Sparse Non-negative Matrix Factorizations via Alternating Non-negativity-constrained Least Squares

Hyunsoo Kim and Haesun Park
College of Computing, Georgia Institute of Technology
801 Atlantic Dr., Atlanta, GA 30332, USA
E-mail: hskim@cc.gatech.edu, hpark@cc.gatech.edu

ABSTRACT

Many practical pattern recognition problems require non-negativity constraints. For example, pixels in digital images and chemical concentrations in bioinformatics are non-negative. Non-negative matrix factorization (NMF) is a useful technique in approximating these high dimensional data. Sparse NMFs are also useful when we need to control the degree of sparseness in non-negative basis vectors or non-negative lower-dimensional representations. In this paper, we introduce novel sparse NMFs via alternating non-negativity-constrained least squares. We applied one of the proposed sparse NMFs to cancer class discovery and gene expression data analysis. Our experimental results illustrate that our proposed method achieves better clustering performance than NMF based on multiplicative update rules and sparse NMFs based on the gradient descent method.

1 Introduction

Given a non-negative matrix A of size $m \times n$, where each column of A corresponds to a data point in the m-dimensional space, and a positive integer $k < \min\{m, n\}$, non-negative matrix factorization (NMF) finds two non-negative matrices $W \in \mathbb{R}^{m \times k}$ and $H \in \mathbb{R}^{k \times n}$ so that $A \approx WH$. A solution to the NMF problem can be obtained by solving the following optimization problem:

$$\min_{W,H} f(W,H) \equiv \frac{1}{2}\|A - WH\|_F^2, \;\; s.t. \;\; W, H \geq 0, \tag{1}$$

where $W \in \mathbb{R}^{m \times k}$ is a basis matrix, $H \in \mathbb{R}^{k \times n}$ is a coefficient matrix, $\| \cdot \|_F$ is the Frobenius norm, and $W, H \geq 0$ means that all elements of W and H are non-negative. Due to $k < m$, dimension reduction is achieved and the lower-dimensional representation is given by H. Since NMF may give us direct interpretation due to non-subtractive combinations of non-negative basis vectors, it has recently received much attention and it has been applied to many interesting problems including text data mining [7, 14] and gene expression data analysis [6, 2, 3]. One of the most interesting properties of NMF is that it usually generates sparse basis vectors that allow us to discover parts-based basis vectors. However, the NMF formulation shown in Eq. (1) does not guarantee sparsity in the factors W or H, and the sparsity depends on specific NMF algorithms. For example, NMF generated holistic basis images instead of parts-based basis images for a facial image dataset in the results presented in [9, 5]. Since it would be useful to control the degree of sparseness explicitly for this situation, there have been several approaches [5, 14, 3, 13] to control the degree to which basis vectors are sparse.

In this paper, we introduce alternative sparse NMFs that can explicitly control sparseness in either of the basis matrix W or the reduced dimensional representation H by using alternating non-negativity-constrained least squares. The rest of this paper is organized as follows. We give brief overviews on sparse NMFs based on the gradient descent method and their mathematical difficulties in Section 2, and NMF based on alternating non-negativity-constrained least squares in Section 3. In Section 4, we introduce sparse NMFs via alternating non-negativity-constrained least squares involving L_1-norm based constraints. Section 5 presents experimental results illustrating properties of the proposed sparse NMFs. Summary is given in Section 6.

2 Sparse NMFs based on the Gradient Descent Method

Lee and Seung [7, 8] suggested NMF algorithms based on multiplicative update rules of W and H. The distance $\|A - WH\|_F$ is nonincreasing under the update rules:

$$H_{qj} \leftarrow H_{qj} \frac{(W^T A)_{qj}}{((W^T W)H)_{qj}},$$

for $1 \le q \le k$ and $1 \le j \le n$,

$$W_{iq} \leftarrow W_{iq} \frac{(AH^T)_{iq}}{(W(HH^T))_{iq}},$$

for $1 \le i \le m$ and $1 \le q \le k$. The divergence is nonincreasing under the different updating rules [8]. Gonzales and Zhang [4] pointed out that these nonincreasing properties of multiplicative update rules may not imply the convergence to a stationary point within realistic amount of run time for problems of meaningful sizes.

