Binary matrix factorization for analyzing gene expression data

**Zhong-Yuan Zhang · Tao Li · Chris Ding ·
Xian-Wen Ren · Xiang-Sun Zhang**

Abstract The advent of microarray technology enables us to monitor an entire genome in a single chip using a systematic approach. Clustering, as a widely used data mining approach, has been used to discover phenotypes from the raw expression data. However traditional clustering algorithms have limitations since they can not identify the substructures of samples and features hidden behind the data. Different from clustering, biclustering is a new methodology for discovering genes that are highly related to a subset of samples. Several biclustering models/methods have been presented and used for tumor clinical diagnosis and pathological research. In this paper, we present a new biclustering model using Binary Matrix Factorization (BMF). BMF is a new variant rooted from non-negative matrix factorization (NMF). We begin by proving a new boundedness property of NMF. Two different algorithms to implement the model and their comparison are then presented. We show that the microarray data biclustering problem can be formulated as a BMF problem and can be solved effectively using our proposed algorithms. Unlike the greedy strategy-based algorithms, our proposed

Z.-Y. Zhang
School of Statistics, Central University of Finance and Economics, Beijing,
People's Republic of China

T. Li (✉)
School of Computing and Information Sciences, Florida International University, Miami, FL, USA
e-mail: taoli@cis.fiu.edu

C. Ding
Department of Computer Science and Engineering, University of Texas, Arlington, TX, USA

X.-W. Ren · X.-S. Zhang
Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing,
People's Republic of China

algorithms for BMF are more likely to find the global optima. Experimental results on synthetic and real datasets demonstrate the advantages of BMF over existing biclustering methods. Besides the attractive clustering performance, BMF can generate sparse results (i.e., the number of genes/features involved in each biclustering structure is very small related to the total number of genes/features) that are in accordance with the common practice in molecular biology.

1 Introduction

The DNA arrays, pioneered in Chee et al. (1996), Fodor et al. (1991), are novel technologies that are designed to measure gene expression of tens of thousands of genes in a single experiment. The ability of measuring gene expression for a very large number of genes, covering the entire genome for some small organisms, raises the issue of characterizing cells in terms of gene expression, that is, using gene expression to determine the fate and functions of the cells. The most fundamental problem of the characterization is identifying a set of genes and its expression patterns that either characterize a certain cell state or predict a certain cell state in the future. The first step of studies in that direction is developing tools for classifying/clustering genes/samples according to the gene expression data (Li et al. 2004).

Many traditional clustering methods such as Hierarchical Clustering (HC) and Self-Organizing Mapping (SOM) have been applied for the purpose of clustering microarray data (Eisen et al. 1998; Tamayo et al. 1999). These traditional methodologies have a significant limitation, that is, they assign some samples into some specific classes based on the genes' expression levels across *ALL* the samples. However, in practice, many genes are only active in some conditions or classes and remain silent under other cases. Such gene-class structures, which are very important for understanding the pathology, can not be discovered using the traditional clustering algorithms. So it is necessary to develop clustering methods that can identify the local structures. Moreover, it has been shown in molecular biology that only a small number of genes are involved in a pathway or biological process on most cases. Specifically, only a small subset of genes are active for one tumor type, so generating sparse biclustering structures (i.e., the number of genes in each biclustering structure is small) is of great interest.

Many biclustering algorithms have been proposed recently to explore the correlations between genes and samples and to identify the local gene-sample structures in microarray data (Prelic et al. 2006). The idea of biclustering is to characterize each sample by a subset of genes and to define each gene in a similar way. As a consequence, biclustering algorithms can select the groups of genes that show similar expression behaviors in a subset of samples that belong to some specific classes such as some tumor types, thus identify the local structures of the microarray matrix data (Cheng and Church 2000). Several biclustering methods have been presented in the literature including BiMax (Prelic et al. 2006), ISA (Iterative Signature Algorithm) Ihmels et al.

2004, 2002), SAMBA (Sharan et al. 2003; Tanay et al. 2002, 2004), and OPSM (Order Preserving Submatrix) (Ben-Dor et al. 2002). A systematic comparison and evaluation of these methods has been studied in Prelic et al. (2006).

Recently, Non-negative Matrix Factorization (NMF), as a useful tool for analyzing datasets with non-negativity constraints, has been receiving a lot of attention (Lee and Seung 1999, 2001). Nonnegative matrix factorization (NMF) factorizes an input non-negative matrix into two nonnegative matrices of lower rank. In particular, NMF with the sum of squared error cost function is equivalent to a relaxed K-means clustering, the most widely used unsupervised learning algorithm (Ding et al. 2005). In addition, NMF with the I-divergence cost function is equivalent to probabilistic latent semantic indexing, another unsupervised learning method popularly used in text analysis (Ding et al. 2006b; Gaussier and Goutte 2005). However, NMF can not produce the biclustering structures explicitly. In this paper, we extend standard NMF to Binary Matrix Factorization (BMF) for solving the biclustering problem: the input binary gene-sample matrix X is decomposed into two binary matrices W and H. The binary matrices W and H preserve the most important integer property of the input matrix and also explicitly designate the cluster memberships for genes and samples (Li 2005). As a result, BMF leads to a new biclustering model. The results generated by BMF are not only better than those of other models/methods, but also very sparse. A preliminary version of the paper was appeared in Proceedings of 2007 IEEE International Conference on Data Mining (Zhang et al. 2007). In this journal submission, we added more discussions on theoretical analysis and on biclustering microarray data and also included more experiments. Details on the difference are discussed in Appendix.

The rest of this paper is organized as follows: Sect. 2 introduces the model of Binary Matrix Factorization (BMF); Sect. 3 presents the algorithms for BMF; Sect. 4 compares the performance of the two BMF algorithms using a set of numerical simulations; Sect. 5 discusses the application of BMF for biclustering microarray data; Sect. 6 shows the experimental results on artificial and real datasets; and finally Sect. 7 concludes.

2 Methods

2.1 Non-negative matrix factorization

Non-negative Matrix Factorization (NMF) is one type of the methods that focus on the analysis of non-negative data matrices which are often originated from text, images and biology. Mathematically, NMF can be described as follows: Given an $n \times m$ matrix X composed of non-negative elements, the task is to factorize X into a non-negative matrix W of size $n \times r$ and another non-negative matrix H of size $r \times m$ such that $X \approx WH$. It is usually written down as an optimization

$$\min_{W \geq 0, H \geq 0} ||X - WH||_F^2 = \min_{W \geq 0, H \geq 0} \sum_{ij} (X - WH)_{ij}^2$$

where r is preassigned and should satisfy $r < nm/(n + m)$.

In general, the derived algorithm of NMF is as follows:

- Randomize W and H with positive numbers in [0, 1]. Select the cost function to be minimized.
- Fixing W, update H, then update W for the updated H and so on until the process converges.

For example, if the conventional non-negative least squares $||X - WH||_F^2$ are selected as cost functions, the corresponding update rules of W and H should be:

$$H_{\mathrm{au}} := H_{\mathrm{au}} \frac{(W^T V)_{\mathrm{au}}}{(W^T W H)_{\mathrm{au}}}, \tag{1}$$

$$W_{\mathrm{ia}} := W_{\mathrm{ia}} \frac{(V H^T)_{\mathrm{ia}}}{(W H H^T)_{\mathrm{ia}}}. \tag{2}$$

Thanks to the data mining ability, NMF has attracted a lot of recent attention and has been used in a variety of fields, including environmetrics (Paatero and Tapper 1994), chemometrics (Xie et al. 1999), pattern recognition (Li et al. 2001), multimedia data analysis (Cooper and Foote 2002), text mining (Pauca et al. 2004; Xu et al. 2003) and DNA gene expression analysis (Berry et al. 2007). Algorithmic extensions of NMF have been developed to accommodate a variety of objective functions (Dhillon and Sra 2005) and a variety of data analysis problems, including classification (Sha et al. 2003) and collaborative filtering (Srebro et al. 2005).

