Simplicial Nonnegative Matrix Tri-factorization: Fast Guaranteed Parallel Algorithm

Duy-Khuong Nguyen[1,3(✉)], Quoc Tran-Dinh[2], and Tu-Bao Ho[1,4]

[1] Japan Advanced Institute of Science and Technology, Nomi, Japan
khuongnd@gmail.com
[2] The University of North Carolina at Chapel Hill, Chapel Hill, USA
[3] University of Engineering and Technology,
Vietnam National University, Hanoi, Vietnam
[4] John von Neumann Institute, Vietnam National University, Ho Chi Minh, Vietnam

Abstract. Nonnegative matrix factorization (NMF) is a linear powerful dimension reduction and has various important applications. However, existing models remain the limitations in the terms of interpretability, guaranteed convergence, computational complexity, and sparse representation. In this paper, we propose to add simplicial constraints to the classical NMF model and to reformulate it into a new model called simplicial nonnegative matrix tri-factorization to have more concise interpretability via these values of factor matrices. Then, we propose an effective algorithm based on a combination of three-block alternating direction and Frank-Wolfe's scheme to attain linear convergence, low iteration complexity, and easily controlled sparsity. The experiments indicate that the proposed model and algorithm outperform the NMF model and its state-of-the-art algorithms.

Keywords: Dimensionality reduction · Nonnegative matrix factorization · Simplicial nonnegative matrix tri-factorization · Frank-wolfe algorithm

1 Introduction

Nonnegative matrix factorization (NMF) has been recognized as a linear powerful dimension reduction, and has a wide range of applications including text mining, image processing, bioinformatics [9]. In this problem, a given nonnegative observed matrix $V \in \mathbb{R}_+^{n\times m}$ consists of m vectors having n dimensions, which is factorized into a product of two nonnegative factor matrices, namely latent components $W^T \in \mathbb{R}_+^{n\times r}$ and new coefficients $F \in \mathbb{R}_+^{r\times m}$. In the classical setting, due to noise and nonnegative constraints, NMF is approximately conducted by minimizing the objective function $D(V\|W^TF) = \|V - W^TF\|_2^2$.

Despite having more than a decade of rapid developments, NMF still has the limitations of interpretability and computation. First, the values of factor matrices in NMF do not concisely represent the roles of latent components over instances and the contributions of attributes over latent components. Simply,

A. Hirose et al. (Eds.): ICONIP 2016, Part II, LNCS 9948, pp. 117–125, 2016.
DOI: 10.1007/978-3-319-46672-9_14

they express the appearances of latent components over instances and attributes over latent components rather than their roles. In other words, it is not reasonable to determine how may percentages each latent component contributes to instances via the values in F. Second, concerning the computation, Wang et al., 2012 [4] proposed one of the most state-of-the algorithms which has sub-linear convergence $O(1/k^2)$, high complexity $\mathcal{O}(r^2)$ at each iteration, and considerable difficulties of parallelability and controlling sparsity because it works on the whole of matrices and all the variable gradients.

To overcome the above mentioned limitations, we introduce a new formulation of NMF as simplicial nonengative matrix tri-factorization (SNMTF) by adding simplicial constraints over factor matrices, and propose a fast guaranteed algorithm which can be conveniently and massively parallelized. In this model, the roles of latent components over instances and the contributions of attributes over latent components are represented via the factor matrices F and W. To the end, this work has the major contributions as follows:

(a) We introduce a new model of NMF which is named by SNMTF using L_2 regularizations on both the latent components W and the new coefficients F. The new model has not only more concise interpretability, but also retains the generalization of NMF.
(b) We propose a fast parallel algorithm based on a combination of three-block alternating direction and Frank-Wolfe algorithm [7] to attain linear convergence, low iteration complexity, and easily controlled sparsity.

2 Problem Formulation

Concerning the interpretability, NMF is a non-convex problem having numerous solutions as stationary points, which has rotational ambiguities [1]. Particularly, if $V \approx W^T F$ is a solution, $V \approx W^T[D_F F'] = [W^T D_F]F'$ are also equivalent solutions; where D_F is a positive diagonal matrix satisfying $F = D_F F'$. Hence, it does not consistently explain the roles of latent components over instances and the contributions of attributes over latent components.

