The Relationships Among Various Nonnegative Matrix Factorization Methods for Clustering

Tao Li
School of Computer Science
Florida International University
Miami, FL 33199
taoli@cs.fiu.edu

Chris Ding
Lawrence Berkeley Nat'l Lab
University of California
Berkeley, CA 94720
chqding@lbl.gov

Abstract

The nonnegative matrix factorization (NMF) has been shown recently to be useful for clustering. Various extensions of NMF have also been proposed. In this paper we present an overview and theoretically analyze the relationships among them. In addition, we clarify previously unaddressed issues, such as NMF normalization, cluster posterior probabilty, and NMF algoritm convergence rate. Experiments are also conducted to empirically evaluate and compare various factorization methods.

Keywords: *matrix factorization, simultaneous clustering, NMF normalization, NMF convergence rate*

1. Introduction

The nonnegative matrix factorization (NMF) has been shown recently to be useful for many applications in environment, pattern recognition, multimedia, text mining, and DNA gene expressions [9, 26, 29, 32]. NMF can be traced back to 1970s (Notes from G. Golub) and is studied extensively by Paatero [29]. The work of Lee and Seung [24, 25] brought much attention to NMF in machine learning and data mining fields. A very recent theoretical analysis [12] shows the equivalence of NMF and spectral clustering and K-means clustering. Various extensions and variations of NMF have been proposed recently [13, 14, 15, 23, 27, 3, 30, 33].

Despite significant research progress in this area, few attempts have been made to establish the connections between various factorization methods while highlighting their differences. In this paper, we aim to provide a comparative study on matrix factorization for clustering. We present an overview and summary on various matrix factorization algorithms and theoretically analyze the relationships among them. Experiments were also conducted to empirically evaluate and compare various factorization methods. In particular, our study tries to address the following important questions for matrix factorizations:

- What are the available forms of matrix factorizations for clustering?
- What are the relations among the matrix factorizations as well as the existing clustering methods?
- How to interpret the cluster posterior obtained from matrix factorizations?
- What are the benefits of simultaneous clustering?
- How to evaluate simultaneous clustering?
- How to choose different factorization methods?

We expect our study would provide insightful guidance on matrix factorization research for clustering. The rest of the paper is organized as follows: Section 2 summarizes various matrix factorizations; Section 3 illustrates the differences among various factorizations using examples; Section 4 introduces the computation algorithms for various factorization methods; Section 5 studies the relationships of various matrix factorization methods; Section 6 discusses the normalization method for eliminating the uncertainty in NMF solutions; Section 7 explains when and why the simultaneous clustering is preferred and presents strategies for evaluating simultaneous clustering; Section 8 shows the experimental results for empirically comparing various matrix factorization methods; and finally Section 9 concludes.

2. Different Matrix Factorizations

In general, matrix factorization algorithms attempt to find the subspace in which the majority of the data points lie.

Let the input data matrix $X = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ contain the collection of n data column vectors. Generally, we factorize X into two matrices,

$$X \approx FG^T, \qquad (1)$$

where $X \in \mathbb{R}^{p\times n}$, $F \in \mathbb{R}^{p\times k}$ and $G \in \mathbb{R}^{n\times k}$. Generally, the rank of matrices F, G is much lower than the rank of X (i.e., $k \ll \min(p,n)$). Here we provide an overview on related matrix factorization methods:

1. **SVD:** The classic matrix factorization is Principal Component Analysis (PCA) which uses the singular value decomposition [**?**, 17], $X \approx U\Sigma V^T$, where we allow U,V to have mixed-signs; the input data could have mixed-signs. absorbing Σ into U, we can write

$$\text{SVD:} \quad X_\pm \approx U_\pm V_\pm$$

2. **NMF:** When the input data is nonnegative, and we restrict F and G to be nonnegative. The standard NMF can be written as

$$\text{NMF:} \quad X_+ \approx F_+ G_+$$

using an intuitive notation for $X, F, G \geq 0$.

3. **Semi-NMF:** When the input data has mixed signs, we can restrict G to be nonnegative while placing no restriction on the signs of F. This is called semi-NMF [13]:

$$\text{semi-NMF:} \quad X_\pm \approx F_\pm G_+.$$

Semi-NMF can be motivated by K-means clustering. Let $F = (\mathbf{f}_1, \cdots, \mathbf{f}_k)$ be the cluster centroids obtained via K-means clustering. Let G be the cluster indicators: i.e., $g_{ki} = 1$ if $\mathbf{x}_i$ belongs to cluster c_k; $g_{ki} = 0$ otherwise. The K-means clustering objective can be written as

$$J_{K-means} = \sum_{i=1}^{n}\sum_{k=1}^{K} g_{ik}\|\mathbf{x}_i - \mathbf{f}_k\|^2 = \|X - FG^T\|^2,$$

where $\|\cdot\|$ is Frobenius norm[1]. Semi-NMF can be thought as a soft clustering by relaxing the element of g from binary to continuous nonnegative values.

4. **Convex-NMF:** In general, the basis vectors

$$F = (\mathbf{f}_1, \cdots, \mathbf{f}_k)$$

can be anything in a large space, in particular, a space that contains the space spanned by the columns of $X = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$. In order for the vectors F to capture the notion of cluster centroids, we restrict them to lie within the space spanned by the columns of X, i.e.,

$$\mathbf{f}_l = w_{1l}\mathbf{x}_1 + \cdots + w_{nl}\mathbf{x}_n = X\mathbf{w}_l, \text{or } F = XW.$$

Furthermore, we restrict $\mathbf{f}_l$ as a convex combination $w_{il} \geq 0$ of the data points. We call this restricted form of factorization as Convex-NMF. Convex-NMF applies to both nonnegative and mixed-sign input data.

5. **Tri-Factorization:** To simultaneously cluster the rows and the columns of the input data matrix X, we consider the following nonnegative 3-factor decomposition [15]

$$X \approx FSG^T. \qquad (2)$$

Note that S provides additional degrees of freedom such that the low-rank matrix representation remains accurate while F gives row clusters and G gives column clusters. More precisely, we solve

$$\min_{F\geq 0, G\geq 0, S\geq 0} \|X - FSG^T\|^2, \;\; s.t.\, F^TF = I,\; G^TG = I. \qquad (3)$$

This form gives a good framework for simultaneously clustering the rows and columns of X [35].