Computationally, NMF can be solved using several methods. For the sum of squares cost function, the traditional *alternative nonnegative least squares* method can be used to solve the problem. The multiplicative updating algorithm of Lee and Seung (1999, 2001) is a simple and effective algorithm. A number of studies have focused on further developing computational methodologies for NMF (Berry et al. 2007; Hoyer 2004). In addition, various extensions and variations of NMF and the complexity proof that NMF is NP-hard have been proposed recently (Berry et al. 2007; Ding et al. 2006a,b; la Torre and Kanade 2006; Sha et al. 2003; Zeimpekis and Gallopoulos 2005; Vavasis 2007).

2.2 Binary matrix factorization

Although NMF has shown its power in many applications, it can not discover the biclustering structures explicitly. In this paper, we extend the standard NMF to Binary Matrix Factorization (BMF), that is, elements of X are either 1 or 0, and we want to factorize X into two binary matrices W and H (thus conserving the most important integer property of the objective matrix X) satisfying $X \approx WH$. We will study both the theoretical and the practical aspects of BMF.

Here we give an example to demonstrate the biclustering capability of BMF.[1] Given the original data matrix

---

[1] We will discuss how to discretize the microarray data into a binary matrix in Sect. 5.

$$X = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

One can see two biclusters, one in the upper-right corner, and one in lower-left corner. Our BMF model gives

$$W = (w_1, w_2) = \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix};$$

$$H = (h_1, h_2)^T = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix};$$

The two discovered biclusters are recovered in a clean way:

$$WH = w_1 h_1^T + w_2 h_2^T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

3 Computational algorithms

In this section, we extend the standard NMF to BMF: given a binary matrix X, we want to factorize X into two binary matrices W, H (thus conserving the most important integer property of the objective matrix X) satisfying $X \approx WH$. This is not straightforward and two parallel methodologies (e.g., penalty function algorithm and thresholding algorithm) have been studied and compared. We show that in this section each of these two methods has its own advantages and disadvantages.

3.1 A property of non-negative matrix factorization

We first give a new property of NMF. This property will be useful in computational algorithms. The theorem was first introduced in Zhang et al. (2007). A standard decomposition of matrix is Singular Value Decomposition (SVD): $X = U \Sigma V^T = U'V'$ where $U' = U \Sigma^{1/2}$ and $V' = V \Sigma^{1/2}$ typically contain mixed sign elements. NMF differs from SVD due to the absence of cancellation of plus and minus signs. But what is the fundamental significance of this absence of cancellation? It is the *Boundedness Property*.

Theorem 1 (Boundedness Property) *Let $0 \le X \le 1$ be the input data matrix. W, H are the nonnegative matrices satisfying*

$$X = WH \tag{3}$$

There exists a diagonal matrix $D \geq 0$ such that

$$X = WH = (WD)(D^{-1}H) = W^*H^* \tag{4}$$

with

$$W_{ij}^* \leq 1, \quad H_{ij}^* \leq 1 \tag{5}$$

If X is symmetric and $W = H^T$, then $H^* = H$.

We note that SVD decomposition does not have the boundedness property. In this case, even if the input data are in the range of $0 \leq X_{ij} \leq 1$, we can find some elements of U' and V' such that $U' > 1$ and $V' > 1$. The proof of the theorem is given in Appendix.

We call Eq. 4 the **Normalization Process**. In NMF, there is a scale flexibility, i.e., for any positive D, if WH is a solution, so is $(WD)(D^{-1}H)$. This theorem assures the existence of an appropriate scale such that both W and H are bounded, i.e., their elements can not exceed the magnitude of the input data matrix. This ensures that W, H are in the same scale, which is crucial for the robustness of our proposed penalty function method and the thresholding method.

3.2 Penalty function method

In terms of nonlinear programming, the problem can be represented as:

$$\begin{aligned}
\min \quad & J(W, H) = \sum_{i,j}(X_{ij} - (WH)_{ij})^2 \\
\text{s.t.} \quad & H_{ij}^2 - H_{ij} = 0 \\
& W_{ij}^2 - W_{ij} = 0
\end{aligned}$$

which can be solved by a penalty function algorithm and is programmed as follows:

Algorithm 1: Penalty function method of BMF

Step 1 Initialize λ, W, H and ϵ. Normalize W, H using Eq. (4).

Step 2 For W and H, alternately solve:
$$\min J^* = \sum_{i,j}(X_{ij} - (WH)_{ij})^2 + \tfrac{1}{2}\lambda[(H_{ij}^2 - H_{ij})^2 + (W_{ij}^2 - W_{ij})^2]$$

Step 3 if
$(H_{ij}^2 - H_{ij})^2 + (W_{ij}^2 - W_{ij})^2 < \epsilon$
$$W = \theta(W - 0.5); \ H = \theta(H - 0.5), \ \textbf{break}$$
else $\quad \lambda =: 10\lambda$, return to 2.

---

where the Heaviside step function is defined as

$$\theta(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0, \end{cases}$$

and $\theta(\bullet)$ is element-wise operation: $\theta(\bullet)$ is a matrix whose (i, j)th element is $[\theta(\bullet)]_{ij} = \theta(\bullet_{ij})$. More details on the derivation of the algorithm can be found in Appendix.

3.3 Thresholding method

The second method is thresholding, in other words, finding the best thresholds w, h for W and H respectively so that the minima of the following problem can be achieved:

$$\min F(w, h) = \frac{1}{2} \sum_{i,j} (X_{ij} - (\theta(W - w)\,\theta(H - h))_{ij})^2$$

Initial values of W, H are given via the original NMF algorithm (Lee and Seung 1999). As we can see, $\theta(x)$ is non-smooth, so the problem is a non-smooth optimization problem. There are two implementations to conquer this difficulty.

1. Discretized Method: We discretize the domain $\{(w, h) : 0 \leq w \leq max(W), 0 \leq h \leq max(H)\}$ and try on every grid point to search for optimal thresholds (w^*, h^*).
2. Gradient Decent Method: We approximate the Heaviside function by the function

$$\theta(x) \approx \phi(x) = \frac{1}{1 + e^{-\lambda x}}, \quad \lambda > 0 \quad \text{is a large constant.}$$

Then one can solve the replaced problem under the gradient decent method framework as Algorithm 2. The derivation of the algorithm can be found in Appendix.

Algorithm 2: Thresholding method of BMF

Step 1 Initialize $w_0, h_0, k = 0$. Normalize W and H using Eq. (4).
Step 2 Compute gradient direction g_k of $F(w, h)$. Select stepsize α_k.
Step 3 $w_{k+1} = w_k - \alpha_k g_k, h_{k+1} = h_k - \alpha_k g_k$.
        if some stop strategy is satisfied
                $W =: \theta(W - w_{k+1}); H =: \theta(H - h_{k+1});$ **break**
        else $k = k + 1$, *turn to step 2*

---

4 Comparisons of the two algorithms

In this section, we perform a set of numerical simulations to examine the performance of the two BMF algorithms on the input matrices with different conditions and to demonstrate the effects of normalization. The input matrix X is generated as follows:

Table 1 Errors (in unit of 10^3) for various factorizations

	P	SVD	NMF	BMF-penalty	BMF-threshold	Diff-W	Diff-H
$r = 3$	0.2	1.2361	1.2361	1.6039	3.2744	0.04	0.0519
	0.5	1.9281	1.9287	3.7254	3.6519	0.0218	0.0186
	0.8	1.2391	1.2406	1.6299	1.6054	0.0033	0.0088
$r = 5$	0.2	1.1970	1.1973	1.5923	4.1197	0.0426	0.0321
	0.5	1.8752	1.8784	3.8725	3.6280	0.0133	0.0147
	0.8	1.1889	1.1939	1.5770	1.5690	0.0249	0.0187
$r = 10$	0.2	1.1208	1.1262	1.6025	4.3570	0.0520	0.0704
	0.5	1.7523	1.7736	3.9580	3.5345	0.0362	0.0275
	0.8	1.1203	1.1449	1.5950	1.5850	0.0333	0.0293
$r = 20$	0.2	0.9803	1.0082	1.6099	4.4127	0.0706	0.1258
	0.5	1.5281	1.6086	3.4371	3.3909	0.0142	0.0273
	0.8	0.9799	1.0550	1.5567	1.5420	0.0169	0.0313

Diff-W = root-mean-square difference between the BMF-penalty solution and BMF-threshold solution on W

- Step 1. Randomize X with positive number in [0, 1].
- Step 2. For the element $X(i, j) > P$, $X(i, j) = 1$, otherwise $X(i, j) = 0$, where P is a pre-assigned parameter that controls the sparsity of X.

