

Gene Selection via Matrix Factorization

Fei Wang

State Key Lab of Intelligent Technologies and Systems
Department of Automation
Tsinghua University

feiwang03@mails.thu.edu.cn

Tao Li

School of Computer Science
Florida International University
Miami, FL 33199, U.S.A.

taoli@cs.fiu.edu

Abstract

The recent development of microarray gene expression techniques have made it possible to offer phenotype classification of many diseases. However, in gene expression data analysis, each sample is represented by quite a large number of genes, and many of them are redundant or insignificant to clarify the disease problem. Therefore, how to efficiently select the most useful genes has been becoming one of the most hot research topics in the gene expression data analysis. In this paper, a novel unsupervised gene selection method is proposed based on matrix factorization, such that the original gene matrix can be optimally reconstructed using those selected genes. To make our algorithm more efficient, we derive a kmeans preclustering approach to accelerate our algorithm, and we also prove theoretically the optimality of this approach. Finally the experimental results on several data sets are presented to show the effectiveness of our method.

Keywords. Gene Selection; Matrix Factorization; Kmeans Clustering

1. Introduction

The DNA arrays, pioneered in *Chee et al.* [4] and *Fodor et al.* [8], are novel technologies that are designed to measure gene expression of tens of thousands of genes in a single experiment. A DNA array reflects the state of the cell with different protein and mRNA compositions. Thus gene expression profiles of samples corresponding to different pathological states of the same tissue provide molecular rather than morphological signature of the malignancy [1][11].

When analyzing those gene expression profiles, an important issue is how to select a small subset of genes that are useful for the classification of the target phenotypes. Typically, there exists much redundancy in the gene expression

data. For example, for a two-way cancer/non-cancer diagnose, usually 50 informative genes are sufficient [11].

From the machine learning perspective, gene selection is just a *feature selection* problem. It is conventional to categorize the feature selection process into *wrapper* and *filter* modes. Wrapper methods contain a well-specified objective function, which should be optimized through the selection procedure. One can often obtain a very small subset of features with relatively high accuracy using the wrapper methods since the usefulness of features is usually directly determined by the estimated accuracy of the learning algorithm [16][20]. Feature filtering is a process of selecting features without referring back to the data classification or any other target function and it ranges from simple methods such as information gain [3], statistical tests (t-test)[11] to more sophisticated methods such as Markov blanket based on conditional independence [15].

However, most of the state-of-the-art gene selection methods are *supervised*, *i.e.* they assume the class information of the samples are already known. The *unsupervised* methods, although which are relatively scarce, are also important to biological data analysis since (1) they are *unbiased*, since they analyze the data purely based on the data themselves, no prior knowledge from the expert or data-analyst is necessary [14]; (2) they can reduce the risk of overfitting the data set (*e.g.* the supervised methods cannot with data of new unknown types[7]).

Based on the above considerations, in this paper, we propose a novel unsupervised feature selection algorithm, *i.e.* our method can select genes without any prior knowledge on the sample class (phenotype) information. Our algorithm is based on the basic assumption that the selected genes should be *representative* enough, *i.e.* the data (gene) matrix can be *reconstructed* from those selected genes with minimum loss. In our algorithm, such loss is measured by the *Frobenius* norm of the difference between the original and reconstructed gene matrix. We show that the minimization of such loss can be carried out by an iterative matrix updating procedure. Moreover, to make our algorithm scale to

the large number of genes, we derive a *kmeans preclustering* method to accelerate the algorithm, and we prove theoretically the optimality of such method from the matrix approximation perspective. Finally, experiments on several gene data sets are conducted to show the effectiveness of our method.

The rest of this paper is organized as follows. Section 2 introduces our algorithm in detail, and the *kmeans preclustering* method is introduced in section 3. The experimental results are presented in section 4, followed by the conclusions and discussions in section 5.

2. The Algorithm

Basically, the target of gene selection is to select a number of most *representative* genes. Here *representative* can be understood from different perspectives, for example, the selected genes should have the maximum *mutual information* with the data labels [5], or the selected genes should maximize the *feature margin* as defined in [9]. However, most of these gene selection methods are *supervised*, i.e., we should know the labels of the data before we run the algorithm. This may make those algorithms impractical since the collection of labelled data is usually expensive and time consuming.

Therefore, we propose a novel unsupervised gene selection method in the following. Here we define *representative* genes as that *the whole data matrix can be optimally reconstructed using those genes*. Mathematically, assume that we are given N data sample, and $\mathcal{X} = \{\mathbf{x}_i\}$ denotes the set of N -dimensional gene vectors with $1 \leq i \leq M$. Our goal is to select $\tilde{\mathbf{x}}_j \in \mathcal{X}$, $1 \leq j \leq K$ such that the loss

$$\mathcal{J} = \sum_{i=1}^M \left\| \mathbf{x}_i - \sum_{j=1}^K \delta_{ij} \tilde{\mathbf{x}}_j \right\|^2 \quad (1)$$

is minimized, where $\delta_{ij} \geq 0$ are the reconstruction coefficients. Written in its matrix form, Eq.(1) can be reexpressed as

$$\mathcal{J} = \left\| \mathbf{X} - \tilde{\mathbf{X}} \Delta^T \right\|_F^2, \quad (2)$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M] \in \mathbb{R}^{N \times M}$ is the *feature matrix*, $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_K] \in \mathbb{R}^{N \times K}$ is the *selected gene matrix*, and Δ is an $M \times K$ nonnegative *coefficients matrix* with its (i, j) -th entry $\Delta_{ij} = \delta_{ij}$, $\|\cdot\|_F$ is the *Frobenius norm* of a matrix. To minimize \mathcal{J} and solve for the optimal $\tilde{\mathbf{X}}$ and Δ , we introduce a set of *auxiliary variables* $\{w_{uv}\}$ ($1 \leq u \leq K$, $1 \leq v \leq M$) and expand $\tilde{\mathbf{x}}_i$ as

$$\tilde{\mathbf{x}}_i = \sum_{j=1}^M w_{ij} \mathbf{x}_j, \quad (3)$$

where $w_{ij} \in \{0, 1\}$, and for each specific i , there is only one j such that $w_{ij} = 1$. Then we can define a 0-1 matrix

