Nonnegative matrix factorization for interactive topic modeling and document clustering

Da Kuang and Jaegul Choo and Haesun Park

Abstract Nonnegative matrix factorization (NMF) approximates a nonnegative matrix by the product of two low-rank nonnegative matrices. Since it gives semantically meaningful result that is easily interpretable in clustering applications, NMF has been widely used as a clustering method especially for document data, and as a topic modeling method.

We describe several fundamental facts of NMF and introduce its optimization framework called block coordinate descent. In the context of clustering, our framework provides a flexible way to extend NMF such as the sparse NMF and the weakly-supervised NMF. The former provides succinct representations for better interpretations while the latter flexibly incorporate extra information and user feedback in NMF, which effectively works as the basis for the visual analytic topic modeling system that we present.

Using real-world text data sets, we present quantitative experimental results showing the superiority of our framework from the following aspects: fast convergence, high clustering accuracy, sparse representation, consistent output, and user interactivity. In addition, we present a visual analytic system called UTOPIAN (User-driven Topic modeling based on Interactive NMF) and show several usage scenarios.

Overall, our book chapter cover the broad spectrum of NMF in the context of clustering and topic modeling, from fundamental algorithmic behaviors to practical visual analytics systems.

Da Kuang
Georgia Institute of Technology e-mail: da.kuang@cc.gatech.edu

Jaegul Choo
Georgia Institute of Technology e-mail: jaegul.choo@cc.gatech.edu

Haesun Park
Georgia Institute of Technology e-mail: hpark@cc.gatech.edu

1 Introduction to Nonnegative Matrix Factorization

Nonnegative matrix factorization (NMF) is a dimension reduction method and factor analysis method. Many dimension reduction techniques are closely related to the low-rank approximations of matrices, and NMF is special in that the low-rank factor matrices are constrained to have only nonnegative elements. The nonnegativity reflects the inherent representation of data in many application areas, and the resulting low-rank factors lead to physically natural interpretations [33]. NMF was first introduced by Paatero and Tapper [43] as positive matrix factorization and subsequently popularized by Lee and Seung [33]. Over the last two decades, NMF has received enormous attention and has been successfully applied to a broad range of important problems in the areas including text mining [45, 52], computer vision [21, 37], bioinformatics [6, 11, 23], spectral data analysis [44], and blind source separation [10], and many others.

Suppose a nonnegative matrix $A \in \mathbb{R}^{m\times n}$ is given. When the desired lower dimension is k, the goal of NMF is to find the two matrices $W \in \mathbb{R}^{m\times k}$ and $H \in \mathbb{R}^{k\times n}$ having only nonnegative entries such that

$$A \approx WH. \tag{1}$$

According to (1), each data point, which is represented as the column of A, can be approximated by an additive combination of the nonnegative basis vectors, which are represented as the columns of W. As the goal of dimension reduction is to discover compact representation in the form of (1), k is assumed to satisfy that $k < \min\{m, n\}$. The matrices W and H are found by solving an optimization problem defined with the Frobenius norm (a distance measure between two given matrices), the Kullback-Leibler (KL) divergence (a distance measure between two probability distributions) [34, 36], or other divergences [12, 36]. In this book chapter, we focus on NMF based on the Frobenius norm, which is the most commonly used formulation:

$$\min_{W\geq 0, H\geq 0} f(W,H) = \|A - WH\|_F^2. \tag{2}$$

The constraints in (2) indicate that all the entries of W and H are nonnegative.

NMF with the formulation (2) has been very successful in partitional clustering, and many variations have been proposed for different settings such as constrained clustering and graph clustering [29, 23, 7, 38]. NMF especially performs well as a document clustering and topic modeling method. Due to an ever increasing amount of document data and the complexity involved with analyzing them in practice, revealing meaningful insights and thus guiding users in their decision-making processes has long been an active area of research. Document clustering is an important task in text mining with the goal of organizing a large text collection into several semantic clusters and helping users browse documents efficiently. Topic modeling is related to soft clustering where the documents are represented as a weighted combination of topics in terms of their proximity to each topic. In addition to its soft clustering aspect, topic modeling also deals with the semantic meaning of

each cluster/topic and models it as a weighted combination of keywords. Because of the nonnegativity constraints in NMF, the result of NMF can be viewed as document clustering and topic modeling results directly, which will be elaborated by theoretical and empirical evidences in this book chapter.

The goal of this book chapter is to provide an overview of NMF used as a clustering and topic modeling method for document data. We present a wide spectrum of material including the theoretical justification of NMF as a clustering method (Section 2), an algorithmic framework and extensions (Section 3), empirical performances and practical issues (Sections 4-5), as well as a visual analytic system called UTOPIAN (Section 6). Our emphasis is placed on NMF in the context of document clustering and topic modeling; however, the presented methodology applies to data types beyond text, for example, DNA microarray and RNA sequencing data in the biological domain.

We recommend the readers be familiar with linear algebra and numerical optimization theory.

Notations: Notations used in this book chapter are as follows. A lower-case letter, such as x, denotes a scalar; an upper-case letter, such as X, denotes a matrix; a bold-face lower-case letter, such as $\mathbf{x}$, denotes a column vector. We typically use i, j as indices: For example, $i \in \{1, \cdots, n\}$. The elements of a sequence of vectors or matrices are denoted by superscripts within parentheses, such as $X^{(1)}, \cdots, X^{(n)}$, and the entire sequence is denoted by $\{X^{(i)}\}$. The entries of a matrix are denoted by subscripts, such as x_{ij} for a matrix X. $X \geq 0$ indicates that the elements of X are nonnegative, i.e., X is a *nonnegative matrix*. $\mathbb{R}$ and $\mathbb{R}_+$ denote the set of real numbers and nonnegative real numbers, respectively. $\|\cdot\|_2$ and $\|\cdot\|_F$ denotes the L_2 norm and the Frobenius norm, respectively. The operator $.*$ denotes entrywise multiplication of matrices.

2 Nonnegative Matrix Factorization for Clustering

Dimension reduction and clustering are closely related. Consider the low-rank approximation in (1), where $A \in \mathbb{R}_+^{m \times n}$, $W \in \mathbb{R}_+^{m \times k}$, $H \in \mathbb{R}_+^{k \times n}$, and $k << \min(m,n)$ is the pre-specified lower rank. The columns of A represent n data points in an m-dimensional space. Each column of H is the k-dimensional representation of a data point. If we can use H to derive an assignment of the n data points into k groups, clustering can be viewed as a special type of dimension reduction. One example is the classical K-means clustering:

$$\min \sum_{i=1}^{n} \|\mathbf{a}_i - \mathbf{w}_{g_i}\|_2^2, \tag{3}$$

where $\mathbf{a}_1, \cdots, \mathbf{a}_n$ are the columns of A, $\mathbf{w}_1, \cdots, \mathbf{w}_k$ are the k centroids, and $g_i = j$ when the i-th data point is assigned to the j-th cluster ($1 \leq j \leq k$). Consider K-

means formulated as a dimension reduction problem [26]:

$$\min_{H\in\{0,1\}^{k\times n}, H^T\mathbf{1}_k=\mathbf{1}_n} \|A-WH\|_F^2, \tag{4}$$

where $\mathbf{1}_k \in \mathbb{R}^{k\times 1}$ and $\mathbf{1}_n \in \mathbb{R}^{n\times 1}$ are the column vectors whose elements are all 1's. In the K-means formulation for (4), the columns of W are the cluster centroids, and the single nonzero element in each column of H indicates the clustering assignment. Another example of dimension reduction is NMF:

$$\min_{W\geq 0, H\geq 0} \|A-WH\|_F^2.$$

In this formulation, the columns of W provide the basis of a latent k-dimensional space, and the columns of the second factor H provide the representation of $\mathbf{a}_1,\cdots,\mathbf{a}_n$ in the latent space. With only the nonnegativity constraints on H, this formulation can still be interpreted as clustering results: The columns of W are interpreted as k cluster representatives, and the i-th column of H contains the soft clustering membership of the i-th data point for the k clusters. NMF is best known for the *interpretability* of the latent space it finds [33]. In the case of document clustering and topic modeling, the basis vectors in W represent k topics, and the coefficients in the i-th column of H indicate the topic proportions for $\mathbf{a}_i$, the i-th document. To obtain a hard clustering result, we can simply choose the topic with the largest weight, i.e., the largest element in each column of H.

Historically, NMF has been extensively compared with K-means and singular value decomposition (SVD). We give several clarifications and caveats regarding using NMF as a clustering method. It has been shown that K-means and NMF have the equivalent form of an objective function, $\|A-WH\|_F^2$ [13]. However, each clustering method has its own conditions under which it performs well. K-means assumes that data points in each cluster follow a spherical Gaussian distribution [16]. In contrast, the NMF formulation (2) provides a better low-rank approximation of the data matrix A than the K-means formulation (4). If $k \leq \text{rank}(A)$, the columns of W are linearly independent due to $\text{rank}(A) \leq \text{nonnegative-rank}(A)$ [1] [3]. Therefore, NMF performs well when different clusters correspond to linearly independent vectors [29].

