On the Equivalence Between Nonnegative Matrix Factorization and Probabilistic Latent Semantic Indexing

Chris Ding [1], Tao Li [2] and Wei Peng [2]
[1] Department of CSE, University of Texas at Arlington, Arlington, TX 76019
[2] School of Computer Science, Florida International University, Miami, FL 33199

January 27, 2008

Abstract

Non-negative Matrix Factorization (NMF) and Probabilistic Latent Semantic Indexing (PLSI) have been successfully applied to document clustering recently. In this paper, we show that PLSI and NMF (with the I-divergence objective function) optimize the same objective function, although PLSI and NMF are different algorithms as verified by experiments. This provides a theoretical basis for a new hybrid method that runs PLSI and NMF alternatively, each jumping out of local minima of the other method successively, thus achieving a better final solution. Extensive experiments on five real-life datasets show relations between NMF and PLSI, and indicate the hybrid method leads to significant improvements over NMF-only or PLSI-only methods. We also show that at first order approximation, NMF is identical to χ^2-statistic.

1 Introduction

Document clustering has been widely used as a fundamental and effective tool for efficient document organization, summarization, navigation, and retrieval of large amount of documents. Generally document clustering problems are determined by three basic tightly-coupled components: a) the (physical) representation of the given dataset; b) The criterion/objective function which the clustering solutions should aim to optimize; and c) The optimization procedure [10].

Among clustering methods, the K-means algorithm has been the most popularly used. A recent development is the Probabilistic Latent Semantic Indexing (PLSI). PLSI is an unsupervised learning method based on statistical latent class models and has been successfully applied to document clustering [7]. (PLSI is further developed into a more comprehensive Latent Dirichlet Allocation model [1].)

Nonnegative Matrix Factorization (NMF) is another recent development for document clustering. The initial work on NMF [8, 9] emphasizes that the NMF factors contain coherent parts of the original data (images). Later works [13, 11] show the

usefulness of NMF for clustering with experiments on documents collections, and a recent theoretical analysis [2] shows the equivalence between NMF and K-means / spectral clustering.

Despite significant research on both NMF and PLSI, few attempts have been made to establish the connections between them while highlighting their differences in the clustering framework. Gaussier and Goutte [4] made the initial connection between NMF and PLSI, by showing that the iterative update procedures of PLSI and NMF are similar in that the fixed-point equations for the converged solutions are the same. However, we emphasize that NMF and PLSI are different algorithms: they converge to different solutions, even if they start from the same initial condition, as verified by experiments (see later sections).

In this paper, we first show that both NMF (with I-divergence objective) and PLSI optimize the same objective function. This fundamental fact and the L_1 normalization NMF ensures that NMF and PLSI are equivalent. In other words, PLSI is equivalent to NMF with I-divergence objective.

Second, we show, by an example and extensive experiments, that NMF and PLSI are different algorithms and they converge to different local minima. This leads to a new insight: NMF and PLSI are different algorithms for optimizing the same objective function.

Third, we give a detailed analysis about the NMF and PLSI solutions. They are local minima of the same landscape in a very high dimensional space. We show that PLSI can jump out of the local minima where NMF converges to and vice versa. Based on this, we further propose a hybrid algorithm to run NMF and PLSI alternatively to jump out a series of local minima and finally reach to a much better minimum. Extensive experiments show this hybrid algorithm improves significantly over the standard NMF-only or PLSI-only algorithms.

A preliminary version of this paper is appeared in [3]. More theoretical analysis and experiments are included in the journal version. The rest of the paper is organized as follows: Section 2 introduces the data representations of NMF and PLSI, Section 3 presents the equivalence between NMF and PLSI, Section 4 shows the column normalized NMF is equivalent to the probability factorization, Section 5 uses an example to illustrate the difference between NMF and PLSI, Section 6 gives the empirical comparison results between NMF and PLSI, Section 7 proposes a hybrid algorithm to run NMF and PLSI alternatively and finally Section 8 concludes.

2 Data Representations of NMF and PLSI

Suppose we have n documents and m words (terms). Let $F = (F_{ij})$ be the word-to-document matrix: $F_{ij} = F(w_i, d_j)$ is the frequency of word w_i in document d_j.