\mathbf{W} of size $M \times K$ and rewrite Eq.(3) as

$$\tilde{\mathbf{X}} = \mathbf{X} \mathbf{W}, \quad (4)$$

where $\mathbf{W}_{ij} = w_{ji}$. Therefore, the optimization problem that we want to solve ultimately becomes

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{v}} \quad & \left\| \mathbf{X} - \mathbf{X} \mathbf{W} \Delta^T \right\|_F^2 \\ \text{s.t.} \quad & \delta_{ij} \geq 0, \\ & w_{ij} \in \{0, 1\}, \\ & \mathbf{W}^T \mathbf{1}_M = \mathbf{1}_K, \end{aligned} \quad (5)$$

where $\mathbf{1}_M \in \mathbb{R}^{M \times 1}$, $\mathbf{1}_K \in \mathbb{R}^{K \times 1}$ are column vectors with all their elements equaling to 1. Clearly, from the matrix factorization perspective, the goal our method is just to seek an optimal factorization of the gene matrix \mathbf{X} such that

$$\mathbf{X} \approx \mathbf{X} \mathbf{W} \Delta^T. \quad (6)$$

Therefore we call our method a *matrix factorization (MF)* approach.

The problem is that Eq.(5) is a complicated combinatorial optimization problem needs *integer programming*, which is known to be NP-hard. Hence we will introduce a relaxation scheme in the following to make it solvable.

2.1 Problem Relaxation

A common trick to make the integer programming problem as Eq.(5) solvable is to drop the constraint that $w_{ij} \in \{0, 1\}$, which means that w_{ij} can take continuous real values. In our approach, we also apply this trick and further constrain that $w_{ij} \geq 0$, such that w_{ij} can be regarded as the *possibility* that \mathbf{x}_j is selected as the i -th representative gene (note that we can also post-normalize the columns of \mathbf{W} to make $\mathbf{W}^T \mathbf{1}_N = \mathbf{1}_K$, then \mathbf{W}_{ij} can be regarded as the *probability* of selecting \mathbf{x}_i as the j -th representative gene). After the relaxation, our optimization problem becomes

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{v}} \quad & \left\| \mathbf{X} - \mathbf{X} \mathbf{W} \Delta^T \right\|_F^2 \\ \text{s.t.} \quad & \delta_{ij} \geq 0, \\ & w_{ij} \geq 0. \end{aligned} \quad (7)$$

Based on the equality that $\|\mathbf{A}\|_F^2 = \text{tr}(\mathbf{A}^T \mathbf{A})$, where $\text{tr}(\cdot)$ denotes the trace of a matrix, we can expand the objective function of Eq.(7) as

$$\begin{aligned} \mathcal{J} &= \left\| \mathbf{X} - \mathbf{X} \mathbf{W} \Delta^T \right\|_F^2 \\ &= \text{tr} \left((\mathbf{X} - \mathbf{X} \mathbf{W} \Delta^T)^T (\mathbf{X} - \mathbf{X} \mathbf{W} \Delta^T) \right) \\ &= \text{tr} \left((\mathbf{I} - \mathbf{W} \Delta^T)^T \mathbf{X}^T \mathbf{X} (\mathbf{I} - \mathbf{W} \Delta^T) \right). \end{aligned}$$

Let $\mathbf{K} = \mathbf{X}^T \mathbf{X}$ be the $M \times M$ gene similarity matrix, then

$$\begin{aligned}\mathcal{J} &= \text{tr} \left((\mathbf{I} - \mathbf{W} \Delta^T)^T \mathbf{K} (\mathbf{I} - \mathbf{W} \Delta^T) \right) \\ &= \text{tr} (\mathbf{K} - 2\Delta \mathbf{W}^T \mathbf{K} + \Delta \mathbf{W}^T \mathbf{K} \mathbf{W} \Delta^T) \\ &= \text{tr}(\mathbf{K}) - 2\text{tr}(\mathbf{W}^T \mathbf{K} \Delta) + \text{tr}(\Delta \mathbf{W}^T \mathbf{K} \mathbf{W} \Delta^T),\end{aligned}$$

which is a standard quadratic form of Δ if we fix \mathbf{W} , and a standard quadratic form of \mathbf{W} if we fix Δ . Therefore, the alternative optimization technique can be adopted to solve for the nonnegative solution that minimizes \mathcal{J} . More concretely, we have the following theorem:

Theorem 1[13]. Define the standard nonnegative quadratic form as

$$\mathcal{F}(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x}, \quad (8)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ is a nonnegative $n \times 1$ vector, \mathbf{A} is an arbitrary $n \times n$ semi-definite positive matrix, $\mathbf{b} = (b_1, b_2, \dots, b_n)^T$ is an arbitrary $n \times 1$ vector. Let $\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-$, where \mathbf{A}^+ and \mathbf{A}^- are two symmetric matrix with all their elements nonnegative. Then the solution \mathbf{x} that minimizes $\mathcal{F}(\mathbf{x})$ can be obtained by the following update rule

$$x_i \leftarrow x_i \left(\frac{-b_i + \sqrt{b_i^2 + 4(\mathbf{A}^+ \mathbf{x})_i (\mathbf{A}^- \mathbf{x})_i}}{2(\mathbf{A}^+ \mathbf{x})_i} \right), \quad (9)$$

where $(\cdot)_i$ represents the i -th element of a vector. For a general quadratic expression $\mathcal{F}(\mathbf{y})$ which does not take the standard form as shown in Eq.(8), we can solve for its corresponding \mathbf{A} and \mathbf{b} for updating \mathbf{y} by

$$\mathbf{A}_{ij} = \frac{\partial^2 \mathcal{F}(\mathbf{y})}{\partial y_i \partial y_j} \quad (10)$$

$$b_i = \left. \frac{\partial \mathcal{F}(\mathbf{y})}{\partial y_i} \right|_{\mathbf{y}=0} \quad (11)$$