To resolve the limitation, we propose a new formulation as SNMTF, in which the data matrix is factorized into a product of three matrices $V \approx W^T DF$, where D is a positive diagonal matrix, $\sum_{k=1}^{r} W_{ki} = 1 \forall i$, and $\sum_{k=1}^{r} F_{kj} = 1 \forall j$.

However, the scaling of factors via the diagonal matrix D can lead to the inconsistency of interpreting the factor matrices W and F, and the existence of bad stationary points such as $\lambda_k = 0 \Rightarrow W_{ki} = 0, F_{kj} = 0$. Hence, we restrict the formulation by adding the condition $\lambda_1 = ... = \lambda_r = \lambda$. This model can be considered as an extension of the probabilistic latent semantic indexing (PLSI) [5] for scaling data with an additional assumption that the weights of latent factors are the same. Remarkably, their roles of latent components over instances and of attributes over latent components are represented via the values of the factor matrices W and F. As a result, it is more easier to recognize these roles than

in the case that the weights of latent factors can be different. Therefore, the objective function of SNMTF with L_2 regularizations is written as follows:

$$\begin{cases} \min\limits_{W,D,F} \left\{ \Phi(W,D,F) := \frac{1}{2}\|V - W^T DF\|_F^2 + \frac{\alpha_1}{2}\|W\|_F^2 + \frac{\alpha_2}{2}\|F\|_F^2 \right\} \\ \text{s.t.} \quad \sum\limits_{k=1}^{r} W_{ki} = 1 \ (i = 1, \cdots, n), \ \sum\limits_{k=1}^{r} F_{kj} = 1 \ \ (j = 1, \cdots, m), \\ \qquad W \in \mathbb{R}_+^{r \times n}, \quad F \in \mathbb{R}_+^{r \times m}, \quad D = \text{diag}(\lambda, \cdots, \lambda). \end{cases} \tag{1}$$

There are three significant remarks in this formulation. Firstly, adding simplicial constraints leads to more concise interpretability of the factor matrices and convenience for post processes such as neural network and support vector machine because the sum of attributes is normalized to 1. Secondly, L_1 regularization is ignored because $\|W\|_1$ and $\|F\|_1$ equal to a constant. Finally, the diagonal of $D = \text{diag}(\lambda, ..., \lambda)$ has the same value because of two main reasons: First, it is assumed that V_{ij} is generated by W_i, F_j and a scale as $V_{ij} \approx \lambda W_i^T F_j$; Second, it still retains the generalization of SNMTF which every solution of NMF can be equivalently represented by SNMTF, which will be proved in Sect. 4.

3 Proposed Algorithm

We note that problem (1) is nonconvex due to the product $W^T DF$. We first propose to use three-block alternating direction method to decouple there three blocks. Then, we decompose the computation onto column of the matrix variables, which can be conducted in parallel. Finally, we apply Frank-Wolfe's algorithm [6,7] to solve the underlying subproblems.

3.1 Iterative Multiplicative Update for Frobenius Norm

We decouple the tri-product $W^T DF$ by the following alternating direction scheme, which is also called iterative multiplicative update:

$$\begin{cases} F^{t+1} := \arg\min\limits_{F \in \mathbb{R}^{r \times m}} \left\{ \Phi(W^t, D^t, F) : F \in \Delta_r^n \right\}, \\ W^{t+1} := \arg\min\limits_{W \in \mathbb{R}^{r \times n}} \left\{ \Phi(W, D^t, F^{t+1}) : W \in \Delta_r^m \right\}, \\ D^{t+1} := \text{argmin}\limits_{\lambda \in \mathbb{R}} \left\{ \Phi(W^{t+1}, D, F^{t+1}) : D = \text{diag}(\lambda, \cdots, \lambda) \right\}. \end{cases} \tag{2}$$

Clearly, both F-problem and W-problem in (2) are convex but are still constrained by simplex, while the D-problem is unconstrained. We now can solve the D-problem in the third line of (2). Due to the constraint $D = \text{diag}(\lambda, \cdots, \lambda)$, this problem turns out to be an univariate convex program, which can be solved in a closed form as follows:

$$\lambda_{t+1} = \text{argmin}_{\lambda \in \mathbb{R}} \|V - {W^{t+1}}^T D {F^{t+1}}\|_2^2 = \frac{\langle V, {W^{t+1}}^T F^{t+1} \rangle}{\langle W^{t+1} {W^{t+1}}^T, F^{t+1} {F^{t+1}}^T \rangle}. \tag{3}$$