Table 1 shows the numerical results where the size of the input binary matrix X is 200×400. In Table 1, the density parameter P is selected from {0.2, 0.5, 0.8}. Note that NMF is a restricted form of matrix factorization. To evaluate the performance of NMF, we compare it with SVD using $||X - X^*||^2$ as the evaluation function where $X^* = W_{m,r} H_{r,n}$ via NMF and $X^* = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T$ via SVD. *NMF* refers to the standard NMF algorithm, *BMF-penalty* refers to the penalty function method, and *BMF-threshold* refers to the gradient decent thresholding method. *Diff-W* and *Diff-H* show the difference between the results of penalty method and thresholding method. From Table 1, we observe that when the input matrix X is dense (i.e., P is small), the penalty function algorithm works better than the thresholding algorithm and the thresholding algorithm is better when the input matrix X is sparse.

One useful consequence of Theorem 1 is the normalization of W, H, which eliminates the bias between W and H. This is especially true when the matrix X is sparse. Table 2 demonstrates the effect of normalization using BMF-penalty.[2] P, again, refers to the sparsity parameter selected from {0.2, 0.5, 0.8}. The values in the bracket are the percentages of non-zero elements for the non-normalized case, and the values outside the bracket are the percentages for the normalized case. One can observe, from Table 2, that the normalization process has effectively eliminated the bias between W and H and made the results more robust. Without normalization, the resulting matrix H is often very sparse (sometimes it even becomes zero matrix) while W is very dense.

---

[2] Note that BMF-threshold generates similar results, i.e., Normalization+BMF-threshold can balance the sparsity of W and H too. But because the bias between W and H obtained from BMF-threshold is not so significant compared with those obtained from BMF-penalty, the normalization process has more significant effects on BMF-penalty.

Table 2 Comparison of the normalized case and non-normalized case (in brackets)

	Sparsity (P)	W (%)	H(%)
$r = 3$	0.2	33.3 (33.3)	71.8 (64.6)
	0.5	46 (77.8)	29.2 (9.8)
	0.8	6.7 (67)	11.3 (0)
$r = 5$	0.2	40 (78.8)	46.8 (20)
	0.5	11.6 (58.2)	37.4 (0)
	0.8	27.1 (46.1)	10 (0)
$r = 10$	0.2	29 (54.9)	29.4 (0.1)
	0.5	10 (45.9)	17.2 (0.03)
	0.8	17.8 (36.5)	7.2 (0)
$r = 20$	0.2	13.1 (45.4)	22.3 (0)
	0.5	7.5 (35.9)	12.7 (0)
	0.8	5.2 (26.5)	10.6 (0)

Shown are percentages of nonzero elements

As a result, much information that should be given via H is lost and this can not be compensated by the resulting dense matrix W.

5 BMF for biclustering of microarray data

5.1 Microarray data

The data obtained from microarray experiments can be represented as a matrix X of $n \times m$, the ith row of which represents the ith gene's expression level across the m different samples. The meaning of the resulting W and H from NMF can be explained as follows: each column of W is a metagene (in fact, metagenes are the linear combination of the measured genes.). Each entry of W denotes the weight of the corresponding gene in the metagene. Each row of H can be viewed as the expression level of the metagene across different samples.

5.2 Discretization

The matrix X is first discretized to a binary matrix in the preprocessing step. In the resulting matrix, the element a_{ij} denotes whether the gene i is active in the sample j or not. So this step helps us understand the essential property of the data. In other words, we can confirm which gene responds in the case samples while remains silence in the control ones. There are several reasons for discretization.

First, the discretization method does not necessarily result in information loss, and sometimes it can even improve the clustering performance. We take lung cancer data (see description in Sect. 6.4.1) as an example, the error rate of NMF applied to the original microarray matrix is 2/32 while that of BMF is 0/32. The reason is that the microarray data's signal to noise ratio is low, the discretization method can effectively reduce the effect of noise which is in accordance with the result of artificial data.

Second, one should note that the biclustering structure is data-dependent. That is, as to the different types of tumors, the active genes' expression modes are different, some are up-regulated while others are down-regulated. The discretization method is data-dependent. It can not only identify which genes are active, but also can reveal the expression mode (up-regulated or down-regulated) of the active genes. This can be seen from the detailed discretization process (Prelic et al. 2006): first, we set two thresholds X, Y such as $X = 2$ cutoff and $Y = 0.5$ cutoff, then we regard the gene whose expression value is strictly above $X \cdot M$ or strictly below $Y \cdot M$ as active, non-active otherwise where M is the gene's average expression level across all the samples (Prelic et al. 2006). If necessary, we normalize the microarray data before discretization such that all the columns (samples) have the same L_2 norm. We will describe more details on thresholding and normalization in Sect. 6.4.2. For example, we use 1 cutoff, 8 cutoff (normalized) to discretize the CNS microarray matrix and 1/9 cutoff, 7 cutoff (non-normalized) to the AML/ALL ($k = 3$) data. This indicates that the active genes in CNS are up-regulated while in AML/ALL, some of the active genes are up-regulated and the others are down-regulated. Note that fold changes are not necessarily the best measure of differential expression (Rocke and Durbin 2001; Ideker et al. 2000; Huber et al. 2002). We want to emphasize that the performance of BMF is good enough using this simplest thresholding method. We will investigate more sophisticated thresholding methods in our future work.

Finally, the discretization can guarantee the sparseness of the results of BMF. Mining localized part-based representation can help to reveal low dimensional and more intuitive structures of observations. Although NMF was presented as a "part of whole" factorization method, it has shown that NMF may give holistic representation instead of part-based representation (Li et al. (2001)). Many efforts have been done to improve the sparseness of NMF in order to identify more localized features that are building parts for the whole representation (Lee and Seung 1999). Our experiments in Sect. 6.4.4 show that discretization can dramatically reduce the density of the results of BMF.

5.3 Gene identification

If the resulting matrices W and H are not binary, it is difficult to select the most representative genes since one has no information to impose a threshold to gene coefficients. Using BMF, we can easily select the genes that the corresponding coefficients are 1 while eliminating the rest of the genes. We use the method in references (Carmona-Saez et al. 2006; Brunet et al. 2004) to identify the class number k and the coherent bicluster structures.

6 Experimental results

6.1 Assess standard

We use match score defined as follows as a standard to assess the biclustering performance:

Definition If M_1, M_2 are two biclustering sets, the match score in gene dimension of M_1 with respect to M_2 is:

$$S(M_1, M_2) = \frac{1}{|M_1|} \sum_{(G_1, C_1) \in M_1} \max_{(G_2, C_2) \in M_2} \frac{|G_2 \bigcap G_1|}{|G_2 \bigcup G_1|}$$

This definition was first presented by Prelic et al. (2006). Obviously, the match score in sample dimension can be obtained in a similar way. As we can see, if M_2 matches M_1 exactly, the match score equals to 1. In general, the higher the score, the better M_2 matches M_1.