An important special case is that the input X contains a matrix of pairwise similarities: $X = X^T = W$. In this case, $F = G = H$. We optimize the symmetric NMF:

$$\min_{W\geq 0, S\geq 0} \|X - HSH^T\|^2, \;\; s.t.\, H^TH = I.$$

6. **Kernel NMF:** Consider a mapping

$$\mathbf{x}_i \to \phi(\mathbf{x}_i), \text{ or } X \to \phi(X) = (\phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_n)).$$

A standard NMF or Semi-NMF like $\phi(X) \approx FG^T$ would be difficult since F, G will depends explicitly on the mapping function $\phi(\cdot)$. However, Convex-NMF provides a nice possibility:

$$\phi(X) \approx \phi(X)WG^T.$$

It is easy to see that the minimization objective

$$\begin{aligned} ||\phi(X) - \phi(X)WG^T||^2 \;\; &= \;\; \mathrm{Tr}[\phi(X)^T\phi(X) - 2G^T\phi^T(X)\phi(X)W \\ &\quad + W^T\phi^T(X)\phi(X)WG^TG] \end{aligned}$$

depends only on the kernel $K = \phi^T(X)\phi(X)$. This kernel extension of NMF is similar to kernel-PCA and kernel K-means .

In summary, various factorizations differ by the restrictions on the matrix forms and signs, we write them collectively as follows:

$$\begin{aligned} \text{SVD:} &\quad X_\pm \approx U_\pm V_\pm \\ \text{NMF:} &\quad X_+ \approx F_+ G_+^T \\ \text{Semi-NMF:} &\quad X_\pm \approx F_\pm G_+^T \\ \text{Convex-NMF:} &\quad X_\pm \approx X_\pm W_+ G_+^T \\ \text{Kernel-NMF:} &\quad \phi(X_\pm) \approx \phi(X_\pm) W_+ G_+^T \\ \text{Tri-Factorization:} &\quad X_+ \approx F_+ S_+ G_+^T \\ \text{Symmetric-NMF:} &\quad W_+ \approx H_+ S_+ H_+^T \end{aligned}$$

[1] Without specifying, all norms in this paper are Frobenius norm.

3. Illustration Examples

In this section, we use examples to illustrate the difference among various NMF methods.

3.1 A Nonnegative Example

In the section, we use a nonnegative example to illustrate the differences in NMF and Tri-Factorization.

The input data matrix X is

$$\begin{pmatrix} 0.185 & 0.326 & 0.761 & 2.799 & 2.375 & 2.970 & 2.585 \\ 0.508 & 0.380 & 0.884 & 2.134 & 2.374 & 2.342 & 2.524 \\ 0.452 & 0.887 & 0.457 & 2.065 & 2.484 & 2.253 & 2.163 \\ 1.486 & 1.843 & 1.858 & 0.566 & 0.103 & 0.417 & 0.269 \\ 1.496 & 1.806 & 1.610 & 0.612 & 0.158 & 0.560 & 0.784 \end{pmatrix}$$

One can see that the first 3 columns should be one cluster and the last 4 columns should be another cluster.

The computed basis vectors F for NMF and Tri-Factorization are:

$$F_{nmf} = \left(\begin{array}{cc} 0.0403 & 0.3695 \\ 0.0889 & 0.3149 \\ 0.1033 & 0.2945 \\ 0.3882 & 0.0002 \\ 0.3794 & 0.0210 \\ \hline 16.83 & 30.64 \end{array}\right), F_{Tri} = \left(\begin{array}{cc} 0.0000 & 0.3704 \\ 0.0215 & 0.3228 \\ 0.0320 & 0.3068 \\ 0.4773 & 0.0000 \\ 0.4692 & 0.0000 \\ \hline 2.6172 & 3.4036 \end{array}\right)$$

Basis vectors are normalized to 1 in L_2-norm (norms are given at the bottom line) for comparison purpose. We can see that F leads to the correct row clustering results: the first three row are in one cluster and the bottom 2 rows are in another cluster.

The matrices G for NMF and Tri-Factorization are listed as follows:

$$G_{nmf} =$$
$$\begin{pmatrix} 0.234 & 0.287 & 0.259 & 0.080 & 0.012 & 0.063 & 0.065 \\ 0.006 & 0.014 & 0.040 & 0.223 & 0.238 & 0.244 & 0.236 \end{pmatrix}$$
$$G_{Tri} =$$
$$\begin{pmatrix} 0.270 & 0.335 & 0.333 & 0.034 & 0.000 & 0.009 & 0.020 \\ 0.000 & 0.000 & 0.000 & 0.239 & 0.248 & 0.264 & 0.250 \end{pmatrix}.$$

In both factorizations, G leads to the correct clustering results: the first three columns are in one cluster and the remaining columns are in another cluster. Note that

$$S_{Tri-factor} = \begin{pmatrix} 4.3626 & 1.0136 \\ 1.4824 & 8.4000 \end{pmatrix}.$$

It absorbs the different scales of X, $F_{Tri-factor}$ and $G_{Tri-factor}$ and thus $F_{Tri-factor}$ provides row clusters (i.e., the attribute clusters).

3.2 A Mixed-sign Example

In this section, we give an example to illustrate the differences in SVD, semi-NMF and convex-NMF. The input data matrix is

$$X = \begin{pmatrix} 1.3 & 1.8 & 4.8 & 7.1 & 5.0 & 5.2 & 8.0 \\ 1.5 & 6.9 & 3.9 & -5.5 & -8.5 & -3.9 & -5.5 \\ 6.5 & 1.6 & 8.2 & -7.2 & -8.7 & -7.9 & -5.2 \\ 3.8 & 8.3 & 4.7 & 6.4 & 7.5 & 3.2 & 7.4 \\ -7.3 & -1.8 & -2.1 & 2.7 & 6.8 & 4.8 & 6.2 \end{pmatrix}$$

One can see that the first 3 columns should be one cluster and the last 4 columns should be another cluster.