Based on the above theorem, we can also derive the updating rule of \mathbf{W}_{ij} and Δ_{ij} for minimizing \mathcal{J} . First let's decompose the gene similarity matrix as

$$\mathbf{K} = \mathbf{K}^+ - \mathbf{K}^-, \quad (12)$$

where \mathbf{K}^+ and \mathbf{K}^- are symmetric matrices with all their elements nonnegative. Then the updating rules of \mathbf{W}_{ij} and Δ_{ij} can be derived as [21]

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} \frac{(\mathbf{K} \Delta)_{ij} + \sqrt{(\mathbf{K} \Delta)_{ij}^2 + 4\mathbf{P}_{ij}^+ \mathbf{P}_{ij}^-}}{2\mathbf{P}_{ij}^+} \quad (13)$$

$$\Delta_{ij} \leftarrow \Delta_{ij} \frac{(\mathbf{K} \mathbf{W})_{ij} + \sqrt{(\mathbf{K} \mathbf{W})_{ij}^2 + 4\mathbf{Q}_{ij}^+ \mathbf{Q}_{ij}^-}}{2\mathbf{Q}_{ij}^+} \quad (14)$$

Table 1. Matrix Factorization Gene Selection

Input: gene matrix \mathbf{X} , number of desired features K , convergence threshold θ .

Output: continuous nonnegative matrix \mathbf{W}^* .

1. Let $\theta_{old} = 10^4$, $\theta_{new} = 0$, $\Delta\theta = |\theta_{new} - \theta_{old}|$;
2. Compute the gene similarity matrix \mathbf{K} ;
3. Initialize \mathbf{W} to be a $M \times K$ nonnegative matrix;
4. Initialize Δ to be a $M \times K$ identity matrix;
5. While $\Delta\theta > \theta$
 - (a). Update \mathbf{W} using Eq.(13);
 - (b). Update Δ using Eq.(14);
 - (c). Compute $\theta_{new} = \|\mathbf{X} - \mathbf{X} \mathbf{W} \Delta^T\|_F^2$;
 - (d). Compute $\Delta\theta = |\theta_{new} - \theta_{old}|$;
4. Output \mathbf{W}^* .

where

$$\mathbf{P}^+ = \mathbf{K}^+ \mathbf{W} \Delta^T \Delta \quad (15)$$

$$\mathbf{P}^- = \mathbf{K}^- \mathbf{W} \Delta^T \Delta \quad (16)$$

$$\mathbf{Q}^+ = \Delta \mathbf{W}^T \mathbf{K}^+ \mathbf{W} \quad (17)$$

$$\mathbf{Q}^- = \Delta \mathbf{W}^T \mathbf{K}^- \mathbf{W} \quad (18)$$

Therefore, we can first initialize \mathbf{W} and Δ to some arbitrary nonnegative matrix, and use Eq.(13) and Eq.(14) to update them until convergence. Table 1 summarizes the main procedure of our matrix factorization gene selection algorithm. However, we have the following non-unique solution theorem.

Theorem 2. For any \mathbf{W}^* , Δ^* that is the optimal solution to problem (7), then $\mathbf{W}^* \mathbf{R}$, $\Delta^* \mathbf{R}$ are also solutions to that problem, where \mathbf{R} is a $K \times K$ square matrix subject to $\mathbf{R}_{ij} \geq 0$, $\mathbf{R} \mathbf{R}^T = \mathbf{I}$.

Proof. Clearly, if \mathbf{W}^* and Δ^* are the solutions to problem (7), then $\mathbf{W}_{ij}^* \geq 0$, $\Delta_{ij}^* \geq 0$. Then for any nonnegative matrix \mathbf{R} , let $\mathbf{W}' = \mathbf{W}^* \mathbf{R}$, $\Delta' = \Delta^* \mathbf{R}$, we have $\mathbf{W}'_{ij} \geq 0$, $\Delta'_{ij} \geq 0$, which satisfies the nonnegative constraints in problem (7). Furthermore, if $\mathbf{R} \mathbf{R}^T = \mathbf{I}$, then

$$\begin{aligned}\mathcal{J}' &= \text{tr}(\mathbf{K}) - 2\text{tr}(\mathbf{W}'^T \mathbf{K} \Delta') + \text{tr}(\Delta' \mathbf{W}'^T \mathbf{K} \mathbf{W}' \Delta'^T) \\ &= \text{tr}(\mathbf{K}) - 2\text{tr}(\mathbf{K} \Delta^* \mathbf{R} \mathbf{R}^T \mathbf{W}^{*T}) \\ &\quad + \text{tr}(\Delta^* \mathbf{R} \mathbf{R}^T \mathbf{W}^{*T} \mathbf{K} \mathbf{W}^* \mathbf{R} \mathbf{R}^T \Delta'^T) \\ &= \text{tr}(\mathbf{K}) - 2\text{tr}(\mathbf{W}^{*T} \mathbf{K} \Delta^*) + \text{tr}(\Delta^* \mathbf{W}^{*T} \mathbf{K} \mathbf{W}^* \Delta'^T) \\ &= \mathcal{J}^*\end{aligned} \quad (19)$$

That is, $\mathbf{W}^* \mathbf{R}$, $\Delta^* \mathbf{R}$ are also the optimal solutions to problem (7). \square .