Algorithm 1. Iterative multiplicative update for Frobenius norm

Input: Data matrix $V = \{V_j\}_{j=1}^m \in \mathbb{R}_+^{n\times m}$, and r, $\alpha_1, \alpha_2 \geq 0, \beta \geq 0$.
Output: Coefficients $F \in R_+^{r\times m}$ and latent components $W \in R_+^{r\times n}$

begin
  Pick an arbitrary initial point $W \in \mathbb{R}_+^{r\times n}$ (e.g., random);
  repeat
    $q = -\lambda WV$; $Q = \lambda^2 WW^T + \alpha_1 \mathbb{I}$;
    /***Inference step:** Fix W and D to find new F;
    for *j = 1 to m* **do**
      $F_j \approx \underset{x\in\Delta_r}{\operatorname{argmin}} \left\{\frac{1}{2}x^T Qx + q_j^T x\right\}$/*Call Algorithm 2 in parallel */;
    $q = -\lambda FV^T$; $Q = \lambda^2 FF^T + \alpha_2 \mathbb{I}$;
    /***Learning step:** Fix F and D to find new W;
    for *i = 1 to n* **do**
      $W_i \approx \underset{x\in\Delta_r}{\operatorname{argmin}} \left\{\frac{1}{2}x^T Qx + q_i^T x\right\}$/*Call Algorithm 2 in parallel */;
    $\lambda = \underset{\lambda\in\mathbb{R}}{\operatorname{argmin}} \|V - W^T DF\|_2^2 = \frac{\langle V, W^T F\rangle}{\langle WW^T, FF^T\rangle}$;
  until *convergence condition is satisfied*;

Then, we form the matrix D^{t+1} as $D^{t+1} = \operatorname{diag}(\lambda_{t+1}, \cdots, \lambda_{t+1})$.

If we look at the F-problem, then fortunately, it can be separated into n independent subproblems ($j = 1, \cdots, m$) of the form:

$$F_j^{t+1} = \arg\min_{F_j} \left\{ \frac{1}{2}\|V_j - (W^t)^T D^t F_j\|_2^2 + \frac{\alpha_2}{2}\|F_j\|_2^2 : F_j \in \Delta_r \right\}. \tag{4}$$

The same trick is applied to the W-problem in the second line of (2). Now, we assume that we apply the well-known Frank-Wolfe algorithm to solve (4), then we can describe the full algorithm for solving (1) into Algorithm 1.

The stopping criterion of Algorithm 1 remains unspecified. Theoretically, we can terminate Algorithm 1 using the optimality condition of (1). However, computing this condition requires a high computational effort. We instead terminate Algorithm 1 if it does not significantly improve the objective value of (1) or the differences $W_{t+1} - W_t$ and $F_{t+1} - F_t$ and the maximum number of iterations.

3.2 Frank-Wolfe's Algorithm for QP over Simplex Constraint

Principally, we can apply any convex optimization method such as interior-point, active-set, projected gradient and fast gradient method to solve QP problems of the form (4). However, this QP problem (4) has special structure and is often sparse. In order to exploit its sparsity, we propose to use a Frank-Wolfe algorithm studied in [7] to solve this QP problem. Clearly, we can write (4) as follows:

$$x \approx \underset{x\in\Delta_r}{\operatorname{argmin}} \frac{1}{2}\|v - A^T x\|_2^2 + \frac{\alpha}{2}\|x\|_2^2 = \underset{x\in\Delta_r}{\operatorname{argmin}} \frac{1}{2}x^T Qx + q^T x \tag{5}$$

where $v = V_j, A = DW$, $Q = AA^T + \alpha$, and $q = -Av$. By applying the Frank-Wolfe algorithm form [7] to solve this problem, we obtain Algorithm 2 below.

