Note on Algorithm Differences Between Nonnegative Matrix Factorization And Probabilistic Latent Semantic Indexing

[1]Zhong-Yuan Zhang, [2]Chris Ding, [3]Jie Tang
*1, Corresponding Author *School of Statistics, Central University of Finance and Economics, P.R.China, zhyuanzh@gmail.com*
[2,] *Department of Computer Science and Engineering, University of Texas at Arlington, USA*
[3] *Department of Computer Science and Technology, Tsinghua University, Beijing, China*

Abstract

NMF and PLSI are two state-of-the-art unsupervised learning models in data mining, and both are widely used in many applications. References have shown the equivalence between NMF and PLSI under some conditions. However, a new issue arises here: why can they result in different solutions since they are equivalent? or in other words, their algorithm differences are not studied intensively yet. In this note, we explicitly give the algorithm differences between PLSI and NMF. Importantly, we find that even if starting from the same initializations, NMF and PLSI may converge to different local solutions, and the differences between them are born in the additional constraints in PLSI though NMF and PLSI optimize the same objective function.

Keywords*: Algorithm, Relation, NMF, PLSI*

1. Introduction

Nonnegative Matrix Factorization (NMF,[[1][2][3]) is evolved from Principal Component Analysis (PCA,[4][5]). PCA is one of the basic techniques for extracting the principal components (factors) from a series of vectors such that each vector is a linear combination of the components. One basic problem with PCA is that there are both positive and negative elements in each of the principal components and also both positive and negative coefficients in linear combinations. However the mixed signs contradict our experience and make it hard to explain the results. In fact, in many applications such as image processing, biology or text mining, nonnegative data analysis is often important and nonnegative constraints on the wanted principal components (basis matrix) and coefficients (coding matrix) can improve interpretability of the results. NMF is thus proposed to address this problem. In particular, NMF aims to find the nonnegative basic representative factors which can be used for feature extraction, dimensional reduction, eliminating redundant information and discovering the hidden patterns behind a series of non-negative vectors. NMF has been successfully applied to the field of unsupervised learning in data mining. In [6] it has been shown that when the least squares error is selected as the cost function, NMF is equivalent to soft K-means model, which establishes the theoretical foundation of NMF used for data clustering. Besides the traditional least squares error (Frobenius norm), there are other divergence functions that can be used as the cost functions for NMF, such as generalized K-L divergence and chi-square statistic ([2][7]). In [7] it has been shown that constrained NMF using with generalized K-L divergence has a close relationship with Probabilistic Latent Semantic Indexing.

Probabilistic Latent Semantic Indexing (PLSI, [8]), another state-of-the-art unsupervised learning model in data mining, is a probabilistic model stemmed from Latent Semantic Analysis (LSA, [9]) . Compared to LSA, PLSI has a more solid theoretical foundation in statistics and thus is a more principled approach for analyzing text, discovering latent topics and information retrieval, etc. The parameters in PLSI model are trained by the Expectation Maximization (EM) algorithm which iteratively increases the objective likelihood function until some convergence condition is satisfied. Interestingly, it is proved that maximizing the objective likelihood function in PLSI is equivalent to minimizing the generalized K-L divergence in NMF. Hence NMF and PLSI optimize the same objective function (generalized K-L divergence).

In [7][10] it has been shown that any local solution of PLSI is also a solution of NMF with generalized K-L divergence and vice versa, i.e., the results of PLSI and NMF are equivalent. Indeed, algorithms of NMF and PLSI are both of gradient descent in nonlinear programming and the solutions of NMF and PLSI both satisfy KKT conditions ([11][12][8]). But this does not mean that the

doi:10.4156/jcit.vol6.issue9.25

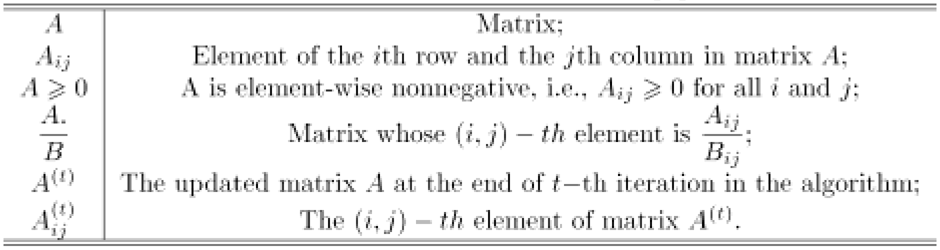
algorithms of NMF and PLSI are identical because there are many local solutions for the objective function (generalized K-L divergence) and even if starting from the same initializations, NMF and PLSI may converge to different local solutions. [10] did not further discuss the differences of the algorithms and the reasons. In this submission, we focus on analyzing the algorithm differences of NMF and PLSI. We will show that the differences are due to the normalization constraints in PLSI. But even if one normalizes the factors of NMF to satisfy the constraints, the normalized version of NMF cannot replace PLSI. Thus we have revealed finely the differences of NMF and PLSI. To our knowledge, this is the first time to address this problem and report it explicitly. Its implications will also be discussed.