6.2 Comparison methods

We compare BMF-based algorithms with six other methods, BiMax (Prelic et al. 2006), ISA (Ihmels et al. 2004, 2002), SAMBA (Sharan et al. 2003; Tanay et al. 2002, 2004), Binary Non-orthogonal Matrix Decomposition(BND) (Koyuturk et al. 2006), SNMF/R (Kim and Park 2007) and nsNMF (Carmona-Saez et al. 2006). The first three algorithms have been reported to be the best among the six biclustering methods (Prelic et al. 2006). ISA and Bimax are implemented by the software BicAT developed in Prelic et al. (2006), SAMBA is implemented by EXPANDER Sharan et al. (2003). The related parameter settings, given in Prelic et al. (2006), are listed in Table 3. BND is implemented by PROXIMUS. SNMF/R (Kim and Park 2007) and nsNMF (Carmona-Saez et al. 2006) are two recent NMF-based methods.

6.3 Experiments on artificial data

We use the method described in Prelic et al. (2006) to generate synthetic datasets. Four datasets are generated with different bicluster structures as shown in Fig. 1. The main advantage of using synthetic datasets is that the detailed bicluster structures are known and hence we can evaluate the performance of our BMF methods with different factors such as noise level and overlap degree systematically. In order to perform systematic evaluation with a large number of experiments, the datasets are kept small and they are of size 100×100. Note that the size of the datasets does not restrict the generality of the experimental results as we are focusing on the inherent structures of the input matrix (Prelic et al. 2006).

Figures 2 and 3 present the results on synthetic datasets. The results are obtained by averaging ten trials. The constant bicluster structures are subsets of rows and subset of columns with constant values and the *additive* bicluster structures are subset of rows

Table 3 Parameter settings of different biclustering methods

	Parameter settings
Samba	Default settings in EXPANDER
ISA	$t_g = 2.0$, $t_c = 2.0$, nr.seeds $= 50$

Fig. 1 Bicluster structures in synthetic datasets

Non−overlap case
(constant biclusters)

Overlap case
(constant biclusters)

Non−overlap case
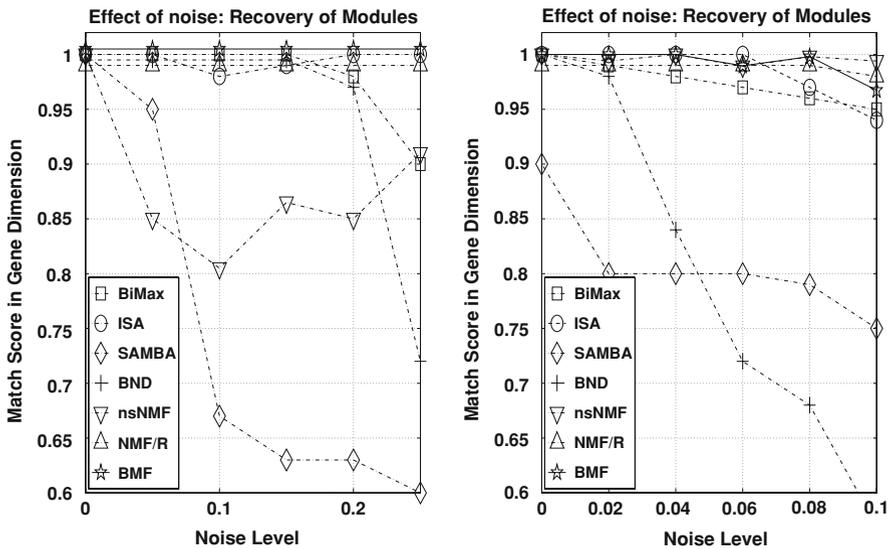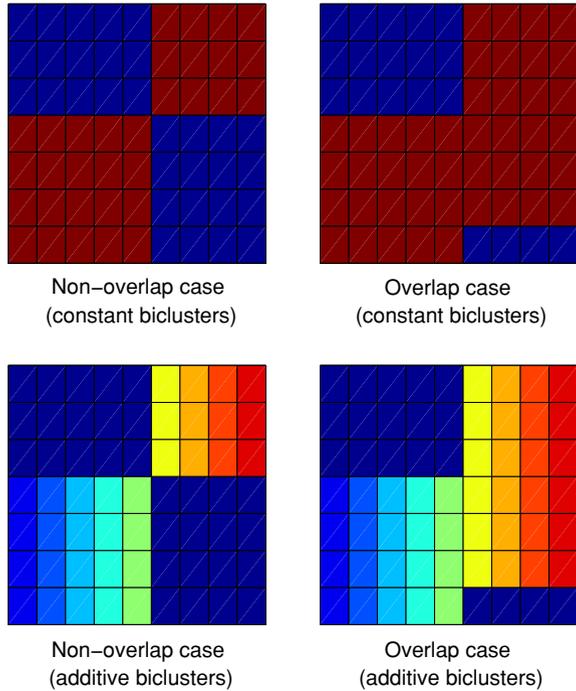(additive biclusters)

Overlap case
(additive biclusters)

Fig. 2 Performance on non-overlap case: We use match score $G(M_{opt}, M_{comp})$ in gene dimension as the standard to assess the performance, where M_{opt} is the implanted biclustering structure, M_{comp} is the computed biclustering structure
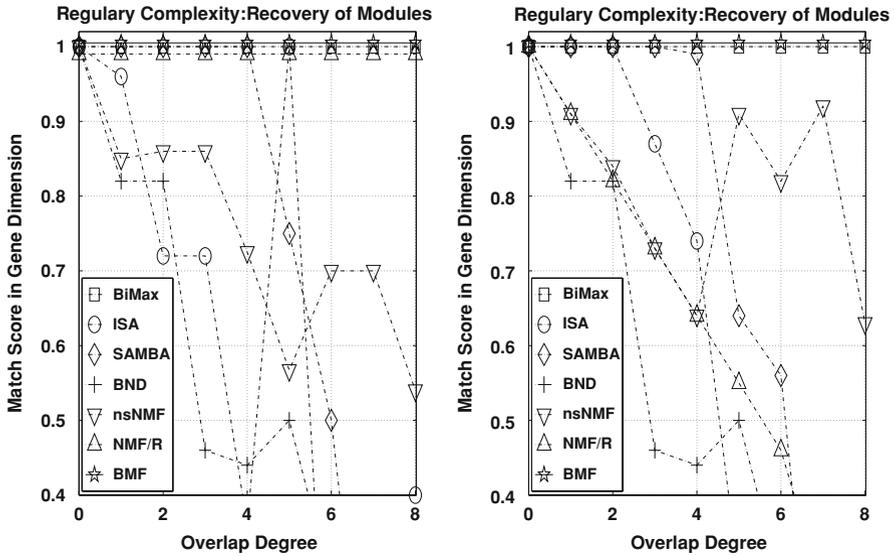
Fig. 3 Performance on overlap case: We use match score $G(M_{opt}, M_{comp})$ in gene dimension as the standard to assess the performance, where M_{opt} is the implanted biclustering structure, M_{comp} is the computed biclustering structure

and subset of columns where each row or column is obtained by adding a constant to another row or column (Madeira et al. 2004). We use match score $G(M_{opt}, M_{comp})$ as the standard to assess the performance, where M_{opt} is the implanted biclustering structure and M_{comp} is the computed biclustering structure. From Figs. 2 and 3, we observe that: (1) the thresholding BMF is almost noise-independent and overlap degree-independent; (2) the thresholding BMF is always the best among the four methods and can nearly identify all the bicluster structures. The main reason is the ability of BMF to correctly discretize original matrix. This is one of the key characteristics of BMF and is very important for identifying the exact bicluster structures. The results also show that, unlike the other greedy search strategy-based algorithms, BMF is more likely to find the global optima.

6.4 Experiments on real data

6.4.1 Dataset description

Real data are also used because artificial data can only be used to test the effect of certain aspects such as noise level and overlap degree of the bicluster problems on different models/methods. We use AML/ALL data (Brunet et al. 2004), lung cancer data (Gordon et al. 2002) and Central Nervous System tumor data (Brunet et al. 2004) to test the performance of BMF. Detailed information can be obtained from the corresponding papers. The basic information is listed in Table 4. All these datasets can be obtained directly from http://sdmc.lit.org.sg/GEDatasets/.