Algorithm 2. Fast Algorithm for NQP with Simplicial Constraint

Input: $Q \in \mathbb{R}^{r\times r}, q \in \mathbb{R}^r$.
Output: New coefficient $x \approx \underset{x\in\Delta_r}{\operatorname{argmin}} f(x) = \frac{1}{2}x^TQx + q^Tx$.

begin
  Choose $k = \underset{k}{\operatorname{argmin}} \frac{1}{2}e_k^TQe_k + q^Te_k$, where e_k is the k^{th} basis vector;
  Set $x = \mathbf{0}^k$; $x_k = 1$; $Qx = Qe_k$; $qx = q^Tx$ and $\nabla f = Qx + q^T$;
  repeat
    Select $k = \underset{k\in\{1..r\}}{\operatorname{argmin}} \{\langle e_k - x, \nabla f\rangle\}$ *or* $\{\langle x - e_k, \nabla f\rangle | x_k > 0\}$;
    Select $\alpha = \underset{\alpha}{\operatorname{argmin}} f(\alpha e_k + (1-\alpha)x)$;
    $\alpha = min(1, max(\alpha, -\frac{x_k}{1-x_k}))$;
    $Qx = (1-\alpha)Qx + \alpha Qe_k$; $\nabla f = Qx + q$;
    $qx = (1-\alpha)qx + \alpha qe_k$;
    $x = (1-\alpha)x$; $x_k = x_k + \alpha$;
  until *converged conditions are staisfied*;

In Algorithm 2, the first derivative of $f(x) = \frac{1}{2}x^TQx + q^Tx$ is computed by $\nabla f = Qx + q$. In addition, the steepest direction in the simplex is selected by this formula: $k = \underset{k\in\{1..r\}}{\operatorname{argmin}} \{\langle e_k - x, \nabla f\rangle\}$ *or* $\{\langle x - e_k, \nabla f\rangle | x_k > 0\}$.

For seeking the best variable α to minimize $f(\alpha x + (1-\alpha)e_k)$ where e_k is the k^{th} unit vector. Let consider $f(\alpha x + (1-\alpha)e_k)$, we have:

$$\begin{aligned} \frac{\partial f}{\partial \alpha}(\alpha = 0) &= (x - \mathbf{e}_k)^T(Qx + q) = x^T(Qx + q) - [Qx]_k - q_k \\ \frac{\partial^2 f}{\partial \alpha^2}(\alpha = 0) &= (x - \mathbf{e}_k)^TQ(x - \mathbf{e}_k) = x^TQx - 2[Qx]_k + Q_{kk}. \end{aligned} \tag{6}$$

Since f is a quadratic function of α, its optimal solution is $\alpha = \underset{\alpha\in[-\frac{x_k}{1-x_k},1]}{\operatorname{argmin}} f((1-\alpha)x + \alpha\mathbf{e}_k) = [-\frac{\nabla f_{\alpha=0}}{\nabla^2 f_{\alpha=0}}]_{[-\frac{x_k}{1-x_k},1]}$. The projection of solution over the interval $[-\frac{x_k}{1-x_k}, 1]$ is to guarantee $x_k \geq 0\ \forall k$. The updates $x = (1-\alpha)x$ and $x_k = x_k + \alpha$ are to retain the simplicial constraint $x \in \Delta_r$.

In Algorithm 2, duplicated computation is removed to reduce the iteration complexity into $\mathcal{O}(r)$ by maintaining Qx and q^Tx. This result is highly competitive with the state-of-the-art algorithm having a sub-linear convergence rate $\mathcal{O}(1/k^2)$ and complexity of $\mathcal{O}(r^2)$ [4].

4 Theoretical Analysis

This section discusses three important aspects of the proposed algorithm as convergence, complexity, and generalization. Concerning the convergence, setting $\alpha_1, \alpha_2 > 0$, based on Theorem 3 in Lacoste-Julien, S., & Jaggi, M. (2013) [7], since $f(x) = \frac{1}{2}x^TQx + q^Tx$ is smoothness and strongly convex, we have:

Theorem 1. *Algorithm 2 linearly converges as* $f(x_{k+1}) - f(x^*) \leq (1 - \rho_f^{FW})^k (f(x_0) - f(x^*))$, *where* $\rho_f^{FW} = \{\frac{1}{2}, \frac{\mu_f^{FW}}{C_f}\}$, C_f *is the curvature constant of the convex and differentiable function* f, *and* μ_f^{FW} *is an affine invariant of strong convex parameter.*

Since Algorithm 2 always linearly converges and the objective function restrictedly decreases, Algorithm 1 always converges stationary points. Regarding the complexity of the proposed algorithm, we have:

Theorem 2. *The complexity of Algorithm 2 is* $\mathcal{O}(r^2 + \bar{t}r)$, *and the complexity of each iteration in Algorithm 1 is* $\mathcal{O}(mnr + (m+n)r^2 + \bar{t}(m+n)r)$.

