CANCER MOLECULAR PATTERN DISCOVERY BY SUBSPACE CONSENSUS KERNEL CLASSIFICATION

Xiaoxu Han

Department of Mathematics and Bioinformatics Program, Eastern Michigan University
Ypsilanti, MI 48197, USA
xiaoxu.han@emich.edu

Cancer molecular pattern efficient discovery is essential in the molecular diagnostics. The characteristics of the gene/protein expression data are challenging traditional unsupervised classification algorithms. In this work, we describe a subspace consensus kernel clustering algorithm based on the projected gradient nonnegative matrix factorization (PG-NMF). The algorithm is a consensus kernel hierarchical clustering (CKHC) method in the subspace generated by the PG-NMF. It integrates convergence-soundness parts-based learning, subspace and kernel space clustering in the microarray and proteomics data classification. We first integrated subspace methods and kernel methods by following our framework of the input space, subspace and kernel space clustering. We demonstrate more effective classification results from our algorithm by comparison with those of the classic NMF, sparse-NMF classifications and supervised classifications (KNN and SVM) for the four benchmark cancer datasets. Our algorithm can generate a family of classification algorithms in machine learning by selecting different transforms to generate subspaces and different kernel clustering algorithms to cluster data.

1. INTRODUCTION

With the development of genomics and proteomics, Molecular diagnostics has appeared as a new tool to diagnose cancers. It picks a patient's tissues or blood samples and uses DNA microarray or mass spectrometry (MS) based proteomics techniques to generate their gene expressions or protein expressions. The gene/protein expressions reflect gene/protein activity patterns in different types of cancerous or precancerous cells. They are molecular patterns or molecular signatures of cancers. Different cancers will have different molecular patterns and the molecular patterns of a normal cell will be different from those of a cancer cell. Clinicians identify the potential biomarkers by analyzing the gene/protein patterns. However, robustly classifying cancer molecular patterns is still a challenge for clinicians and bioiformaticans.

Many classification methods from statistical and machine learning are proposed for cancer molecular pattern classification. These methods can be generally classified as supervised classification methods, such as k-nearest neighborhood (kNN), linear discriminant anayalsis (LDA), neural networks (NN), support vector machines (SVM);[1-3] unsupervised classification (clustering) methods, such as hierarchical clustering (HC), self-organizing maps (SOM), principal component analysis (PCA); and their variants, such as particle swarm optimization support vector machines (PSO-SVM), kernel principal component analysis (KPCA) etc. [4-7] We are particularly interested in the unsupervised molecular pattern discovery algorithms, because they do not need or have prior knowledge about data. They also have potentials to explore the latent structure of data. However, the traditional clustering algorithms: HC and SOM were already proved unstable for gene and protein expression data although they are widely used in the cancer molecular pattern discovery community. [4,8,15]

Actually, the characteristics of gene and protein expression data are challenging the traditional unsupervised classification algorithms. These high dimensional data can be represented by an $n \times m$ matrix after preprocessing. The row data in the matrix are the expression levels of a gene across different experiments or intensity values of a measured data point in different samples (observations) corresponding to an m/z ratio. The column data are the gene expression levels of a genome under an experiment or intensity values of all measured data points in a sample corresponding to m/z ratios. Usually, $n >> m$; that is, the number of variables

in a dataset is much greater than the number of observations/experiments. For the gene expression data, the column number in the matrix is <100 and the row number > 5000 usually; for the proteomics data, the matrix column number is < 200 and the matrix row number is in the order of $10^5 \sim 10^6$ generally. These data are not noise free data because their raw data have noise and preprocessing algorithms can't remove them completely. Although there are a large number of variables in these data, only a small set of variables account for most of data variations.

1.1. Nonnegative matrix factorization

It is obvious that dimension reduction / feature selection should be conducted to reduce data to a much lower dimension before classification. Several well-known global feature selection methods, such as principal component analysis (PCA), singular value decomposition (SVD), and independent component analysis (ICA) have been applied in the cancer molecular pattern classifications. [9,10,11,12] However, the holistic feature selection mechanism from these methods prevents from the alternative local feature selection. For example, PCA can only capture the global characteristics of data and each principal component (PC) contains information from all input variables. This leads to the hard time to interpret PCs intuitively. Data representation in PCA is not "purely additive". Each PC has both positive and negative entries, which are likely to cancel each other partly in the feature selection.

On the other hand, there is a local feature selection algorithm: nonnegative matrix factorization (NMF) with parts-based learning mechanism.[13] In contrast to the global feature selection algorithms, NMF can capture variables contributing to local characteristics of data with obvious interpretations. It makes the global characteristics as the simple "addition/combinations" of the local characteristics. In fact, data representation in NMF is purely additive is because of the nonnegative constraints in the NMF.

Given an nonnegative matrix $X \in R^{n\times m}$ and a rank $r < \min(n, m)$, NMF is an nonlinear programming problem to find two optimal nonnegative matrices $W \in R^{n\times r}$ and $H \in R^{r\times m}$ that minimize the reconstruction error, which can be measured by a distance metric, between the matrices X and WH : $E(W,H) = \|X - WH\|$; that is, $X \sim WH$. We name W as a basis matrix and H as a feature matrix. The columns of W (a set of bases) set up a new coordinate system and all elements of H are the coordinates of X in this new coordinate system. The feature matrix H is the prototype dataset of X after the feature selection, where each column is the prototype of an observation. After NMF, each column (observation) of X can be represented as a linear combination of r bases W_i, $i = 1,2,\cdots r$ approximately,

$$X_j \approx \sum_{i=1}^{r} h_{ij} W_i = h_{1j} W_1 + h_{2j} W_2 + \cdots h_{ir} W_r \qquad (1)$$

That is, each observation is expressed as the product of the basis matrix and its corresponding prototype after feature selection.