Recalling that the final goal of our method is to find an optimal 0-1 matrix \mathbf{W} satisfying the constraints in problem

(5), we need to discretize the continuous \mathbf{W} after solving problem (7). In the next section we will introduce an optimal discretization method to achieve this goal.

2.2 Optimal Discretization

From theorem 2 we can see that the problem of optimal discretization of \mathbf{W}^* (assuming \mathbf{W}^* is the optimal solution to problem (7)) is just to solve the following problem

$$\begin{aligned} \min_{\tilde{\mathbf{W}}, \mathbf{R}} \quad & \varepsilon = \|\tilde{\mathbf{W}} - \mathbf{W}^* \mathbf{R}\|_F^2 \\ \text{s.t.} \quad & \tilde{\mathbf{W}}_{ij} \in \{0, 1\}, \\ & \tilde{\mathbf{W}}^T \mathbf{1}_M = \mathbf{1}, \\ & \mathbf{R}_{ij} \geq 0, \\ & \mathbf{R} \mathbf{R}^T = \mathbf{I} \end{aligned} \quad (20)$$

Clearly, this is a *bilinear* program with two matrix variables ($\tilde{\mathbf{W}}$ and \mathbf{R}), therefore we can also adopt the *alternative optimization* scheme to solve for a local optimum of this problem.

Step 1. Fix $\mathbf{R} = \mathbf{R}^*$, solve $\tilde{\mathbf{W}}$. Given a specific \mathbf{R}^* , problem (23) becomes

$$\begin{aligned} \min_{\tilde{\mathbf{W}}, \mathbf{R}^*} \quad & \varepsilon_1 = \|\tilde{\mathbf{W}} - \mathbf{W}^* \mathbf{R}^*\|_F^2 \\ \text{s.t.} \quad & \tilde{\mathbf{W}}_{ij} \in \{0, 1\}, \\ & \tilde{\mathbf{W}}^T \mathbf{1}_M = \mathbf{1} \end{aligned} \quad (21)$$

Let $\hat{\mathbf{W}} = \mathbf{W}^* \mathbf{R}^*$, then the optimal solution to problem (21) is just given by

$$\tilde{\mathbf{W}}_{ij}^* = \langle i = \arg \max_u \bar{\mathbf{W}}_{uv} \rangle \quad (22)$$

where $\langle e \rangle = 1$ if e is *true*, and $\langle e \rangle = 0$ if e is *false*.

Step 2. Fix $\tilde{\mathbf{W}} = \tilde{\mathbf{W}}^*$, solve \mathbf{R} . Given a specific $\tilde{\mathbf{W}}^*$, problem (23) becomes

$$\begin{aligned} \min_{\tilde{\mathbf{W}}^*, \mathbf{R}} \quad & \varepsilon_2 = \|\tilde{\mathbf{W}}^* - \mathbf{W}^* \mathbf{R}\|_F^2 \\ \text{s.t.} \quad & \mathbf{R}_{ij} \geq 0, \\ & \mathbf{R} \mathbf{R}^T = \mathbf{I}. \end{aligned} \quad (23)$$

This problem has been well studied in [6], and the elements in \mathbf{R} can be updated by

$$\mathbf{R}_{ij} \leftarrow \mathbf{R}_{ij} \frac{(\mathbf{W}^{*T} \tilde{\mathbf{W}}^*)_{ij}}{(\mathbf{W}^{*T} \tilde{\mathbf{W}}^* \mathbf{R}^T \mathbf{R})_{ij}} \quad (24)$$

It can be easily shown that if we keep on alternating the above two steps, then the loss function ε will decrease monotonically so that the difference between $\tilde{\mathbf{W}}$ and $\mathbf{W}^* \mathbf{R}$ will become smaller and smaller. Finally we can get the optimal $\tilde{\mathbf{W}}^*$ and \mathbf{R}^* when such an alternative procedure converges. The basic algorithm of discretizing \mathbf{W} is summarized in table 2.

Table 2. Discretization

Input: continuous nonnegative matrix \mathbf{W}^* , convergence threshold θ .

Output: discretized 0-1 matrix $\tilde{\mathbf{W}}$.

1. Let $\theta_{old} = 10^4$, $\theta_{new} = 0$, $\Delta\theta = |\theta_{new} - \theta_{old}|$;
2. Let $\omega_{old} = 10^4$, $\omega_{new} = 0$, $\Delta\omega = |\omega_{new} - \omega_{old}|$;
2. Initialize \mathbf{R} to be a $K \times K$ identity matrix;
3. While $\Delta\theta > \theta$
 - (a). Solve $\tilde{\mathbf{W}}^*$ using Eq.(22);
 - (b). While $\Delta\omega > \theta$
 - (i). Update \mathbf{R} using Eq.(24);
 - (ii). Compute $\omega_{new} = \|\tilde{\mathbf{W}}^* - \mathbf{W}^* \mathbf{R}\|_F^2$;
 - (iii). Compute $\Delta\omega = |\omega_{new} - \omega_{old}|$;
 - (c). Compute $\theta_{new} = \|\tilde{\mathbf{W}} - \mathbf{W}^* \mathbf{R}^*\|_F^2$;
 - (d). Compute $\Delta\theta = |\theta_{new} - \theta_{old}|$;
4. Output $\tilde{\mathbf{W}}^*$.