One caveat is that NMF does not always perform well as a clustering method. Consider the example in Fig. 1, where the two cluster centers are along the same direction and thus the two centroid vectors are linearly dependent. While NMF still approximates all the data points well in this example, no two linearly independent vectors in a two-dimensional space can represent the two clusters shown in Fig. 1. Since K-means and NMF have different conditions under which each of them does clustering well, they may generate very different clustering results in practice.

In contrast to NMF, rank-k SVD provides the best rank-k approximation but allows negative entries:

[1] The nonnegative rank of a matrix $X \in \mathbb{R}_+^{m\times n}$ is the smallest number $\hat{k}$ such that $X = WH$ where $W \in \mathbb{R}_+^{m\times \hat{k}}$ and $H \in \mathbb{R}_+^{\hat{k}\times n}$.

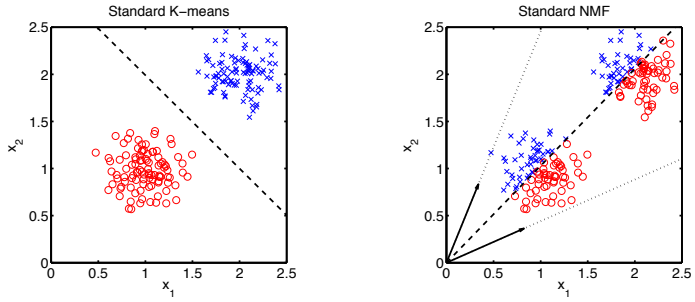


Fig. 1: An example with two ground-truth clusters, and the different clustering results given by K-means and NMF. The "o" and "x" markers in each figure indicate the cluster membership of the data points given by the clustering algorithm. The left figure shows that K-means correctly identifies the two clusters where the two centroids are linearly dependent. The right figure shows that NMF, on the contrary, uses two linearly independent vectors as cluster representatives marked by the two thick arrows, and leads to incorrect clustering results.

$$\min_{U^TU=I, V^TV=I} \|A - WY\|_F = \|A - U\Sigma V^T\|_F, \tag{5}$$

where $U \in \mathbb{R}^{m\times k}$, $\Sigma \in \mathbb{R}^{k\times k}$, and $V \in \mathbb{R}^{n\times k}$. Thus we cannot interpret the coefficients in the lower-dimensional space spanned by the columns of U as clustering assignments. In other words, the rows of V cannot be used as cluster indicators directly. Instead, an additional clustering method such as K-means has to be applied to a lower-dimensional representation of the data such as the rows of V to generate clusters.

The success of NMF as a clustering method depends on the underlying data set, and its greatest success has been in the area of document clustering [52, 45, 46, 38, 26, 15]. In a document data set, data points are often represented as unit-length vectors [40] and embedded in a linear subspace. For a term-document matrix A, a basis vector $\mathbf{w}_j$ is interpreted as the keyword-wise distribution of a single topic. When these distributions of k topics are linearly independent, which is usually the case, NMF can properly extract the ground-truth clusters determined by the true cluster labels.

Recently, NMF has been applied to topic modeling, a task related to document clustering, and achieved satisfactory results [2, 1]. Both document clustering and topic modeling can be considered as dimension reduction processes. Compared to standard topic modeling methods such as probabilistic latent semantic indexing (p-LSI) [20] and latent Dirichlet allocation (LDA) [5], NMF essentially gives the same output types: A keyword-wise topic representation (the columns of W), and a topic-wise document representation (the columns of H). The only difference, however, is that the columns of W and H do not have a unit L_1-norm unlike the p-LSI and LDA outputs. Nonetheless, such a difference is negligible in that (1) can be manipulated

via diagonal scaling matrices as

$$A \approx WH = (WD_W)(D_W^{-1}H) = \hat{W}\hat{H}, \tag{6}$$

where the diagonal component of the diagonal matrix $D_W \in \mathbb{R}_+^{k\times k}$ corresponds to the column sums of W. Now the new matrix $\hat{W}$ is column-normalized, giving an output analogous to the first outputs from p-LSI and LDA, but the second output $\hat{H}$ is still not column-normalized. The column normalization on H does not affect the interpretation of each document in terms of its relative relationships to topics. In this sense, NMF can be used as an alternative to topic modeling methods.

Note that NMF with KL-divergence is another commonly used formulation for topic modeling. It has a probabilistic interpretation and can be shown to be equivalent to p-LSI under certain constraints. However, algorithms for solving NMF with KL-divergence and p-LSI are typically much slower than those for solving NMF based on the Frobenius norm (2) [51]. Therefore, we focus on (2) in this chapter since there are many justified and efficient optimization algorithms developed for (2) in the literature.

3 Optimization Framework for Nonnegative Matrix Factorization

Although NMF is known as an NP-hard problem [49], one can still hope to find a local minimum as an approximate solution for NMF. In this section, we introduce an algorithmic framework to optimize the objective function (2), namely the *block coordinate descent* (BCD) framework. Multiplicative updating (MU) is another popular framework for solving NMF [33]. However, it has slow convergence and may lead to inferior solutions [18, 39]. In the later section, we will compare the solutions given by these two frameworks empirically and show the better clustering quality given by BCD.

3.1 Block Coordinate Descent Framework

The BCD framework is a widely-applicable strategy in nonlinear optimization problems. It divides variables into several disjoint subgroups and iteratively minimize the objective function with respect to the variables of each subgroup at a time. In the formulation of NMF (2), A is given as an input and the entries of W and H are the variables to be solved. A natural partitioning of the variables is the two blocks representing W and H, respectively. That is to say, we take turns solving

$$W \leftarrow \arg\min_{W \geq 0} f(W,H), \tag{7a}$$

$$H \leftarrow \arg\min_{H \geq 0} f(W,H). \tag{7b}$$

These subproblems can be written as

$$\min_{W \geq 0} \|H^T W^T - A^T\|_F^2, \tag{8a}$$

$$\min_{H \geq 0} \|WH - A\|_F^2. \tag{8b}$$

The subproblems (8) are called nonnegativity constrained least squares (NLS) problems [32], and the BCD framework has been called the alternating nonnegative least squares (ANLS) framework [39, 24]. It is summarized in Algorithm 1.

Algorithm 1 The BCD framework for solving NMF: $\min_{W,H \geq 0} \|A - WH\|_F^2$

1: Input: Matrix $A \in \mathbb{R}^{m \times n}$, tolerance parameter $0 < \varepsilon << 1$, upper limit of the number of iterations T
2: Initialize H
3: **repeat**
4: Obtain the optimal solution of subproblem (8a)
5: Obtain the optimal solution of subproblem (8b)
6: **until** A particular stopping criterion based on W, H, ε is satisfied *or* the number of iterations reaches upper limit T
7: Output: W, H

Note that we need to initialize H and solve (8a) and (8b) iteratively, as stated in Algorithm 1. Alternatively, we can also initialize W and solve (8b) and (8a) iteratively. Different initializations may lead to different solutions for NMF. A common strategy is to run an NMF algorithm starting from different random initializations and pick the solution with the smallest objective function value. Other strategies for initializing W and/or H were also proposed in the literature. For example, in the context of clustering, we can run spherical K-means, i.e. K-means with $1 - \mathbf{a}_i^T \mathbf{a}_j$ (one minus cosine similarity) as the distance function, and use the resulting centroids as the initialization of W [50].

Even though the subproblems are convex, they do not have a closed-form solution, and a numerical algorithm for the subproblems has to be provided. Many approaches for solving the NLS subproblems have been proposed in the NMF literature, e.g., an active-set method [24], a block principal pivoting [27, 28], a projected gradient descent [39], a quasi-Newton method [22]. We skip these details in this book chapter, but refer the readers to several software packages that solve NLS efficiently.[2,3]

[2] `http://www.cc.gatech.edu/hpark/nmfsoftware.php`

[3] `http://www.csie.ntu.edu.tw/cjlin/nmf/index.html`

3.1.1 Convergence Property

The objective function of NMF is a fourth-order polynomial with respect to W and H, and thus is nonconvex. For a nonconvex optimization problem, most algorithms only guarantee the stationarity of a limit point [4], not necessarily a local minimum. In practice, we often run an NMF algorithm in the BCD framework for multiple times with different initializations of W and H, and select the output with the smallest objective function value.

We have the following theorem regarding the convergence property of the BCD framework:

Theorem 1 *If a minimum of each subproblem in (8) is attained at each step, every limit point of the sequence $\{(W,H)^{(i)}\}$ generated by the BCD framework is a stationary point of (2).*

The BCD framework requires that the optimal solution be returned for each NLS subproblem. Note that the minimum of each subproblem is not required to be unique for the convergence result to hold because the number of blocks is two, as proved by Grippo and Sciandrone [19].