In summary, the algorithms of NMF and PLSI are very similar, as will be shown later, but different. Analysis also shows that though NMF and PLSI give equivalent solutions, NMF is faster.

The rest of the paper is organized as follows: Sect. 2 briefly reviews Nonnegative Matrix Factorization (NMF) using generalized K-L divergence, Sect. 3 briefly reviews Probabilistic Latent Semantic Indexing (PLSI), Sect. 4 discusses the normalization of NMF, Sect. 5 gives the relations between NMF and PLSI,, and Sect. 6 concludes.

To easy explanation, Table 1 lists the notations used throughout this paper.

Table 1. Notations used in this paper



A	Matrix;
A_{ij}	Element of the ith row and the jth column in matrix A;
$A \geqslant 0$	A is element-wise nonnegative, i.e., $A_{ij} \geqslant 0$ for all i and j;
$\frac{A.}{B}$	Matrix whose $(i,j)-th$ element is $\frac{A_{ij}}{B_{ij}}$;
$A^{(t)}$	The updated matrix A at the end of $t-$th iteration in the algorithm;
$A^{(t)}_{ij}$	The $(i,j)-th$ element of matrix $A^{(t)}$.

2. NMF using Generalized K-L divergence

In general, NMF can be formulated as follows: given a nonnegative matrix X[1] of size $n \times m$, try to find two nonnegative matrices F of size $n \times k$ and G of size $m \times k$, where k is a predefined parameter satisfying $k \ll m, n$, such that $X \approx FG^T$. This is typically an optimization problem which can be expressed as:

$$\min_{F \geqslant 0, G \geqslant 0} J(X, FG^T),$$

where $J(A, B)$ is some divergence function that measures the dissimilarity between A and B. Specifically, if the generalized K-L divergence is selected, the problem is:

$$\min_{F \geqslant 0, G \geqslant 0} \sum_{i,j} (X_{ij} \log \frac{X_{ij}}{(FG^T)_{ij}} - X_{ij} + (FG^T)_{ij}).$$

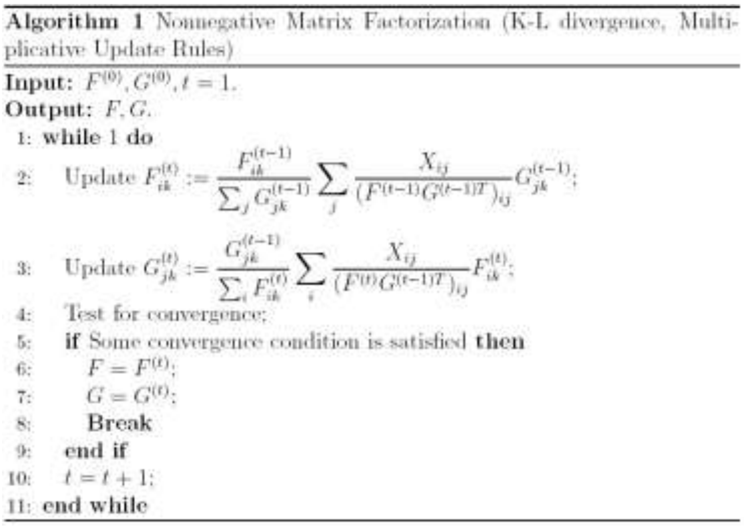
The corresponding algorithm is got by gradient descent and can be summarized as Algorithm 1[2]

3. Probabilistic Latent Semantic Indexing

In this section, we briefly review the PLSI model. PLSI is one of the topic models and, given a joint probabilistic matrix X (i.e., $\sum_{i,j} X_{ij} = 1$.), aims to get three nonnegative matrices C, diagonal S and H such that CSH^T is approximation of X. As a stochastic model, PLSI maintains the column normalization property of C, S and H at each step ($\sum_i C_{ik} = 1, \sum_k S_{kk} = 1, \sum_j H_{jk} = 1$).

[1] Unless otherwise defined in this submission, X satisfies $\sum_{i,j} X_{ij} = 1$.

[2] We mainly consider the multiplicative update algorithm of NMF in this paper.