3 Acceleration via Kmeans Preclustering

Till now we have introduced our basic algorithm on selecting representative genes. However, in typical *microarray data analysis* problems each sample is usually represented by a large number of genes, *i.e.* M is very large. This may cause our algorithm very time consuming (*e.g.* we need to compute the $m \times M$ gene similarity matrix \mathbf{K} , $M \times K$ gene significance matrix \mathbf{W}) and inefficient in practical problems. In the following, we will introduce a *kmeans preclustering* method to accelerate our algorithm and we also prove theoretically the optimality of it.

As its name suggests, *kmeans preclustering* approach is just to first clustering the gene set into C clusters $\{\pi_i\}_{i=1}^C$ using the *kmeans* algorithm, and then select a representative gene \mathbf{r}_i for each cluster π_i , finally we can apply our *matrix factorization* algorithm to select K representative genes from $\mathcal{R} = \{\mathbf{r}_i\}_{i=1}^C$. Although this approach seems intuitive and heuristic, we can prove the optimality of this approach in the following from the feature similarity matrix approximation perspective. First we will introduce the definition of *duplicate data set (DDS)*.

Definition (Duplicate Data Set DDS). Denote the data set as $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^M$, which has been clustered into C clusters $\{\pi_i\}_{i=1}^C$, and assume we have constructed a representative \mathbf{r}_i for cluster π_i , ($1 \leq i \leq C$). Then the duplicate data set for \mathcal{X} is a just data set with all the data points in the same cluster replaced by the same corresponding cluster representative.

Let $\tilde{\mathcal{X}}$ be the DDS of the feature set \mathcal{X} , and \mathcal{R} be the cluster representative set of \mathcal{X} , then it would be equivalent to select genes on $\tilde{\mathcal{X}}$ and on \mathcal{R} , since there are unique genes

of $\bar{\mathcal{X}}$ are the same as the genes in \mathcal{R} .

Let's return to our matrix factorization algorithm. The only thing that our algorithm needs to precompute is just the gene similarity matrix \mathbf{K} (the matrix \mathbf{W} and Δ is first randomly initialized and then updated successively). Therefore, if we want to select genes from the duplicate data set $\bar{\mathbf{X}}$, then we should select the cluster representatives such that the resultant feature similarity matrix $\bar{\mathbf{K}}$ constructed on $\bar{\mathcal{X}}$ can have a good approximation to \mathbf{K} . Consider the special structure of $\bar{\mathcal{X}}$, after careful re-ordering (e.g., the data in the same cluster are indexed successively), $\bar{\mathbf{K}}$ should have the following form

$$\bar{\mathbf{K}} = \begin{bmatrix} a & a & a & b & b & \cdots & c & c \\ a & a & a & b & b & \cdots & c & c \\ a & a & a & b & b & \cdots & c & c \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ e & e & e & f & f & \cdots & g & g \end{bmatrix}, \quad (25)$$

which is *block-constant*, since the similarities of the data in the same cluster π_i are all $\mathbf{r}_i^T \mathbf{r}_i$, ($1 \leq i \leq C$), and the data similarities between cluster π_i and π_j are all $\mathbf{r}_i^T \mathbf{r}_j$, ($1 \leq i \leq C$, $1 \leq j \leq C$). Then we have the following theorem.

Theorem 3. *Assume the feature set \mathcal{X} has already been partitioned into C clusters $\{\pi_1, \dots, \pi_C\}$, then the representative point \mathbf{r}_i for each π_i satisfying*

$$\mathbf{r}_i = \frac{1}{n_i} \left(\sum_{\mathbf{x}_j \in \pi_i} \mathbf{x}_j \right), \quad (26)$$

where $n_i = |\pi_i|$ is the cardinality of π_i , can make $\eta = \|\bar{\mathbf{K}} - \mathbf{K}\|_F^2$ minimum.

Proof. Since

$$\begin{aligned} \eta &= \|\bar{\mathbf{K}} - \mathbf{K}\|_F^2 = \sum_{i,j} (\mathbf{K}_{ij} - \bar{\mathbf{K}}_{ij})^2 \\ &= \sum_{p,q} \sum_{\mathbf{x}_i \in \pi_p, \mathbf{x}_j \in \pi_q} (\mathbf{K}_{ij} - S_{pq})^2, \end{aligned} \quad (27)$$

where S_{pq} represents the similarity between π_p and π_q (i.e. $\mathbf{r}_p^T \mathbf{r}_q$). Then letting $\partial J / \partial S_{pq} = 0$, we can get

$$\begin{aligned} S_{pq} &= \frac{1}{n_p n_q} \sum_{\mathbf{x}_i \in \pi_p, \mathbf{x}_j \in \pi_q} \mathbf{K}_{ij} \\ &= \left(\frac{1}{n_p} \sum_{\mathbf{x}_i \in \pi_p} \mathbf{x}_i \right)^T \left(\frac{1}{n_q} \sum_{\mathbf{x}_j \in \pi_q} \mathbf{x}_j \right) \end{aligned}$$

Since $S_{pq} = \mathbf{r}_p^T \mathbf{r}_q$, then $\mathbf{r}_i = \frac{1}{n_i} \left(\sum_{\mathbf{x}_j \in \pi_i} \mathbf{x}_j \right)$ can make J minimum. \square

Theorem 3 tells us that the cluster means are the best representatives from the feature similarity matrix approximation perspective. However, the mean of a gene cluster is just a linear combination of all the genes in that cluster,

which is meaningless in real cases. Therefore, we select the gene in each cluster that is closest to the cluster mean as the representative of that cluster, which is usually referred to as the *cluster medoid*.

Now the only remaining problem is how to partition the data set into clusters. We have the following theorem.