The computed basis vectors F are:

$$F_{svd} = \left(\begin{array}{cc} -0.41 & 0.50 \\ 0.35 & 0.21 \\ 0.66 & 0.32 \\ -0.28 & 0.72 \\ -0.43 & -0.28 \\ \hline 25.5 & 15.6 \end{array}\right),$$

$$F_{semi} = \left(\begin{array}{cc} 0.05 & 0.27 \\ 0.40 & -0.40 \\ 0.70 & -0.72 \\ 0.30 & 0.08 \\ -0.51 & 0.49 \\ \hline 20.3 & 23.0 \end{array}\right), F_{conv} = \left(\begin{array}{cc} 0.31 & 0.53 \\ 0.42 & -0.30 \\ 0.56 & -0.57 \\ 0.49 & 0.41 \\ -0.41 & 0.36 \\ \hline 31.0 & 39.3 \end{array}\right),$$

Basis vectors are normalized to 1 in L_2-norm (norms are given at the bottom line) for comparison purpose.

The matrix G are listed as follows:

$$G_{svd} = \begin{pmatrix} 0.25 & 0.05 & 0.22 & -0.45 & -0.44 & -0.46 & -0.52 \\ 0.50 & 0.60 & 0.43 & 0.30 & -0.12 & 0.01 & 0.31 \end{pmatrix}$$

$$G_{semi} = \begin{pmatrix} 0.61 & 0.89 & 0.54 & 0.77 & 0.14 & 0.36 & 0.84 \\ 0.12 & 0.53 & 0.11 & 1.03 & 0.60 & 0.77 & 1.16 \end{pmatrix}$$

$$G_{conv} = \begin{pmatrix} 0.31 & 0.31 & 0.29 & 0.02 & 0 & 0 & 0.02 \\ 0 & 0.06 & 0 & 0.31 & 0.27 & 0.30 & 0.36 \end{pmatrix}$$

Both semi-NMF and convex-NMF give the correct clustering results. However, convex-NMF gives sharper cluster indicators, while semi-NMF gives a soft clustering. The residual values, the level of low-rank approximations, are

$$\|X - FG^T\| = 0.27940, 0.27944, 0.30877,$$

for SVD, semi-NMF, and convex-NMF respectively. We see that semi-NMF has a good quality approximation close to SVD.

4. Algorithms for Various Matrix Factorization Methods

The algorithms for matrix factorizations are generally iterative updating procedures: updating one factor while fixing the other factors. The algorithms for various matrix

Factorizations	Updating Rules
NMF	$F_{ik} \leftarrow F_{ik}\frac{(XG)_{ik}}{(FG^TG)_{ik}}$ $G_{jk} \leftarrow G_{jk}\frac{(X^TF)_{jk}}{(GF^TF)_{jk}}$
Semi-NMF	$F = XG(G^TG)^{-1}$ $G_{ik} \leftarrow G_{ik}\sqrt{\frac{(X^TF)^+_{ik}+[G(F^TF)^-]_{ik}}{(X^TF)^-_{ik}+[G(F^TF)^+]_{ik}}}$
Convex-NMF	$G_{ik} \leftarrow G_{ik}\sqrt{\frac{[(X^TX)^+W]_{ik}+[GW^T(X^TX)^-W]_{ik}}{[(X^TX)^-W]_{ik}+[GW^T(X^TX)^+W]_{ik}}}$ $W_{ik} \leftarrow W_{ik}\sqrt{\frac{[(X^TX)^+G]_{ik}+[(X^TX)^-WG^TG]_{ik}}{[(X^TX)^-G]_{ik}+[(X^TX)^+WG^TG]_{ik}}}$
Tri-Factorization	$G_{jk} \leftarrow G_{jk}\sqrt{\frac{(X^TFS)_{jk}}{(GG^TX^TFS)_{jk}}}$ $F_{ik} \leftarrow F_{ik}\sqrt{\frac{(XGS^T)_{ik}}{(FF^TXGS^T)_{ik}}}$ $S_{ik} \leftarrow S_{ik}\sqrt{\frac{(F^TXG)_{ik}}{(F^TFSG^TG)_{ik}}}$
Kernel-NMF	replace X^TX by $\langle \phi(X)^T\phi(X)\rangle$ in Convex-NMF

Table 1. Updating rules for different matrix factorizations.

factorizations are summarized in Table 1. In the table, we separate the positive and negative parts of a matrix A as $A^+_{ik} = (|A_{ik}| + A_{ik})/2$, $A^-_{ik} = (|A_{ik}| - A_{ik})/2$.

In the Literature there is some question [16] on whether Lee-Seung algorithm converge to a local minima. However, it is easy to show that at convergence, the solution satisfy the well-known KKT complementarity condition in the theory of constrained optimization, which is,

$$(XG - FG^TG)_{ik}F_{ik} = 0,\ (X^TF - GF^TF)_{jk}G_{jk} = 0, \quad (4)$$

for the objective $J = ||X - FG^T||^2$. For example, at convergence, $F^*_{ik} = F^*_{ik}(XG^*)_{ik}/\ (F^*G^{*T}G^*)_{ik}$ which is identical to first condition in Eq.(4). Therefore, Lee-Seung algorithm does converge to a local minima according to KKT theory. It has been proved that at convergence, solutions of all algorithms listed in Table 1 satisfy the KKT conditions in their respective cases.

We can let $\Theta = \begin{pmatrix} F \\ G \end{pmatrix}$ and view the updating algorithms as mapping $\Theta^{(t+1)} = M(\Theta^{(t)})$. At convergence, $\Theta^* = M(\Theta^*)$. The objectives for all cases in Table 1 have been proved to be non-increasing, $J(\Theta^{(t+1)}) \leq J(\Theta^{(t)})$. Following Xu & Jordan [31], we expand[2] $\Theta \simeq M(\Theta^*) + (\partial M/\partial\Theta)(\Theta - \Theta^*)$. Therefore,

$$||\Theta^{(t+1)} - \Theta^*|| \leq ||\frac{\partial M}{\partial \Theta}|| \cdot ||\Theta^{(t)} - \Theta^*||$$

under appropriate matrix norm. In general, $\partial M/\partial\Theta \neq 0$. Thus these updating algorithms have first order convergence rate, same as the EM algorithm [31].