The objective function $E(W,H) = \|X - WH\|$ can be expressed as Euclidean distance or Kullback-Leibler (K-L) divergence between X and WH . For example, the Euclidean distance objective function is defined as follows.

$$\|X - WH\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{m} (X_{ij} - (WH)_{ij})^2 \qquad (2)$$

Lee and Seung gave a multiplicative update algorithm for NMF by conducting a dynamic step based gradient descent learning with respect to W and H .[13] The iteration schemes for the Euclidean distance objective function are as follows (The iteration schemes for the K-L divergence are similar). In the iteration, W and H are initialized randomly.

$$W_{il}^{(k+1)} = W_{il}^{(k)} \frac{(XH^t)_{il}^{(k)}}{(WHH^t)_{il}^{(k)}} \qquad (3)$$

$$H_{uj}^{(k+1)} = H_{uj}^{(k)} \frac{((W^{(k+1)})^t X)_{uj}}{((W^{(k+1)})^t W^{(k+1)} H^{(k)})_{uj}} \qquad (4)$$

The multiplicative update algorithm works well experimentally. However, there is no guarantee that it can converge to local minimum points of the objective function, because the limit of the non-increasing sequence $\{W^{(k)}, H^{(k)}\}$ generated from the multiplicative update algorithm may not be a stationary point; [14] that is, it lacks "convergence-soundness".

Brunet *et al*. used NMF to classify cancer molecular patterns by conducting NMF based clustering for gene expression data.[15] Their NMF clustering consists of three steps. First, decompose gene expression data X

under a rank r by the multiplicative update algorithm, i.e. each observation is represented as the linear combination of bases by Eq. (1), where h_{ij} is the i-th element of the H_j, which is the prototype of the j-th observation X_j after feature selection. Second, clustering is conducted by the following query asked by each sample: 'which basis has the largest expression level in my prototype? I will belong to the cluster associative with that basis'. For example, suppose h_{ij} is the largest value in H_j, then sample X_j will be assigned to the cluster i because the i^{th} basis has the largest expression level in its prototype H_j. The number of clusters is just the decomposition rank r. Finally, the rank leading to the most meaningful clustering is decided by a Monte Carlo based model selection mechanism by finding a rank with the maximum cophenetic correlation coefficient in the hierarchical clustering. The cophenetic correlation coefficient is a measure to evaluate the stability of hierarchical clustering. It is the correlation between the pairwise distance and linkage distance in the hierarchical clustering. A large cophenetic correlation coefficient value will indicate the high stability of a hierarchical clustering.

Brunet *et al* proved this method was superior to HC and SOM methods for three benchmark cancer datasets.[15] Inspired by this work, Gao and Church developed a sparse nonnegative matrix factorization to cluster the cancer samples by adding sparseness control in the basic NMF formulation (sparse-NMF).[16, 17] They demonstrated the sparse-NMF based clustering was superior to the basic NMF clustering method for the same datasets.

However, Brunet *et al* 's NMF based clustering method has following weak points. 1. The multiplicative update algorithm in the NMF lacks the convergence soundness. The model selection mechanism in the NMF clustering is expensive, because it requires to compute cophenetic correlation coefficients for the hierarchical clustering conducted at all possible ranks to decide the final optimal decomposition rank.

1.2. Contributions

In this study, we describe a subspace consensus kernel clustering technique based on the projected gradient nonnegative matrix factorization (PG-NMF), which was developed by Lin,[14] to conduct cancer molecular pattern classification for microarray and proteomics data. The projected gradient nonnegative matrix factorization (PG-NMF) has sound convergence and converges faster than the basic NMF.[14] In addition, we present the ideas of input space, subspace and kernel space clustering before elaborating on our PG-NMF based classification method under the framework of subspace and kernel space clustering.

The idea of our method is to transform a gene/protein expression data set $X \in \Re^n$ into a subspace $S \subset \Re^n$ by using the PG-NMF algorithm. Then, a consensus kernel hierarchical clustering algorithm (CKHC) is developed to cluster the projections of a dataset X in the subspace S to infer the latent structure of the data. We have showed that the PG-NMF based subspace kernel clustering (PG-NMF-CKHC) is superior to the basic NMF, sparse-NMF clustering and supervised clustering (KNN and SVM) in the cancer molecular pattern discovery for four benchmark cancer datasets.

This paper is organized as follows. Section 2 presents the concepts of input space, subspace and kernel space clustering before introducing our PG-NMF based consensus kernel hierarchical clustering in the section 3. Section 4 shows the experimental results of our algorithm. Finally, we discuss the possible algorithm generalizations and draw conclusions.

2. INPUT SPACE, SUBSPACE AND KERNEL CLUSTERING

For a given data set $X = (x_1, x_2, \ldots x_n)^T \in \Re^{n \times m}$, clustering is to find an implicit classification function $f : X \to \Gamma$ that maps each data sample x_i, to its target function value y_j (label) in a set Γ according to some dissimilarity metric ($j = 1,2 \cdots |\Gamma|$). Data samples with a same target function value (label) after classification will claim to share a same cluster.