Theorem 4. *The optimal partition for minimizing $\eta = \|\bar{\mathbf{K}} - \mathbf{K}\|_F^2$ can be achieved by k -means clustering.*

Proof. Based on theorem 3, we can expand $J = \|\mathbf{W} - \mathbf{W}'\|_F^2$ as follows.

$$\begin{aligned} J &= \|\bar{\mathbf{K}} - \mathbf{K}\|_F^2 = \sum_{p,q} \sum_{\mathbf{x}_i \in \pi_p, \mathbf{x}_j \in \pi_q} (\mathbf{K}_{ij} - S_{pq})^2 \\ &= \sum_{p,q} \sum_{\mathbf{x}_i \in \pi_p, \mathbf{x}_j \in \pi_q} (\mathbf{x}_i^T \mathbf{x}_j - \mathbf{c}_p^T \mathbf{c}_q)^2 \\ &= \|\mathbf{X}^T \mathbf{X} - (\mathbf{XPP}^T)^T (\mathbf{XPP}^T)\|_F^2, \end{aligned}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M]$ is the feature matrix, and \mathbf{P} is an $M \times C$ scaled cluster indication matrix such that $\mathbf{P}_{ij} = 1/\sqrt{n_j}$ if $\mathbf{x}_i \in \pi_j$, and $\mathbf{P}_{ij} = 0$ otherwise, $\mathbf{c}_p, \mathbf{c}_q$ are the *cluster means* of π_p, π_q . Using the fact that $\mathbf{P}^T \mathbf{P} = \mathbf{I}$ and $\text{trace}(\mathbf{A}^T \mathbf{A}) = \|\mathbf{A}\|_F^2$, we can derive that

$$J = \text{trace}((\mathbf{X}^T \mathbf{X})^T \mathbf{X}^T \mathbf{X} - (\mathbf{P}^T \mathbf{X}^T \mathbf{X} \mathbf{P})^T (\mathbf{P}^T \mathbf{X}^T \mathbf{X} \mathbf{P})).$$

Therefore, the minimization of J is just equivalent to the maximization of $J' = \text{trace}(\mathbf{P}^T \mathbf{X}^T \mathbf{X} \mathbf{P})$. According to the analysis in [19], this criterion is equivalent to the relaxed criterion of k -means. Therefore kernel k -means is an optimal choice for partitioning the data set in the sense of minimizing J . \square

Therefore, combining theorem 3 and theorem 4, we can draw the conclusion that k -means clustering together with the cluster medoids are the optimal choices for constructing the duplicate data set $\bar{\mathcal{X}}$ if we want to select genes from $\bar{\mathcal{X}}$. As we have discussed before, selecting genes from $\bar{\mathcal{X}}$ is equivalent to select genes from the data set that is composed of the cluster medoids, in such a way the data scale is greatly decreased so that our algorithm can run more efficiently.

4. Experiments

In this section we will present the experimental results of our algorithm on several public data sets. First let's describe the basic information of those data sets.

4.1. The Data Sets

- **ALL.** The ALL data set [18] is a data set that covers six subtypes of acute lymphoblastic leukemia: BCR (15), E2A (27), Hyperdip (64), MLL (20), T (43), and TEL (79), where the numbers in the parentheses are the numbers of samples. The dataset is available at <http://www.stjuderesearch.org/data/ALL1/>.

Table 3. Data Set Information

| Data set | #Samples | #Genes | #Classes |
|----------|----------|--------|----------|
| ALL | 248 | 12558 | 2 |
| GCM | 198 | 16063 | 14 |
| LYM | 62 | 4026 | 3 |
| NCI60 | 60 | 1123 | 9 |
| MLL | 72 | 12582 | 3 |
| HBC | 22 | 3226 | 3 |

- **GCM.** The GCM data set [17] consists of 198 human tumor samples of 15 different types.
- **LYM.** The lymphoma data set is a data set of the three most prevalent adult lymphoid malignancies and available at <http://genome-www.stanford.edu/lymphoma>. The data set was first studied in [1].
- **NCI60.** The NCI60 data set was first studied in [12] cDNA microarrays were used to examine the variation in gene expression among the 60 cell lines from the National Center Institutes anticancer drug screen. The dataset spans nine classes and can be downloaded at <http://genomewww.stanford.edu/nci60/>.
- **MLL.** The MLL-leukemia data set consists of three classes and can be downloaded at http://research.dfci.harvard.edu/korsmeyer/Supp_pubs/Supp_Armstrong_Main.html. The data set was first studied in [2].
- **HBC.** The HBC data set consists of 22 hereditary breast cancer samples and was first studied in [10] The dataset has three classes and can be downloaded at <http://www.columbia.edu/xy56/project.htm>.

The basic information of the above data sets are summarized in table 3.

4.2. Experimental Setup

In our experiments, all observed genes are normalized to have the zero mean and unit variance. We first select genes from the gene data set, and then split each data set into a training set and a testing set with size 4:1. Then the following two classifiers will be trained on the training set and then classifying the testing data:

- *Support Vector Machine* with *RBF* kernel (SVM);
- Nearest Neighbor classifier (NN).

The splitting of training and testing set for each data set will repeat 100 times independently. Finally the average classification accuracy is used as the performance measures, and

the parameters in SVM are tuned to achieve the highest average classification accuracies. All the experiments are performed on a P4 2 GHz machine with 512M memory running MS Windows.

For the feature selection process, besides our matrix factorization approach, we also conduct experiments based on two other methods as following.

- **Kmeans.** We first use kmeans to cluster the gene set into K genes, and then select the cluster medoids as the representative genes of the whole data set.
- **No feature selection.** We directly run SVM on the whole gene data set without gene selection.

For our matrix factorization (MF) method, we first use kmeans to cluster the data set into $\lfloor n/10 \rfloor$ clusters (n is the size of the data set, $\lfloor x \rfloor$ denotes the maximum integer that is not larger than x), and then apply MF to select genes from the cluster medoids.