Table 4 Description of real datasets

Datasets	# Samples	# Genes	# Class	# Data source	Data pre-processing (discretization, normalization)
ALL/AML	38	5000	$k = 2$	Brunet et al. (2004)	1/7, 5 non-normalized)
ALL/AML	38	5000	$k = 3$	Brunet et al. (2004)	1/9, 7 (non-normalized)
Lung Cancer	32	5000	$k = 2$	Gordon et al. (2002)	9, 1/5 (non-normalized)
CNS	34	5597	$k = 4$	Brunet et al. (2004)	1, 8 (normalized)

ALL-AML: This dataset, as a golden standard in the cancer classification community, includes two types of human tumor-acute myelogenous leukemia (AML, 11 samples) and acute lymphoblastic leukemia (ALL, 27 samples). Also ALL can be divided into two subtypes-ALL-T(eight samples) and ALL-B(19 samples).

Central Nervous System (CNS): This dataset consists of 34 samples: ten classic medulloblastomas, ten malignant, ten rhabdoids, and four normals.

Lung cancer (LC): This dataset is composed of 32 samples which are about malignant pleural mesothelioma (MPM, 16 samples) and adenocarcinoma (ADCA, 16 samples) of the lung.

6.4.2 Effects of different thresholds

The real data is first discretized to a binary matrix in the preprocessing step. It is interesting to know how variations in the thresholding or normalization affect the biclustering results. In this section, we provide a systematic examination on the effects of data preprocessing. Table 5 presents the clustering results of real data under different pre-process conditions.

We note that the results vary a lot as the conditions change. Also, biclustering structures are data-dependent. In other words, for different types of tumors, the active genes' expression modes are different. So the discretization method and normalization should also be data-dependent. In our experiments, we use the clustering result of the original NMF as a reference to decide the concrete discretization method and normalization. In particular, we select the pre-process criteria (e.g., the cutoff and

Table 5 Shown are the match scores in sample dimension under different conditions

	1/7, 5 N-N	1/7, 7 N-N	1/9, 7 N-N	1/5, 9 N-N	1/5, 7 N-N	1, 8 Normalized	NMF
AML/ALL ($k = 2$)	96.3%	88.43%	87.96%	88.43%	88.43%	42.39%	93.67%
AML/ALL ($k = 3$)	89.55%	91.3%	91.3%	86%	82.2%	21.38%	95.3%
Lung cancer ($k = 2$)	96.88%	96.88%	96.88%	1	1	72.61%	88.19%
CNS ($k = 4$)	41.41%	54%	28.97%	51%	53.04%	82.5%	95.23%

Note that only a small part of the results are given due to space limitation. "N-N" means "Non-normalized", "NMF" column represents the results obtained by applying standard NMF algorithm on the original data

normalization) so that the clustering result of BMF is close to that of standard NMF applied to the original matrix. The information-theoretic measure proposed in Strehl and Ghosh (2003) is used to measure the consistency between two clustering results.

In particular, we regard the clustering result of NMF applied to the original matrix as "benchmark", and then try different pre-process criteria, each producing a binary matrix. Then, we apply BMF on these binary matrices and compare the results with the "benchmark" using some assess standards, such as Normalized Mutual Information (NMI). Finally the best result is selected. NMI is computed as follows:

$$ \text{NMI} = \frac{\sum_{i,j} P(i,j) \log_2 \frac{P(i,j)}{P(i)P(j)}}{\sqrt{\left(\sum_i P(i) \log_2 P(i)\right)\left(\sum_j P(j) \log_2 P(j)\right)}} $$

where $P(i)$ is the probability that the data points belong to cluster i for BMF results, and $P(j)$ is the probability that the data points belong to cluster j for NMF results, $P(i,j)$ is the joint probability that data points belong to cluster i (from BMF results) and j (from NMF results). The larger the NMI, the better the coincidence is.

6.4.3 Results analysis

First, the input microarray matrix is pre-processed for BMF and Bimax as shown in Table 4. Then we use the thresholding BMF (Sect. 3.3) because we observed that the penalty method is more likely falling into the local minima when the objective matrix is sparse.

Table 6 illustrates the results of real data where k is the class number. As one can see, the match score in sample dimension of thresholding BMF is consistently higher than that of Samba. Although the detailed bicluster structures are unknown, we should still regard BMF as a promising model compared with others since the good performance in sample dimension is the preconditions of being an excellent biclustering tool. On the contrary, Samba gives many bicluster structures among which some are obviously meaningless, the best are selected to compute the match score, but this is somewhat unfair to BMF. From above analysis, we can see that BMF is a better model and the

Table 6 The match scores of different biclustering methods

	AML/ALL ($k = 2$)	AML/ALL ($k = 3$)	Lung cancer ($k = 2$)	CNS ($k = 4$)
BMF	96.3%	91.2%	100%	82.5%
Samba	75.7%	75.7%	81.8%	66.1%
nsNMF	88.6%	90%	100%	95.23%
NMF/R	87.46%	90.76%	88.19%	95.23%

We use the match score in sample dimension as the standard. The results of Bimax and ISA are not included for two reasons: (1) the two methods are time-consuming for large dataset; (2) they do not give any biclustering result under the default parameter settings

result of BMF-based algorithms can identify the bicluster structures more exactly and clearly compared with other model-based greedy algorithms.

6.4.4 Sparsity of BMF

To illustrate the sparseness effect of BMF, we compare the results of BMF with those obtained by applying SNMF/R (Kim and Park 2007) and nsNMF (Carmona-Saez et al. 2006) on the original datasets. Table 7 shows the percentages of non-zero elements of W and H obtained respectively on the three datasets. From this table, we can see that our method can dramatically reduce the density of the results. Later analysis will show the biological significance of our results.

6.4.5 Biological analysis

To validate the biological significance of our results, we take ALL/AML dataset as an example to have a detailed case study. Figure 4 is the result obtained from BMF on ALL/AML($k = 3$). Two samples are misclassified (ALL_14749_B-cell is misclassified into AML and ALL_16415_T-cell is not assigned into any class). To illustrate

Table 7 Sparsity of the results obtained from three variations of NMF: BMF, SNMF/R and nsNMF

	AML/ALL ($k = 2$) W (%)/H (%)	AML/ALL ($k = 3$) W (%)/H (%)	Lung cancer ($k = 2$) W (%)/H (%)	CNS ($k = 4$) W (%)/H (%)
BMF	2.6/47.37	2.1/32.46	1.95/50	67.2/25
SNMF/R	61.57/73.68	59.61/55.26	57.41/79.69	84.88/71.32
nsNMF	91/79	90.4/69.3	94.8/76.5	76.7/53.68

The parameter β in SNMF/R is 0.1 and θ in nsNMF is 0.5. Shown are the percentages of non-zero elements of W and H on three datasets
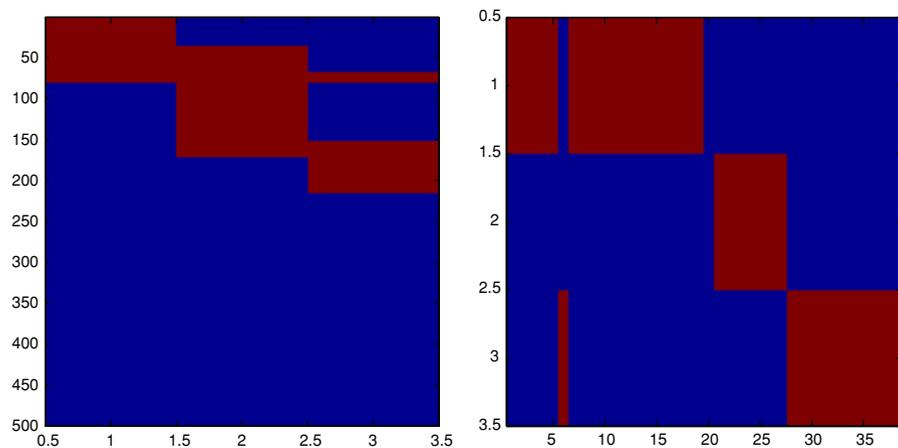