Hoyer [5] devised a sparse NMF based on the projected gradient descent method (SNMF/PGD) in order to constrain NMF to find solution with desired sparseness of W and H. To impose sparseness constraints on only one matrix W or H, this algorithm uses a multiplicative update rule for the counter matrix, which suffers from slow convergence. More practical difficulties of this algorithm will be discussed in Section 5.2.

Pauca *et al.* [13] proposed a constrained NMF (CNMF) optimization problem,

$$\min_{W,H}\{\|A - WH\|_F^2 + \alpha\|W\|_F^2 + \beta\|H\|_F^2\}, \ \ s.t. \ W, H \geq 0, \tag{2}$$

and suggested the following multiplicative updating rules:

$$H_{qj} \leftarrow H_{qj}\frac{(W^TA)_{qj} - \beta H_{qj}}{((W^TW)H)_{qj}},$$

for $1 \leq q \leq k$ and $1 \leq j \leq n$,

$$W_{iq} \leftarrow W_{iq}\frac{(AH^T)_{iq} - \alpha W_{iq}}{(W(HH^T))_{iq}},$$

for $1 \leq i \leq m$ and $1 \leq q \leq k$, where α and β are regularization parameters (zero or positive real values) that are used to balance the trade-off between the accuracy of approximation and the sparseness of W and H, respectively. However, note that H or W may have negative elements during iterations when we are using a large positive α or a large positive β. When $\alpha = 0$, Eq. (2) can be rewritten as

$$\min_{W,H}\{\|A - WH\|_F^2 + \beta\|H\|_F^2\}, \ \ s.t. \ W, H \geq 0. \tag{3}$$

This formulation contains the minimization of L_2-norm of each column of H in order to increase the sparseness of H. The following least squares formulation without non-negativity-constraints on H,

$$\min_{H}\{\|A - WH\|_F^2 + \beta\|H\|_F^2\}, \tag{4}$$

has appeared in [14, 3]. Any negative values in H obtained from Eq. (4) during iterations were set to zero in [14, 3]. However, setting negative values to zero for imposing non-negativity cannot be recommended for several reasons: first of all, one does not obtain least squares estimates, which means that there is no guarantee for the quality of the model. Another problem with this approximate approach is that when included in a multiway algorithm, it can cause the algorithm to diverge, *i.e.* successive iterations yield models that describe the data progressively more poorly. This can happen because the approximate estimates are not truly least squares [1]. Moreover, L_1-norm based formulations would be more appropriate than L_2-norm based formulations so as to control sparsity [15]. These are our motivations for proposing alternative sparse NMFs based on minimizing L_1-norm of columns of W^T or H via alternating non-negativity-constrained least squares.

3 NMF based on Alternating Non-negativity-constrained Least Squares (NMF/ANLS)

Given $A \in \mathbb{R}^{m \times n}$, NMF based on alternating non-negativity-constrained least squares (NMF/ANLS) starts with an initialization of $H \in \mathbb{R}^{k \times n}$ with non-negative values. Then, it iterates the following two non-negativity-constrained least squares until convergence:

$$\min_{W} \|H^T W^T - A^T\|_F^2, \;\; s.t. \;\; W \geq 0, \tag{5}$$

which fixes H and solves the optimization with respect to W, and

$$\min_{H} \|WH - A\|_F^2, \;\; s.t. \;\; H \geq 0, \tag{6}$$

which fixes W and solves the optimization with respect to H. Similarly, one may initialize $W \in \mathbb{R}^{m \times k}$ and alternate the above in the order of solving Eq. (6) and Eq. (5). Paatero and Tapper [12] originally proposed using the constrained alternating least squares method to solve Eq. (1). We used a fast algorithm for large scale non-negativity-constrained least squares problems [16] to solve Eqs. (5)-(6). Lin [10] discussed the convergence property of alternating non-negativity-constrained least squares and showed that any limit point of the sequence (W,H) generated by alternating non-negativity-constrained least squares is a stationary point of Eq. (1).

4 Sparse NMFs based on Alternating Non-negativity-constrained Least Squares

In order to enforce sparseness constraints on W or H in $A \approx WH$, we propose two sparse NMFs, *i.e.* SNMF/L for sparse W (where 'L' denotes that we control the sparseness of the left side factor) and SNMF/R for sparse H (where 'R' denotes that we control the sparseness of the right side factor). These sparse NMFs are based on alternating non-negativity constrained least squares.