We remark that at a stationary point solution, the Karush-Kuhn-Tucher (KKT) condition is satisfied:

$$W \geq 0, \quad H \geq 0, \tag{9a}$$

$$\nabla f_W = 2WHH^T - 2AH^T \geq 0, \quad \nabla f_H = 2W^TWH - 2W^TA \geq 0, \tag{9b}$$

$$W .* \nabla f_W = 0, \quad H .* \nabla f_H = 0. \tag{9c}$$

In contrast, the MU algorithm does not have the convergence property stated in Theorem 1. We consider the MU algorithm proposed by Lee and Seung [34]. This algorithm has an advantage of being simple and easy to implement, and it has contributed greatly to the popularity of NMF. Though it also has a form of updating W and H alternately, it is different from the BCD framework in the sense that its solutions for the subproblems (8) are not optimal. That is, the update rule of MU is:

$$W \leftarrow W .* \frac{AH^T}{WHH^T}, \quad H \leftarrow H .* \frac{W^TA}{W^TWH}, \tag{10}$$

where the division operator indicates entrywise division. This update rule can be seen as a gradient descent algorithm with specifically chosen step lengths. The step lengths are conservative enough so that the result is always nonnegative. However, we cannot achieve the optimal solution of every NLS subproblem using this update rule.

Lee and Seung [34] showed that under the update rule (10), the objective function in NMF (2) is non-increasing. However, it is unknown whether it converges to a stationary point or a local minimum [18]. In fact, even though the papers using MU algorithms claimed that the solution satisfied the KKT condition, such as in [14], often their proofs did not include all the components of the KKT condition in (9), for example, the sign of the gradients (9b).

Furthermore, since the values are updated only through multiplications in MU algorithms, the entries of W and H typically remain nonzero. Hence, their solution matrices typically are denser than those from algorithms in the BCD framework empirically, and thus it is harder to interpret the solution matrices as clustering results.

3.1.2 Stopping Criterion

Iterative methods have to be equipped with a criterion for stopping iterations. A naive approach is to stop when the decrease of the objective function becomes smaller than a pre-defined threshold:

$$|f(W^{(i-1)},H^{(i-1)}) - f(W^{(i)},H^{(i)})| \leq \varepsilon. \tag{11}$$

Although this method is commonly adopted, it is potentially misleading because the decrease of the objective function may become small before a stationary point is achieved. A more principled criterion was proposed by Lin [39] as follows. Recall the KKT condition (9) for the objective function of NMF. Let us define the projected gradient $\nabla^P f_W \in \mathbb{R}^{m\times k}$ as

$$(\nabla^P f_W)_{ij} = \begin{cases} \nabla(f_W)_{ij}, & \text{if } (\nabla f_W)_{ij} < 0 \text{ or } W_{ij} > 0; \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

for $i = 1,\cdots,m$ and $j = 1,\cdots,k$, and $\nabla^P f_H$ similarly. Then, conditions (9) can be rephrased as

$$\nabla^P f_W = 0 \text{ and } \nabla^P f_H = 0. \tag{13}$$

Let us denote the projected gradient matrices at the i-th iteration by $\nabla^P f_W^{(i)}$ and $\nabla^P f_H^{(i)}$ and define

$$\Delta(i) = \sqrt{\|\nabla^P f_W^{(i)}\|_F^2 + \|\nabla^P f_H^{(i)}\|_F^2}. \tag{14}$$

Using this definition, the stopping criterion is written by

$$\frac{\Delta(i)}{\Delta(1)} \leq \varepsilon, \tag{15}$$

where $\Delta(1)$ is from the first iterate of (W,H). Unlike (11), (15) guarantees the stationarity of the final solution. For caveats when using (15), see [25].

3.2 Extension 1: Sparse NMF

With only nonnegativity constraints, the resulting factor matrix H of NMF contains the fractional assignment values corresponding to the k clusters represented by the columns of W. Sparsity constraints on H have been shown to facilitate the interpre-

tation of the result of NMF as a hard clustering result and improve the clustering quality [21, 23, 26]. For example, consider two different scenarios of a column of $H \in \mathbb{R}_+^{3\times n}$: $(0.2, 0.3, 0.5)^T$ and $(0, 0.1, 0.9)^T$. Clearly, the latter is a stronger indicator that the corresponding data point belongs to the third cluster.

To incorporate extra constraints or prior information into the NMF formulation (2), various regularization terms can be added. We can consider an objective function

$$\min_{W,H\geq 0} \|A - WH\|_F^2 + \phi(W) + \psi(H), \tag{16}$$

where $\phi(\cdot)$ and $\psi(\cdot)$ are regularization terms that often involve matrix or vector norms. The L_1-norm regularization can be adopted to promote sparsity in the factor matrices [47, 23]. When sparsity is desired on H, the L_1-norm regularization can be set as

$$\phi(W) = \alpha\|W\|_F^2 \text{ and } \psi(H) = \beta\sum_{i=1}^{n}\|H(:,i)\|_1^2, \tag{17}$$

where $H(:,i)$ represents the i-th column of H. The L_1-norm term of $\psi(H)$ in (17) promotes sparsity on the columns of H while the Frobenius norm term of $\phi(W)$ is needed to prevent W from growing too large. Scalar parameters α and β are used to control the strength of regularization.

The sparse NMF can be easily computed using the BCD framework. We can reorganize the terms in the sparse NMF formulation (16) and (17) and the two subproblems in the BCD framework become:

$$\min_{W\geq 0} \left\| \begin{pmatrix} H^T \\ \sqrt{\alpha} I_k \end{pmatrix} W^T - \begin{pmatrix} A^T \\ 0_{k\times m} \end{pmatrix} \right\|_F^2, \tag{18a}$$

$$\min_{H\geq 0} \left\| \begin{pmatrix} W \\ \sqrt{\beta}\mathbf{1}_k^T \end{pmatrix} H - \begin{pmatrix} A \\ \mathbf{0}_n^T \end{pmatrix} \right\|_F^2. \tag{18b}$$

where $\mathbf{1}_k \in \mathbb{R}^{k\times 1}, \mathbf{0}_n \in \mathbb{R}^{n\times 1}$ are the column vectors whose elements are all 1's and 0's, respectively, and I_k is a $k \times k$ identity matrix. Hence, the two subproblems (18a) and (18b) for the sparse NMF can be solved as NLS problems, similar to Algorithm 1 for the original NMF.

3.3 Extension 2: Weakly-Supervised NMF

The flexible BCD framework allows another important variant called weakly-supervised NMF (WS-NMF).[4] WS-NMF can incorporate a variety of user inputs so that that the clustering and topic modeling results of NMF can be improved in a user-driven way. In this section, we describe the formulation and the algorithm of

[4] The term "weakly-supervised" can be considered similar to semi-supervised clustering settings, rather than supervised learning settings such as classification and regression problems.

WS-NMF. Later in Section 6, we further discuss how WS-NMF can be utilized to support various types of user interactions in a visual analytics environment.

In WS-NMF, such user inputs are manifested in the form of reference matrices for W and H. These reference matrices play a role of making W and H become similar to them. That is, given reference matrices $W_r \in \mathbb{R}_+^{m\times k}$ for W and $H_r \in \mathbb{R}_+^{k\times n}$ for H, diagonal mask/weight matrices $M_W \in \mathbb{R}_+^{k\times k}$ and $M_H \in \mathbb{R}_+^{n\times n}$, a data matrix $A \in \mathbb{R}_+^{m\times n}$, and an integer $k \ll \min(m,n)$, WS-NMF has the additional regularization terms that penalize the differences between H_r and H (up to a column-wise scaling via D_H) and those between W_r and W as

$$f(W,H,D_H) = \min_{W,H,D_H} \|A-WH\|_F^2 + \|(W-W_r)M_W\|_F^2 + \|(H-H_rD_H)M_H\|_F^2 \tag{19}$$

for $W \in \mathbb{R}_+^{m\times k}$ and $H \in \mathbb{R}_+^{k\times n}$ and a diagonal matrix $D_H \in \mathbb{R}_+^{n\times n}$.

Through these regularization terms, WS-NMF can incorporate various types of users' prior knowledge. Each column of H_r specifies the soft clustering membership of data items. A diagonal matrix D_H accounts for a possible scaling different between H_r and H and is a variable to be computed. For example, two vectors, $(0.1, 0.3, 0.6)$ and $(0.2, 0.6, 1.2)$, are interpreted the same in terms of cluster membership coefficients, and D_H allows them to be treated as same. WS-NMF also supports partial supervision on a subset of column (or data items) in H_r. The diagonal matrix M_H achieves this by masking/down-weighting the columns or data items in H_r with no prior information.

Next, W_r supervise the basis representations. In document clustering and topic modeling, the columns of W_r specify the keyword-wise topic representations in W. Similar to the role of M_H for the partial supervision on H, the diagonal matrix M_W allows the supervision on a subset of columns in W by masking/down-weighting those columns in W_r with no supervision. However, unlike the supervision on H, the regularization on W via W_r does not involve any diagonal matrix analogous to D_H because scaling on either W or H suffices due to the relationship (6), which indicates that if W and H are the solution of a particular NMF problem, then so are WD and $D^{-1}H$ for an arbitrary diagonal matrix D.