4.3. Experimental Results

The experimental results are shown in figure 1 and figure 2. In all the figures, the x-axis represents the number of selected genes, and the y-axis represents the average classification accuracies of 100 independent runs. From those figures we can observe that

- Generally using feature selection can produce better results than directly classifying on the original data sets. This is because there exists redundancies and noises in the original data set.
- Feature selection using our matrix factorization method can produce better results compared to using kmeans, especially when the number of selected genes is small.

5. Conclusions

We propose a novel unsupervised gene selection method in this paper. Based on the minimum data information loss principle, our method is able to select the most representative genes via matrix factorization. Moreover, to make our method scale well to the large gene data set, we also derive a kmeans preclustering method to accelerate our approach and we prove theoretically the optimality of such method from the gene similarity matrix approximation perspective.

References

- [1] A. A. Alizadeh *et al.* Distinct Types of Diffuse Large B-Cell Lymphoma Identified by Gene Expression Profiling. *Nature*, 403, 503-511. 2000.

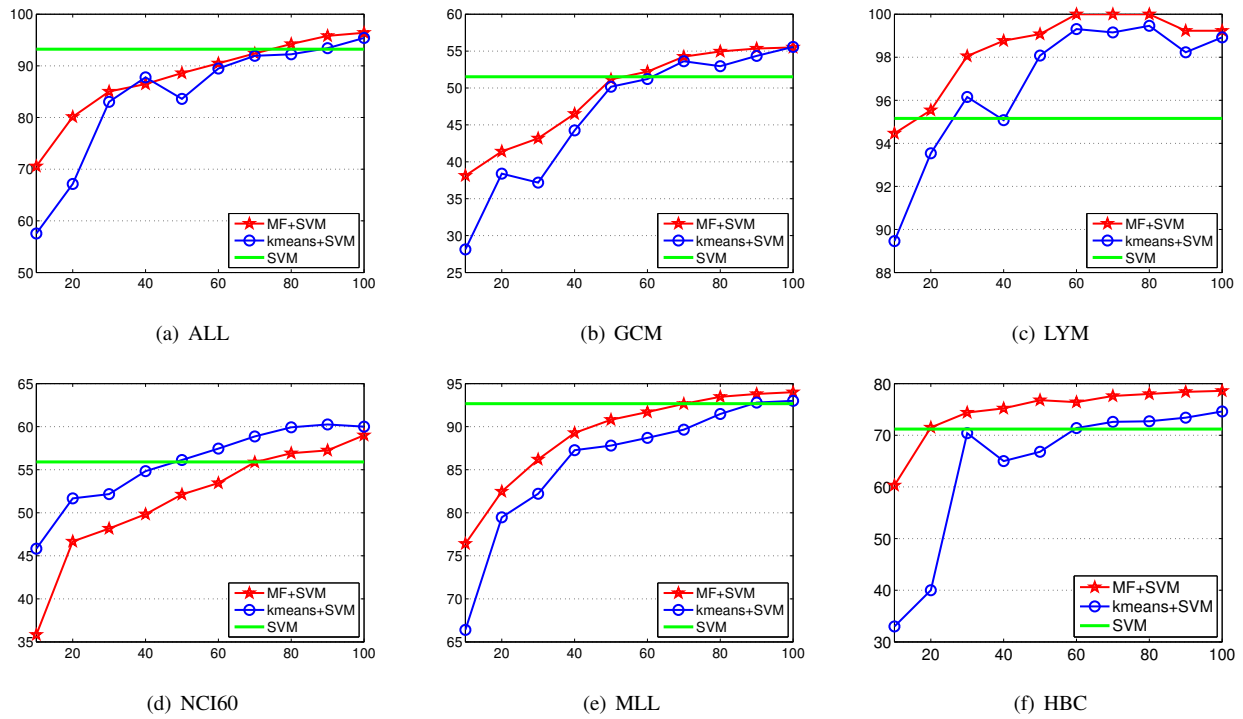


Figure 1. Average classification accuracy results using SVM (%).