4.1 SNMF/L

To impose sparseness constraints on W, we deal with the following optimization problem:

$$\min_{W,H}\{\|A - WH\|_F^2 + \alpha \sum_{i=1}^{m} \|W(i,:)\|_1^2\}, \;\; s.t. \;\; W, H \geq 0, \tag{7}$$

where $W(i,:)$ is the i-th row vector of W. The regularization parameter α is a real non-negative value to balance the trade-off between accuracy of the approximation and sparseness of W. SNMF/L begins with an initialization of non-negative matrix W. Then, it iterates the following ANLS until convergence:

$$\min_{H} \|WH - A\|_F^2, \;\; s.t. \;\; H \geq 0, \tag{8}$$

$$\min_{W} \left\| \begin{pmatrix} H^T \\ \sqrt{\alpha}\mathbf{e}_{1\times k} \end{pmatrix} W^T - \begin{pmatrix} A^T \\ \mathbf{0}_{1\times m} \end{pmatrix} \right\|_F^2, \;\; s.t. \;\; W \geq 0, \tag{9}$$

where $\mathbf{e}_{1\times k} \in \mathbb{R}^{1\times k}$ is a row vector whose elements are all ones and $\mathbf{0}_{1\times m} \in \mathbb{R}^{1\times m}$ is a zero vector whose elements are all zeros. The rows of the coefficient matrix H are normalized to unit L_2-norm, *i.e.* $\|H(q,:)\|_2 = 1$ for $1 \leq q \leq k$, after Eq. (8) at each iteration so that rows of H have constant energy. Eq. (9) can be simplified as

$$\begin{aligned}\min\nolimits_W \quad & \{\|H^TW^T(:,1)-A^T(:,1)\|_2^2 \\ & +\alpha\left(\textstyle\sum_{q=1}^k W^T(q,1)\right)^2 + \quad \cdots \\ & +\|H^TW^T(:,m)-A^T(:,m)\|_2^2 \\ & +\alpha\left(\textstyle\sum_{q=1}^k W^T(q,m)\right)^2\Bigg\} \quad s.t.\ \ W \geq 0.\end{aligned}$$

Since all elements in W are non-negative, we obtain the following formulation by the definition of L_1-norm of a vector:

$$\begin{aligned}\min\nolimits_W \quad & \{\|H^TW^T(:,1)-A^T(:,1)\|_2^2 \\ & +\alpha\|W^T(:,1)\|_1^2 + \quad \cdots \\ & +\|H^TW^T(:,m)-A^T(:,m)\|_2^2 \\ & +\alpha\|W^T(:,m)\|_1^2\}, \quad s.t.\ \ W \geq 0,\end{aligned}$$

which involves the minimization of L_1-norm of each column of W^T.

4.2 SNMF/R

To apply sparseness constraints on H, we deal with the following optimization problem:

$$\min_{W,H}\{\|A-WH\|_F^2+\beta\sum_{j=1}^n\|H(:,j)\|_1^2\}, \quad s.t.\ \ W,H \geq 0, \tag{10}$$

where $H(:,j)$ is the j-th column vector of H. The regularization parameter β is a real non-negative value to balance the trade-off between accuracy of the approximation and sparseness of H. SNMF/R begins with the initialization of H with non-negative values. Then, it iterates the following ANLS until convergence:

$$\min_W \|H^TW^T-A^T\|_F^2, \quad s.t.\ \ W \geq 0, \tag{11}$$

$$\min_H \left\| \begin{pmatrix} W \\ \sqrt{\beta}\mathbf{e}_{1\times k} \end{pmatrix} H - \begin{pmatrix} A \\ \mathbf{0}_{1\times n} \end{pmatrix} \right\|_F^2, \quad s.t.\ \ H \geq 0, \tag{12}$$

where $\mathbf{e}_{1\times k} \in \mathbb{R}^{1\times k}$ is a row vector with all components equal to one and $\mathbf{0}_{1\times n} \in \mathbb{R}^{1\times n}$ is a null vector whose elements are all zeros. The columns of the basis matrix W are normalized to unit L_2-norm, *i.e.* $\|W(:,q)\|_2 = 1$ for $1 \leq q \leq k$, after Eq. (11) at each iteration so that columns of W have constant energy. Eq. (12) minimizes L_1-norm of columns of $H \in \mathbb{R}^{k\times n}$.

