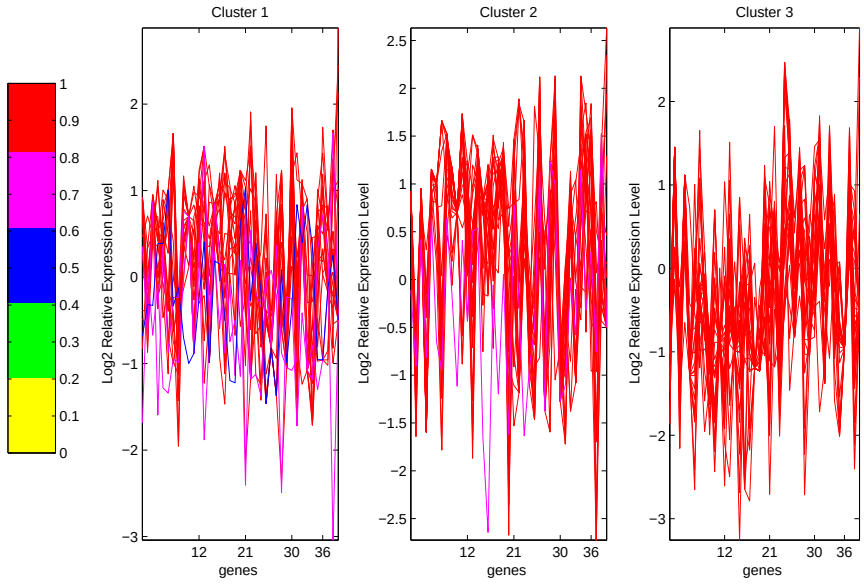

Figure 6: Gene expression trajectory of FCM clustering on Leukemia data set

4.2. Mixed-Lineage Leukemia data

Experiments were also conducted on the MLL data set. The MLL data set consists of 72 samples with 12582 gene expressions in three types of leukemia: 24 acute lymphoblastic leukemia (ALL), 20 mixed-lineage leukemia (MLL) and 28 acute myelogenous leukemia (AML) [29]. The original data set is of high dimensions. For the convenience of data analysis, feature selections and dimension reduction were adopted to remove redundant and irrelevant features before validations were carried out. After these processing steps, the final data set rendered for analysis consisting of 72 samples with 39 genes.

The clustering results were presented in Figure 6, Figure 7 and Figure 8, in which the trajectory clustering using FCM, KFCM and FMP were shown respectively. The expression pattern of each gene was plotted in different colors corresponding to the different values of memberships of the resulting clusters, which was the same with that in the validation on yeast gene data set. From these Figures, it can be observed that most of the trajectories were plotted in

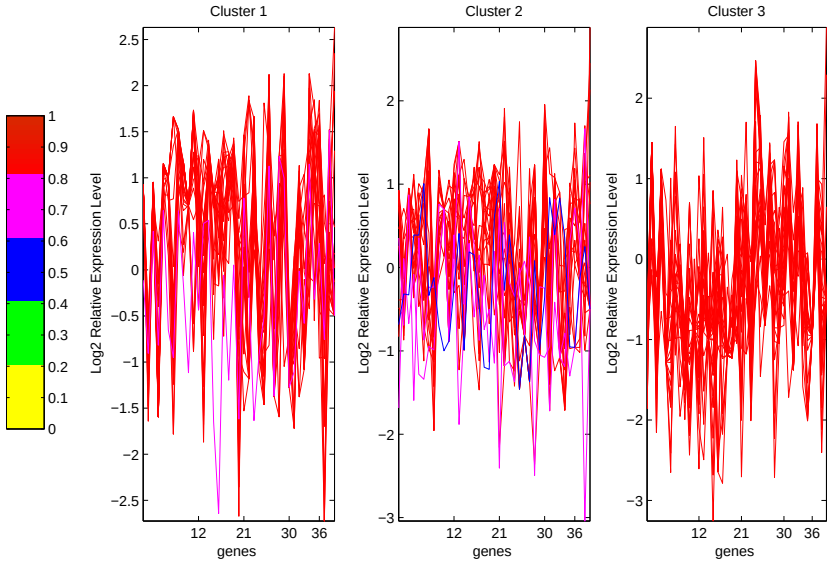

Figure 7: Gene expression trajectory of KFCM clustering with Gaussian kernel on Leukemia data set