5. Relations Among Various Factorizations

In this section, we theoretically analyze the relationships among various matrix factorization methods.

5.1. NMF and K-means Clustering

Lee and Seung [24] emphasizes the difference between NMF and vector quantization (which is K-means clustering). Later experiments [22, 26] empirically show that NMF has clear clustering effects. Theoretically, NMF is inherently related to kernel K-means clustering.

Theorem 1. Orthogonal NMF,

$$\min_{F\geq 0, G\geq 0} ||X - FG^T||^2,\ \ s.t.\ G^TG = I. \quad (5)$$

is equivalent to K-means clustering.

This theorem has been previously proved [12] with additional normalization conditions. Here we give a more general proof, which will generalize to bi-orthogonality.

Proof. We write $J = ||X - FG^T||^2 = \mathrm{Tr}(X^TX - 2F^TXG + F^TF)$. The zero gradient condition $\partial J/\partial F = -2XG + 2F = 0$ gives $F = XG$. Thus $J = \mathrm{Tr}(X^TX - G^TX^TXG)$. Since $\mathrm{Tr}(X^TX)$ is a constant, the optimization problem becomes

$$\min_{G\geq 0} \mathrm{Tr}(G^TX^TXG)\ \ s.t.\ G^TG = I. \quad (6)$$

According to Theorem 2 below, this is identical to K-means clustering. ■

We note that Theorem 1 holds even if X and F are not nonnegative, i.e., X and F have mixed-sign entries.

Theorem 2[11, 34]. The K-means clustering

$$J = \sum_{k=1}^{K} \sum_{i\in C_k} ||\mathbf{x}_i - \mathbf{f}_k||^2 \quad (7)$$

[2] Nonnegativity constraint need be enforced.

where $\mathbf{f}_k$ is the cluster centroid of the k-th cluster, and more generally, the Kernel K-means with mapping $\mathbf{x}_i \to \phi(\mathbf{x}_i)$

$$J_\phi = \sum_{k=1}^{\kappa} \sum_{i \in C_k} ||\phi(\mathbf{x}_i) - \overline{\phi}_k||^2 \tag{8}$$

where $\overline{\phi}_k$ is the centroid in the feature space. This can be solved via the optimization problem

$$\max_{G^T G = I,\ G \geq 0} \mathrm{Tr}(G^T W G), \tag{9}$$

where G are the cluster indicators and $W_{ij} = \phi(\mathbf{x}_i)^\tau \phi(\mathbf{x}_j)$ is the kernel. For K-means, $\phi(\mathbf{x}_i) = \mathbf{x}_i$, $W_{ij} = \mathbf{x}_i^\tau \mathbf{x}_j$.

NMF has clustering capabilities which is generally better than the K-means. In fact, PCA is effectively doing K-means clustering [11, 34]. Let G be the cluster indicators for the k clusters then (1) $GG^T \simeq VV^T$; (ii) the principal directions, UU^T, project data points into the subspace spanned by the k cluster centroids.

5.2. NMF, Semi-NMF, Convex-NMF and Kernel-NMF

In fact, NMF, semi-NMF, convex-NMF and kernel-NMF all have K-means clustering interpretations when the factor G is orthogonal. Being orthogonal and nonnegative, implies each row of G has only one nonnegative elements, i.e., G is a bona fide cluster indicator. We have

Theorem 3. G-orthogonal NMF, semi-NMF, convex-NMF and Kernel-NMF is identical to relaxed K-means clustering.

Proof. For NMF, semi-NMF and convex-NMF, we first eliminate F. The objective is $J = \|X - FG^T\|^2 = \mathrm{Tr}(X^TX - 2X^TFG^T + FF^T)$. Setting $\partial J/\partial F = 0$, we obtain $F = XG$. Thus we obtain

$$J = \mathrm{Tr}(X^TX - G^TX^TXG).$$

For Kernel-NMF, we have

$$J = \|\phi(X) - \phi(X)WG^T\|^2 = \mathrm{Tr}(K - G^TKW + W^TKW),$$

where K is the kernel. Setting $\partial J/\partial W = 0$, we have $KG = KW$. Thus

$$J = \mathrm{Tr}(X^TX - G^TKG).$$

In all the above cases, the first term are constant and are ignored. The minimization problem thus becomes

$$\max_{G^T G = I} \mathrm{Tr}(G^TKG),$$

where K is either a linear kernel X^TX or $\langle \phi(X), \phi(X) \rangle$. It is known [34] that this is identical to (kernel-) K-means clustering. □

In the definitions of NMF, semi-NMF, convex-NMF, G is not restricted to be orthogonal; these NMF varieties are *soft* versions of K-means clustering. From NMF/semi-NMF, and to convex-NMF, the successive restrictions make them different levels of soft clustering.

This situation is similar to the mixture of Gaussian generalization of K-means . K-means is a mixture of spherical Gaussians with same variance. The first step is to generalize to spherical Gaussians with individual variance. The second step is to generalize to Gaussians with individual full covariance matrix, etc. Each generalization have more model parameters and fits the data better.

5.3. Tri-Factorization

First, we emphasize the role of orthogonality in Tri-Factorization [3] Considering the unconstrained 3-factor NMF

$$\min_{F \geq 0, G \geq 0, S \geq 0} \|X - FSG^T\|^2, \tag{10}$$

we note that this 3-factor NMF can be reduced to the unconstrained 2-factor NMF by mapping $F \leftarrow FS$. Another way to say this is that the degree of freedom of FSG^T is the same as FG^T.