We classify clustering as the input space, subspace and kernel space clustering according to where the implicit classification function f is computed. In the input space clustering, the implicit classification function f is computed in the input space $\Re^{n \times m}$ of the dataset. Hierarchical clustering (HC), K-means clustering and expectation maximization (EM) clustering all belong to the input space clustering. In the kernel space clustering, the classification function f is computed in a kernel space Ω of the input space, which is a high dimensional Hilbert space generated by a feature map function $\Phi : X \to \Omega$, $\dim(\Omega) >> \dim(X)$. That is, the clustering is conducted for the high

dimensional data $\Phi(X)$. On the other hand, in the subspace clustering, the classification function *f* is computed in a subspace S of the input space, generated by a linear or nonlinear transform ϕ, $\dim(S) \le \dim(X)$. Generally, almost all input-space clustering methods can be used in the subspace clustering to cluster the feature data in the subspace. However, not all input space clustering algorithms can have corresponding kernel space clustering algorithms. In the following work, we use the HC as an example to demonstrate the input space, subspace and kernel space clustering.

2.1. Subspace clustering

A subspace S is generated from a linear or nonlinear transform $\phi: X \in \Re^{n\times m} \to X^* \in \Re^{r\times m}$ and clustering is conducted through the transformed data X^*. For example, SOM and PCA based clustering are typical subspace clustering approaches. Most likely, the subspace has the lower dimensionality than the original dataset, i.e. $\dim(S) < \dim(X)$. Each transform ϕ applied to X can be represented as $TX = X^*$, where T is the matrix representation of transform ϕ. Writing it as a matrix decomposition form of X, we have $X = WX^*$, where the matrix W is the inverse or pseudo-inverse of the matrix T. We still call W as a basis matrix and X^* as a feature matrix.

The columns of the basis matrix span the subspace: $S = span(W_1, W_2, \ldots W_r)$. Dependent on the properties of the transform ϕ, the basis matrix may not be unique and the corresponding matrix decomposition may not be unique also. Geometrically, each column of X^* is the coordinates of each observation/column of X in the subspace S, which can be viewed as a new coordinate system.

Self-organizing map clustering can be viewed as a simple subspace clustering, where the target function value of each sample is determinated by the location of its corresponding reference vector of the best matching unit (BMU) on the SOM plane. In the nonlinear transform conducted by a self-organizing map (SOM), the feature matrix X^* is called the prototype data including all reference vectors on the SOM plane. The subspace bases $(W_1, W_2, \ldots W_r)$ can be obtained by solving r least square problems, where r is the number of neurons on the SOM plane.

Actually, the transform ϕ can be implemented by any linear or nonlinear feature selection methods, such as principal component analysis (PCA), independent component analysis (ICA), self-organizing map (SOM) and nonnegative matrix factorization (NMF). The spectral analysis methods like fast Fourier transform, wavelet transform can also implement ϕ. That is, any input space clustering algorithms can be employed to cluster the feature data X^*. For example, clustering the data principal components (PCA clustering) by HC or other input space clustering methods is a typical subspace clustering, where the subspace generated by the PCA transform is an orthogonal space.[18] Similarly are the hierarchical clustering of the independent components of data (ICA clustering) and the FFT coefficients of data (FFT clustering). [19]

2.2. Kernel space clustering: conduct clustering in a high dimension space with kernel tricks

Kernel space clustering conducts clustering in the kernel/feature space Ω of a data set $X \in \Re^{m\times n}$. The motivation to conduct kernel space clustering is because classification/learning in a high dimensional space can have desirable results. We use the kernel tricks to avoid the huge computing complexity from clustering in the feature space Ω. To apply the kernel tricks in clustering, we need to formulate an input space clustering algorithm into inner product forms at first. Then a kernel function $k(x, y) = (\Phi(x) \bullet \Phi(y))$ is employed to evaluate all the inner products. The kernel function has to satisfy the mercer theorem.[20] Through the kernel tricks, classification/clustering can be conducted in a high dimensional space by only paying input space level computing complexity, and the feature map Φ is unnecessary to be explicit. Although several input-space clustering methods have their corresponding kernel extensions, we give the kernelization of the hierarchical clustering (HC) in this work. Qin *et al* mentioned the applications of the kernel hierarchical clustering in the gene expression data. [21] However, they only gave an approximation based kernel extension rather than a rigorous kernel extension of the classic hierarchical clustering.

Kernelization of the general hierarchical clustering algorithm consists of two steps: kernelize pairwise distance and linkage computing. In the kernelization of the pairwise distances, we focus on the Euclidean and correlation distances because they are mostly used

dissimilarity metrics in HC. The Euclidean distance between samples x_i and x_j in the kernel space can be which can be kernalized as:

$$d(\Phi(x_i),\Phi(x_j)) = (K_{ii} - 2K_{ij} + K_{jj})^{1/2} \quad (5)$$

where $K_{ij} = K(x_i, x_j) = ((\Phi(x_i) \bullet \Phi(x_j))$.