Finally, note that (19) does not have typical regularization parameters that balance between different terms since M_W and M_H can account for the effects of the parameters. In other words, $\alpha\|(W-W_r)M_W\|_F^2$ is equivalent to $\left\|(W-W_r)M_W^{new}\right\|_F^2$ when $M_W^{new} = \alpha M_W$, and the same argument holds for M_H.

The optimization of (19) follows the BCD framework by iteratively solving W, H, and D_H. Given initial values for these variables, W is updated as

$$W \leftarrow \arg\min_{W\geq 0} \left\| \begin{bmatrix} H^T \\ M_W \end{bmatrix} W^T - \begin{bmatrix} A^T \\ M_W W_r^T \end{bmatrix} \right\|_F^2. \tag{20}$$

Next, each column of H is updated one at a time by solving

$$H(:,i) \leftarrow \arg\min_{H(:,i)\geq 0} \left\| \begin{bmatrix} W \\ M_H(i)I_k \end{bmatrix} H(:,i) - \begin{bmatrix} A(:,i) \\ M_H(i)D_H(i)H_r(:,i) \end{bmatrix} \right\|_F^2, \tag{21}$$

where $(:, i)$ indicates the i-th column of a matrix. Finally, the i-th diagonal component $D_H(i)$ of D_H is obtained in a closed form as

$$D_H(i) \leftarrow \begin{cases} \frac{H_r(:,i)^T \cdot H(:,i)}{\|H_r(:,i)\|_2^2} & \text{if } M_H(i) \neq 0 \\ 0 & \text{otherwise} \end{cases}. \tag{22}$$

4 Choosing the Number of Clusters

The number of clusters k is one of the input parameters that NMF requires. It is often difficult to choose an appropriate k to achieve the best clustering quality. In this section, we introduce our method to choose k based on random sampling and consensus clustering.

Monti et al. [42] proposed a model selection method that used the notion of *stability* of the clusters with respect to random sampling of the data points. Let $\mathscr{A}_1$ and $\mathscr{A}_2$ be two subsets sampled randomly from a data set of n data points. Suppose two data points $\mathbf{a}_i$ and $\mathbf{a}_j$ appear in both subsets generated by random sampling, that is to say, $\mathbf{a}_i, \mathbf{a}_j \in \mathscr{A}_1 \cap \mathscr{A}_2$. Let us run a clustering algorithm on both $\mathscr{A}_1$ and $\mathscr{A}_2$, and the correct number of clusters k is given. Conceptually, we expect that if $\mathbf{a}_i$ and $\mathbf{a}_j$ belong to the same cluster derived from $\mathscr{A}_1$, they also belong to the same cluster derived from $\mathscr{A}_2$. Based on this reasoning, Monti et al. [42] proposed *consensus clustering* to aggregate the results of a clustering method over many runs and achieve a consensus partitioning of data points.

We formulate the idea of a *consensus matrix* in the context of NMF-based document clustering. For a data set with n documents, the (i, j)-th entry of a consensus matrix $\tilde{C} \in \mathbb{R}^{n \times n}$ is the co-clustered frequency of the i-th and j-th documents over multiple runs of NMF. More rigorously, let r be the sampling rate, the fraction of documents selected in each random sample. We generate T subsets $\mathscr{A}_1, \cdots, \mathscr{A}_T$ by random sampling, each with sampling rate r, and run an NMF algorithm on each subset with the same number of clusters k. Define the matrices $C^{(t)}$ and $S^{(t)}$ as the following ($1 \leq t \leq T$):

$$c_{ij}^{(t)} = \begin{cases} 1, \text{ if the } i\text{-th and the } j\text{-th documents belong to the same cluster using } \mathscr{A}_t; \\ 0, \text{ otherwise,} \end{cases} \tag{23}$$

$$s_{ij}^{(t)} = \begin{cases} 1, \text{ if both the } i\text{-th and the } j\text{-th documents appear in } \mathscr{A}_t; \\ 0, \text{ otherwise.} \end{cases} \tag{24}$$

Clearly, $c_{ij}^{(t)} = 1$ implies $s_{ij}^{(t)} = 1$. Then we can define the consensus matrix $\tilde{C}$:

$$\tilde{c}_{ij} = \frac{\sum_{t=1}^{T} c_{ij}^{(t)}}{\sum_{t=1}^{T} s_{ij}^{(t)}}. \tag{25}$$

The entries of $\tilde{C}$ have values in the interval $[0,1]$. In the ideal scenario where no ambiguity exists for the co-membership of any two documents, the entries of $\tilde{C}$ could be 0 or 1 only. To measure the dispersion of a consensus matrix $\tilde{C}$, we define the dispersion coefficient ρ as:

$$\rho = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} 4(\tilde{c}_{ij} - 0.5)^2. \tag{26}$$

For an ideal consensus matrix where all the entries are 0 or 1, we have $\rho = 1$; for a scattered consensus matrix, $0 \leq \rho < 1$. After obtaining ρ_k values for various k's, we can determine the number of clusters as the one with the maximal ρ_k.

Now we illustrate the above method for choosing the number of clusters with a real-world text data set. We extracted the three largest clusters based on ground-truth labels from the entire TDT2 data set. For running NMF, we applied the ANLS algorithm with block principal pivoting [27, 28] solving the NLS subproblems. To construct the consensus matrix, we used the parameters $T = 50$ and $r = 0.8$ for $k = 2,3,4,5$. Table 1 shows the dispersion coefficients for these k's. We can see that $k = 3$ corresponds to the largest ρ and is thus chosen as the most appropriate number of clusters.

Table 1: Dispersion coefficients for $k = 2,3,4,5$ using the three largest clusters based on ground-truth labels from the TDT2 data set.

	$k=2$	$k=3$	$k=4$	$k=5$
$\rho(k)$	0.5642	0.9973	0.8515	0.9411

Note that our method for choosing the number of clusters differs from the work of Brunet et al. [6] in two aspects. First, the authors of [6] assessed the stability of clustering results with respect to random initialization of NMF. In contrast, our method reveals the stability of the cluster structure by examining whether the clusters can be reproduced using a random sample of the data points. Second, the rows and the columns of the consensus matrix were reordered in [6], and if the reordered matrix exhibited a block-diagonal structure, the number of clusters was determined to be appropriate. However, the optimal reordering was obtained by a hierarchical clustering of the items using the consensus matrix as similarity values between all the item pairs. Thus, it was very expensive to compute for large-scale data sets. We experienced difficulty in computing the optimal reordering for a few thousand documents. Therefore, we did not adopt the model selection method in [6] but rather used the dispersion coefficient (26) to assess the stability of clusters.

5 Experimental Results

In this section, we present the empirical evidences that support NMF as a successful document clustering and topic modeling method. We compare the clustering quality between K-means and NMF; Within the NMF algorithms, we compare the multiplicative updating (MU) algorithm and the alternating nonnegative least squares (ANLS) algorithm in terms of their clustering quality and convergence behavior, as well as sparseness and consistency in the solution.

5.1 Data Sets and Algorithms

We used the following text corpora in our experiments. All these corpora have ground-truth labels for evaluating clustering quality but not given as an input to the clustering algorithms.

1. **TDT2** contains 10,212 news articles from various sources (e.g., NYT, CNN, and VOA) in 1998.
2. **Reuters**[5] contains 21,578 news articles from the Reuters newswire in 1987.
3. **20 Newsgroups**[6] (20News) contains 19,997 posts from 20 Usenet newsgroups. Unlike previous indexing of these posts, we observed that many posts have duplicated paragraphs due to cross-referencing. We discarded cited paragraphs and signatures in a post by identifying lines starting with "$>$" or "$--$". The resulting data set is less tightly-clustered and much more difficult to apply clustering or classification methods.
4. From the more recent Reuters news collection **RCV1**[7] [35] that contains over 800,000 articles in 1996-1997, we selected a subset of 23,149 articles. Labels are assigned according to a topic hierarchy, and we only considered leaf topics as valid labels.
5. The research paper collection **NIPS14-16**[8] contains NIPS papers published in 2001-2003 [17], which are associated with labels indicating the technical area (algorithms, learning theory, vision science, etc).

For all these data sets, documents with multiple labels are discarded in our experiments. In addition, the ground-truth clusters representing different topics are highly unbalanced in their sizes for TDT2, Reuters, RCV1, and NIPS14-16. We selected the largest 20, 20, 40, and 9 ground-truth clusters from these data sets, respectively. We constructed term-document matrices using tf-idf features [40], where each row

[5] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[6] http://qwone.com/jason/20Newsgroups/

[7] http://jmlr.csail.mit.edu/papers/volume5/lewis04a/lyrl2004rcv1v2README.htm

[8] http://robotics.stanford.edu/gal/data.html

corresponds to a term and each column to a document. We removed any term that appears less than three times and any document that contains less than five words. Table 2 summarizes the statistics of the five data sets after pre-processing. For each data set, we set the number of clusters to be the same as the number of ground-truth clusters.