4.3 Stopping Criterion

Once we have a non-negative decomposition ($A \approx WH \;\; s.t. \;\; W, H \geq 0$), we can use the basis matrix W to divide the m genes into k gene-clusters and the coefficient matrix H to divide the n samples into k sample-clusters. Typically, gene i is assigned to gene-cluster q if the $W(i,q)$ is the largest element in $W(i,:)$ and sample j is assigned to sample-cluster q if the $H(q,j)$ is the largest element in $H(:,j)$. We tested convergence at every five iterations by using these positions of the largest elements in rows of W and columns of H. We assumed that NMFs are converged if both the positions of the largest elements in rows of W, *i.e.* $\tilde{\mathbf{w}} = (\tilde{w}_1, \ldots, \tilde{w}_m)$, and the positions of the largest elements in columns of H, *i.e.* $\tilde{\mathbf{h}} = (\tilde{h}_1, \ldots, \tilde{h}_n)$, have not changed during 11 convergence tests, where $\tilde{w}_i$ is the position of the largest element in the i-th row of W and $\tilde{h}_j$ is the positions of the largest element in the j-th column of H. Brunet *et. al.* [2] used a connectivity matrix $\hat{C} = [\hat{c}_{ij}]$ of size $n \times n$ for convergence tests, whose entry is $\hat{c}_{ij} = 1$ if samples i and j belong to the same sample-cluster, and $\hat{c}_{ij} = 0$ if they belong to different sample-clusters. However, this convergence criterion does not include the change of W. Considering W is also important since $\tilde{\mathbf{w}}$ can change even if $\tilde{\mathbf{h}}$ has not changed for many iterations. Thus, we took account of the convergence of $\tilde{\mathbf{w}}$ as well as the convergence of $\tilde{\mathbf{h}}$. Our stopping criterion is suitable for biclustering obtained from NMF.

5 Experiments and Discussion

5.1 Datasets Description

We used the leukemia gene expression dataset (ALLAML) and the central nervous system tumors dataset (CNS) [2]. The ALLAML dataset contains acute lymphoblastic leukemia (ALL) that has B and T cell subtypes, and acute myelogenous leukemia (AML) that occurs more commonly in adults than in children. This gene expression dataset consists of 38 bone marrow samples (19 ALL-B, 8 ALL-T, and 11 AML) with 5,000 genes. The central nervous system dataset is composed of four categories of CNS tumors with 5,597 genes. It consists of 34 samples representing four distinct morphologies: 10 classic medulloblastomas, 10 malignant gliomas, 10 rhabdoids, and 4 normals. All datasets we used contain only nonnegative entries. We implemented algorithms in Matlab 6.5 [11]. All our experiments were performed on a P3 600MHz machine with 512MB memory.

5.2 Properties of Sparse NMFs

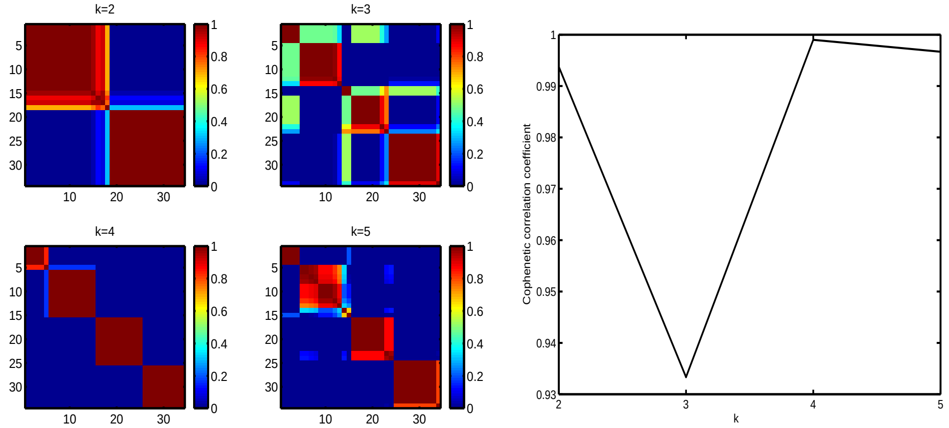
To measure the clustering performance, we used purity and entropy. Suppose we are given l categories (true class labels), while NMF generates k clusters. Purity is given by