Therefore, 3-factor NMF is interesting only when it can not be transformed into 2-factor NMF. This happens when certain constraints are applied to the 3-factor NMF. However, not all constrained 3-factor NMF differ from their 2-factor NMF counterpart. For example, the following 1-sided orthogonal 3-factor NMF

$$\min_{F \geq 0, G \geq 0, S \geq 0} \|X - FSG^T\|^2, \text{ s.t. } F^TF = I \tag{11}$$

is no different from its 2-factor counterpart, because the mapping $F \leftarrow FS$ reduces one to another. It is clear that

$$\min_{F \geq 0, G \geq 0, S \geq 0} \|X - FSG^T\|^2, \ s.t. \ F^TF = I, \ G^TG = I. \tag{12}$$

has no corresponding 2-factor counterpart. This is a genuine new factorization, which we call 3-factor NMF. The update rules are given in Table 1.

An important special case is that the input X contains a matrix of pairwise similarities: $X = X^T = W$. In this case, $F = G = H$. We optimize the symmetric NMF:

$$\min_{W \geq 0, S \geq 0} \|X - HSH^T\|^2, \ s.t. \ H^TH = I. \tag{13}$$

When the orthogonality of $H^TH = I$ is enforced, we can use the update rules of Tri-Factorization of Eq.(12) with appropriate substitutions. When $H^TH = I$ is not enforced, the update rules are :

$$S_{ik} \leftarrow S_{ik} \frac{(H^\tau WH)_{ik}}{(H^THSH^TH)_{ik}}. \tag{14}$$

[3] Sometimes we also use 3-factor NMF to represent Tri-Factorization.

$$H_{ik} \leftarrow H_{ik}\left(1-\beta+\beta\frac{(WHS)_{ik}}{(HSH^THS)_{ik}}\right). \qquad (15)$$

where $0 < \beta \leq 1$. In practice, we find $\beta = 1/2$ is a good choice.

5.4. NMF and PLSI

Here we show that NMF is related to another relevant unsupervised learning method: Probabilistic Latent Semantic Indexing (PLSI). So far, the cost function we used for computing NMF is the sum of squared errors, $||X - FG^T||^2$. Another cost function KL divergence:

$$J_{\text{NMF-KL}} = \sum_{i=1}^{m}\sum_{j=1}^{n}\left[X_{ij}\log\frac{X_{ij}}{(FG^T)_{ij}} - X_{ij} + (FG^T)_{ij}\right] \qquad (16)$$

Probabilistic Latent Semantic Indexing (PLSI) is a unsupervised learning method based on statistical latent class models and has been successfully applied to document clustering [21]. (PLSI is further developed into a more comprehensive Latent Dirichlet Allocation model [5].)

PLSI maximize the likelihood

$$J_{\text{PLSI}} = \sum_{i=1}^{m}\sum_{j=1}^{n} X(w_i, d_j)\log P(w_i, d_j) \qquad (17)$$

where the joint occurrence probability is factorized (i.e., parameterized or approximated) as

$$\begin{aligned} P(w_i, d_j) &= \sum_k P(w_i, d_j|z_k)P(z_k) \\ &= \sum_k P(w_i|z_k)P(d_j|z_k)P(z_k), \end{aligned} \qquad (18)$$

assuming that w_i and d_j are conditionally independent given z_k.

Proposition 1. Objective function of PLSI is identical to the objective function of NMF, i.e., $J_{\text{PLSI}} = -J_{\text{NMF-KL}} + constant$.

The proposition can be easily proved by setting $(FG^T)_{ij} = P(w_i, d_j)$. Therefore, the NMF update algorithm and the EM algorithm in training PLSI are alternative methods to optimize the same objective function [14].

6. Normalization of Nonnegative Matrix Factorizations

In this section, we try to interpret the cluster posterior obtained form matrix factorization.

Given a solution (F, G) of NMF: $X = FG^T$, it is usually assumed that G is the cluster posterior and thus G_{ik} gives the posterior probability that $\mathbf{x}_i$ belongs to the k-th column cluster. However, the NMF solutions are not unique. Suppose (F, G) is solution of NMF. There exist many matrices (A, B) such that $AB^T = I$, $FA \geq 0$, $GB \geq 0$. Thus (FA, GB) is also the solution with the same residue $||X - FG^T||$.

A way to resolve this is to assume NMF follows a certain distribution. We can think of the rectangular input data X as a word-document matrix and perform a PLSI type probabilistic decomposition. as in Eq. 18, where z_k is the latent cluster variable, and the probability factors follow the probability normalization

$$\sum_{i=1}^{m} p(w_i|z_k) = 1,\ \sum_{j=1}^{n} p(d_j|z_k) = 1,\ \sum_{k=1}^{K} p(z_k) = \sum_{ij} X_{ij} = 1.$$

We assume the data is normalized such that $\sum_{ij} X_{ij} = 1$. With this, the cluster posterior probability for column d_j is then

$$p(z_k|d_j) = p(d_j|z_k)p(z_k)/p(d_j) \propto p(z_k)p(d_j|z_k).$$

Translating to F, G, the equivalent probabilistic decomposition is

$$X = FG^T = (FD_F^{-1})(D_F D_G)(GD_G^{-1})^T,$$

where $D_F = \text{diag}(\mathbf{e}^T F)$, $D_G = \text{diag}(\mathbf{e}^T G)$, and

$$\sum_{i=1}^{m}(FD_F^{-1})_{ik} = 1,\ \sum_{j=1}^{n}(GD_G^{-1})_{jk} = 1,\ \sum_{k=1}^{K}(D_F D_G)_{kk} = 1.$$

Thus for standard NMF, the cluster posterior probability for column $\mathbf{x}_i$ is

$$\text{NMF:} \quad p(z_k|\mathbf{x}_i) \propto [(D_F D_G)(GD_G^{-1})^T]_{ik}^T = (GD_F)_{ik}$$

For Convex-NMF, the centroid interpretation of $F = XW$ implies W should have a L_1 normalization. Thus we write

$$X = XWG^T = (XWD_W^{-1})(D_W D_G)(GD_G^{-1})^T.$$

Therefore, the cluster posterior probability for column $\mathbf{x}_i$ is

$$\text{Convex-NMF:} \quad p(z_k|\mathbf{x}_i) \propto (GD_W)_{ik},\ D_W = \text{diag}(\mathbf{e}^T W).$$