Table 2: Data sets used in our experiments.

Data set	# Terms	# Documents	# Ground-truth clusters
TDT2	26,618	8,741	20
Reuters	12,998	8,095	20
20 Newsgroups	36,568	18,221	20
RCV1	20,338	15,168	40
NIPS14-16	17,583	420	9

We further process each term-document matrix A in two steps. First, we normalize each column of A to have a unit L_2-norm, i.e., $\|\mathbf{a}_i\|_2 = 1$. Conceptually, this makes all the documents have equal lengths. Next, following [52], we compute the normalized-cut weighted version of A:

$$D = \text{diag}(A^T A \mathbf{1}_n), \quad A \leftarrow AD^{-1/2}, \tag{27}$$

where $\mathbf{1}_n \in \mathbb{R}^{n\times 1}$ is the column vector whose elements are all 1's, and $D \in \mathbb{R}_+^{n\times n}$ is a diagonal matrix. This column weighting scheme was reported to enhance the clustering quality of both K-means and NMF [52, 30].

For K-means clustering, we used the standard K-means with Euclidean distances. The Matlab `kmeans` function has a batch-update phase that re-assigns the data points all at once in each iteration, as well as a more time-consuming online-update phase that moves a single data point each time from one cluster to another if such a movement reduces the sum of squared error [16]. We used both phases and rewrote this function using BLAS3 operations and boosted its efficiency substantially.[9]

For the ANLS algorithm for NMF, we used the block principal pivoting algorithm[10] [27, 28] to solve the NLS subproblems (8) and the stopping criterion (15) with $\varepsilon = 10^{-4}$. For the MU algorithm for NMF, we used the update formula in (10). The MU algorithm is not guaranteed to converge to a stationary point and thus could not satisfy the stopping criterion in (15) after a large number of iterations in our experiments. Therefore, we used another stopping criterion

$$\|H^{(i-1)} - H^{(i)}\|_F / \|H^{(i)}\|_F \leq \varepsilon \tag{28}$$

with $\varepsilon = 10^{-4}$ to terminate the algorithm.

For the sparse NMF, we used the formulations (16) and (17). The choice of the parameters α, β that control the regularization strength and the sparsity of the so-

[9] `http://www.cc.gatech.edu/dkuang3/software/kmeans3.html`

[10] `https://github.com/kimjingu/nonnegfac-matlab`

lution can be determined by cross validation, for example, by tuning α, β until the desired sparseness is reached. Following [23, 24], we set α to the square of the maximum entry in A and $\beta = 0.01$ since these choices have been shown to work well in practice.

5.2 Clustering Quality

We used two measures to evaluate the clustering quality against the ground-truth clusters.

Clustering accuracy is the percentage of correctly clustered items given by the maximum bipartite matching (see more details in [52]). This matching associates each cluster with a ground-truth cluster in an optimal way and can be found by the Kuhn-Munkres algorithm [31].

Normalized mutual information (NMI) is an information-theoretic measure of the similarity between two flat partitionings [40], which, in our case, are the ground-truth clusters and the generated clusters. It is particularly useful when the number of generated clusters is different from that of ground-truth clusters or when the ground-truth clusters have highly unbalanced sizes or a hierarchical labeling scheme. It is calculated by:

$$\text{NMI} = \frac{I(C_{\text{ground-truth}}, C_{\text{computed}})}{\left[H(C_{\text{ground-truth}}) + H(C_{\text{computed}})\right]/2} = \frac{\sum_{h,l} n_{h,l} \log \frac{n \cdot n_{h,l}}{n_h n_l}}{\left(\sum_h n_h \log \frac{n_h}{n} + \sum_l n_l \log \frac{n_l}{n}\right)/2}, \tag{29}$$

where $I(\cdot,\cdot)$ denotes mutual information between two partitionings, $H(\cdot)$ denotes the entropy of a partitioning, and $C_{\text{ground-truth}}$ and C_{computed} denote the partitioning corresponding to the ground-truth clusters and the computed clusters, respectively. n_h is the number of documents in the h-th ground-truth cluster, n_l is the number of documents in the l-th computed cluster, and $n_{h,l}$ is the number of documents in both the h-th ground-truth cluster and the l-th computed cluster.

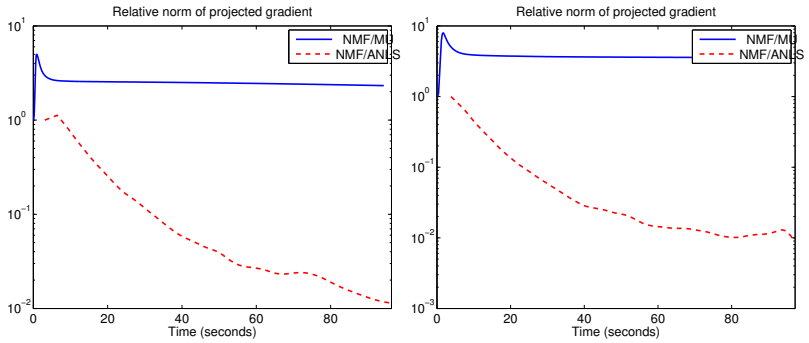
Tables 3 and 4 show the clustering accuracy and NMI results, respectively, averaged over 20 runs with random initializations. All the NMF algorithms have the same initialization of W and H in each run. We can see that all the NMF algorithms consistently outperform K-means except one case (clustering accuracy evaluated on the Reuters data set). Considering the two algorithms for standard NMF, the clustering quality of NMF/ANLS is either similar to or much better than that of NMF/MU. The clustering quality of the sparse NMF is consistently better than that of NMF/ANLS except on the 20 Newsgroups data set and always better than NMF/MU.

Table 3: The average clustering accuracy given by the four clustering algorithms on the five text data sets.

	K-means	**NMF/MU**	**NMF/ANLS**	**Sparse NMF/ANLS**
TDT2	0.6711	0.8022	0.8505	0.8644
Reuters	0.4111	0.3686	0.3731	0.3917
20News	0.1719	0.3735	0.4150	0.3970
RCV1	0.3111	0.3756	0.3797	0.3847
NIPS14-16	0.4602	0.4923	0.4918	0.4923

Table 4: The average normalized mutual information given by the four clustering algorithms on the five text data sets.

	K-means	**NMF/MU**	**NMF/ANLS**	**Sparse NMF/ANLS**
TDT2	0.7644	0.8486	0.8696	0.8786
Reuters	0.5103	0.5308	0.5320	0.5497
20News	0.2822	0.4069	0.4304	0.4283
RCV1	0.4092	0.4427	0.4435	0.4489
NIPS14-16	0.4476	0.4601	0.4652	0.4709



(a) 20 Newgroups (b) RCV1

Fig. 2: The convergence behavior of NMF/MU and NMF/ANLS on the 20 Newsgroups data set ($k = 20$) and RCV1 data set ($k = 40$).

5.3 Convergence Behavior

Now we compare the convergence behaviors of NMF/MU and NMF/ANLS. Fig. 2 shows the relative norm of projected gradient $\Delta/\Delta(1)$ as the algorithms proceed on the 20 Newsgroups and RCV1 data sets. The quantity $\Delta/\Delta(1)$ is not monotonic in general but is used to check stationarity and determine whether to terminate the algorithms. On both data sets, the norm of projected gradient for NMF/ANLS has a decreasing trend and eventually reached the given tolerance ε, while NMF/MU did not converge to stationary point solutions. This observation is consistent with

the result that NMF/ANLS achieved better clustering quality and sparser low-rank matrices.

5.4 Sparseness

We also compare the sparseness in the W and H matrices between the solutions of NMF/MU, NMF/ANLS, and the sparse NMF/ANLS. Table 5 shows the percentage of zero entries for the three NMF versions. [11] Compared to NMF/MU, NMF/ANLS does not only lead to better clustering quality and smaller objective values, but also facilitates sparser solutions in terms of both W and H. Recall that each column of W is interpreted as the term distribution for a topic. With a sparser W, the keyword-wise distributions for different topics are more orthogonal, and one can select important terms for each topic more easily. A sparser H can be interpreted as clustering indicators more easily. Table 5 also validates that the sparse NMF generates an even sparser H in the solutions and often better clustering results.

Table 5: The average sparseness of W and H for the three NMF algorithms on the five text data sets. $\%(\cdot)$ indicates the percentage of the matrix entries that satisfy the condition in the parentheses.