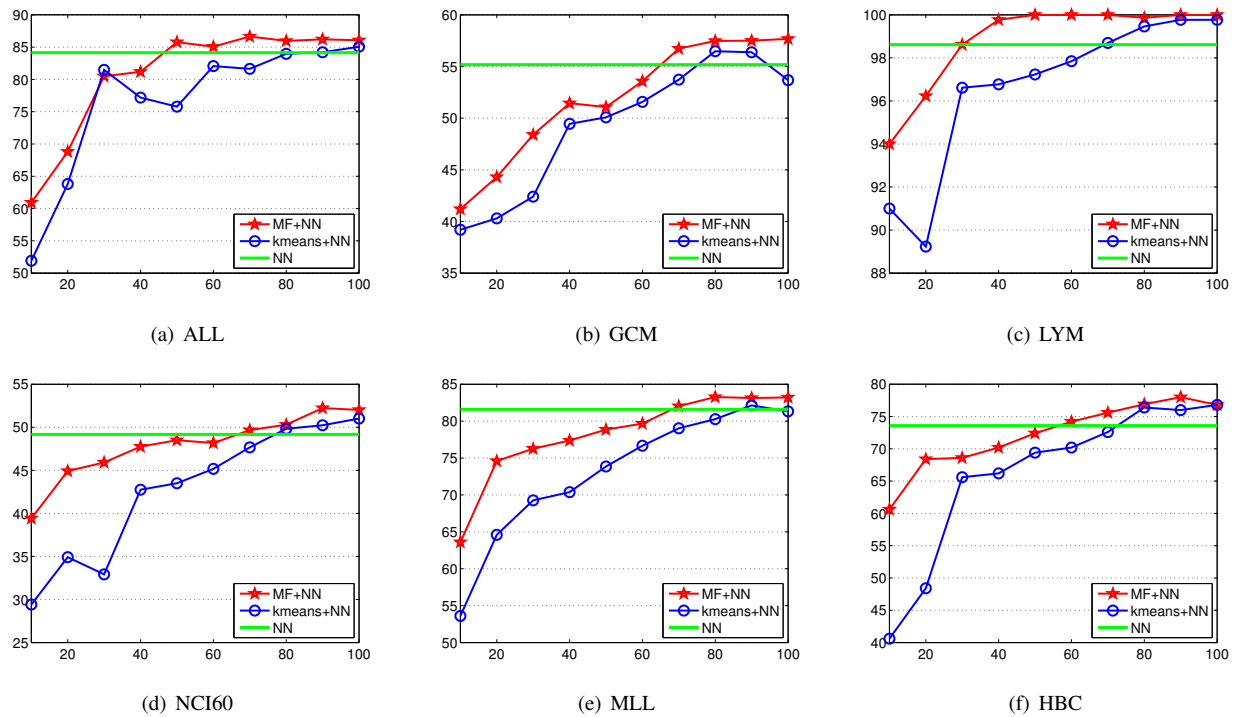


Figure 2. Average classification accuracy results using NN (%).

- [2] 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 and S. J. Korsmeyer. MLL Translocations Specify a Distinct Gene Expression Profile That Distinguishes a Unique Leukemia. *Nat. Genet.*, 30, 41-47. 2002.
- [3] U. Alon *et al.* Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays. *Proc. National Academic Science, USA*, 96, 6745-6750. 1999.
- [4] M. Chee, R. Yang, E. Hubbell, A. Berno, X. C. Huang, D. Stern, J. Winkler, D. J. Lockhardt, M. S. Morris and S. P. Fodor. Accessing Genetic Information with High-Density DNA Arrays. *Science*, 274, 610-614. 1996.
- [5] I. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3:1265-1287, 2003.
- [6] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal Non-negative Matrix Tri-factorizations for Clustering. In *Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, Pages 126-135. 2006.
- [7] C. Ding. Unsupervised Feature Selection via Two-Way Ordering in Gene Expression Analysis. *Bioinformatics*, v.19, 1259-1266, 2003.
- [8] S. P. Fodor, J. L. Read, M. C. Pirrung, L. Stryer, A. T. Lu, and D. Solas. Light-Directed, Spatially Addressable Parallel Chemical Synthesis. *Science*, 251, 767-783. 1991.
- [9] R. Gilad-Bachrach, A. Navot and N. Tishby. Margin based feature selection-theory and algorithms. In *Proc. of the 21st International Conference on Machine Learning (ICML)*, 43-50, 2004.
- [10] I. Hedenfalk, D. Duggan, Y. Chen, M. Radmacher, M. Bittner, R. Simon, P. Meltzer, B. Gusterson, M. Esteller, O. P. Kallioniemi *et al.* Gene-Expression Profiles in Hereditary Breast Cancer. *N. Engl. J. Med.*, 344, 539-548. 2001.
- [11] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield and E. S. Lander. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, 286, 531-537. 1999.
- [12] D. T. Ross, U. Scherf, M. B. Eisen, C. M. Perou, C. Rees, P. Spellman, V. Iyer, S. S. Jeffrey, M. Van de Rijn, M. Waltham *et al.* Systematic variation in gene expression patterns in human cancer cell lines. *Nat. Genet.*, 24, 227-235. 2000.
- [13] F. Sha, Y. Lin, L. K. Saul and D. D. Lee. Multiplicative Updates for Nonnegative Quadratic Programming. *Neural Computation*, to appear. 2007.
- [14] R. Varshavsky, A. Gottlieb, M. Linial, D. Horn. Novel Unsupervised Feature Filtering of Biological Data. *Bioinformatics*, Vol. 22, No. 14, pp. e507-e513(1). 2006.
- [15] E. P. Xing and R. M. Karp. CLIFF: Clustering of Highdimensional Microarray Data via Iterative Feature Filtering Using Normalized cuts. *Bioinformatics*, 17, 306-315. 2001.
- [16] M. Xiong, X. Fang and J. Zhao. Biomarker Identification by Feature Wrappers. *Genome Res.*, 11, 1878-1887. 2001.
- [17] C. H. Yeang, S. Ramaswamy, P. Tamayo, S. Mukherjee, R. M. Rifkin, M. Angelo, M. Reich, E. Lander, J. Mesirov and T. Golub. Molecular Classification of Multiple Tumor Types. *Bioinformatics*, 11, 1C7. 2001.
- [18] E. J. Yeoh, M. E. Ross, S. A. Shurtleff, W. K. Williams, D. Patel, R. Mahrouz, and F. G. Behm. Classification, subtype discovery, and prediction of outcome in pediatric lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1(2):133-143, 2002.
- [19] H. Zha, C. Ding, M. Gu, X. He and H. Simon. Spectral Relaxation for K-means Clustering. *Advances in Neural Information Processing Systems 14 (NIPS)*, 1057-1064, Vancouver, Canada. Dec. 2001.
- [20] H. H. Zhang, J. Ahn, X. Lin, and C. Park. Gene Selection Using Support Vector Machines with Non-Convex Penalty. *Bioinformatics*, 22: 88-95. 2006.
- [21] W. Xu and Y. Gong. Document Clustering by Concept Factorization. In *Proc. of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 202-209. 2004.
- [22] X. Zhu, Z. Ghahramani, J. Lafferty. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. *Proc. of the 20th International Conference on Machine Learning (ICML)*, 912-919. 2003.