$$\text{Purity} = \sum_{q=1}^{k} \frac{n_q}{n} P(\tilde{\Omega}_q), \quad P(\tilde{\Omega}_q) = \frac{1}{n_q} \max_j (n_q^j),$$

where $\tilde{\Omega}_q$ is a particular cluster of size n_q, n_q^j is the number of samples in $\tilde{\Omega}_q$ that belong to original class Ω_j ($1 \leq \Omega_j \leq l$), k is the number of clusters, and n is the total number of samples. The larger values of purity, the better clustering performance. Entropy is defined as follows:

$$\text{Entropy} = \sum_{q=1}^{k} \frac{n_q}{n} E(\tilde{\Omega}_q),$$

Figure 1: CNS tumors clustering by NMF based on divergence-based update rules. (Left) The reordered consensus matrices on the CNS tumors dataset. (Right) The corresponding Cophenetic correlation coefficients.

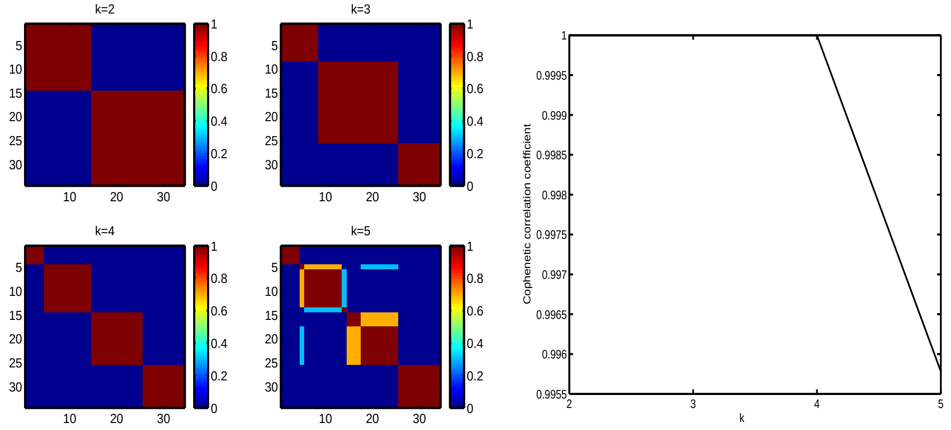


$$E(\tilde{\Omega}_q) = -\frac{1}{\log_2 l}\sum_{j=1}^{l}\frac{n_q^j}{n_q}\log_2\frac{n_q^j}{n_q},$$

where l denotes the number of original class labels. The smaller values of entropy, the better clustering quality.

Tables 1 and 2 show the results of SNMF/L and SNMF/R under various parameters of $\alpha \in \{0.001, 0.01, 0.1, 1.0\}$ and $\beta \in \{0.001, 0.01, 0.1, 1.0\}$ on the ALLAML dataset with $k = 3$ and on the CNS tumors dataset with $k = 4$, respectively. We compared sparse NMFs with a NMF algorithm based on divergence-based multiplicative update rules [8, 2]. The averages of sparseness, purity and entropy were computed by repeating NMFs five times with different random initializations. By increasing α, we could enhance the sparsity of W, while reducing sparsity of H. By increasing β, we could achieve a sparser H, while diminishing the sparsity of W. SNMF/R produced better clustering performance (higher purity, lower entropy) than NMF based on multiplicative update rules. On the other hand,

Figure 2: CNS tumors clustering by SNMF/R. (Left) The reordered consensus matrices on the CNS tumors dataset. (Right) The corresponding Cophenetic correlation coefficients. The correlation coefficient drops when k increases from 4 to 5, indicating a four-cluster split of the data is more stable than a five-cluster split.



SNMF/L can be applied to obtain parts-based basis vectors. NMF based on multiplicative updating rules generated holistic basis images for a facial image dataset [9, 5], while SNMF/L could yield parts-based basis images since it could control the degree to which basis vectors are sparse (Result parts-based basis images are not shown here due to space limitation).