	NMF/MU		**NMF/ANLS**		**Sparse NMF/ANLS**	
	$\%(w_{ij}=0)$	$\%(h_{ij}=0)$	$\%(w_{ij}=0)$	$\%(h_{ij}=0)$	$\%(w_{ij}=0)$	$\%(h_{ij}=0)$
TDT2	21.05	6.08	55.14	50.53	52.81	65.55
Reuters	40.92	12.87	68.14	59.41	66.54	72.84
20News	46.38	15.73	71.87	56.16	71.01	75.22
RCV1	52.22	16.18	77.94	63.97	76.81	76.18
NIPS14-16	32.68	0.05	50.49	48.53	49.90	54.49

5.5 Consistency from Multiple Runs

We analyze the consistency of the clustering results obtained from multiple runs of a particular method. We have chosen three methods: K-means, LDA,[12] and NMF/ANLS. The detailed procedure is as follows. First, we run each method multiple times, e.g., 30 times in our experiment. Second, for each pair of different runs, e.g., 435 cases, we measure the relative number of documents of which the (hard) clustering membership results differ from each other. To solve the correspondence

[11] Results given by the sparseness measure based on L_1 and L_2 norms in [21] are similar in terms of comparison between the three NMF versions.

[12] For LDA, we used Mallet [41], a widely-accepted software library based on a Gibbs sampling method.

between the two set of cluster indices generated independently from multiple runs, we apply the Kuhn-Munkres algorithm [31] before comparing the clustering memberships. Finally, we compute the consistency measure as the average value of the relative numbers of documents over all the pairs of runs, e.g., 435 cases.

Table 6 shows the results of these consistency measures for the five text data sets. It can be seen that NMF/ANLS generates the most consistent results from multiple runs compared to K-means and LDA. Combined with the accuracy and NMI results shown in Tables 3 and 4, this indicates that NMF generally produces the best clustering result with the least amount of variance. On the other hand, K-means or LDA may require users to check many results by running them multiple times until finding satisfactory results.

Table 6: The consistency measure of three clustering algorithms on the five text data sets.

	K-means	**LDA**	**NMF/ANLS**
TDT2	0.6448	0.7321	0.8710
Reuters	0.6776	0.6447	0.8534
20News	0.7640	0.6166	0.7244
RCV1	0.6636	0.5996	0.7950
NIPS14-16	0.6421	0.5352	0.8399

6 UTOPIAN: User-driven Topic Modeling via Interactive NMF

In this section, we present a visual analytics system called UTOPIAN (User-driven Topic Modeling Based on Interactive NMF)[13] [8], which utilizes NMF as a main tool to steer topic modeling results in a user-driven manner. As seen in Fig. 3, UTOPIAN provides a visual overview of the NMF topic modeling result as a 2D scatter plot where dots represent documents. The color of each dot corresponds to the topic/clustering membership computed by NMF. The position of each dot is determined by running a modified version [8] of t-distributed stochastic neighborhood embedding [48] to the cosine similarity matrix of bag-of-words vectors of documents. Additionally, the topics are summarized as their representative keywords.

Beyond the visual exploration of the topic modeling result in a passive manner, UTOPIAN provides various interaction capabilities that can actively incorporate user inputs to topic modeling processes. The interactions supported by UTOPIAN include topic keyword refinement, topic merging/splitting, and topic creation via seed documents/keywords, all of which are built upon WS-NMF. In the following, we describe each interaction in more detail.

[13] `http://tinyurl.com/2013utopian`

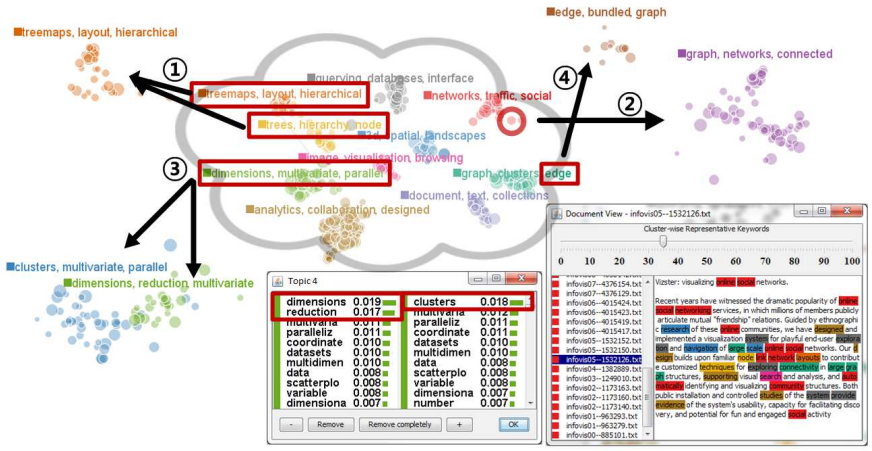


Fig. 3: An overview of UTOPIAN. Given a scatter plot visualization generated by a modified t-distributed stochastic neighborhood embedding, UTOPIAN provides various interaction capabilities: (1) topic merging, (2) document-induced topic creation, (3) topic splitting, and (4) keyword-induced topic creation. Additionally, the user can refine topic keyword weights (not shown here). The document viewer highlights the representative keywords from each topic.

Topic keyword refinement. In the topic modeling using NMF, the i-th topic, which corresponds to the i-th column vector $W^{(i)}$ of W is represented as a weighted combination of keywords. This interaction allows users to change the weights corresponding to keywords, corresponding to each component of the vector $W^{(i)}$, so that the meaning of the topic can be refined. For instance, users might want to remove some of the uninformative terms by setting its weight value to zero. In addition, users could increase/decrease the weight of a particular keyword to make the corresponding topic more/less relevant to the keyword. In turn, this refined vector $W^{(i)}$ is placed in the the corresponding i-th column vector of W_r in (19) as the reference information during the subsequent WS-NMF. We also set a nonzero value of $M_W^{(l)}$ to make this reference vector in effect.

Topic merging. This interaction merges multiple topics into one. To this end, we utilize the reference information H_r for H as follows. We first interpret the columns of H as hard clustering results and identify the set of documents clustered to one of the merged topics. For these documents, we obtain their $H^{(i)}$'s and merge the values corresponding to the merged topics by adding them up to a single value, and set the corresponding columns of H_r to the resulting $H^{(i)}$'s. For example, suppose two documents, whose $H^{(i)}$'s are represented as (0.6, 0.3, 0.1) and (0.4, 0.5, 0.1), respectively, corresponding to the three original topics. The corresponding column of H_r will be set to (0.6+0.4, 0.1) and (0.3+0.5, 0.1), respectively, where the first component corresponds to the merged topic. Alternatively, for topic merging, one

could use the reference information for W, but we found our approach empirically works better.

Topic splitting. UTOPIAN also support a topic splitting interaction. It splits a particular topic, e.g., $W^{(i)}$ of W , into the two topics. To guide this splitting process, users can assign the reference information for the two topics as follows. First, both vectors are initialized as the same as $W^{(i)}$. Now users can specify these two topics differently using the topic keyword refinement interaction.. In this manner, the topic splitting process in WS-NMF is guided by users based on the differently weighted keyword-wise representations of the two topics.

Document-induced topic creation. This interaction creates a new topic by using user-selected documents as seed documents. For example, such seed documents can be a person's own research papers and s/he might want to see the topic formed by them as well as other papers relevant to this topic. To achieve this interaction, we utilize the reference information for documents. That is, for those documents specified by the user, their corresponding vectors in H_r in 19 are initialized to zero vectors but are set to one for the corresponding component to the newly created topic. This generates the reference information such that these documents are related only to the newly created topic. WS-NMF then creates a new topic based on it, which is represented as a keyword-wise distribution, and as a result, other relevant documents can be included.

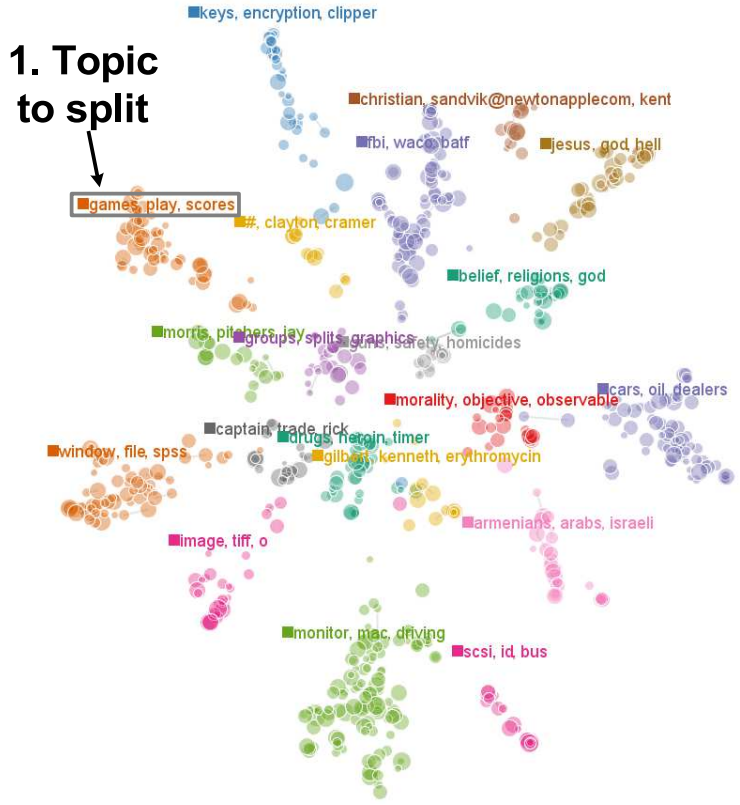
Keyword-induced topic creation. It creates a new topic via user-selected keywords. For instance, given the summary of topics as their representative keywords, users might want to explore more detailed (sub-)topics about particular keywords. A new topic created using these keywords would reveal such information. This interaction works similarly to document-induced topic creation except that we now use the reference information for keywords. Given user-selected keywords, the reference information of a new topic, i.e., a newly added column vector of W_r, is set to a zero vector, but the components corresponding to the keywords are set to ones. Accordingly, WS-NMF will include related documents in this topic, which, in turn, reveals the details about this topic.