Proof. The complexity of the initial computation $Qx + q$ is $O(r^2)$, and each iteration in Algorithm 2 is $O(r)$. Hence, the complexity of Algorithm 2 is $\mathcal{O}(r^2 + \bar{t}r)$. The complexity of main operators in Algorithm 1 as WV, FV^T, FF^T, and WW^T is $O(mnr + (m+n)r^2)$. Overall, the complexity of each iteration in Algorithm 1 is $\mathcal{O}(mnr + (m+n)r^2 + \bar{t}(m+n)r)$.

This is highly competitive with the guaranteed algorithm [4] having the complexity of $\mathcal{O}(mnr + (m+n)r^2 + \bar{t}(m+n)r^2)$. Furthermore, adding the simplicial constants into NMF does not reduce the generalization and flexibility of NMF:

Theorem 3. *Each solution of NMF can be equivalently transfered into SNMTF.*

Proof. Assume that $V \approx W^TF$, which leads to the existence of λ large enough to satisfy $W^TF = W'^TD'F'$, where $D' = \text{diag}(\lambda, .., \lambda)$, $\sum_{k=1}^{r} W_{ki} < 1 \ \forall i$, and $\sum_{k=1}^{r} W_{kj} < 1 \ \forall j$. Therefore, $\exists W''$, D'', and F'': $W''^TD''F'' = W'^TD'F' = W^TF$, where $W''_{ij} = W'_{ij}$, $F''_{ij} = F'_{ij}$, $W''_{r+1,i} = 0, W''_{r+2,i} = 1 - \sum_{k=1}^{r+1} W''_{ki} \ \forall i$, $F''_{r+1,j} = 1 - \sum_{k=1}^{r} F''_{kj}, F''_{r+2,j} = 0 \ \forall j, D'' = \text{diag}(\lambda, .., \lambda)$. $W''^TD''F''$ is SNMTF of V.

The generalization is crucial to indicate the robustness and high flexibility of the proposed model in comparison with NMF models, although many constraints have been added to enhance the quality and interpretability of the NMF model.

5 Experimental Evaluation

This section investigates the effectiveness of the proposed algorithm via three significant aspects of convergence, classification performance, and sparsity. The proposed algorithm **SNMTF** is compared with the following methods:

- **NeNMF** [4]**:** It is a guaranteed method, each alternative step of which sub-linearly converges at $\mathcal{O}(1/k^2)$ that is highly competitive with the proposed algorithm.
- **LeeNMF** [8]**:** It is the original gradient algorithm for NMF.
- **PCA:** It is considered as a based-line method in dimensionality reduction, which is compared in classification and sparsity.

Datasets: We compared the selected methods in three typical datasets with different size, namely Faces[1], Digits[2] and Tiny Images [3].

Environment Settings: We develop the proposed algorithm SNMTF in Matlab with embedded code C++ to compare them with other algorithms. We set system parameters to use only six threads in the machine Mac Pro 6-Core Intel Xeon E5 3 GHz 32 GB. The initial matrices W^0 and F^0 are set to the same values, the maximum number of iterations is 500. The source code is published on our homepage [4] (Table 1).

Table 1. Dataset Information

Datasets	n	m	Testing size	r	#class
Faces	361	6,977	24,045	30	2
Digits	784	6.10^4	10^4	60	10
Cifar-10	3,072	5.10^4	10^4	90	10

5.1 Convergence

We investigate the convergence of the compared algorithms by $\frac{f_1}{f_k}$ because they have the different formulations and objective functions. Figure 1 clearly shows that the proposed algorithm converges much faster than the other algorithms. The most steepest line of the proposed algorithm represents its fast convergence. This result is reasonable because the proposed algorithm has a faster convergence rate and lower complexity than the state-of-the-art algorithm NeNMF.

5.2 Classification

Concerning the classification performance, the training datasets with labels are used to learn gradient boosting classifiers [2,3], one of the robust ensemble methods, to classify the testing datasets. The proposed algorithm outperforms the other algorithms and PCA over all the datasets. For the small and easy dataset

[1] http://cbcl.mit.edu/cbcl/software-datasets/FaceData.html.
[2] http://yann.lecun.com/exdb/mnist/.
[3] http://horatio.cs.nyu.edu/mit/tiny/data/index.html.
[4] http://khuongnd.appspot.com/.