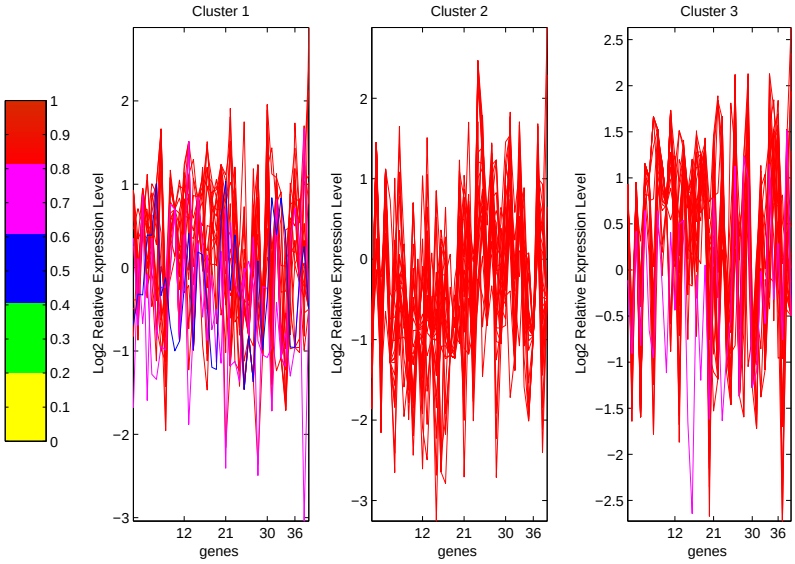

Figure 8: Gene expression trajectory of FMP clustering on Leukemia data set

Table 3: Maximum change of membership of different clustering algorithms on MLL data set

Iteration \ Algorithms	KFCM	FCM	FMP
1	0.63643	0.54446	**0.45763**
2	0.43713	0.46082	**0.26999**
3	0.44732	0.52015	**0.41393**
4	0.71430	0.30488	**0.47539**
5	0.22010	0.08204	**0.10172**
6	0.07556	0.01454	**0.01017**
7	0.02331	0.00263	**0.00165**
8	0.00739	0.00045	**0.00040**
9	0.00238	0.00013	**0.00013**

Table 4: Random index of different clustering algorithms on MLL data set

Cluster numbers \ Algorithms	FCM	KFCM	FMP
2	0.94718	0.76213	**0.76213**
3	0.90141	0.94718	**0.94718**
4	0.80869	0.89789	**0.94092**
5	0.81612	0.92175	**0.93623**
6	0.83059	0.85603	**0.81455**
7	0.80869	0.82394	**0.82394**
8	0.81064	0.90493	**0.82433**
9	0.81025	0.81455	**0.81455**
10	0.81808	0.82121	**0.83412**
11	0.81495	0.82786	**0.82786**
12	0.80634	0.83059	**0.83059**
13	0.81182	0.81338	**0.82042**
14	0.81182	0.83059	**0.81534**

Table 5: Adjusted random index of different clustering algorithms on MLL data set

Cluster numbers \ Algorithms	FCM	KFCM	FMP
2	0.52652	0.52652	**0.88034**
3	0.88034	0.88034	**0.88286**
4	0.86286	0.75665	**0.76347**
5	0.85251	0.81717	**0.50386**
6	0.52155	0.64486	**0.52297**
7	0.54918	0.54918	**0.56841**
8	0.55577	0.77738	**0.50386**
9	0.52155	0.52155	**0.50978**
10	0.58455	0.53950	**0.50860**
11	0.56599	0.56599	**0.52886**
12	0.57446	0.57446	**0.51976**
13	0.53622	0.51403	**0.49638**
14	0.52357	0.56870	**0.51130**