In all the above-described interaction capabilities, UTOPIAN provides slider user interfaces with which users can interactively control how strongly the supervision is imposed. The values specified via these user interfaces are used as those for the nonzero elements in M_W and M_H.

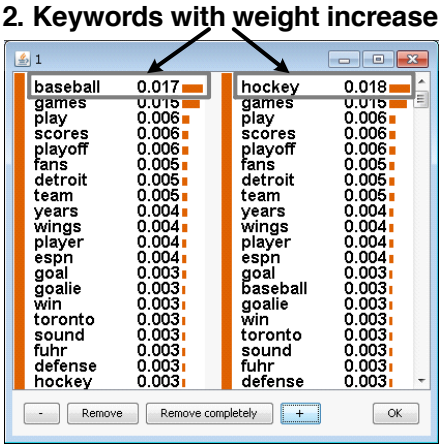
6.1 Usage Scenarios

We show usage scenarios of the above-described interactions using the 20 Newsgroups data set. For efficient interactive visualization in real time, we randomly sampled 50 data items per each of the 20 categories. Figs. 4-7 shows a sequence of interactions with these data in UTOPIAN.

Fig. 4(a) shows an initial visualization result generated by UTOPIAN, which gives a nice visual overview about generated topic clusters. One can see that se-



(a) The initial visualization



(b) The topic refinement of the two split topics.

Fig. 4: The initial visualization and the topic splitting interaction for the subset of the 20 Newsgroup data set. From the topic 'games, play, scores,' we increase the weight of the keyword 'baseball' in one topic while increasing that of 'hockey' in the other.

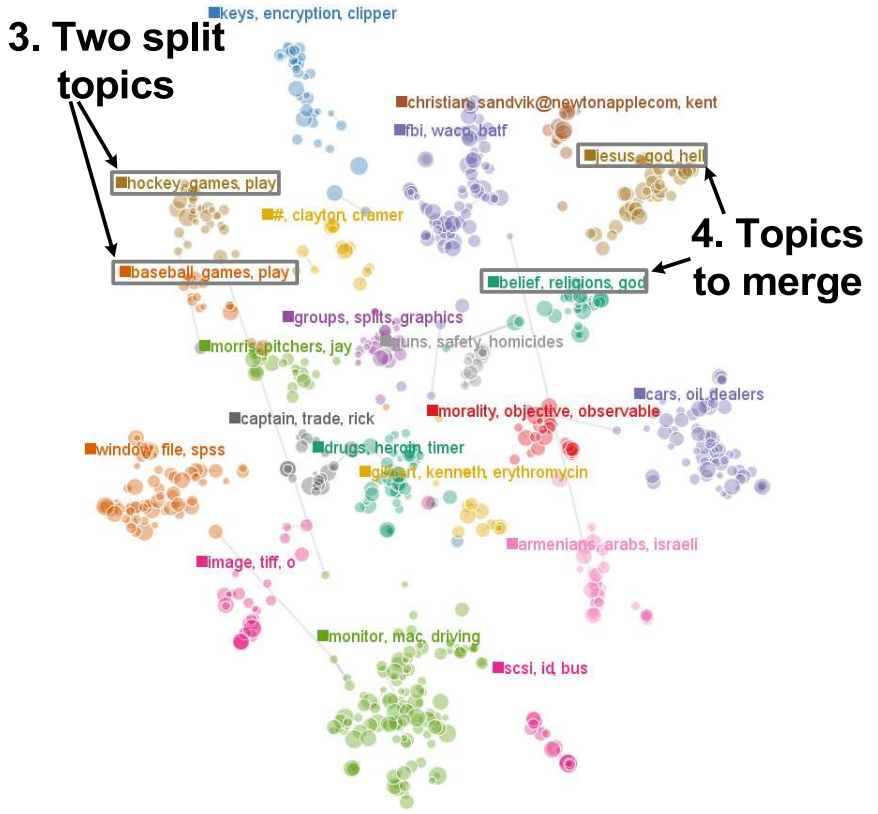


Fig. 5: The two splitted topics, one of which is mainly related to baseball and the other to hockey, are visualized. Next, the two topics ‘jesus, god, hell’ and ‘belief, religion, god’ are to be merged.

mantically similar topics are placed closely, e.g., the topics ‘christian, ...’ and ‘jesus, god, hell’ (top right) and the topics ‘monitor, mac, driving’ and ‘scsi, id, bus’ (bottom middle) while unrelated topics far from each other, e.g., the topics ‘games, play, scores’ (top left) and ‘armenian, arabs, israeli’ (bottom right) and the topics ‘window, file, spss’ (lower left) and ‘cars, oil, dealers’ (upper right).

Now, we perform a topic splitting interaction on the topic ‘games, play, scores.’ Initially, both the keywords ‘baseball’ and ‘hockey’ are shown to be highly ranked in this topic, but we aim at distinguishing the two topics with respect to these keywords. Thus, as shown in Fig. 4(b), we increase the weight of the former keyword in the left topic and that of the latter in the right topic. This interaction generates the two split topics, as shown in Fig. 5, and the documents included in each topic properly reflect such user intention. Next, we merge semantically similar topics. We select the two topics ‘jesus, god, hell’ and ‘belief, religion, god’ to merge. The merged topic is generated as ‘god, jesus, sin,’ as shown in Fig. 6.

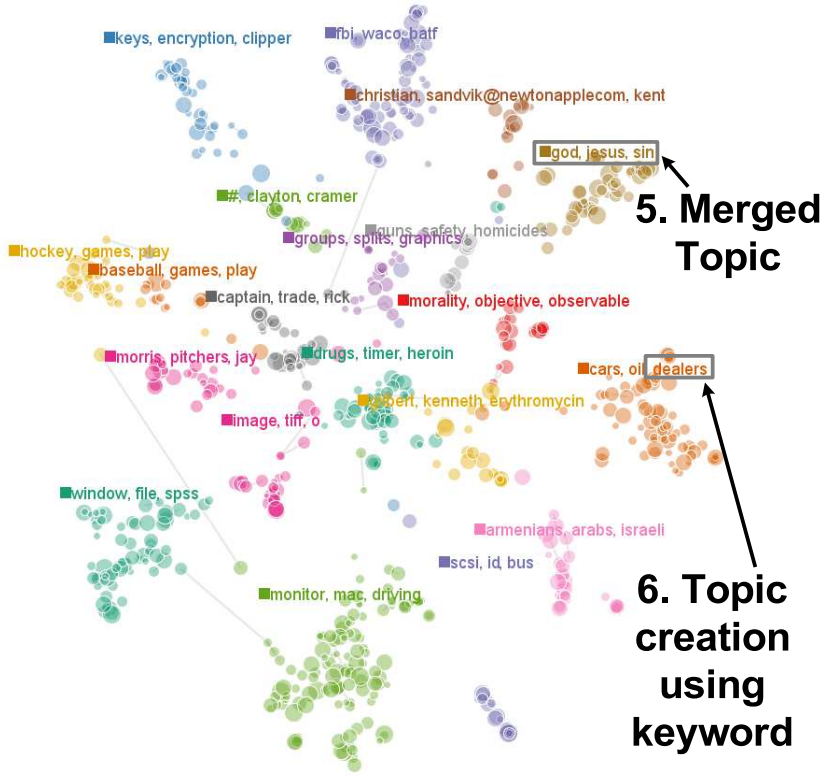


Fig. 6: The merged topic ‘god, jesus, sin’ is generated. Next, a new topic based on the keyword ‘dealers’ from the topic ‘cars, oil, dealers’ is to be generated.

Finally, we create a new topic via a keyword-induced topic creation interaction. To this end, we select a keyword ‘dealers’ from the topic ‘cars, oil, dealers.’ As shown in Fig. 7, the newly created topic ‘dealers, invoice, cost’ reveals the detailed information about the relevant topic to this keyword.