Semi-NMF does not have a probability interpretation because F could be negative signs. For this reason, the L_2 normalization is most natural. Let $F = (\mathbf{f}_1, \cdots, \mathbf{f}_k)$ and $Z_F = \text{diag}(||\mathbf{f}_1||, \cdots, ||\mathbf{f}_k||)$. We write

$$X = FG^T = (FZ_F^{-1})(Z_F D_G)(GD_G^{-1})^T.$$

Thus the cluster posterior probability is

$$\text{Semi-NMF:} \quad p(z_k|\mathbf{x}_i) \propto (GZ_F)_{ik}.$$

7. Simultaneous Clustering

Consider the nonnegative Tri-Factorization $X \simeq FSG^T$. For the objective of the function approximation, we optimize

$$\min_{F\geq 0, G\geq 0, S\geq 0} ||X - FSG^T||^2,\ s.t.\ F^T F = I,\ G^T G = I.$$

We note $X \in \mathbb{R}_+^{p\times n}$, $F \in \mathbb{R}_+^{p\times k}$ and $S \in \mathbb{R}_+^{k\times \ell}$ and $G \in \mathbb{R}_+^{n\times \ell}$. This allows the number of row cluster (k) differ from the number of column cluster (ℓ). In most cases, we set $k = \ell$. This form gives a good framework for simultaneously clustering the rows and columns of X. Recently, simultaneous clustering has been extensively studied [10, 8, 2, 7, 28]. However, two questions are still largely unaddressed in the literature:

- Why do we prefer the simultaneous clustering to single-side clustering?
- How to evaluate the simultaneous clustering?

In this section, we attempt to provide our insights for the above questions.

7.1. Why Simultaneous Clustering?

First, simultaneous clustering is preferred for applications in high dimensional spaces. Most clustering algorithms do not work efficiently in high dimensional spaces due to the *curse of dimensionality*. It has been shown that in a high dimensional space, the distance between every pair of points is almost the same for a wide variety of data distributions and distance functions [4]. Many feature selection techniques have been applied to reduce the dimensionality of the space. However, as demonstrated in [1], the correlations among the dimensions are often specific to data locality; in other words, some data points are correlated with a given set of features and others are correlated with respect to different features. As pointed out in [20], all methods that overcome the dimensionality problems use a metric for measuring neighborhoods, which is often implicit and/or adaptive. Simultaneous clustering performs an implicit feature selection at each iteration and provides an adaptive metric for measuring the neighborhood.

Second, simultaneous clustering is preferred when there is an association relationship between the data and the features (i.e., the columns and the rows). A case is the binary data. A distinctive characteristic of the binary data is that the features (attributes) they include have the same nature as the data they intend to account for: both are binary. Another case is block diagonal clustering where both data points and features have the same number of clusters. In this case, after appropriate permutation of the rows and columns, the cluster structure takes the form of a block diagonal matrix [18].

It should be noted that simultaneous clustering can also be interpreted using a probabilistic view similar to the PLSI model. Instead of assuming that the variables w_i and d_j are conditionally independent given z_k in Eq. 18, we assume that the variable w_i only depends on its cluster variable f_k and the variable d_j only depends on its cluster variable g_l in the probabilistic model of simultaneous clustering. Therefore we have

$$\begin{aligned} P(w_i,d_j) &= \sum_{k,l} P(w_i,d_j|f_k,g_l)P(f_k,g_l) \\ &= \sum_{k,l} P(w_i|f_k)P(d_j|g_l)P(f_k,g_l). \end{aligned} \tag{19}$$

Here, $P(w_i|f_k)$ corresponds to F, $P(d_j|g_l)$ to G and $P(f_k,g_l)$ to S.

7.2. An Illustrative Example

The example is based on a simple dataset which contains six system log messages from two different situations: **Start** and **Create**.

After removing stop words and words only appear once, we get the binary document-term matrix as shown in Table 2. For this example, using one-side clustering, e.g., k-means, it usually does get perfect clustering results. However, using simultaneous clustering, we could correctly obtain the message clusters. The reason is that using simultaneous clustering, in the iteration process, we could adaptively measure the distance between the data points: if the words have similar distributions across multiple clusters, it can be treated as outliers and does not contribute to the distance computation. In the example, Term 3 and 4 (i.e., column 3 and 4) can be thought as feature noises as they have similar distributions across multiple clusters.

Terms/Messages	S1	S2	S3	C1	C2	C3
T1	1	1	1	0	0	1
T2	1	1	1	1	0	0
T3	0	1	1	1	1	0
T4	0	1	1	1	1	0
T5	0	0	0	1	1	1
T6	1	0	0	1	1	1

Table 2. Log message example: The 6 terms are *start*, *application*, *version*, *service*, *create*, *temporary* **respectively.**

7.3. Evaluate Simultaneous Clustering

Simultaneous clustering performs clustering of row and column clustering simultaneously, where the factor F is the cluster indicator for words (i.e., rows). Quantitatively, we can view the i-th row of the cluster indicator F as the posterior probability that word i belongs to each of the K word clusters. We can assign a word to the cluster that has the largest probability value. However, row clustering has no clear *a prior* labels to compare with. For example, for document clustering, we usually have labels for each document class and we have no label information about word clusters.

Here we provide a systematic way for analyzing and evaluating the clustering of rows (i.e.,words). Let this row of F be $(p_1,\cdots,p_k)$, which has been normalized to $\sum_k p_k = 1$. Suppose a word has a posterior distribution of

$$(0.93,\ 0.01,\ 0.04,\cdots,0.02);$$

it is obvious that this word is cleanly clustered into one cluster. We say this word has a 1-peak distribution. Suppose another word has a posterior distribution of

$$(0.52,\ 0.46,\ 0.01,\cdots,0.01);$$

obviously this word is clustered into two clusters. We say this word has a 2-peak distribution. In general, we wish to characterize each word as belonging to 1-peak, 2-peak, 3-peak etc. For K word clusters, we set K prototype distributions:

$$(1,0,\cdots,0),\ (\frac{1}{2},\frac{1}{2},\cdots,0),\ \cdots,\ (\frac{1}{K},\cdots,\frac{1}{K}).$$

For each word, we assign it to the closest prototype distribution based on the Euclidean distance, allowing all possible permutations of the clusters. For example, $(1,0,0,\cdots,0)$ is equivalent to $(0,1,0,\cdots,0)$. In practice, we first sort the row such that the components decrease from the left to the right, and then assign it to the closest prototype. Generally speaking, the less peaks of the posterior distribution of the word, the more unique content of the word has. This multi-peak distribution approach provides the capability of evaluating row (e.g., word) clusterings and enables the systematic analysis of word content.