In the kernelization of the correlation distance between samples x_i and x_j, we assume the mapped vectors $\Phi(x_i), \Phi(x_j)$ are zero mean data in the kernel space Ω, then the correlation distance between $\Phi(x_i)$ and $\Phi(x_j)$ can be formulated as the following inner product form in Eq. (6), where $c_{ij} = c(\Phi(x_i), \Phi(x_j))$.

$$c_{ij} = 1 - \frac{(\Phi(x_i) \bullet \Phi(x_j))}{(\Phi(x_i) \bullet \Phi(x_j))^{1/2} (\Phi(x_i) \bullet \Phi(x_j))^{1/2}} \quad (6)$$

However, we shall drop this assumption in the kernel space for more general practice. We use the expectation of all feature data to center each feature data,

$$\overline{\Phi}(x_i) = \Phi(x_i) - \frac{1}{m}\sum_{i=1}^{m} \Phi(x_i) \quad (7)$$

Then the corresponding correlation distance can be formulated as the similar form as in the Eq. (6). Let $K_{ij}^{'} = (\overline{\Phi}(x_i) \bullet \overline{\Phi}(x_j))$, then we have the following result:

$$K_{ij}^{'} = K_{ij} - \frac{1}{m}\sum_{n=1}^{m} K_{in} - \frac{1}{m}\sum_{l=1}^{m} K_{lj} + \frac{1}{m^2}\sum_{l=1}^{m}\sum_{n=1}^{m} K_{\ln} \quad (8)$$

Since the kernel matrix K is a semi-positive definite matrix, summarizing previous results, we have the correlation distance in the kernel space between $\Phi(x_i)$ and $\Phi(x_j)$ can be computed as

$$c(\Phi(x_i),\Phi(x_j)) = \frac{(K_{ii}^{'} K_{jj}^{'})^{1/2} - K_{ij}^{'}}{(K_{ii}^{'} K_{jj}^{'})^{1/2}} \quad (9)$$

The extension of the single, complete and average linkage in the kernel space is trivial but not for the centroid linkage. The centroid linkage between two clusters is defined as the Euclidean distance between the centroid of two clusters. We give the centroid linkage d_{rs} between the clusters C_r and C_s in the Eq. (10).

$$d_{rs} = (\frac{1}{|C_r|^2}\sum_{i,j=1}^{|C_r|} k_{ij}^{(r)} - \frac{1}{|C_r||C_s|}\sum_{i=1}^{|C_r|}\sum_{j=1}^{|C_s|} k_{ij}^{(r,s)} + \frac{1}{|C_s|^2}\sum_{i,j=1}^{|C_s|} k(x_{ij}^{(s)}))^{1/2} \quad (10)$$

Where $x_i^{(r)}$ is the i^{th} sample in the cluster C_r; The $|C_r|, |C_s|$ are the number of samples in the clusters C_r and C_s; $k_{ij}^{(r)} = k(x_i^{(r)}, x_j^{(r)})$, $k_{ij}^{(s)} = k(x_i^{(s)}, x_j^{(s)})$ and $k_{ij}^{(r,s)} = k(x_i^{(r)}, x_j^{(s)})$.

2.3. What's the ideal unsupervised classification algorithm for the high dimensional gene/protein expression data?

We believe that an ideal unsupervised classification or clustering algorithm for the high dimensional gene and protein data should satisfy following criteria. 1. Some feature selection methods ought to be applied to reduce data dimensions such that data are "clean and compact". 2. The feature selection method employed should have the part-base learning property to maintain the data locality well; that is, the feature selection method can conduct local feature selection. 3. Kernel tricks are desirable to be applied in the clustering of the data after feature selection to achieve better classification results in a kernel space.

According to the criteria, we give our subspace consensus kernel classification algorithm based on the projected gradient NMF (PG-NMF). The basic idea is to apply a convergent soundness local feature algorithm: PG-NMF to the gene/protein expression dataset X, which is equivalent to project the dataset X into the subspace S generated by the PG-NMF: $X \sim WH$, where W is the basis matrix generating the subspace. Then kernel hierarchical clustering is applied to column data the feature matrix H, which are the prototype data of the original data. Since the basis matrix and feature matrix are not unique in the NMF. We develop the consensus kernel hierarchical clustering algorithm (CKHC) to get the final classification.

3. PG-NMF SUBSPACE KERNEL HIERARCHICAL CLASSIFICATION

PG-NMF based subspace kernel classification is to conduct consensus kernel hierarchical clustering (CKHC) to each feature matrix H in a subspace S generated by the PG-NMF. The CKHC is an algorithm to run the kernel hierarchical clustering in a Monte Carlo simulation approach and compute the final classification by building a consensus tree. It consists of two general steps. 1. Build a consensus tree for the expression dataset X at each rank by conducting CKHC

to feature matrices H from the PG-NMF. 2. Then the best consensus tree, which is the final classification, is selected by our novel model selection method. The following algorithm describes the consensus kernel hierarchical clustering (CKHC) at rank r**.**

Algorithm 1 Consensus kernel hierarchical clustering at rank r
Input: nonnegative matrix X (n×m), rank r,
PG-NMF running times N>=100,
Kernel function $k(x, y)$, linkage metric l
Output: the consensus tree T at rank r

// Run PG-NMF X~WH to do feature selection at rank r N times
1. For run=1:N
2. Initialize W and H randomly
3. Compute X~WH, $W \in R^{n\times r}, H \in R^{r\times m}$ by PG-NMF
4. Compute the kernel pairwise distances between columns of feature matrix H in the kernel space by Eq. (5)/(9)
5. Record the kernel pairwise distances in an m(m-1)/2 ×1 vector: d
6. Concatenate all such kernel distance vectors for N feature matrices in a matrix D: D=[D, d];
7. End
8. Compute a consensus kernel distance vector $d_{consensus}$ by weighting the ratios of the sum of each column in D over the sum of the elements of matrix D

$$d_{consensus} = \sum_{j=1}^{N} \frac{D(:, j)}{\sum_{i=1}^{m(m-1)/2} \sum_{j=1}^{N} D(i, j)} \times D(:, j)$$

9. Build the consensus tree T from the consensus kernel distance vector under the linkage metric l
10. Return T

We still need to answer the following question: 'What is the model selection method to find the most robust consensus tree (classification)?' To avoid the exhaustive search on all possible ranks, we give a singular-value based rank selection method to find an optimal rank search interval $[2, r^*]$. The idea can be described as follows.