7 Conclusions and Future Directions

In this book chapter, we have presented nonnegative matrix factorization (NMF) for document clustering and topic modeling. We have first introduced the NMF formulation and its applications to clustering. Next, we have presented the flexible algorithmic framework based on block coordinate descent (BCD) as well as its convergence property and stopping criterion. Based on the BCD framework, we discussed two important extensions for clustering, the sparse and the weakly-supervised NMF,

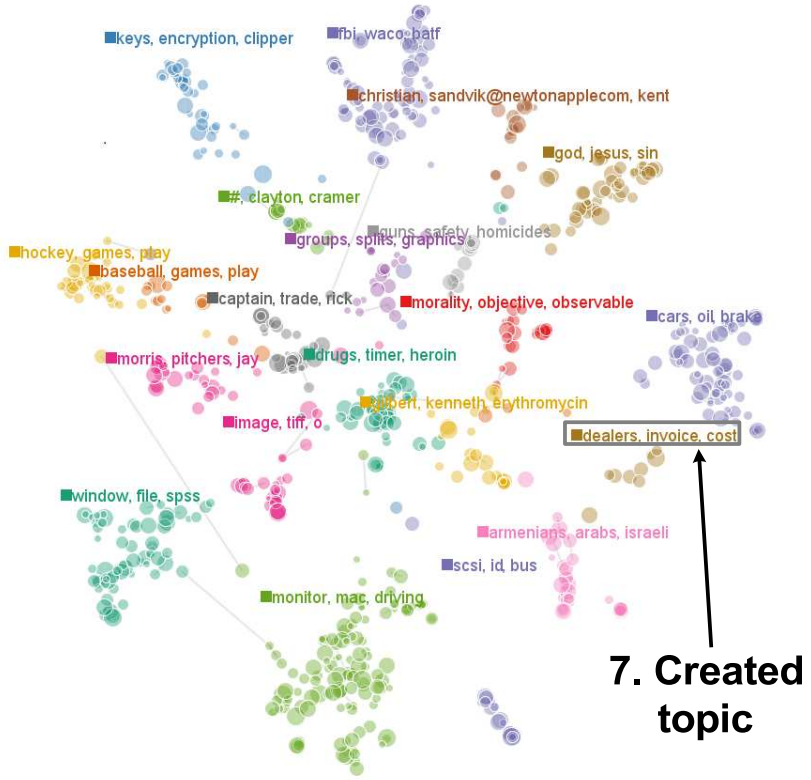


Fig. 7: The newly created topic ‘dealers, invoice, cost’ is shown.

and our method to determine the number of clusters. Experimental results on various real-world document data sets show the advantage of our NMF algorithm in terms of clustering quality, convergence behavior, sparseness, and consistency. Finally, we presented a visual analytics system called UTOPIAN for interactive visual clustering and topic modeling and demonstrated its interaction capabilities such as topic splitting/merging as well as keyword-/document-induced topic creation.

The excellence of NMF in clustering and topic modeling poses numerous exciting research directions. One important direction is to improve the scalability of NMF. Parallel distributed algorithms are essential for this purpose, but at the same time, the real-time interaction capability can also be considered from the perspective of a human perception [9]. Another direction is to allow users to better understand clustering and topic modeling outputs. In practice, the semantic meaning of document clusters and topics is understood based on several representative keywords and/or documents. However, significant noise in real-world data often makes it difficult to understand the resulting clusters and topics. In this sense, how to provide

additional information such as the cluster/topic quality as well as contextual meaning of given keywords has to be addressed.

Acknowledgements The work of the authors was supported in part by the National Science Foundation (NSF) grants CCF-0808863 and the Defense Advanced Research Projects Agency (DARPA) XDATA program grant FA8750-12-2-0309. 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 NSF or the DARPA.

References

1. S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu. A practical algorithm for topic modeling with provable guarantees. *Journal of Machine Learning Research (JMLR) W&CP*, 28(2):280–288, 2013.
2. S. Arora, R. Ge, R. Kannan, and A. Moitra. Computing a nonnegative matrix factorization – provably. In *Proc. the 44th Symposium on Theory of Computing (STOC)*, pages 145–162, 2012.
3. A. Berman and R. J. Plemmons. *Nonnegative matrices in the mathematical sciences*. SIAM, 1994.
4. D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, 2nd edition, 1999.
5. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research (JMLR)*, 3:993–1022, 2003.
6. J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Science (PNAS)*, 101(12):4164–4169, 2004.
7. D. Cai, X. He, J. Han, and T. S. Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(8):1548–1560, 2011.
8. J. Choo, C. Lee, C. K. Reddy, and H. Park. UTOPIAN: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 19(12):1992–2001, 2013.
9. J. Choo and H. Park. Customizing computational methods for visual analytics with big data. *IEEE Computer Graphics and Applications (CG&A)*, 33(4):22–28, 2013.
10. A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. Wiley, 2009.
11. K. Devarajan. Nonnegative matrix factorization: An analytical and interpretive tool in computational biology. *PLoS Computational Biology*, 4(7):e1000029, 2008.
12. I. S. Dhillon and S. Sra. Generalized nonnegative matrix approximations with bregman divergences. In *Advances in Neural Information Processing Systems (NIPS) 18*, pages 283–290, 2005.
13. C. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proc. SIAM International Conference on Data Mining (SDM)*, pages 606–610, 2005.
14. C. Ding, T. Li, and M. Jordan. Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding. In *Proc. the 8th IEEE International Conference on Data Mining (ICDM)*, pages 183–192, 2008.
15. C. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(1):45–55, 2010.
16. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

17. A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research (JMLR)*, 8:2265–2295, 2007.
18. E. F. Gonzales and Y. Zhang. Accelerating the Lee-Seung algorithm for non-negative matrix factorization. Technical Report TR05-02, Rice University, 2005.
19. L. Grippo and M. Sciandrone. On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. *Operations Research Letters*, 26:127–136, 2000.
20. T. Hofmann. Probabilistic latent semantic indexing. In *Proc. the 22nd Annual International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR)*, 1999.
21. P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research (JMLR)*, 5:1457–1469, 2004.
22. D. Kim, S. Sra, and I. Dhillon. Fast Newton-type methods for the least squares nonnegative matrix approximation problem. In *Proc. SIAM International Conference on Data Mining (SDM)*, pages 343–354, 2007.
23. H. Kim and H. Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 2007.
24. H. Kim and H. Park. Nonnegative matrix factorization based on alternating non-negativity-constrained least squares and the active set method. *SIAM Journal on Matrix Analysis and Applications*, 30(2):713–730, 2008.
25. J. Kim, Y. He, and H. Park. Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework. *Journal of Global Optimization*, 2013.
26. J. Kim and H. Park. Sparse nonnegative matrix factorization for clustering. Technical Report GT-CSE-08-01, Georgia Institute of Technology, 2008.
27. J. Kim and H. Park. Toward faster nonnegative matrix factorization: A new algorithm and comparisons. In *Proc. the 8th IEEE International Conference on Data Mining (ICDM)*, pages 353–362, 2008.
28. J. Kim and H. Park. Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM Journal on Scientific Computing*, 33(6):3261–3281, 2011.
29. D. Kuang, C. Ding, and H. Park. Symmetric nonnegative matrix factorization for graph clustering. In *Proc. SIAM International Conference on Data Mining (SDM)*, pages 106–117, 2012.
30. D. Kuang and H. Park. Fast rank-2 nonnegative matrix factorization for hierarchical document clustering. In *Proc. the 19th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 739–747, 2013.
31. H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.
32. C. L. Lawson and R. J. Hanson. *Solving least squares problems*. Prentice-Hall, 1974.
33. D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.
34. D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS) 14*, pages 556–562, 2001.
35. D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research (JMLR)*, 5:361–397, 2004.
36. L. Li, G. Lebanon, and H. Park. Fast Bregman divergence NMF using taylor expansion and coordinate descent. In *Proc. the 18th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 307–315, 2012.
37. S. Li, X. W. Hou, H. J. Zhang, and Q. S. Cheng. Learning spatially localized, parts-based representation. In *Proc. the 2001 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 207–212, 2001.
38. T. Li, C. Ding, and M. I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Proc. the 7th IEEE International Conference on Data Mining (ICDM)*, pages 577–582, 2007.
39. C.-J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19(10):2756–2779, 2007.

40. C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
41. A. K. McCallum. MALLET: A machine learning for language toolkit. `http://mallet.cs.umass.edu`, 2002.
42. S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1-2):91–118, 2003.
43. 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.
44. V. P. Pauca, J. Piper, and R. J. Plemmons. Nonnegative matrix factorization for spectral data analysis. *Linear Algebra and its Applications*, 416(1):29–47, 2006.
45. V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons. Text mining using non-negative matrix factorizations. In *Proc. SIAM International Conference on Data Mining (SDM)*, pages 452–456, 2004.
46. F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42:373–386, 2006.
47. R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58:267–288, 1994.
48. L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9:2579–2605, 2008.
49. S. A. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM J. on Optimization*, 20(3):1364–1377, 2009.
50. S. Wild, J. Curry, and A. Dougherty. Improving non-negative matrix factorizations through structured initialization. *Pattern Recognition*, 37:2217–2232, 2004.
51. B. Xie, L. Song, and H. Park. Topic modeling via nonnegative matrix factorization on probability simplex. In *NIPS Workshop on Topic Models: Computation, Application, and Evaluation*, 2013.
52. W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proc. the 26th Annual International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR)*, pages 267–273, 2003.