Fig. 4 Results of W and H obtained by BMF on the dataset ALL/AML. For W, only 500 genes are shown. From W and H we can observe the sparse structures of the results

Table 8 Functional enrichment of genes in ALL/AML

Factor	Biological process	Gene number	*P* value
Factor 1: co-expressed in ALL_T (136 genes)	Cell activation	1 (0.78%)	0
	Immune response regulating cell surface receptor signaling pathway	1 (0.78)	0
	Negative regulation of chemokine biosynthetic process	2 (1.56%)	0
	Immune response	20 (15.83%)	0
	Defense response	5 (3.91%)	7E−5
	T-cell costimulation	1 (0.78%)	7.7E−4
	Response to virus	4 (3.13%)	0.0017
	T-cell differentiation	1 (0.78%)	0.00449
	Apoptosis	6 (4.69%)	0.00738
Factor 2: co-expressed in ALL_B (90 genes)	Regulation of T cell differentiation	1 (1.15%)	0
	Immune response	14 (16.09%)	0
	Defense response to bacterium	5 (5.75%)	0
	Immunoglobulin mediated immune response	2 (2.3%)	0
	Inflammatory response	9 (10.34%)	1.0E−5
	Cell surface receptor linked signal transduction	6 (6.9%)	1E−4
	Anti−apoptosis	5 (5.75%)	1.5E−4
	Induction of apoptosis	4 (4.6%)	6.4E−4
	Cell-cell signaling	6 (6.9%)	0.0030
	Response to virus	4 (4.6%)	0.00314
	B-cell activation	2 (2.3%)	0.0036
Factor 3: co-expressed in AML (87 genes)	Cell activation	1 (1.18%)	0
	Regulation of T cell differentiation	1 (1.18%)	0
	Immune response	12 (14.12%)	2E−5
	Regulation of transcription	4 (4.71%)	1.8E−4
	Platelet activation	2 (2.35%)	2.8E−4
	Cell-cell adhesion	3 (3.53%)	4E−4
	Natural killer cell activation	1 (1.18%)	0.00125
	Cell growth	2 (2.35%)	0.00164
	Cell motility	4 (4.71%)	0.0056
	Cell surface receptor linked signal transduction	4 (4.71%)	0.00651
	Anti-apoptosis	3 (3.53%)	0.01

The results are obtained from Onto-Express. *Gene number* here means the number of genes in the corresponding bicluster structures that are included in the corresponding biological process. The numbers in the bracket is the fraction of relevant genes, i.e., (gene number)/(total number of the genes in the corresponding factor)

the biological meaning of our result, we use Onto-Express (Draghici et al. 2003; Khatri et al. 2002) to investigate the enrichment of functional annotations of genes co-expressed in bicluster structures. Onto-Express reads one input file containing a list of GenBank accession numbers of the considered genes and uses another input file containing the full set of genes in the array as reference. The function enrichment results of the three datasets are listed in Tables 8, 9, and 10, respectively. We have analyzed the genes that are active in one and only one type of tumors and provided some significant biological processes for each type of tumors.

Some genes are dominantly co-expressed with single type of leukemia. For instance: EGR-1(Early Growth Response-1, GenBank Accession ID: X52541) is active in ALL_T. It has inhibited function in cancer growth and also has multiple roles in prostate tumor cell growth and survival, cell differentiation, tumor progression, angiogenesis and apoptosis. RUNX3(GenBank Accession ID: Z35278) is active in ALL_B. RUNX3 methylation is associated with gastric cancers and is also related to pathogenesis of testicular yolk sac tumors in infants, lung cancer, hepatocellular carcinogenesis and stomach carcinogenesis. FGFR-1(GenBank Accession ID: X66945) is active in AML and it is expressed in early hernatopoietic precursor cells, as well as in a subpool of endothelial cells in tumor vessels. Some genes are co-expressed in multiple types. For example, CD24 (GenBank Accession ID: L33930) is active in both AML and

Table 9 Functional enrichment Of genes in lung cancer

Factor	Biological process	Gene number	P value
Factor 1: co-expressed in Mesothelioma (101 genes)	Caspase activation	1 (0.99%)	0.02
	Induction of apoptosis via death domain receptors	1 (0.99)	0.02
	Leukotriene biosynthetic process	1 (0.99%)	0.02
	Positive regulation of granulocyte macrophage colony-stimulating factor	1 (0.99%)	0.02
	Positive regulation of interleukin-3 biosynthetic process	1 (0.99%)	0.02
	Positive regulation of mast cell degranulation	1 (0.99%)	0.02
	Positive regulation of type 1 hypersensitivity	1 (0.99%)	0.02
	Immune response	3 (2.97%)	0.02
Factor 2: co-expressed in ADSA (94 genes)	Negative regulation of blood vellel endothelial cell migration	1 (1.06%)	0.017
	Negative regulation of angiogenesis	1 (1.06%)	0.017
	Negative regulation of caspase activity	1 (1.06%)	0.035
	Apoptotic program	1 (1.06%)	0.035
	Lymph node development	1 (1.06%)	0.035
	Skin development	1 (1.06%)	0.035
	Tissue remodeling	1 (1.06%)	0.035
	Response to drug	1 (1.06%)	0.05

Table 10 Functional enrichment of genes in CNS

Factor	Biological process	Gene number	*P* value
Factor 1: co-expressed in classic medulloblastomas (15 genes)	RNA-mediated gene silencing	1 (6.67%)	0.0027
	Sphingolipid metabolic process	1 (6.67%)	0.0027
	Glycosphingolipid metabolic process	1 (6.67%)	0.0027
	Lipid metabolic process	2 (13.33%)	0.017
	Generation of precursor metabolites and energy	1 (6.67%)	0.087
	Nervous systems development	1 (6.67%)	0.225
Factor 2: co-expressed in malignant gliomas (23 genes)	Protein sumoylation	1 (4.35%)	0.0046
	Ribosome assembly	1 (4.35%)	0.01
	Response to UV	1 (4.35%)	0.046
	Response to oxidative stress	1 (4.35%)	0.134
	Induction to apoptosis	1 (4.35%)	0.175
	DNA repair	1 (4.35%)	0.205
	Anti-apoptosis	1 (4.35%)	0.238
Factor 3: co-expressed in rhabdoids (148 genes)	Glutamate signaling pathway	2 (1.35%)	0.0121
	Phosphoinositide phosphorylation	2 (1.35%)	0.025
	Behavior response to ethanol	1 (0.68%)	0.025
	Detection of glucose	1 (0.68%)	0.025
	Detection of light stimulus	1 (0.68%)	0.025
	Generation of ovulation cycle rhythm	1 (0.68%)	0.025
	DNA repair	4 (2.7%)	0.0335

ALL_T. It is reported to have high correlation with invasiveness. The gene is used as an assessment of expression on bone marrow neutrophilic granulocyles: a marker for myelocytic leukemia staging. It is also expressed in ovarian cancer, non-small cell lung cancer, and intrahepatic. PRDx2(GenBank Accession ID: Z22548) is active in both ALL_B and AML. Loss of PRDx2 during tumor development may involve in tumor progression and metastasis cholangiocarcinoma. MS4A1(GenBank Accession ID: X12530) and CD27(GenBank Accession ID: M63928) are co-expressed in all the three types and they are all related to immune response. As we can see from Table 8, a large amount of the selected genes are involved in immune response. This is consistent with the fact that Leukemia is cancer of the blood forming tissues and it often results in producing excessive amounts of blood cells which are unable to work properly and weakening the immune systems.