Algorithm 1 Nonnegative Matrix Factorization (K-L divergence, Multiplicative Update Rules)

```
Input: $F^{(0)}, G^{(0)}, t = 1$.
Output: $F, G$.
1: while 1 do
2:   Update $F_{ik}^{(t)} := \frac{F_{ik}^{(t-1)}}{\sum_j G_{jk}^{(t-1)}} \sum_j \frac{X_{ij}}{(F^{(t-1)}G^{(t-1)T})_{ij}} G_{jk}^{(t-1)}$;
3:   Update $G_{jk}^{(t)} := \frac{G_{jk}^{(t-1)}}{\sum_i F_{ik}^{(t)}} \sum_i \frac{X_{ij}}{(F^{(t)}G^{(t-1)T})_{ij}} F_{ik}^{(t)}$;
4:   Test for convergence;
5:   if Some convergence condition is satisfied then
6:     $F = F^{(t)}$;
7:     $G = G^{(t)}$;
8:     Break
9:   end if
10:  $t = t + 1$;
11: end while
```

For simplifying explanation, we take the document analysis task as an example. Given a document collection $X_{n\times m}$ of m documents and a vocabulary of n words, where each element x_{ij} indicates whether a word w_i occurs in document d_j, the learning task in PLSI is to find three matrices C, H and S, such that X is approximated by CSH^T, where C_{ik} is the probability of $P(w_i|z_k)$ [3], H_{jk} is the probability of $P(d_j|z_k)$ and S is diagonal matrix with diagonal element $S_{kk} = P(z_k)$.

To learn the PLSI model, we can consider maximizing the log-likelihood of the PLSI model $L = \sum_{i,j} n(i,j) log P(w_i, d_j)$, where $n(i,j)$ is the co-occurrence number of word i and document j, and $P(w_i, d_j) = \sum_k P(w_i|z_k)P(z_k)P(d_j|z_k) = \sum_k C_{ik}S_{kk}H_{jk}$. Here we normalize X to satisfy $\sum_{i,j} X_{ij} = 1$, and the log-likelihood function can be rewritten as: $L = \sum_{i,j} X_{ij} \log P(w_i, d_j)$. The parameters C, S and H are then iteratively got by Expectation-Maximization (EM) algorithm.

The EM algorithm begins with some initial values of C, H, S and iteratively updates them according to the following formulas:

$$C_{ik} := \frac{\sum_j X_{ij}P_{ij}^k}{\sum_{i,j} X_{ij}P_{ij}^k}, \quad S_{kk} := \sum_{i,j} X_{ij}P_{ij}^k, \quad H_{jk} := \frac{\sum_i X_{ij}P_{ij}^k}{\sum_{i,j} X_{ij}P_{ij}^k}. \tag{1}$$

where P_{ij}^k is the probability of

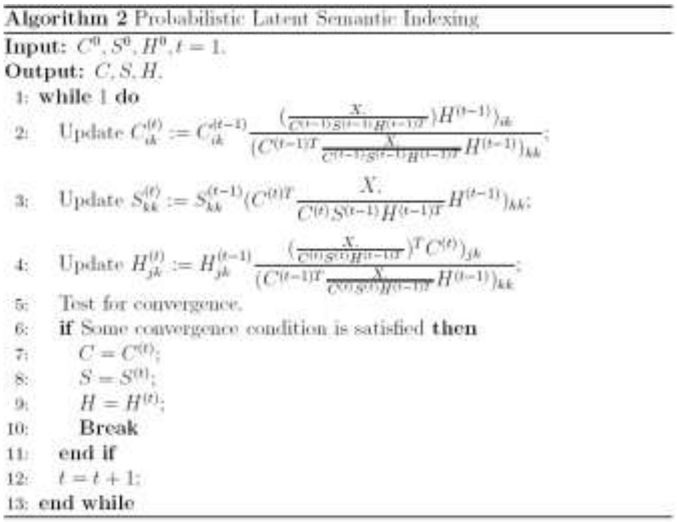
$$P(z_k|w_i, d_j) = \frac{S_{kk}C_{ik}H_{jk}}{\sum_k S_{kk}C_{ik}H_{jk}} \tag{2}$$

By combining (1) and (2), one can get:

$$\begin{aligned} C_{ik} &:= \frac{\sum_j X_{ij}\frac{S_{kk}C_{ik}H_{jk}}{\sum_k S_{kk}C_{ik}H_{jk}}}{\sum_{i,j} X_{ij}\frac{S_{kk}C_{ik}H_{jk}}{\sum_k S_{kk}C_{ik}H_{jk}}} & H_{jk} &:= \frac{\sum_i X_{ij}\frac{S_{kk}C_{ik}H_{jk}}{\sum_k S_{kk}C_{ik}H_{jk}}}{\sum_{i,j} X_{ij}\frac{S_{kk}C_{ik}H_{jk}}{\sum_k S_{kk}C_{ik}H_{jk}}} & S_{kk} &:= S_{kk}\frac{\sum_{ij} X_{ij}C_{ik}H_{jk}}{\sum_k S_{kk}C_{ik}H_{jk}} \\ &= C_{ik}\frac{(\frac{X.}{CSH^T})H)_{ik}}{(C^T\frac{X.}{CSH^T}H)_{kk}}; & &= H_{jk}\frac{(\frac{X.}{CSH^T})^T C)_{jk}}{(C^T\frac{X.}{CSH^T}H)_{kk}}; & &= S_{kk}(C^T\frac{X.}{CSH^T}H)_{kk}. \end{aligned} \tag{3}$$

[3] z_k means the latent topic is k.