Given a threshold ε ($\varepsilon \in [0.90,1)$), we compute the importance ratio of first r^* singular values such that the important ration >= the threshold. The importance ratio of first r^* singular values is defined as the ratio of the sum of the first r^* singular values over the sum of all singular values (Eq.11).

$$\rho_{r^*} = \sum_{j=1}^{r^*} \sigma_j / \sum_{i=1}^{m} \sigma_i \,. \tag{11}$$

That is, PG-NMF is only conducted in the optimal rank search interval $[2, r^*]$ and we only search the best consensus tree from the r* consensus trees.

The most robust consensus tree will be from which rank in the interval $[2, r^*]$? It is reasonable that the most robust consensus tree should be from a rank, where the bases of its subspace generated by the PG-NMF each time represent all levels of patterns inherent in the dataset. From the point of view of data variability, it is a rank where the ratio between the largest data variability and the smallest data variability of the bases data reaches its maximum value.

We propose a measure robust index δ to find the most robust consensus tree according to the previous considerations. The robust index δ is the condition number of the covariance matrix of the average basis matrix $E(W)$ from the N times running of the PG-NMF. The average basis matrix is defined as:

$$E(W) = \frac{1}{N} \sum_{i=1}^{N} W^{(i)} \tag{12}$$

The condition number of the covariance matrix of the average basis matrix $E(W)$ is the ratio between the maximum eigenvalue and the minimum eigenvalue of $E(W)$: $\delta = \lambda_{\max} / \lambda_{\min}$. The $\lambda_{\max}$ is the variance of the 1st principal component of the average basis matrix: the largest data variability of the basis data. The $\lambda_{\min}$ is the variance of the last principal component of the average basis matrix: the smallest variability of the basis data. The robust index can be huge but it is impossible to reach infinite because $\lambda_{\min}$ is the smallest positive eigenvalue of the covariance matrix of $E(W)$. The final classification is just the consensus tree with the largest robust index number. The PG-NMF based consensus kernel hierarchical clustering algorithm (PG-NMF-CKHC) can be described as follows.

Algorithm 2 PG-NMF based Consensus kernel hierarchical clustering
Input: a $n \times m$ nonnegative data matrix X, Importance ratio threshold $\varepsilon \geq 0.90$
Output: the final consensus tree T

1. Decide the rank search interval $[2, r^*]$ by the important ratio threshold ε
2. For r=2: r*
3. Conduct consensus kernel hierarchical clustering at rank r to get a consensus tree T_r at rank r
4. Compute the robust index δ of the consensus tree T_r
5. End
6. $T \leftarrow T_r$ with the maximum robust index

4. EXPERIMENTS

We apply the PG-NMF-CKHC algorithm to discover the cancer molecular patterns for several bench-mark cancer datasets. We use a measure called classification rate $C_r = \sum_{i=1}^{m} \delta(i)/m$ to evaluate the accuracy of the unsupervised classification for a dataset with m samples, where $\delta(i) = 1$ if the sample i is assigned in a correct cluster; otherwise $\delta(i) = 0$. We use three kernel functions in our algorithms: linear, polynomial and Gaussian kernel. The dissimilarity measures in the kernel hierarchical clustering are chosen as Euclidean and correlation distances. We choose the average linkage metric in the kernel hierarchical clustering. The PG-NMF algorithm is run N=100 times in each optimal rank search interval with tolerance 10e-9.

The first dataset is **Leukemia dataset,** a benchmark dataset consisting of 38 samples in the cancer research. It can be classified as 27 acute lymphoblastic leukemia (ALL) and 11 acute myelogenous leukemia (AML) marrow samples. The ALL samples can be further divided into 19 'B' and 8 'T' subtypes. HC and SOM were proved to be unstable for this dataset.[15] The optimal search interval for this dataset is [2,6] under the importance ratio threshold 0.90. The robust index in PG-NMF-CKHC reaches its largest number at rank 5 for a Gaussian kernel under the correlation distance (Figure 2). Figure 1 is the visualization of the final consensus tree. It is clear that there are three clusters, AML, ALL-B, and ALL-T in the final consensus tree.

There is just only one misclassification i.e. *ALL_14749_B-cell* was assigned to AML. We have found the combinations of the Gaussian kernel function and correlation/Euclidean distance under the average linkage metric both can reach the best performance in the classification. Under the linear kernel, we can see that classification results under the correlation distance are better than those of Euclidean distance (Figure 3). The NMF clustering has two misclassified samples*:* *ALL_21302_B-cell* and *ALL_14749_B-cell*. Sparse-NMF clustering has one misclassified sample: *AML_12*. However, the running time of NMF and sparse-NMF clustering are twice more than that of our algorithm.

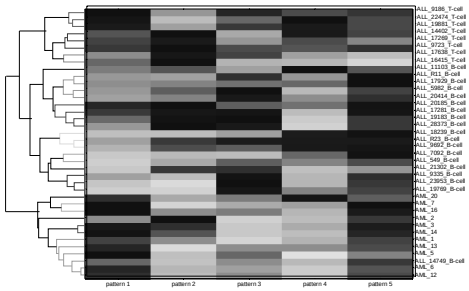

Fig. 1. The visualization of the consensus tree at rank 5 for a Gaussian kernel under the correlation distance and average linkage metric.