In our experiments, CNMF multiplicative updating rules [13] could not control the sparsity of W and H well. Some difficulties associated with this method were already discussed in Section 2. We also tested Hoyer's sparse NMF based on the projected gradient descent method by his Matlab implementation. Although it worked when we applied sparseness constraints only on W, it failed when we tried to impose sparse constraints only on H. We could overcome this problem by dividing the dataset by a large value or applying normalization in order to avoid values that are too large in the dataset. Table 3 shows performance comparison between SNMF/PGD [5] having sparseness constraints only on H with a desired sparseness parameter $s_H = 0.4$ and SNMF/R with $\beta = 0.01$ on the same scaled dataset ($A_s = 0.01 * A$). After five runs with different random initializations, we observed the average percentages of zero elements in W and H, mean approximation error, mean purity, mean entropy, mean iteration, and total computing time. In general, sparse NMFs generate larger errors when we apply stronger sparsity constraints. Although SNMF/R achieved greater sparseness both in W and H, its approximation error was less than that of SNMF/PGD. More importantly, SNMF/R showed significantly better clustering performance than SNMF/PGD. Although we tested various s_H values, SNMF/PGD did not show better clustering performance than SNMF/R. Specifically, the maximal purity was only 0.895 and the minimal entropy was 0.280. Since many practical applications apply NMFs to clustering problems, the superior clustering power of SNMF/R is one of the major advantages. Moreover, SNMF/R required an order of magnitude shorter computing time and smaller number of iterations than SNMF/PGD.

For the CNS tumors dataset, we repeated non-negative matrix factorizations 50 times to obtain the average connectivity matrix (*i.e.* consensus matrix) whose entries reflect the probability that samples i and j belong to the same cluster. We can measure the dispersion of the consensus matrix by the Cophenetic correlation coefficient (ρ) [2]. The value of coefficient is $\rho = 1$ for a perfect consensus matrix (all entries = 0 or 1) and $0 \leq \rho < 1$ for a scattered consensus matrix. After obtaining ρ_k values for various k, we can determine the number of clusters from the maximal ρ_k. Figures 1 and 2 illustrate that NMFs find the number of clusters in the CNS tumors dataset with the maximal ρ_k at $k = 4$. Figure 2 shows that SNMF/R with $\beta = 0.01$ finds perfect consensus matrices for $k = 2, 3, 4$. In other words, SNMF/R generated H matrices that have the same cluster structure with different random initializations of H. By using SNMF/R, we could obtain finer consensus matrices (higher ρ_k) for various k values as well as the number of clusters in the CNS tumors dataset.

6 Summary

We present novel sparse NMFs via alternating non-negativity-constrained least squares involving L_1-norm minimization. These sparse NMFs can also be considered as unsupervised dimension reduction methods that can control the degree of sparseness of basis matrix or coefficient matrix under non-negativity constraints. SNMF/L is helpful in obtaining parts-based basis vectors. SNMF/R can be used for cancer class discovery and gene expression data analysis due to its good clustering performance. These algorithms can be applied to many practical problems in bioinformatics and computational biology, for instance, biomedical text mining, gene/protein microarray data analysis, *etc.*

Acknowledgment

This material is based upon work supported in part by the National Science Foundation Grants CCR-0204109 and ACI-0305543. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] R. Bro and S. de Jong. A fast non-negativity-constrained least squares algorithm. *J. Chemometrics*, 11:393–401, 1997.

[2] J. P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proc. Natl Acad. Sci. USA*, 101(12):4164–4169, 2004.

[3] Y. Gao and G. Church. Improving molecular cancer class discovery through sparse non-negative matrix factorization. *Bioinformatics*, 21(21):3970–3975, 2005.

[4] E. F. Gonzales and Y. Zhang. Accelerating the Lee-Seung algorithm for non-negative matrix factorization. Technical report, Department of Computational and Applied Mathematics, Rice University, 2005.

[5] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[6] P. M. Kim and B. Tidor. Subsystem identification through dimensionality reduction of large-scale gene expression data. *Genome Research*, 13:1706–1718, 2003.

[7] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[8] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proceedings of Neural Information Processing Systems*, pages 556–562, 2000.

[9] S. Z. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized parts-based representations. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 207–212, 2001.

[10] C. J. Lin. Projected gradient methods for non-negative matrix factorization. Technical Report Information and Support Service ISSTECH-95-013, Department of Computer Science, National Taiwan University, 2005.

[11] MATLAB. *User's Guide*. The MathWorks, Inc., Natick, MA 01760, 1992.

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

[13] V. P. Pauca, J. Piper, and R. J. Plemmons. Nonnegative matrix factorization for spectral data analysis, 2006. *Linear Algebra and Applications*, to appear.