8. Experiments

In this section, experiments are conducted to empirically compare the clustering results of various NMF algorithms. In our experiments, documents are represented using the binary vector-space model where each document is a binary vector in the term space. Our comparative experimental study includes the following six methods: K-means , NMF, Semi-NMF, Convex-NMF, Tri-Factorization, and PLSI.

8.1. Datasets

We use a variety of datasets, most of which are frequently used in the information retrieval research. Table 3 summarizes the characteristics of the datasets.

CSTR This is the dataset of the abstracts of technical reports (TRs) published in the Department of Computer Science at a research university. The dataset contained 476 abstracts, which were divided into four research areas: Natural Language Processing(NLP), Robotics/Vision, Systems, and Theory.

Datasets	# documents	# class
CSTR	476	4
WebKB4	4199	4
Reuters	2,900	10
WebACE	2,340	20
Log	1367	9

Table 3. Document Datasets Descriptions.

WebKB The WebKB dataset contains webpages gathered from university computer science departments. There are about 8280 documents and they are divided into 7 categories: student, faculty, staff, course, project, department and other. The raw text is about 27MB. Among these 7 categories, student, faculty, course and project are four most populous entity-representing categories. The associated subset is typically called **WebKB4**.

Reuters The Reuters-21578 Text Categorization Test collection contains documents collected from the Reuters newswire in 1987. It is a standard text categorization benchmark and contains 135 categories. In our experiments, we use a subset of the data collection which includes the 10 most frequent categories among the 135 topics and we call it **Reuters-top 10**.

WebACE The K-dataset was from WebACE project and has been used for document clustering [6, 19]. The K-dataset contains 2340 documents consisting news articles from Reuters new service via the Web in October 1997. These documents are divided into 20 classes.

Log The log data used in our experiments are collected from several different machines with different operating systems using logdump2td (NT data collection tool) developed at IBM T.J. Watson Research Center. The data in the log files describe the status of each component and record system operational changes, such as the starting and stopping of services, detection of network applications, software configuration modifications, and software execution errors.

To pre-process the datasets, we remove the stop words using a standard stop list, all HTML tags are skipped and all header fields except subject and organization of the posted articles are ignored. In all our experiments, we first select the top 1000 words by mutual information with class labels.

8.2. Result Analysis

The above document datasets are standard labeled corpora widely used in the information retrieval literature. We view the labels of the datasets as the objective knowledge on the structure of the datasets. We use accuracy as the clustering performance measure. Accuracy discovers the one-to-one relationship between clusters and classes and measures the extent to which each cluster contained data points from the corresponding class. It sums up the whole matching de-

Datasets/Methods	K-Means	NMF	Semi-NMF	Convex-NMF	Tri-Factorization	PLSI
CSTR	0.4256	0.5713	0.5628	0.5340	0.604	0.587
WebKB4	0.3888	0.4418	0.4578	0.4658	0.483	0.503
Reuters	0.4448	0.4947	0.4867	0.4789	0.554	0.4870
WebACE	0.4001	0.4761	0.4162	0.4089	0.510	0.4890
Log	0.6876	0.7805	0.7385	0.7257	0.801	0.778

Table 4. Clustering Accuracy. Each entry is the clustering accuracy of the column method on the corresponding row dataset. The results obtained by averaging 5 trials.

gree between all pair class-clusters. Accuracy can be represented as:

$$Accuracy = Max(\sum_{C_k,L_m} T(C_k,L_m))/N, \qquad (20)$$

where C_k denotes the k-th cluster, and L_m is the m-th class. $T(C_k,L_m)$ is the number of entities which belong to class m are assigned to cluster k. Accuracy computes the maximum sum of $T(C_k,L_m)$ for all pairs of clusters and classes, and these pairs have no overlaps. The greater accuracy means the better clustering performance.

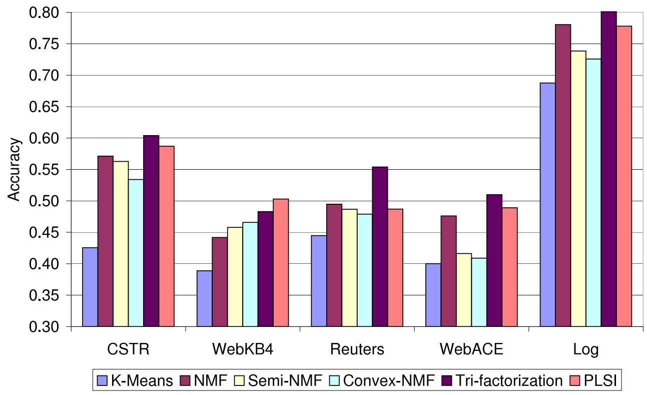

Figure 1. Clustering Accuracy Comparison

The experimental results are shown in Table 4 and Figure 1. From the experimental comparisons, we observe that:

- NMF-like algorithms generally outperform K-mean clustering algorithms. As we showed in Section 5.4, NMF is equivalent to soft K-means and the soft relaxation improves clustering performance.

- On most of the datasets, NMF gives somewhat better accuracy than semi-NMF and convex-NMF. The differences are modest, however, suggesting that the more highly-constrained semi-NMF and convex-NMF may be worthwhile options if interpretability is viewed as a goal of the data analysis.

- The experimental comparisons empirically verify the equivalence between NMF and PLSI. It can be observed from the table that NMF and PLSI usually lead to similar clustering results.

- Tri-Factorization generally is better than K-means and NMF-like algorithms on most of the datasets. The document datasets are of high dimension. Tri-Factorization provides a good framework for simultaneously clustering the rows and columns. Simultaneous clustering performs an implicit feature selection at each iteration, provides an adaptive metric for measuring the neighborhood, and thus tends to yield better clustering results.