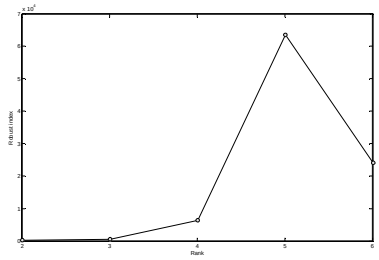

Fig. 2. The largest robust index reached at rank 5 for the Gaussian kernel with correlation distance.

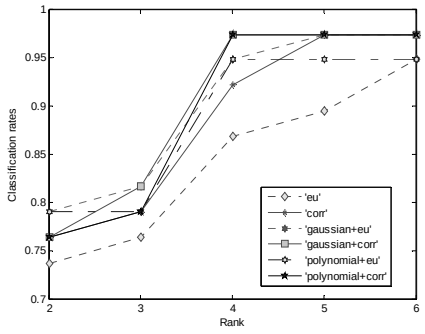

Fig. 3. The classification rates under linear, polynomial and Gaussian kernel for Euclidean and correlation distances.

The second dataset is **Medulloblastoma** dataset, the gene expression data from childhood brain tumors

known as medulloblastomas. The pathogenesis about these tumors is still not well understood yet by investigators. However, there are two generally accepted histological sub-classes: classic and desmoplatic. These sampled are divided as 25 classic and 9 desmoplastic medulloblastomas. General HC and SOM failed to reveal the classifications of these samples.[15] The robust index reaches its maximum in the optimal rank search interval [2,10] at rank 7 for a polynomial kernel under the correlation distance. Figure 4 is the visualization of the final classification. There are 8 desmoplastic samples clustered and total 2 samples are misclassified: sample 25 and sample 33.

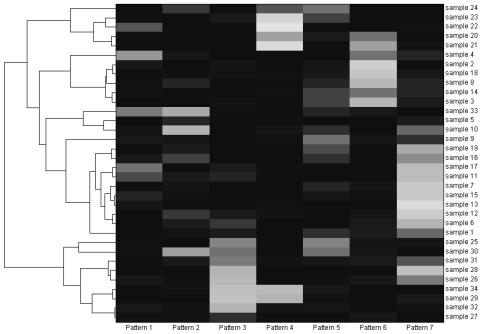

Fig. 4. Visualization of the final consensus tree of the medulloblastomas dataset at the rank 7 under the polynomial kernel under the average linkage metric and correlation distance.

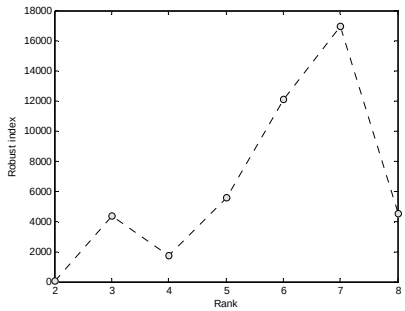

Fig. 5. The largest robust index reached at rank 7 for the polynomial kernel with correlation distance.

The NMF has 2 samples misclassified at its best decomposition rank 5.[15] However, it only gets 7 desmophlastic samples clustered. Although our algorithm also have 2 misclassified samples, we have better clustering structure since there are 8 desmophlastic samples clustered. On the other hand, sparse-NMF has 7 misclassified at its best rank 5. [16] It seems sparseness constraints do not contribute to the improving classification rates for this dataset. Since the pathogenesis of medulloblastoma is still not well-understood, we did not compute the classification rates for this dataset.

The third dataset is an **ovarian cancer dataset**, a MS proteomics dataset consisting of 20 cancer and 20 normal samples, which presents as a 15142×40 positive matrix. This data set is a subset of *Ovarian Dataset 8-7-02* that was generated using the WCX2 protein array, which includes 91 controls and 162 ovarian cancers. For this dataset, we try supervised classification first. We randomly pick other 40 samples (20 cancer and 20 normal) from the original dataset as a training set; then we use kNN under Euclidean and correlation distance to classify the MS data. We have found the best classification rate from kNN is 92%. But it can't classify sample 3, 12, 36 correctly. Our algorithm reaches the best classification at rank 7 in the optimal rank search interval [2,10]. There is only one misclassified sample :sample 36 (Figure 6).

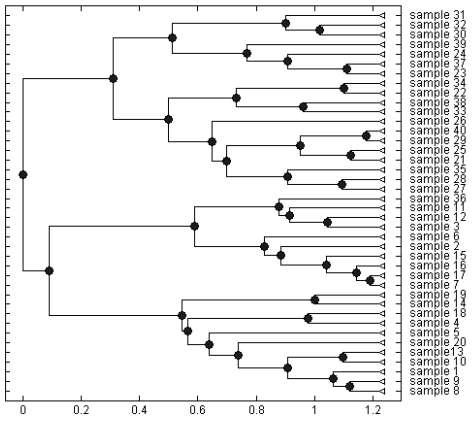

Fig. 6. The final consensus tree at rank 7 under Gaussian kernel with correlation distance.

Figure 7 shows the performance of linear, Gaussian and polynomial kernel in the classification. The combination of the polynomial kernel and correlation distance has the best performance under the average

linkage metric. Classification rates generally decrease after the rank 7 and the correlation distance generally performs better than the Euclidean distance in the classification.