[14] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons. Text mining using non-negative matrix factorizations. In *Proc. SIAM Int'l Conf. Data Mining (SDM'04)*, April 2004.

[15] R. Tibshirani. Regression shrinkage and selection via LASSO. *J. Roy. Statist. Soc. B*, 58:267–288, 1996.

[16] M. H. van Benthem and M. R. Keenan. Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems. *J. Chemometrics*, 18:441–450, 2004.

Table 1: Performance dependency of SNMF/L and SNMF/R against various α and β values on the leukemia data matrix of size $5,000 \times 38$. We present the average percentages of zero elements in W and H over five runs with different random initializations. Average purity and average entropy are also presented. *For NMF based on divergence-based update rules (NMF/DUR), the average percentages of the number of very small non-negative elements that are smaller than 10^{-8} in W and H are presented. We also present total computing time (in seconds) for five runs and the average number of iterations.

Leukemia ($k = 3$)	NMF/DUR	SNMF/L			
α	-	0.001	0.01	0.1	1.0
#($W = 0$) (%)	0.17%*	2.75%	3.26%	12.85%	45.52%
#($H = 0$) (%)	0.00%*	18.42%	15.79%	6.14%	0.00%
Purity	0.953	0.947	0.947	0.947	0.842
Entropy	0.141	0.169	0.169	0.158	0.350
# of iterations	602.0	102.0	105.0	105.0	92.0
Total computing time	331.8	51.8	49.8	54.4	61.7
Leukemia ($k = 3$)	-	SNMF/R			
β	-	0.001	0.01	0.1	1.0
#($W = 0$) (%)	-	2.68%	2.50%	1.70%	0.39%
#($H = 0$) (%)	-	18.42%	23.68%	38.60%	59.82%
Purity	-	0.974	0.974	0.947	0.926
Entropy	-	0.095	0.095	0.158	0.173
# of iterations	-	99.0	96.0	80.0	79.0
Total computing time	-	49.6	48.9	42.2	39.5

Table 2: Performance dependency of SNMF/L and SNMF/R against various α and β values on the CNS tumors data matrix of size $5,597 \times 34$. We present the average percentages of zero elements in W and H over five runs with different random initializations. Average purity and average entropy are also presented. *For NMF based on divergence-based update rules (NMF/DUR), the average percentages of the number of very small non-negative elements that are smaller than 10^{-8} in W and H are presented. We also present total computing time (in seconds) for five runs and the average number of iterations.

CNS tumors ($k = 4$)	NMF/DUR	SNMF/L			
α	-	0.001	0.01	0.1	1.0
#($W = 0$) (%)	1.98%*	9.27%	11.48%	29.24%	59.95%
#($H = 0$) (%)	5.29%*	25.0%	19.12%	11.03%	0.00%
Purity	0.947	0.971	0.971	0.882	0.882
Entropy	0.112	0.071	0.071	0.230	0.230
# of iterations	1001.0	114.0	114.0	240.0	147.0
Total computing time	617.8	74.0	76.0	209.2	179.6

CNS tumors ($k = 4$)	-	SNMF/R			
β	-	0.001	0.01	0.1	1.0
#($W = 0$) (%)	-	8.94%	8.19%	3.45%	0.31%
#($H = 0$) (%)	-	25.0%	26.47%	48.53%	71.32%
Purity	-	0.971	0.971	0.971	0.865
Entropy	-	0.071	0.071	0.071	0.232
# of iterations	-	107.0	104.0	83.0	93.0
Total computing time	-	72.1	69.1	50.2	49.6

Table 3: Performance comparison between SNMF/R and Hoyer's sparse NMF based on the projected gradient descent method (SNMF/PGD) [5] on the scaled leukemia data matrix $A_s = 0.01 * A$ with $k = 3$. After five runs with different random initializations, we present total computing time for five runs and the average values of percentage of zero elements in W and H, approximation error, purity, entropy and the number of iterations.

Algorithms	SNMF/PGD	SNMF/R
Parameter	$s_H = 0.4$	$\beta = 0.01$
#($W = 0$) (%)	0.22%	2.50%
#($H = 0$) (%)	21.75%	23.68%
$f = \|A_s - WH\|_F$	2.385×10^3	2.368×10^3
Purity	0.895	0.974
Entropy	0.280	0.095
# of iterations	662.0	91.0
Total computing time	671.9 sec.	45.3 sec.