The algorithm of PLSI is summarized in Algorithm 2.



```
Algorithm 2 Probabilistic Latent Semantic Indexing
Input: $C^0, S^0, H^0, t = 1.$
Output: $C, S, H.$
while 1 do
    Update $C_{ik}^{(t)} := C_{ik}^{(t-1)} \frac{(\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{ik}}{(C^{(t-1)T}\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{kk}}$;
    Update $S_{kk}^{(t)} := S_{kk}^{(t-1)}(C^{(t)T}\frac{X.}{C^{(t)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{kk}$;
    Update $H_{jk}^{(t)} := H_{jk}^{(t-1)} \frac{(\frac{X.}{C^{(t)}S^{(t)}H^{(t-1)T}})^T C^{(t)})_{jk}}{(C^{(t-1)T}\frac{X.}{C^{(t)}S^{(t)}H^{(t-1)T}}H^{(t-1)})_{kk}}$;
    Test for convergence.
    if Some convergence condition is satisfied then
        $C = C^{(t)}$;
        $S = S^{(t)}$;
        $H = H^{(t)}$;
        Break
    end if
    $t = t + 1$;
end while
```

4. Normalization of NMF

In this section, we will continue to study the normalization of NMF, in other words, we revise NMF to column normalize F and G at each step. The main reason of this consideration is that we want to compare the algorithm differences between NMF and PLSI more easily.

Obviously, in Algorithm 1, it holds that $F^{(t)}G^{(t-1)T} = (F^{(t)}A)(G^{(t-1)}B)^T$ for any two matrices A and B as long as $AB^T = I$ and $F^{(t)}A \geqslant 0, G^{(t-1)}B \geqslant 0$. If we select special A and B such that A is diagonal with $A_{kk} = \sum_i F_{ik}$ and $B = A^{-1}$, then $(F^{(t)}A)$ is column normalization of $F^{(t)}$. Similarly, we can get the column normalization of $G^{(t)}$.

Based on these observations, we revise the standard NMF algorithm as follows: after line 2 in Algorithm 1, we firstly column normalize $F^{(t)}$, and then replace $G^{(t-1)}$ by $(G^{(t-1)}B)^T$, consequently update $G^{(t-1)}$, then normalize $G^{(t)}$ and so on. Thus we get the normalization version of NMF algorithm.

Consequently, we give a conclusion on normalization of NMF. This conclusion can help us understand the algorithm differences between PLSI and NMF more clearly.

Theorem 1: For NMF, at the t th iteration, given the triple factors $C^{(t-1)}$, diagonal matrix $S^{(t-1)}$ and $H^{(t-1)}$, which satisfy $\sum_i C_{ik}^{(t-1)} = 1, \sum_k S_{kk}^{(t-1)} = 1$ and $\sum_j H_{jk}^{(t-1)} = 1,$ as initializations such that $F^{(t-1)} = C^{(t-1)}S^{(t-1)}$ and $G^{(t-1)} = H^{(t-1)}$ or $F^{(t-1)} = C^{(t-1)}$ and $G^{(t-1)} = H^{(t-1)}S^{(t-1)}$, the result $F^{(t)}$ can be equivalently formulated as

$$C_{ik}^{(t)} := C_{ik}^{(t-1)} \frac{(\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{ik}}{(C^{(t-1)T}\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{kk}}, \tag{4}$$

$$S_{kk}^{(t)} := S_{kk}^{(t-1)}(C^{(t-1)T}\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{kk} \tag{5}$$

such that

$$F^{(t)} = C^{(t)}S^{(t)}. \tag{6}$$

Proof: Without loss of generality, suppose that $F^{(t-1)} = C^{(t-1)}S^{(t-1)}$ and $G^{(t-1)} = H^{(t-1)}$. Using the update rules of NMF, one has:

$$F_{ik}^{(t)} \quad := \quad \frac{F_{ik}^{(t-1)}}{\sum\limits_j G_{jk}^{(t-1)}} \sum_j \frac{X_{ij}}{(F^{(t-1)}G^{(t-1)T})_{ij}} G_{jk}^{(t-1)}.$$

Obviously,

$$F^{(t)} = (F^{(t)}D_F^{-1})(D_F), \tag{7}$$

where D_F is diagonal matrix and the diagonal element $(D_F)_{kk}$ is $\sum_i F_{ik}^{(t)}$.