In this paper, we re-scale the term frequency F_{ij} by $F_{ij} \leftarrow F_{ij}/T_w$, where $T_w = \sum_{ij} F_{ij}$ is the total number of words. With this stochastic normalization, $\sum_{ij} F_{ij} = 1$. The joint occurrence probability $p(w_i, d_j) = F_{ij}$.

The general form of NMF is

$$F = CH^T, \tag{1}$$

where the matrices $C = (C_{ik}), H = (H_{jk})$ are nonnegative matrices. They are determined by minimizing

$$J_{\text{NMF}} = \sum_{i=1}^{m}\sum_{j=1}^{n} F_{ij}\log\frac{F_{ij}}{(CH^T)_{ij}} - F_{ij} + (CH^T)_{ij} \tag{2}$$

PLSI maximizes the likelihood

$$\max J_{\text{PLSI}}, \quad J_{\text{PLSI}} = \sum_{i=1}^{m}\sum_{j=1}^{n} F_{ij}\log P(w_i, d_j) \tag{3}$$

where $P(w_i, d_j)$ is the factorized (i.e., parameterized or approximated) joint occurrence probability

$$\begin{aligned} P(w_i, d_j) &= \sum_k P(w_i, d_j|z_k)P(z_k) \\ &= \sum_k P(w_i|z_k)P(d_j|z_k)P(z_k), \end{aligned} \tag{4}$$

assuming that w_i and d_j are conditionally independent given z_k. The probability factors follow the normalization of probabilities

$$\sum_{i=1}^{m} p(w_i|z_k) = 1, \sum_{j=1}^{n} p(d_j|z_k) = 1, \sum_{k=1}^{K} p(z_k) = 1. \tag{5}$$

3 Equivalence of NMF and PLSI

In this section, we present our main results:

Theorem 1. PLSI and NMF are equivalent.

The proof of Theorem 1 is better described by the following two propositions.

Proposition 1. The objective function of PLSI is identical to the objective function of NMF, i.e.,

$$\max J_{\text{PLSI}} \quad \Longleftrightarrow \quad \min J_{\text{NMF}} \tag{6}$$

Proposition 2. Column normalized NMF of Eq.(1) is equivalent to the probability factorization of Eq.(4), i.e., $(CH^T)_{ij} = P(w_i, d_j)$.

Proof of Theorem 1: By Proposition 2, NMF (with L_1-normalization, see §4) is identical to PLSI factorization. By Proposition 1, they minimize the same objective function. Therefore, NMF is identical to PLSI. □

We proceed to prove Proposition 1 in this section. The Proposition 2 will be proved in the §4.

Proof of Proposition 1:

First, we note that the PLSI objective function Eq.(3) can be written as

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n} -F_{ij}\log P(w_i, d_j).$$

Adding a constant, $\sum_{i=1}^{m}\sum_{j=1}^{n} F_{ij}\log F_{ij}$, PLSI is equivalent to solve

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n} F_{ij}\log \frac{F_{ij}}{P(w_i, d_j)}.$$

Now since

$$\sum_{i=1}^{m}\sum_{j=1}^{n}[P(w_i, d_j) - F_{ij}] = [1-1] = 0,$$

we can add this constant to the summation; thus PLSI is equivalent to minimize

$$\sum_{i=1}^{m}\sum_{j=1}^{n} F_{ij}\log \frac{F_{ij}}{P(w_i, d_j)} - F_{ij} + P(w_i, d_j) \tag{7}$$

This is precisely the objective function for NMF. □

3.1 NMF and χ^2-statistic

J_{NMF} of Eq.(2) has a somewhat complicated expression. It is related to the Kullback-Leibler divergence. We give a better understanding by relating it to the familiar χ^2 test in statistics. Assume $\frac{|(CH^T)_{ij}-F_{ij}|}{F_{ij}}$ is small. We can write

$$J_{\text{NMF}} = \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{[(CH^T)_{ij} - F_{ij}]^2}{2F_{ij}} - \frac{[(CH^T)_{ij} - F_{ij}]^3}{3F_{ij}^2} + \cdots \tag{8}$$