Table 6: Mirkin index of different clustering algorithms on Leukemia data set

Cluster numbers \ Algorithms	FCM	KFCM	FMP
2	0.24282	0.23787	**0.23787**
3	0.05908	0.05282	**0.05282**
4	0.09859	0.10211	**0.05908**
5	0.19131	0.07825	**0.06377**
6	0.18388	0.14397	**0.18545**
7	0.16941	0.17606	**0.17606**
8	0.19131	0.09507	**0.17567**
9	0.18936	0.18545	**0.18545**
10	0.18975	0.17879	**0.16588**
11	0.18192	0.17214	**0.17214**
12	0.18505	0.16941	**0.16941**
13	0.19366	0.18662	**0.17958**
14	0.18818	0.16941	**0.18466**

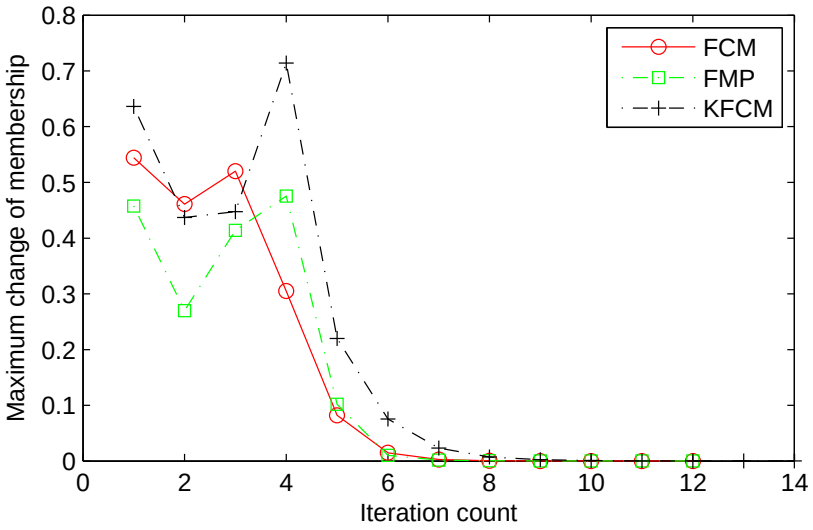

Figure 9: Rate of convergence of different clustering algorithms on MLL-Leukemia data set

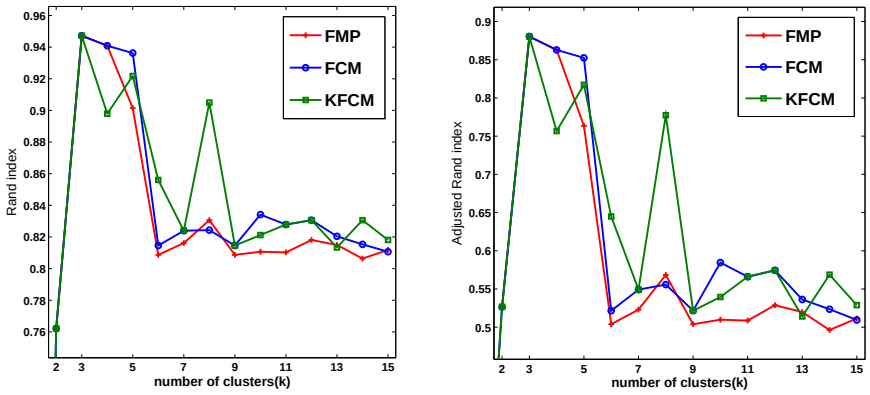

Figure 10: Cluster quality of FCM, KFCM and FMP on MLL-Leukemia data set with different cluster numbers measured by Rand index (left) and adjusted Rand index(right)

red color, which indicates that the expression patterns were clustered with high confidence.