Let

$$C_{ik}^{(t)} = (F^{(t)}D_F^{-1})_{ik}, \tag{8}$$

then one has:

$$\begin{aligned} C_{ik}^{(t)} \quad &:= \quad F_{ik}^{(t)}/(\sum_i F_{ik}^{(t)}) \\ &= \quad \frac{\frac{F_{ik}^{(t-1)}}{\sum\limits_j G_{jk}^{(t-1)}} \sum_j \frac{X_{ij}}{(F^{(t-1)}G^{(t-1)T})_{ij}} G_{jk}^{(t-1)}}{\sum_i \frac{F_{ik}^{(t-1)}}{\sum\limits_j G_{jk}^{(t-1)}} \sum_j \frac{X_{ij}}{(F^{(t-1)}G^{(t-1)T})_{ij}} G_{jk}^{(t-1)}} \\ &= \quad C_{ik}^{(t-1)} \frac{(\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}} H^{(t-1)})_{ik}}{(C^{(t-1)T} \frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}} H^{(t-1)})_{kk}}. \end{aligned}$$

Secondly, let

$$S_{kk}^{(t)} := (D_F)_{kk}, \tag{9}$$

then one has:

$$\begin{aligned} S_{kk}^{(t)} \quad &:= \quad \sum_i F_{ik}^{(t)} \\ &= \quad S_{kk}^{(t-1)}(C^{(t-1)T} \frac{X.}{C^{(t-1)}S^{(t-1)}G^{(t-1)T}} G^{(t-1)})_{kk}. \end{aligned}$$

Thus the conclusion is proved.

From the above theorem, we can see that $C^{(t)}$ is column normalization of $F^{(t)}$, and the update rule of C is given. In corollary 2, we give an interesting property of $S^{(t)}$. Note that it is different from the conclusion in **Sec. Normalizations of NMF** in ref. [7] in that our property holds at each step in NMF algorithm.

Corollary 2: For NMF, at the tth iteration, $\sum\limits_i C_{ik}^{(t)} = 1$ and $\sum\limits_k S_{kk}^{(t)} = 1$.

Proof: $\sum\limits_i C_{ik}^{(t)} = 1$ is obviously true.

Secondly,

$$\begin{aligned}\sum_k S_{kk}^{(t)} &= \sum_k S_{kk}^{(t-1)}(C^{(t-1)T}\frac{X.}{C^{(t-1)}S^{(t-1)}G^{(t-1)T}}G^{(t-1)})_{kk}\\ &= \sum_k S_{kk}^{(t-1)}(\sum_{ij} C_{ik}^{(t-1)}\frac{X_{ij}}{(C^{(t-1)}S^{(t-1)})G^{(t-1)T})_{ij}}H_{jk}^{(t-1)})\\ &= 1.\end{aligned}$$

For G in NMF, we have similar result.

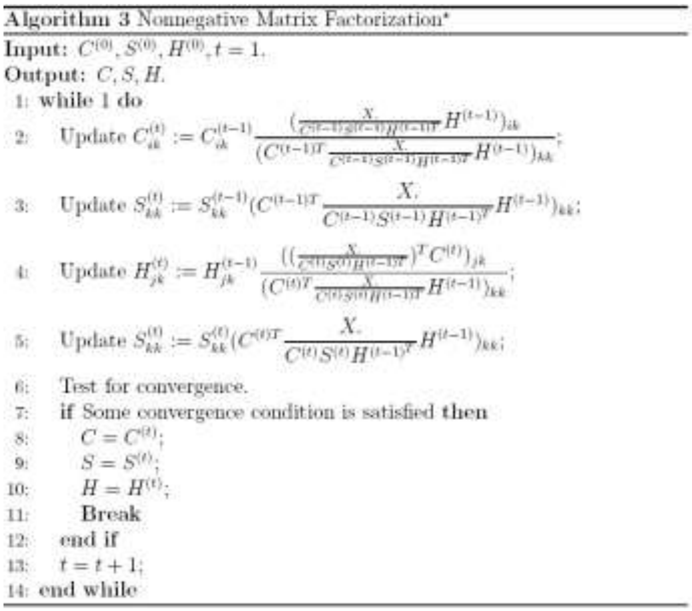
Corollary 3: For NMF, at the t th iteration, given the triple factors $C^{(t-1)}$, diagonal matrix $S^{(t-1)}$ and $H^{(t-1)}$, which satisfy $\sum_i C_{ik}^{(t-1)} = 1, \sum_k S_{kk}^{(t-1)} = 1$ and $\sum_j H_{jk}^{(t-1)} = 1,$ as initializations such that $F^{(t-1)} = C^{(t-1)}S^{(t-1)}$ and $G^{(t-1)} = H^{(t-1)}$ or $F^{(t-1)} = C^{(t-1)}$ and $G^{(t-1)} = H^{(t-1)}S^{(t-1)}$, the result $G^{(t)}$ can be equivalently formulated as

$$\begin{aligned}H_{jk}^{(t)} &:= H_{jk}^{(t-1)}\frac{((\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}})^T C^{(t-1)})_{jk}}{(C^{(t-1)T}\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{kk}},\\ S_{kk}^{(t)} &:= S_{kk}^{(t-1)}(C^{(t-1)T}\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{kk}\end{aligned}$$