7 Conclusion

In this paper, we propose BMF to identify the biclustering structures in microarray data. In fact, several papers (Carmona-Saez et al. 2006; Brunet et al. 2004) have discussed about the biclustering aspect of NMF. But the key difficulty is that one can not identify the binary relationships between genes and samples exactly since the resulting matrices W and H are not binary. This can be solved via BMF. In addition, since W

and H are binary, BMF offers a framework for simultaneously clustering the genes and samples. The framework is able to perform implicit feature selection and provide adaptive metrics for biclustering. All of these properties are preferable for clustering in high-dimensional data.

As for future work, we will investigate more sophisticated discretization methods on real data to improve numerical performance. Our current discretization method is time-consuming, especially for data having complex structures. We will develop more efficient strategies for discretizing original microarray data.

Appendix

Summary of difference

A preliminary version of the paper was appeared in Proceedings of 2007 IEEE International Conference on Data Mining. In this journal submission, we added more discussions on theoretical analysis and on biclustering microarray data and added more experiments (Zhang et al. 2007). In particular,

1. The introduction section is re-written to motivate binary matrix factorizations for biclustering microarray data;
2. An example of BMF for biclustering is added in Sect. 2.2;
3. We added more discussions and analysis in Sect. 3.1;
4. Section 5 is added to discuss the issues of the application of BMF for biclustering microarray data;
5. We added the following experiments: (1) sparsity of BMF; (2) biological analysis;
6. More references on biclustering microarray data are added.

Proof of Theorem 1

First of all, rewrite $W = (w_1, w_2, \ldots, w_r)$, $H = (h_1, h_2, \ldots, h_r)^T$. Let

$$D_W = \text{diag}(\max(w_1), \max(w_2), \ldots, \max(w_r))$$
$$D_H = \text{diag}(\max(h_1), \max(h_2), \ldots, \max(h_r))$$

where $\max(w_i)$, $1 \leq i \leq r$ is the largest element of the i-th column of W and $\max(h_j)$, $1 \leq j \leq r$ is the largest element of the jth row of H.

Note

$$D_W = D_W^{1/2} D_W^{1/2}, \quad D_H = D_H^{1/2} D_H^{1/2}.$$
$$D_W^{-1} = D_W^{-1/2} D_W^{-1/2}, \quad D_H^{-1} = D_H^{-1/2} D_H^{-1/2}.$$

We obtain

$$X = WH = (WD_W^{-1})(D_W D_H)(D_H^{-1} H)$$
$$= (WD_W^{-1/2} D_H^{1/2})(D_H^{-1/2} D_W^{1/2} H).$$

Construct D as $D = D_H^{-1/2} D_W^{1/2}$, then

$$W^* = WD_W^{-1/2} D_H^{1/2}, \quad H^* = D_H^{-1/2} D_W^{1/2} H.$$

Thus Eq. (4) is proved.

Furthermore,

$$(WD_W^{-1/2} D_H^{1/2})_{ij} = W_{ij} \cdot \sqrt{\frac{\max(H_j)}{\max(W_j)}}$$
$$= \frac{W_{ij}}{\max(W_j)} \cdot \sqrt{\max(W_j) \max(H_j)}.$$

Without loss of generality, assuming that

$$\max(W_j) = W_{tj}, \quad \max(H_j) = H_{jl},$$

then we have

$$\max(W_j) \cdot \max(H_j) \leq W_{t1} H_{1l} + \cdots W_{tj} H_{jl} + \cdots + W_{tr} H_{rl}$$
$$= \sum_k W_{tk} H_{kl} = X_{tl} \leq 1,$$

So $0 \leq W_{ij}^* \leq 1$ and $0 \leq H_{ij}^* \leq 1$.

If X is symmetric and $W = H^T$,

$$H_{ij}^* = H_{ij} \cdot \sqrt{\frac{\max(H_i)}{\max(H_i)}} = H_{ij}.$$

which implies $H^* = H$.   □

Penalty function method

The derivative of the cost function $J(W, H)$ with respect to H is:

$$\frac{\partial}{\partial H_{au}} J^* = -\sum_i (X_{iu} - (WH)_{iu}) W_{ia} + \lambda((2H_{au} - 1)(H_{au}^2 - H_{au})).$$

Let the step size $\alpha_{au} = H_{au}/(W^T(WH))_{au} + 2\lambda H_{au}^3 + \lambda H_{au})$, then

$$H_{au} = H_{au} - \alpha_{au} \frac{\partial}{\partial H_{au}} J(W, H)$$

$$= H_{au} \left(\frac{(W^T X)_{au} + 3\lambda H_{au}^2}{(W^T W H)_{au} + 2\lambda H_{au}^3 + \lambda H_{au}} \right).$$

By reversing the roles of W and H, one can easily get the update rule of W. Similarly the update formula can be obtained when X is symmetric. The convergence of the algorithm is guaranteed as long as the minima of Step 2 can be achieved.

In Step 3, we can see that if the stop strategy is satisfied, W_{ij} and H_{ij} will be sufficiently close to 0 or 1, then we use Heaviside step function θ to get the binary results.

Thresholding method

In Step 1, w_0, h_0 are given by the optimal solution of discretized method. In Step 2, the gradient direction g_k is:

$$g_k(1) = \partial F(w, h)/\partial w$$

$$= \partial \sum_{a,b} F(w, h)/\partial W_{ab}^* \cdot \partial W_{ab}^*/\partial w$$

$$= \sum_{a,b} ((XH^{*T})_{ab} - (W^* H^* H^{*T})_{ab}) \cdot \frac{e^{-\lambda(W_{ab}-w)} \cdot \lambda}{(1 + e^{-\lambda(W_{ab}-w)})^2},$$

$$g_k(2) = \partial F(w, h)/\partial h$$

$$= \partial \sum_{a,b} F(w, h)/\partial H_{ab}^* \cdot \partial H_{ab}^*/\partial h$$

$$= \sum_{a,b} ((W^{*T} X)_{ab} - (W^{*T} W^* H^*)_{ab}) \cdot \frac{e^{-\lambda(H_{ab}-h)} \cdot \lambda}{(1 + e^{-\lambda(H_{ab}-h)})^2}.$$

where $W^* = \phi(W - w_k)$, $H^* = \phi(H - h_k)$. α_k can be selected by minimizing $F(w_k - \alpha_k g_k(1), h_k - \alpha_k g_k(2))$, but this is time-consuming. In practice, Wolfe line search method can be applied which requires α_k satisfying:

$$F(w_{k+1}, h_{k+1}) - F(w_k, h_k) \leq \delta \alpha_k g_k^T d_k,$$
$$g_{k+1}^T d_k \geq \sigma g_k^T d_k,$$

where $d_k = -g_k$ and δ, σ are constants, $0 < \delta < \sigma < 1$. It can be proved that the stepsize α_k is well-defined in this way, that is, α_k exists as long as $g_k^T d_k < 0$.

An example to illustrate the limitations of NMF for discovering bicluster structures

Suppose that we want to discover the biclustering structures of

$$X = \begin{pmatrix} 0.8 & 0.8 & 0.8 & 0.64 & 0.64 & 0.64 \\ 0.76 & 0.76 & 0.76 & 0.68 & 0.68 & 0.68 \\ 0.64 & 0.64 & 0.64 & 0.8 & 0.8 & 0.8 \\ 0.68 & 0.68 & 0.68 & 0.76 & 0.76 & 0.76 \\ 0.64 & 0.64 & 0.64 & 0.8 & 0.8 & 0.8 \end{pmatrix}.$$

Each row of X is a feature and each column of X is a sample.