The convergence of the proposed method on MLL-Leukemia data set was

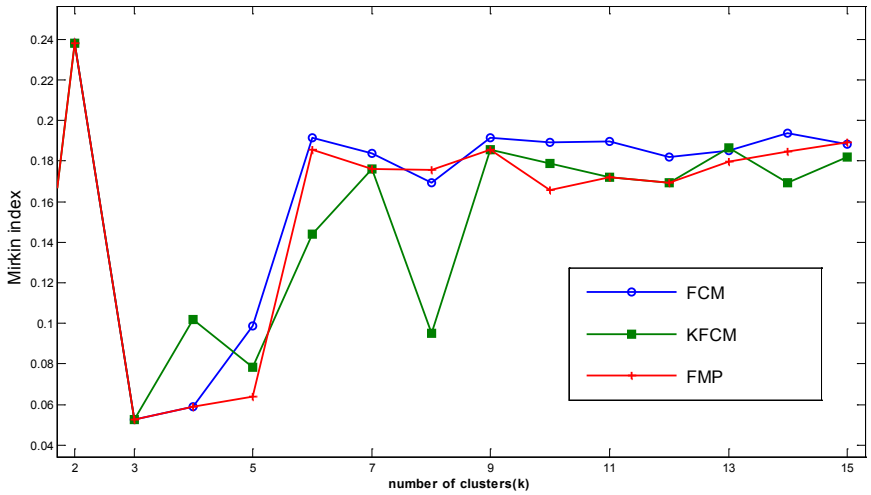

Figure 11: Cluster quality of FCM, KFCM and FMP on MLL-Leukemia data set with different cluster numbers measured by Mirkin index

also validated and compared with that of the FCM and the KFCM. The results were shown in Figure 9 and in Table 3. From the Figure, it can be seen that the values of maximum change in membership of these three methods reduce at first and then climbs to a local high after couples of iterations. After that, the values in these three methods decline sharply, the FMP and the FCM reached convergence first followed by the KFCM. Similar trend can be observed from Table 3, the FCM and the FMP reached convergence at similar pace, followed by the KFCM.

The quality of the clustering results of the proposed FMP was also measured in different indices, which includes Rand index [30], adjusted Rand index [31] and Mirkin index [32]. The Rand index is a measure of similarity between two data clusterings which is basically the fraction of agreement, a high Rand index value indicates a better clustering. The value is calculated according to

$$R(S,V) = \frac{2(a+d)}{(a+b+c+d)} \tag{48}$$

where S is the standard labels of the samples produced by expert, V the label generated by clustering algorithm, a the number of gene pairs belonging to the same clusters in S, and V, b the number of gene pairs belonging to the same cluster in S but different clusters in V, c the number of gene pairs belonging to the same cluster in V but different clusters in S, and d the number of gene pairs belonging to different clusters in both S and V. The Rand index is a value between 0 and 1, where a higher Rand index indicates a better clustering. When the two partitions agree perfectly, the Rand index is 1.

In the experiments, the adjusted Rand index was also adopted to validate the clustering accuracy. The adjusted Rand index is an improved version of the Rand index which assesses the degree of agreement between two partitions of the same set of objects. The value is calculated according to

$$AR(S,V) = \frac{2(ad-bc)}{(a+b)(b+d)+(a+c)(c+d)} \tag{49}$$

where the S,V,a,b,c,d share the same meaning with that in Rand index.

Figure 10, Table 4 and Table 5 show the quality of resulting clusters of FCM, KFCM and FMP with different cluster numbers measured by Rand index and adjusted Rand index, respectively. From the Figure, it can be observed that for all of the three methods, the value of Rand index and adjusted Rand index climb to a peak when the cluster number is set to 3, which corresponds to the Table 4 and Table 5. The value then endures a sharp plunge when the number of clusters is set over 5. The Figure shows that the clustering results are consistent with the fact that there are three different types of leukemia, which demonstrates the effectiveness of the proposed method.

The clustering results were also evaluated by Mirkin metric. The Mirkin metric is defined as

$$M(S,V) = 2N_{disagree}(S,V) \tag{50}$$

where $N_{disagree}$ is defined as the number of point pairs which are in the same cluster under S but in different clusters under V or viceversa. The lower Mirkin value, the better clustering results.