- As we discussed in Section 7, Tri-Factorization enables simultaneous clustering of rows and columns and the multi-peak distribution evaluation approach enables the systematic analysis of word content. We take a closer look at the Log dataset and obtain the words in 1-peak, 2-peak, 3-peak and 4-peak categories respectively. The raw log files contain a free-format ASCII description of the event. We can derive meaningful common situations (i.e., row clustering) from the word cluster results. For example, situation **start** can be described by 1-peak words such as *start*, and *service*, and 2-peak words such as *version*. The situation **configure** can be described by 1-peak words such as *configuration*, two-peak words such as *product*, and 3-peak words such as *professional*. To summarize, the word clustering is capable of distinguishing the contents of words. The results of peak words are consistent with what we would expect from a systematic content analysis.

9. Summary

In this paper we provide a comparative study on (nonnegative) matrix factorization for clustering. Attempts have been made to establish the relations among various matrix factorization methods while highlighting their difference. Previously unaddressed yet important questions such as the interpretation and normalization of cluster posterior, convergence issues, and the benefits and evaluation of simultaneous clustering have also been studied. We expect our study could provide insightful guidances on matrix factorization research for clustering. In particular, the extensive research and experiments show that NMF provides a new paradigm for unsupervised learning.

Acknowledgments. We would like to Dr. Shenghuo Zhu for useful comments and discussion. T. Li is partially supported by a IBM Faculty Research Award, and the NSF CAREER Award IIS-0546280. C. Ding is partially supported by the US Dept of Energy, Office of Science.

References

[1] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. In *Proc. SIGMOD'99*, pages 61–72. ACM Press, 1999.

[2] D. Baier, W. Gaul, and M. Schader. Two-mode overlapping clustering with applications to simultaneous benefit segmentation and market structuring. In R. Klar and O. Opitz, editors, *Classification and Knowledge Organization*, pages 577–566. Springer, 1997.

[3] M. Berry, M. Browne, A. Langville, P. Pauca, and R. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *To Appear in Computational Statistics and Data Analysis*, 2006.

[4] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is nearest neighbor meaningful? In *Procs Int'l Conf. on Database Theory (ICDT'99)*, pages 217–235. 1999.

[5] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[6] D. Boley. Principal direction divisive partitioning. *Data mining and knowledge discovery*, 2:325–344, 1998.

[7] W. Castillo and J. Trejos. Two-mode partitioning: Review of methods and application and tabu search. In K. Jajuga, A. Sokolowski, and H.-H. Bock, editors, *Classification, Clustering and Data Analysis*, pages 43–51. Springer, 2002.

[8] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra. Minimum sum-squared residue co-clustering of gene experssion data. In *Proceedings of the SIAM Data Mining Conference*, 2004.

[9] M. Cooper and J. Foote. Summarizing video using nonnegative similarity matrix factorization. In *Proc. IEEE Workshop on Multimedia Signal Processing*, pages 25–28, 2002.

[10] I. S. Dhillon, S. Mallela, and S. S. Modha. Information-theoretic co-clustering. In *Proc. KDD 2003*, pages 89–98, 2003.

[11] C. Ding and X. He. K-means clustering and principal component analysis. *Int'l Conf. Machine Learning (ICML)*, 2004.

[12] C. Ding, X. He, and H. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. *Proc. SIAM Data Mining Conf*, 2005.

[13] C. Ding, T. Li, and M. Jordan. Convex and semi-nonnegative matrix factorizations for clustering and low-dimension representation. Technical Report LBNL-60428, Lawrence Berkeley National Laboratory, 2006.

[14] C. Ding, T. Li, and W. Peng. Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence, chi-square statistic, and a hybrid method. In *Proc. of National Conf. on Artificial Intelligence (AAAI-06)*, 2006.

[15] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proc SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 2006.

[16] E.F. Gonzales and Y. Zhang. Accelarating ithe Lee-Seung algorithms for nonnegative matrix factorization. *Dept. of Comp. and Applied Math., Rice University Tech Report*, 2005.

[17] G. Golub and C. Van Loan. *Matrix Computations, 3rd edition*. Johns Hopkins, Baltimore, 1996.

[18] G. Govaert. Simultaneous clustering of rows and columns. *Control and Cybernetics*, 24(4):437–458, 1995.

[19] E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. WebACE: A web agent for document categorization and exploration. In *Proc. Int'l Conf. on Autonomous Agents (Agents'98)* 1998.

[20] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, Prediction*. Springer, 2001.

[21] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 289–296, 1999.

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

[23] F. D. La Torre and T. Kanade. Discriminative cluster analysis. In *Proc. Int'l Conf. on Machine Learning (ICML 2006)*, 2006.

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

[25] D. Lee and H. S. Seung. Algorithms for non-negatvie matrix factorization. *Advances in Neural Information Processing Systems 13*, 2001.

[26] S. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. In *Proce IEEE Computer Vision and Pattern Recognition*, pages 207–212, 2001.

[27] B. Long, Z. Zhang, and P. Yu. Co-clustering by block value decomposition. In *KDD '05*, pages 635–640, 2005.

[28] V. Maurizio. Double k-means clustering for simultaneous classification of objects and variables. In S. Borra, R. Rocci, M. Vichi, and M. Schader, editors, *Advances in Classification and Data Analysis*, pages 43–52. Springer, 2001.

[29] 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.

[30] F. Sha, L. Saul, and D. Lee. Multiplicative updates for non-negative quadratic programming in support vector machines. In *Advances in Neural Information Processing Systems 15*, pages 1041–1048. 2003.

[31] L. Xu and M.I. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural Computation*, pages 129–151, 1996.

[32] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *SIGIR'03)*, pages 267–273, 2003.

[33] D. Zeimpekis and E. Gallopoulos. CLSI: A flexible approximation scheme from clustered term-document matrices. *Proc. SIAM Data Mining Conf*, pages 631–635, 2005.

[34] 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'01)*.

[35] H. Zha, X. He, C. Ding, M. Gu, and H. Simon. Bipartite graph partitioning and data clustering. *Proc. Int'l Conf. Information and Knowledge Management (CIKM 2001)*, 2001.