such that

$$G^{(t)} = H^{(t)}S^{(t)}.$$

Based on the above discussions, we can revise Algorithm 1 to Algorithm 3.



```
Algorithm 3 Nonnegative Matrix Factorization*
Input: $C^{(0)}, S^{(0)}, H^{(0)}, t = 1.$
Output: $C, S, H.$
1: while 1 do
2:   Update $C_{ik}^{(t)} := C_{ik}^{(t-1)}\frac{(\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{ik}}{(C^{(t-1)T}\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{kk}}$;
3:   Update $S_{kk}^{(t)} := S_{kk}^{(t-1)}(C^{(t-1)T}\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{kk}$;
4:   Update $H_{jk}^{(t)} := H_{jk}^{(t-1)}\frac{((\frac{X.}{C^{(t)}S^{(t)}H^{(t-1)T}})^T C^{(t)})_{jk}}{(C^{(t)T}\frac{X.}{C^{(t)}S^{(t)}H^{(t-1)T}}H^{(t-1)})_{kk}}$;
5:   Update $S_{kk}^{(t)} := S_{kk}^{(t)}(C^{(t)T}\frac{X.}{C^{(t)}S^{(t)}H^{(t-1)T}}H^{(t-1)})_{kk}$;
6:   Test for convergence.
7:   if Some convergence condition is satisfied then
8:     $C = C^{(t)}$;
9:     $S = S^{(t)}$;
10:    $H = H^{(t)}$;
11:    Break
12:  end if
13:  $t = t + 1$;
14: end while
```

Note that the normalization version of NMF will converge to a different local optimum from the standard NMF. But the revised version has a close relation with the standard one: any stationary point of Algorithm 3 is also a stationary point of Algorithm 1, and vice versa. More discussions will come in Sect. 5.

Theorem 4：Any stationary point of Algorithm 3 is also a stationary point of Algorithm 1.

Proof： This is obviously true by joining line 2 and line 3, line 4 and line 5 in Algorithm 3 at convergence.

After studying normalization of NMF carefully, we can now have a better insight into the algorithm differences between PLSI and NMF.

5. Algorithm Relations Between PLSI and NMF

The following conclusions give the relations of C (in PLSI) and F (in NMF), H (in PLSI) and G (in NMF).

Theorem 5 (One step equivalence): For PLSI and NMF, at the t th iteration, given the triple factors $C^{(t-1)}, S^{(t-1)}$ and $H^{(t-1)}$ as initializations of PLSI and $F^{(t-1)}, G^{(t-1)}$ as initializations of NMF such that $C^{(t-1)}S^{(t-1)} = F^{(t-1)}$ and $H^{(t-1)} = G^{(t-1)}$ or $C^{(t-1)} = F^{(t-1)}$ and $H^{(t-1)}S^{(t-1)} = G^{(t-1)}$ (i.e., $C^{(t-1)}S^{(t-1)}H^{(t-1)T} = F^{(t-1)}G^{(t-1)T}$), the update rules of C and F have the following relations: except for additional normalization, the update rule of C is identical with that of F in NMF, i.e., $C^{(t)} = F^{(t)}D_F^{-1}$,where D_F is diagonal matrix and the diagonal element $(D_F)_{kk} = \sum\limits_i F_{ik}^{(t)}$.

Proof: The result is obviously true from (3), (4), (5) and (6).

Corollary 6: For PLSI and NMF, at the t th iteration, given the triple factors $C^{(t-1)}, S^{(t-1)}$ and $H^{(t-1)}$ as initializations of PLSI and $F^{(t-1)}, G^{(t-1)}$ as initializations of NMF such that $C^{(t-1)}S^{(t-1)} = F^{(t-1)}$ and $H^{(t-1)} = G^{(t-1)}$ or $C^{(t-1)} = F^{(t-1)}$ and $H^{(t-1)}S^{(t-1)} = G^{(t-1)}$ (i.e., $C^{(t-1)}S^{(t-1)}H^{(t-1)T} = F^{(t-1)}G^{(t-1)T}$), the update rules of H and G have the following relations: except for additional normalization, the update rule of H is identical with that of G in NMF, i.e., $H^{(t)} = G^{(t)}D_G^{-1}$, where D_G is diagonal matrix and the diagonal element $(D_G)_{kk} = \sum\limits_j G_{jk}^{(t)}$.

Hence, NMF with normalization at each iteration has close relationship with PLSI. But this does not mean that PLSI can be replaced by NMF by normalizing F and G at each step, which can be observed from Algorithm 3 and Algorithm 2.