From the Figure 11 and the Table 6, it can be seen that the values of the Mirkin index plunge from 0.24 to about 0.05 with the cluster numbers increase from 1 to 3, which is the minimum for both FCM and FMP. After that, the values of Mirkin index climb generally with the increasing of the cluster numbers, which means that the clustering results become worse. For KFCM, the value of Mirkin index reaches its minimum when cluster number is 5, and then the value fluctuates up with the increase of the cluster numbers. From these Figures and results, it can be seen that the proposed clustering method is of great effectiveness.

5. Conclusion and Future Work

We have presented a fuzzy mix-prototype clustering algorithm in this paper. By combining hyperplane-based data analysis and fuzzy c-means clustering algorithm, we formulated the objective function of the FMP. Minimizing the objective function under given constraints was then considered into an optimization problem. By using Lagrangian multiplier method, the necessary conditions for minimizing the objective function were obtained. Based on these necessary conditions, an iterative numerical solution was then formulated. The FMP algorithm was applied to perform microarray data analysis on yeast data set and MLL leukemia data set. The experimental results were compared with those obtained from the FCM and the KFCM, in which the effectiveness of the proposed FMP was demonstrated.

Despite that the FMP has been successful developed and applied, how to compare the performance of the FMP with some of the latest fuzzy clustering techniques, effects of different parameters and the complexity analysis of the FMP are still worth for further investigation, we herein list them as directions for future research.

Acknowledgment

The paper is supported by the Fundamental Research Funds for the Central Universities (Grant No. 2012QNB17), National Natural Science Foundation of China (Grant No. 61303182), the Natural Science Foundation of Jiangsu Province (Grant No. BK20130210), the Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20120095120026), the Postdoctoral Science Foundation of China (Grant No. 2012M521144), the Postdoctoral Science Foundation of Jiangshu Province (Grant No. 1301120C).

References

[1] J. C. Bezdek, C. Coray, R. Gunderson, J. Watson, Detection and characterization of cluster substructure i. linear structure: fuzzy c-lines, SIAM Journal on Applied Mathematics 40 (2) (1981) 339–357.

[2] P. Maji, Fuzzy-rough supervised attribute clustering algorithm and classification of microarray data, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 41 (1) (2011) 222–233.

[3] U. Maulik, Analysis of gene microarray data in a soft computing framework, Applied Soft Computing 11 (6) (2011) 4152 – 4160.

[4] P. Maji, P. S., Rough-fuzzy clustering for grouping functionally similar genes from microarray data, IEEE/ACM Trans Comput Biol Bioinform 10 (2) (2013) 286–99.

[5] M. Kumar, S. K. Rath, Classification of microarray using mapreduce based proximal support vector machine classifier, Knowledge-Based Systems 89 (2015) 584–602.

[6] K. Dhyani, L. Liberti, Mathematical programming formulations for the bottleneck hyperplane clustering problem, in: modelling, computation and optimization in information systems and management sciences, Vol. 14, Springer, Berlin, 2008, pp. 87–96.

[7] Y. H. Shao, C. H. Zhang, X. B. Wang, N. Y. Deng, Improvements on twin support vector machines, IEEE Transactions on Neural Networks 22 (6) (2011) 962–968.

[8] A. Mukhopadhyay, U. Maulik, Towards improving fuzzy clustering using support vector machine: Application to gene expression data, Pattern Recognition 42 (11) (2009) 2744 – 2763.

[9] H. Zhu, X. Liu, R. Lu, H. Li, Efficient and privacy-preserving online medical prediagnosis framework using nonlinear svm, IEEE Journal of Biomedical and Health Informatics 21 (3) (2017) 838–850.

[10] H. J. Dai, P. T. Lai, R. T. H. Tsai, Multistage gene normalization and svm-based ranking for protein interactor extraction in full-text articles, IEEE/ACM Transactions on Computational Biology and Bioinformatics 7 (3) (2010) 412–420.

[11] Z. Qi, Y. Tian, Y. Shi, Robust twin support vector machine for pattern classification, Pattern Recognition 46 (1) (2013) 305 – 316.

[12] X. Yang, S. Chen, B. Chen, Z. Pan, Proximal support vector machine using local information, Neurocomputing 73 (1-3) (2009) 357–365.