We get the factor matrices W and H as follows:

$$\mathbf{W} = \begin{pmatrix} 0.8 & 0.4 \\ 0.7 & 0.5 \\ 0.4 & 0.8 \\ 0.5 & 0.7 \\ 0.4 & 0.8 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} 0.8 & 0.8 & 0.8 & 0.4 & 0.4 & 0.4 \\ 0.4 & 0.4 & 0.4 & 0.8 & 0.8 & 0.8 \end{pmatrix}$$

One can easily observe the clustering structures of the columns from H, but when identifying the biclustering structures, he(or she) has difficulties to identify an appropriate threshold to select which features should be involved in biclustering structures. From this small example we can see that standard NMF has limitations to discovery biclustering structures explicitly.

References

Ben-Dor A, Chor B, Karp R, Yakhini Z (2002) Discovering local structure in gene expression data: the order-preserving submatrix problem. In: RECOMB '02: proceedings of the 6th annual international conference on computational biology. ACM, New York, pp 49–57

Berry M, Browne M, Langville A, Pauca P, Plemmons R (2007) Algorithms and applications for approximate nonnegative matrix factorization. Comput Stat Data Anal 52(1):155–173

Brunet J-P, Tamayo P, Golub TR, Mesirov JP (2004) Metagenes and molecular pattern discovery using matrix factorization. Proc Natl Acad Sci USA 101(12):4164–4169

Carmona-Saez P, Pascual-Marqui RD, Tirado F, Carazo JM, Pascual-Montano A (2006) Biclustering of gene expression data by non-smooth non-negative matrix factorization. BMC Bioinformatics 7(1):78

Chee M, Yang R, Hubbell E, Berno A, Huang X, Stern D, Winkler J, Lockhart D, Morris M, Fodor S (1996) Accessing genetic information with high density DNA arrays. Science 274:610–614

Cheng Y, Church G (2000) Biclustering of expression data. In: Proceedings of the 8th international conference on intelligent systems for molecular biology, pp 93–103

Cooper M, Foote J (2002) Summarizing video using non-negative similarity matrix factorization. In: Proceedings of IEEE workshop on multimedia signal processing, pp 25–28

Dhillon I, Sra S (2005) Generalized nonnegative matrix approximations with Bregman divergences. In: Advances in neural information processing systems, vol 17. MIT Press, Cambridge

Ding C, He X, Simon H (2005) On the equivalence of nonnegative matrix factorization and spectral clustering. In: Proceedings of SIAM data mining conference

Ding C, Li T, Jordan M (2006) Convex and semi-nonnegative matrix factorizations for clustering and low-dimension representation. Technical Report LBNL-60428, Lawrence Berkeley National Laboratory, University of California, Berkeley

Ding C, Li T, Peng W (2006) Nonnegative matrix factorization and probabilistic latent semantic indexing: equivalence, chi-square statistic, and a hybrid method. In: Proceedings of national conference on artificial intelligence (AAAI-06)

Draghici S, Khatri P, Bhavsar P, Shah A, Krawetz SA, Tainsky MA (2003) Onto-tools, the toolkit of the modern biologist: onto-express, onto-compare, onto-design and onto-translate. Nucleic Acids Res 31(13):3775–3781

Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci 95:14863–14868

Fodor S, Read J, Pirrung M, Stryer L, Lu A, Solas D (1991) Light-directed, spatially addressable parallel chemical synthesis. Science 251:767–783

Gaussier E, Goutte C (2005) Relation between plsa and nmf and implications. In: SIGIR '05, pp 601–602

Gordon GJ, Jensen RV, Hsiao L-L, Gullans SR, Blumenstock JE, Ramaswamy S, Richards WG, Sugarbaker DJ, Bueno R (2002) Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. Cancer Res 62:4963–4967

Hoyer PO (2004) Non-negative matrix factorization with sparseness constraints. J Mach Learn Res 5:1457–1469

Huber W et al (2002) Variance stabilization applied to microarray data calibration and to the quantification of differential expression. Bioinformatics 18(Suppl 1):S96–S104

Ideker T et al (2000) Testing for differentially-expressed genes by maximum-likelihood analysis of microarray data. J Comput Biol 7(6):805–817

Ihmels J, Friedlander G, Bergmann S, Sarig O, Ziv Y, Barkai N (2002) Revealing modular organization in the yeast transcriptional network. Nature Genet 31:370–377

Ihmels J, Bergmann S, Barkai N (2004) Defining transcription modules using large-scale gene expression data. Bioinformatics 20(13):1993–2003

Khatri P, Draghici S, Ostermeier G, Krawetz S (2002) Profiling gene expression using onto-express. Genomics 79(2):266–270

Kim H, Park H (2007) Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. Bioinformatics 23(12):1495–1502

Koyuturk M, Grama A, Ramakrishnan N (2006) Non-orthogonal decomposition of binary matrices for bounded-error data compression and analysis. ACM Trans Math Softw 32(1):33–69

la Torre FD, Kanade T (2006) Discriminative cluster analysis. In: Proceedings of the 23rd international conference on machine learning (ICML 2006)

Lee D, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401:788–791

Lee D, Seung HS (2001) Algorithms for non-negative matrix factorization. In: Dietterich TG, Tresp V (eds) Advances in neural information processing systems, vol 13. MIT Press, Cambridge

Li T (2005) A general model for clustering binary data. In: Proceedings of the 11th ACM SIGKDD international conference, pp 188–197

Li S, Hou X, Zhang H, Cheng Q (2001) Learning spatially localized, parts-based representation. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 207–212

Li T, Zhang C, Ogihara M (2004) A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. Bioinformatics 20(15):2429–2437

Madeira SC et al (2004) Biclustering algorithms for biological data analysis: a survey. IEEE Trans Comput Biol Bioinformatics 1:24–45

Paatero P, Tapper U (1994) Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. Environmetrics 5:111–126

Pauca VP, Shahnaz F, Berry M, Plemmons R (2004) Text mining using non-negative matrix factorization. In: Proceedings of SIAM international conference on data mining, pp 452–456

Prelic A, Bleuler S, Zimmermann P, Wille A, Buhlmann P, Gruissem W, Hennig L, Thiele L, Zitzler E (2006) A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22(9):1122–1129

Rocke D, Durbin B (2001) A model for measurement error for gene expression arrays. J Comput Biol 8(6):557–569

Sha F, Saul L, Lee D (2003) Multiplicative updates for nonnegative quadratic programming in support vector machines. In: Advances in neural information processing systems, vol 15, pp 1041–1048

Sharan R, Maron-Katz A, Shamir R (2003) Click and expander: a system for clustering and visualizing gene expression data. Bioinformatics 19(14):1787–1799

Srebro N, Rennie J, Jaakkola T (2005) Maximum margin matrix factorization. In: Advances in neural information processing systems. MIT Press, Cambridge

Strehl A, Ghosh J (2003) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J Mach Learn Res 3:583–617

Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander E, Golub T (1999) Interpreting patterns of gene expression with self-organizing maps. In: Proceedings of the national academy of sciences of USA, vol 96

Tanay A, Sharan R, Shamir R (2002) Discovering statistically significant biclusters in gene expression data. Bioinformatics 18(90001):S136–S144

Tanay A, Sharan R, Kupiec M, Shamir R, Karp RM (2004) Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genome-wide data. Proc Natl Acad Sci USA 101(9):2981–2986

Vavasis SA (2007) On the complexity of nonnegative matrix factorization. http://arxiv.org/abs/0708.4149

Xie Y-L, Hopke P, Paatero P (1999) Positive matrix factorization applied to a curve resolution problem. J Chemom 12(6):357–364

Xu W, Liu X, Gong Y (2003) Document clustering based on non-negative matrix factorization. In: Proceedings of ACM conference on research and development in IR(SIGIR), Toronto, pp 267–273

Zeimpekis D, Gallopoulos E (2005) Clsi: a flexible approximation scheme from clustered term-document matrices. Proceedings of SIAM data mining conference, pp 631–635

Zhang Z, Li T, Ding C, Zhang X (2007) Binary matrix factorization and applications. In: Proceedings of 2007 IEEE international conference on data mining