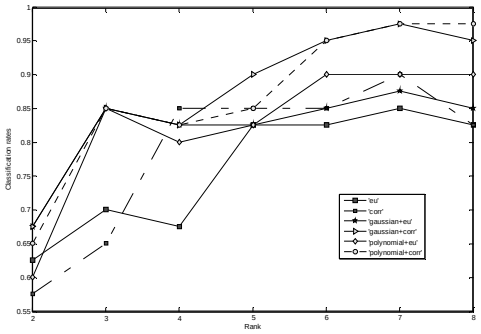

Fig. 7. The classification rates of the PG-NMF-CKHC for this dataset: polynomial kernel + correlation distance reaches the best classification rate.

We also apply NMF and sparse-NMF classification for the proteomics data, although they were developed under the context of gene expression data. There are 8 samples misclassified from NMF clustering and 12 samples misclassified from the Sparse NMF clustering for our ovarian cancer dataset. Both algorithms indicate there are 2 clusters from their cophenetic coefficients. Since a proteomics dataset generally has much higher dimensionalities than a gene expression dataset, NMF and sparse NMF clustering have large time complexity for a proteomics dataset. For this dataset, NMF clustering takes >78 hours and sparse-NMF clustering takes >153 hours running under two PCs with 3.0 GHZ CPU and 504 RAM running under WIN-XP OS. It seems that NMF based clustering/classification mechanism can't work well in the context of the proteomics data.

4.1. Comparing classification results from kNN, sparse-NMF and support vector machines (SVM)

We compare PG-NMF-CKHC for the four datasets (the leukemia, medulloblatoma, ovarian cancer dataset and a colon cancer dataset, which consists of 22 controls and 40 cancer data samples) with the classic NMF clustering, sparse-NMF clustering, and SVM and kNN classifications. In kNN and SVM, We run classification 10 times under holdout cross-validation with 50% hold-out percentage for each case. We take the average classification rates as the final classification rates. In the SVM classification, we also use linear, polynomial and Guassian kernel. We select the best final classification rate from three kernels as the final classification rate of SVM. In the leukemia data, we use SVM/kNN to classify ALL and AML types instead of all three types. Although the pathogenesis of medulloblatoma is not well established, we still compute the classification rates of this dataset based on the general assumption that samples are divided as 25 classic and 9 desmoplastic medulloblastomas, for the convenience of comparisons. Table 1 shows the classification rates for the four benchmark datasets from kNN, PG-NMF-CKHC, NMF, sparse-NMF and SVM classifications.

We have found that our algorithm is superior to the NMF, sparse-NMF and supervised SVM classification algorithms for these datasets; The NMF classification has better performance than SVM and kNN for three gene expression datasets. Sparse-NMF has averagely better performance than kNN for three gene expression datasets. However, the NMF and sparse-NMF can't compete with kNN and SVM for the proteomics data.

According to our classification results, it seems that sparseness constraint on the NMF may not always contribute to the improvement in the classifications for some datasets. Besides the ovarian dataset, for the medulloblatoma dataset, the classic NMF clustering seems to perform better in classifying desmoplastic medulloblastomas than the sparse-NMF clustering at rank 5, where both algorithms reaches the most robust reproducibility partitions. We also noticed the NMF and sparse-NMF clustering can not compete with SVM classification for the ovarian dataset. It is interesting to see that sparseness constraint may not lead to the better classification results for the colon cancer dataset. The classic NMF clustering reaches its largest cophenetic correlation coefficient at rank 2 (2 clusters) and its corresponding classification rate is 0.9355. However, the sparse NMF clustering reaches its largest cophenetic correlation coefficient at rank 4 (4 clusters) and its corresponding classification rate is 0.7581. It is possible due to the fact that the expression patterns of those dominant co-expressed genes such as, oncogenes, tumor suppressor genes are not extracted out in the sparse representation. This may also indicate that sparseness

control may not always lead to a better classification results for some dataset. Figure 8 and 9 give the visualization of the NMF and sparse-NMF clustering from the rank 2-5 for the colon cancer dataset. Probability of two samples clustered together is indicated by color. Generally, blue indicates a numeric value near 0 and a red color indicates a numeric values near 1. The deep blue standing for 0 indicates samples are never assigned in one cluster and dark red standing for 1 indicates samples are assigned in one cluster.

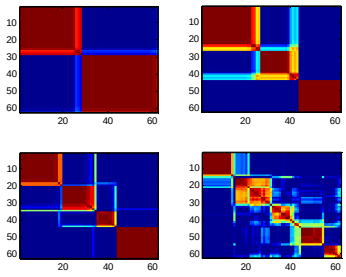

Fig. 8. The visualization of the NMF clustering from rank 2-5 for the colon dataset

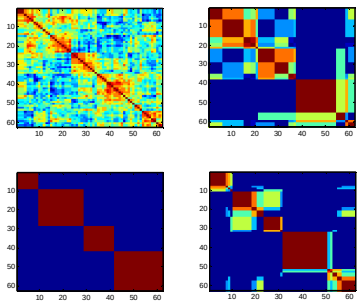

Fig. 9. The visualization of the sparse-NMF clustering from rank 2-5 for the colon dataset

5. CONCLUSIONS

As a part-based learning machine learning algorithm, NMF has found its application successfully in image analysis, document clustering and cancer molecular pattern discovery. In this study, we present an NMF based subspace kernel clustering algorithm: PG-NMF-CKHC based on the input space, subspace and kernel space clustering framework. We have shown that PG-NMF-CKHC improves the cancer molecular pattern discovery for the well-studied four datasets. It can work well for both gene expression data and protein expression data according to out current results.