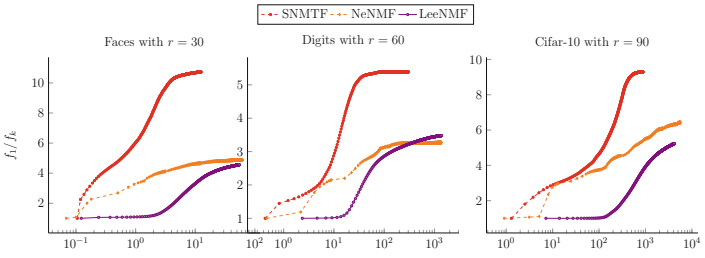

Fig. 1. Convergence of loss information f_1/f_k versus time

Face, the result of the proposed algorithm is close to the results of NeNMF. However, for larger and more complex datasets Digit and Tiny Images, the proposed algorithm has much better accuracy than the other algorithms. Noticeably, the result of Tiny Images is much worse than the result of the other datasets because it is highly complicated and contains backgrounds. This classification result obviously represents the effectiveness of the proposed model and algorithm.

5.3 Sparsity

We investigate the sparsity of factor matrices F, W, and the sparsity average in both F and W. For the dataset Digit, the proposed algorithm outperforms in all these measures. For the other datasets Faces and Tiny Images, it has better representation F and more balance between sparsity of F and W. Frankly speaking, in these datasets, achieving more sparse representation F is more meaningful than achieving more sparse model W because F is quite dense but W is highly sparse, which is a reason to explain why SNMTF has the best classification result (Tables 2 and 3).

Table 2. Classification inaccuracy

Dataset	PCA	LeeNMF	NeNMF	SNMTF
Faces	6.7	3.3	2.7	**2.6**
Digits	30.15	12.16	3.9	**3.6**
Tiny Images	59.3	71.4	51.4	**50.1**

Table 3. Sparsity of factor matrices (%) of F, W, and (both F and W)

Dataset	PCA	LeeNMF	NeNMF	SNMTF
Faces	(0, 0.26, 0.24)	(0.58, 0, 0.55)	(7.64, **60.33**, 10.23)	(**9.78**, 59.98, **12.24**)
Digits	(1.37, 0, 0.02)	(31.24, 50.47, 31.49)	(41.20, 92.49, 41.86)	(**50.49**, **93.47**, **51.04**)
Tiny Images	(0, 0, 0)	(0.02, 0, 0.02)	(9.97, **86.58**, 14.40)	(**11.71**, 85.29, **15.97**)

6 Conclusion

This paper proposes a new model of NMF as SNMTF with L_2 regularizations, which has more concise interpretability of the role of latent components over instances and attributes over latent components while keeping the generalization in comparison with NMF. We design a fast parallel algorithm with guaranteed convergence, low iteration complexity, and easily controlled sparsity to learn SNMTF, which is derived from Frank-Wolfe algorithm [7]. Furthermore, the proposed algorithm is convenient to massively parallelize, and control sparsity of both new representation F and model W. Based on the experiments, the new model and the proposed algorithm outperform the NMF model and its state-of-the-art algorithms in three significant aspects of convergence, classification, and sparsity. Therefore, we strongly believe that SNMTF is highly potential for many applications and extensible for nonnegative tensor factorization.

Acknowledgments. This work was supported by Asian Office of Aerospace R&D under agreement number FA2386-15-1-4006.

References

1. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.I.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. Wiley, Hoboken (2009)
2. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning. Springer Series in Statistics, vol. 1. Springer, New York (2001)
3. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**(5), 1189–1232 (2001)
4. Guan, N., Tao, D., Luo, Z., Yuan, B.: Nenmf: an optimal gradient method for nonnegative matrix factorization. IEEE Trans. Signal Process. **60**(6), 2882–2898 (2012)
5. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57. ACM (1999)
6. Jaggi, M.: Revisiting frank-wolfe: projection-free sparse convex optimization. In: Proceedings of the 30th International Conference on Machine Learning (ICML 2013), pp. 427–435 (2013)
7. Lacoste-Julien, S., Jaggi, M.: An affine invariant linear convergence analysis for frank-wolfe algorithms (2013). arXiv preprint arXiv:1312.7864
8. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems, pp. 556–562 (2001)
9. Wang, Y.X., Zhang, Y.J.: Nonnegative matrix factorization: a comprehensive review. IEEE Trans. Knowl. Data Eng. **25**(6), 1336–1353 (2013)