The key reason is that PLSI imposes normalization conditions on the factors explicitly. In [7] it has been shown that PLSI and NMF optimize the same objective function, hence PLSI can be seen as NMF-based model with additional normalization constraints ($\sum_i C_{ik} = 1, \sum_j H_{jk} = 1, \sum_k S_{kk} = 1$). The derivation process of PLSI update rules of C and H can be separated into two steps. Take the update rule of C while fixing S and H for example: firstly one gets the un-normalized C by gradient descent (identical with NMF), and then normalizes C to satisfy the constraint $\sum_i C_{ik} = 1$. The update rule of H is got in a similar way. The update rule of S can be got even more simply, just by gradient descent, and the normalization constraints will be satisfied automatically. In detail, at the t th iteration, firstly, the derivative of the cost function $J(X, CSH^T)$ with respect to S while fixing C and H is:

$$\begin{aligned}\frac{\partial}{\partial S_{kk}} J &= -\sum_{ij} \frac{X_{ij}C_{ia}H_{ja}}{\sum\limits_k S_{kk}C_{ik}H_{jk}} + \sum_{ij} C_{ia}H_{ja} \\ &= -\sum_{ij} \frac{X_{ij}C_{ia}H_{ja}}{\sum\limits_k S_{kk}C_{ik}H_{jk}} + 1.\end{aligned}$$

Let the step size $\eta_{kk} = S_{kk}$, then the update rule of S is:

$$\begin{aligned}S_{kk} &= S_{kk} + \eta_{kk}(\sum_{ij} \frac{X_{ij}C_{ia}H_{ja}}{\sum\limits_k S_{kk}C_{ik}H_{jk}} - 1) \\ &= S_{kk}(C^T \frac{X.}{CSH^T} H)_{kk}.\end{aligned}$$

In [10] it has been shown that any local optimum solution of PLSI is also a solution of NMF with K-L divergence, and vice versa. Theorem 4 has shown similar results between normalized NMF and standard NMF. These results mean that given the same initializations, PLSI, NMF and normalized

NMF will give equivalent solutions. Furthermore, we observe that their solution values are always identical:

$$CSH^T = FG^T = F^*G^{*T}. \qquad (10)$$[4]

Indeed, this phenomenon is very common in NMF. Roughly speaking, the standard NMF algorithm can be expressed like this: update F, then update G and so on. Now we revise it to: $\underbrace{\texttt{update } F,\ \texttt{update } F,\ \cdots,\texttt{update } F}_{\texttt{m times}}$, then $\underbrace{\texttt{update } G,\texttt{update } G,\cdots,\texttt{update } G}_{\texttt{n times}}$, and so on. Choosing different m and n, we can get infinitely many solutions even if given the same initializations. But these solutions are all having the same solution values.

6. Implications and Conclusion

In this paper we show that though NMF and PLSI optimize the same objective function and can generate equivalent solutions, still their algorithms are different and PLSI cannot be replaced by NMF. The reason is that the EM algorithm for PLSI iteratively update the factors C, S and H such that C, S and H satisfy the normalization constraints at each iteration while NMF does not include these constraints. But since NMF with generalized K-L divergence has the interesting fixed row sum and fixed column sum property, it can also give solutions of PLSI.

This work, combined with the previous results [10][7], has the implications as follows:

Firstly, we have revealed finely the algorithm differences of NMF and PLSI. By comparing Algorithm 2 with Algorithm 3, one can see that even if we revise the stand NMF algorithm 1 to normalized NMF, it cannot yet replace PLSI.

Secondly, Algorithms $1 \sim 3$ can generate equivalent solutions, but NMF is faster. In other words, NMF and PLSI optimize the same objective function (generalized K-L divergence), and PLSI model can be solved by NMF Algorithm 1 and vice versa. Both NMF and PLSI can be used to extract the data structures, such as clusters or latent topics, and their performance are comparable ([7]). This means that: on the one hand, similar to NMF, PLSI can also be used to analyze cluster structures in documents; on the other hand, besides PLSI, the results of NMF can also have a probability interpretation and from numerical optimization point of view, NMF can generate comparable good results with PLSI. Note that like PLSI, NMF also has hierarchical extensions, which has been successfully applied to a small subset of scientific abstracts from PubMed [16][12][17]. Hence this work makes it more easily for people coming from different backgrounds, who may familiar with NMF or PLSI, understand each other and enhances communications among different fields. For example, a problem of natural language processing, which may be solved traditionally by PLSI, can also be solved by NMF now and even can be done more effectively.

Thirdly, NMF is more flexible regarding the choices of its objective functions and the algorithms employed to solve it. For example, symmetric NMF with least squares error is equivalent to soft K-means [6]. NMF using posterior probability normalization (PPC) is more explainable [18]. In particular, NMF can be viewed as a general unsupervised learning model, and K-means and PLSI are just variations of NMF. Furthermore, NMF-based algorithms are more powerful to solve them (more accuracy and robust than greedy algorithm for K-means [6] and faster than EM for PLSI).