Our algorithm can be generalized to a family of subspace kernel classification/clustering algorithms in machine learning by selecting different transforms to generate subspaces and different kernel clustering algorithms to cluster data. For example, conduct kernel k-means clustering in a subspace generated by the independent component analysis (ICA) applied to a high dimensional dataset, or conduct the kernel Fisher discriminant analysis (KFDA) [22] in a subspace generated by principal component analysis (PCA).

Despite its promising features, it is also worthy to point out that PG-NMF based consensus kernel hierarchical clustering has the limitation of greater algorithmic complexity, especially compared with the traditional hierarchical clustering (HC). However, it is clear that our algorithm is easy to fit in a parallel computing structure due to its Monte Carlo simulation mechanism. Thus, we plan to implement the parallel version of the subspace based kernel classification algorithm for the cancer molecular pattern classification in the following work.

Table 1. Compare PG-NMF-CKHC classification results with those of the NMF, sparse-NMF, SVM and KNN classifications

Cancer Data Information			Algorithm Classification Rates				
Cancer Name	Data Size	#type	kNN	PGNMF-CKHC	NMF	Sparse-NMF	SVM
Leukamia	5000×38	3	0.8860	0.9737	0.9470	0.9737	0.9132
Medulloblastoma	5893×34	2	0.7611	0.9412	0.9412	0.8235	0.8300
Ovarian	15142×40	2	0.8990	0.9750	0.8000	0.7000	0.9474
Colon	2000×62	2	0.7667	0.9355	0.9032	0.7581	0.8542

Acknowledgments

Author wants to thank the support from the New Faculty Research Award at Eastern Michigan University for this research.

References

1. Lilien, R. and Farid, H. Probabilistic Disease Classification of Expression-dependent Proteomic Data from Mass Spectrometry of Human Serum, *Journal of Computational Biology* 2003; **10** (6), 925-946.
2. Golub, T. et al. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, *Science* 1999; **286**: 531-537.
3. Furey T., Cristianini N., Duffy N, Bednarski D., Schummer M. and Haussler D. Support vector machine classification and validation of cancer tissue samples using microarray expression data, *Bioinformatics* 2000; **16** (10): 906-914.
4. Hautaniemi, S. , Yli-Harja, O., Jaakko Astola, J., Kauraniemi, P. et al. Analysis and Visualization of Gene Expression Microarray Data in Human Cancer Using Self-Organizing Maps, *Machine Learning* 2003; **52**: 45-66.
5. Ressom, H., Varghese, R., Saha, D., Orvisky, R. et al. Analysis of mass spectral serumprofiles for biomarker selection. *Bioinformatics* 2005; **21**: 4039-4045.
6. Liu Z., Chen D. and Bensmail H. Gene expression data classification with Kernel principal component analysis. *J Biomed Biotechnol.* 2005 (2) 155–159.
7. Eisen,M. *et al*. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA* 1998; **95**: 14863–14868.
8. Tamayo, P. et al. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA* 1999; **96**: 2907–2912.
9. Bicciato,S. *et al*. PCA disjoint models for multiclass cancer analysis using gene expression data. *Bioinformatics* 2003; **19**: 571–578
10. Wall, M., Andreas, R., Rocha, L. Singular value decomposition and principal component analysis. *A Practical Approach to Microarray Data Analysis*. Berrar, D., W. Dubitzky, W., Granzow, M. eds. Kluwer: Norwell, 2003; 91-109.
11. Tan, Y., Shi, L., Tong, W., and Wang, C. Multi-class cancer classification by total principal component regression using microarray gene expression data. *Nucleic Acids Res*. 2005; **33**(1) 56-65.
12. Zhang, X., Yap, Y., Wei, D., Chen, F. and Danchin, A. Molecular diagnosis of humancancer type by gene expression profiles and independent component analysis, *European Journal of Human Genetics* 2005; **1–9**: 1018-4813.
13. Daniel D. Lee and H. Sebastian Seung.: Learning the parts of objects by non-negative matrix factorization. *Nature* 1999; **401**: 788–791.
14. Lin, C. Projected gradient methods for non-negative matrix factoriz*ation*, *Neural Computation* 2007; In Press.
15. Brunet, J., Tamayo, P., Golub, T. and Mesirov., J. Molecular pattern discovery using matrix factorization. *Proc. Natl Acad. Sci. USA*, 2004, **101**,12: 4164–4169.
16. Gao, Y. and Church, G. Improving molecular cancer class discovery through sparse nonnegative matrix factorization, *Bioinformatics* 2005; **21** (21):, 3970–3975.
17. Patrik O. Hoyer: Non-negativematrix factorization with sparseness constraints. *Journal of Machine Learning Research* 2004, **5**: 1457–1469.
18. Yeung, K. and Ruzzo, W.: Principal Component Analysis for clustering gene expression data, *Bioinformatics*, 2001; **17** (9): 763-774.
19. Lee, S. and Batzoglou, S. ICA-Based Clustering of Genes from Microarray Expression Data, *Neural Information Processing Systems(NIPS)* 2003.
20. Vapik, V. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.
21. Qin. J. *et al*.: Kernel hierarchical gene clustering from microarray expression data, *Bioinformatics*, 2003, **19** (16), 2097-2104.
22. Mika, S., Rätsch, G., Weston, J., Schölkopf, B. and Müller, KR. Fisher discriminant analysis with kernels, *Neural Networks for Signal Processing IX*,. 1999; 41-48.