[13] S. Ghorai, A. Mukherjee, P. K. Dutta, Nonparallel plane proximal classifier, Signal Processing 89 (4) (2009) 510–522.

[14] P. S. Bradley, O. L. Mangasarian, k-plane clustering, J. Global Optimization 16 (1) (2000) 23–32.

[15] S. Javanmardi, M. Shojafar, S. Shariatmadari, S. S. Ahrabi, Fr trust: a fuzzy reputation-based model for trust management in semantic p2p grids, International Journal of Grid and Utility Computing 6 (1) (2015) 57–66.

[16] S. S. K. Mohammad Javadian, Saeed Bagheri Shouraki, A novel density-based fuzzy clustering algorithm for low dimensional feature space, Fuzzy Sets and Systems 318 (2017) 34–55.

[17] A. Jahangirzadeh, S. Shamshirband, S. Aghabozorgi, S. Akib, H. Basser, N. B. Anuar, M. L. M. Kiah, A cooperative expert based support vector regression (co-esvr) system to determine collar dimensions around bridge pier, Neurocomputing 140 (2014) 172 – 184.

[18] S. Sajjadi, S. Shamshirband, M. Alizamir, P. Yee, Z. Mansor, A. Manaf, T. Altameem, A. Mostafaeipour, Extreme learning machine for prediction of heat load in district heating systems, Energy and Buildings 122 (2016) 222–227.

[19] S. Sadi-Nezhad, K. Khalili-Damghani, A. Norouzi, A new fuzzy clustering algorithm based on multi-objective mathematical programming, TOP 23 (1) (2015) 168–197.

[20] A. Pathak, N. R. Pal, Clustering of mixed data by integrating fuzzy, probabilistic, and collaborative clustering framework, International Journal of Fuzzy Systems 18 (3) (2016) 339–348.

[21] D. Graves, W. Pedrycz, Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study, Fuzzy Sets and Systems 161 (1) (2010) 522–543.

[22] Y.-I. Kim, D.-W. Kim, D. Lee, K. H. Lee, A cluster validation index for GK cluster analysis based on relative degree of sharing, Inf. Sci. Inf. Comput. Sci. 168 (1-4) (2004) 225–242.

[23] C. J. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (2) (1998) 121–167.

[24] G. M. Fung, O. L. Mangasarian, Multicategory proximal support vector machine classifiers, Mach. Learn. 59 (1-2) (2005) 77–97.

[25] O. L. Mangasarian, E. W. Wild, Multisurface proximal support vector machine classification via generalized eigenvalues, IEEE Trans. Pattern Anal. Mach. Intell. 28 (1) (2006) 69–74.

[26] H. N. Jan R. Magnus, Matrix Differential Calculus with Applications in Statistics and Econometrics, Wiley, 1999.

[27] J. L. DeRisi, V. R. Iyer, P. O. Brown, Exploring the metabolic and genetic control of gene expression on a genomic scale, Science 278 (5338) (1997) 680–686.

[28] J. Liu, T. D. Pham, A spatially constrained fuzzy hyper-prototype clustering algorithm, Pattern Recognition 45 (4) (2012) 1759–1771.

[29] S. A. Armstrong, J. E. Staunton, L. B. Silverman, R. Pieters, M. L. den Boer, M. D. Minden, S. E. Sallan, E. S. Lander, T. R. Golub, S. J. Korsmeyer, Mll translocations specify a distinct gene expression profile that distinguishes a unique leukemia, Nature 30 (1) (2002) 41–7.

[30] W. M. Rand, Objective criteria for the evaluation of clustering methods, Journal of the American Statistical Association 66 (336) (1971) 846–850.

[31] K. Y. Yeung, W. L. Ruzzo, Principal component analysis for clustering gene expression data, Bioinformatics 17 (9) (2001) 763–774.

[32] M. Meila, Comparing clusterings: an axiomatic view, in: L. D. Raedt, S. Wrobel (Eds.), Proceedings of the 22nd International Conference on Machine Learning (ICML-05), 2005, pp. 577–584.