Fourthly, since PLSI has been further developed into Latent Dirichlet Allocation (LDA), which overcomes the difficulty of assigning probability to the new documents [20], one can also consider to develop NMF into some LDA-like model.

Finally, we give an open problem related to this paper: why (10) holds? In other words, since they converge to different local solutions, why the solution values are always identical?

[4]
CSH^T: Results by PLSI
FG^T: Results by NMF
F^*G^{*T}: Results by normalized NMF

7. Acknowledgements

This work is supported by the Foundation of Academic Discipline Program at Central University of Finance and Economics, P.R.China.

8. References

[1] D. D. Lee, H. S. Seung, “Learning the parts of objects by non-negative matrix factorization”, Nature, vol. 401, no. 6755, pp. 788--791, 1999.

[2] D. D. Lee, H. S. Seung, “Algorithms for non-negative matrix factorization”, Annual Conference on Neural Information Processing Systems, pp. 556--562, 2000.

[3] P. Paatero, U. Tapper, “Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values”, Environmetrics, vol. 5, no. 2, pp. 111--126, 1994.

[4] I. T. Jolliffe, “Principal Component Analysis”, Springer, USA, second edition, 2002

[5] T.Hastie, R. Tibshirani, J. Friedman, “The Elements of Statistical Learning: Data Mining, Inference, and Prediction”, Springer, USA, second edition, 2009

[6] C. Ding, X.He, H. D. Simon, “On the equivalence of nonnegative matrix factorization and spectral clustering”, SIAM Data Mining Conf, pp. 606--610, 2005.

[7] C. Ding, T. Li, W. Peng, “Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method”, Proceedings of the National Conference on Artificial Intelligence, pp. 342-347, 2006.

[8] T. Hofmann, “Probabilistic latent semantic indexing”, SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 50--57, 1999.

[9] S. C. Deerwester, S. T. Dumais, T.K. Landauer, G. W. Furnas, R. A. Harshman, “Indexing by latent semantic analysis”, Journal of the American Society for Information Science, vol. 41, no. 6, pp. 391--407, 1990.

[10] E. Gaussier, C. Goutte, “Relation between PLSI and NMF and implications”, SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 601--602, 2005.

[11] A. Cichocki, H. Lee, Y. D. Kim, S. Choi, “Non-negative matrix factorization with α-divergence”, Pattern Recogn. Lett., vol. 29, no. 9, pp. 1433--1440, 2008.

[12] I. S. Dhillon, S. Sra, “Generalized nonnegative matrix approximations with Bregman divergences”, Neural Information Proc. Systems (NIPS), pp. 283--290, 2005.

[13] A. Strehl, J. Ghosh, “Cluster ensembles: a knowledge reuse framework for combining partitionings”, Eighteenth national conference on Artificial intelligence, pp. 93--98, 2002.

[14] W. Xu, X. Liu, Y.Gong, “Document clustering based on non-negative matrix factorization”, SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 267--273, 2003.

[15] N. D. Ho, P. V. Dooren, “Non-negative matrix factorization with fixed row and column sums”, Linear Algebra and its Applications, vol. 429, no. 5-6, pp. 1020--1025, 2008.

[16] A. Cichocki, R. Zdunek, “Multilayer nonnegative matrix factorization using projected gradient approaches”, Int. J. Neural Syst., vol. 17, no. 6, pp. 431--446, 2007.

[17] F. A. Nielsen, D. Balslev, L.K. Hansen, “Mining the posterior cingulate: Segregation between memory and pain components”, NeuroImage, vol. 27, no.3, pp. 520--532, 2005.

[18] C. Ding, T. Li, D. Luo, W. Peng, “Posterior probabilistic clustering using nmf”, SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pp.831--832, 2008.

[19] C.Ding, T. Li,M.Jordan, “Convex and Semi-Nonnegative Matrix Factorization”, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 32, no. 1, pp. 45--55, 2010.

[20] D. M. Blei, A. Y. Ng, M. I. Jordan, “Latent dirichlet allocation”, J. Mach. Learn. Res., vol. 3, no. 4-5, pp. 993--1022, 2003.

[21] S. Zafeiriou, M. Petrou, “Nonnegative tensor factorization as an alternative Csiszar–Tusnady procedure: algorithms, convergence, probabilistic interpretations and novel probabilistic tensor latent variable analysis algorithms”, Data Mining and Knowledge Discovery, vol. 22, no. 3, pp. 419--466, 2011.

[22] G Li, Y Wang, "A New Method for Privacy-Preserving Data Mining Based on Weighted Singular Value Decomposition", JCIT: Journal of Convergence Information Technology, vol. 6, no. 3, pp. 28--34, 2011

[23] A.N.M. Rezaul Karim, Thwarique, "Face Recognition using Matrix Decomposition Technique Eigenvectors and SVD", IJACT: International Journal of Advancements in Computing Technology, vol. 2, no. 1, pp. 64--71, 2010