Let $\delta_{ij} = (CH^T)_{ij} - F_{ij}$, $z = \delta_{ij}/F_{ij}$. Since $\log(1+z) = z - z^2/2 + z^3/3\cdots$; the ij-th term in J_{NMF} becomes

$$\delta_{ij} - F_{ij}\log\left(1 + \frac{\delta_{ij}}{F_{ij}}\right) = \frac{1}{2}\frac{\delta_{ij}^2}{F_{ij}} - \frac{1}{3}\frac{\delta_{ij}^3}{F_{ij}^2} + \cdots$$

Clearly, the first term in J_{NMF} is the χ^2 statistic,

$$\chi^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{[(CH^T)_{ij} - F_{ij}]^2}{F_{ij}^2}, \tag{9}$$

since F_{ij} is the data and $(CH^T)_{ij}$ is the model fit to it. Therefore, to first order approximation, NMF objective function is a χ^2 statistic. As a consequence, we can associate a confidence to NMF factorization by utilizing the statistic.

The χ^2 form of NMF naturally relates to another NMF cost function, i.e., the sum of squared errors

$$J'_{\text{NMF}} = \sum_{i=1}^{m}\sum_{j=1}^{n}[(CH^T)_{ij} - F_{ij}]^2. \tag{10}$$

4 Normalizations of NMF

For any given NMF solution (C, H), there exist a large number of matrices (A, B) such that $AB^T = I,\ CA \geq 0,\ HB \geq 0$.. Thus (CA, HB) is also a solution with the same cost function value. Normalization is a way to eliminate this uncertainty. We mostly consider the normalization of columns of C, H. Specifically, let the columns be expressed explicitly, $C = (\mathbf{c}_1, \ldots, \mathbf{c}_k),\ H = (\mathbf{h}_1, \ldots, \mathbf{h}_k)$. [1]

We consider column normalization. Let the normalized columns be

$$\tilde{C} = (\widetilde{\mathbf{c}}_1, \ldots, \widetilde{\mathbf{c}}_k),\ \tilde{H} = (\widetilde{\mathbf{h}}_1, \ldots, \widetilde{\mathbf{h}}_k). \tag{11}$$

With this normalization, we can write

$$CH^T = \widetilde{\mathbf{C}} S \widetilde{\mathbf{H}}^T, \tag{12}$$

where

$$\widetilde{\mathbf{C}} = CD_C^{-1},\ \widetilde{\mathbf{H}} = HD_H^{-1},\ S = D_C D_H. \tag{13}$$

D_C, D_H are diagonal matrices. Depending on the normalizations in the Hilbert space, the L_p-normalization, the diagonal elements are given by

$$(D_C)_{kk} = ||\widetilde{\mathbf{c}}_k||_p, \quad (D_H)_{kk} = ||\widetilde{\mathbf{h}}_k||_p.$$

For the standard Euclidean distance normalization, i.e., the L_2-norm

$$||\widetilde{\mathbf{c}}_k||_2 = 1, \quad ||\widetilde{\mathbf{h}}_k||_2 = 1, \tag{14}$$

This is the same as in singular value decomposition where the non-negativity constraint is ignored.

For probabilistic formulations, such as PLSI, we use the L_1 norm.

$$||\widetilde{\mathbf{c}}_k||_1 = 1, \quad ||\widetilde{\mathbf{h}}_k||_1 = 1, \tag{15}$$

Due to the non-negativity, these are just the condition that columns sums to 1. D_C contains the column sums of C and D_H contains the column sums of H.

With these clarification, we now prove Proposition 2.

Proof of Proposition 2:

Using L_1-norm, we obviously have

$$\sum_{i=1}^{m} \widetilde{\mathbf{C}}_{ik} = 1,\ \sum_{j=1}^{n} \widetilde{\mathbf{H}}_{jk} = 1,\ \sum_{k=1}^{n} S_{kk} = 1,$$

where the last equality is proved as

$$1 = \sum_{ij} F_{ij} = \sum_{i=1}^{m} \sum_{k=1}^{K} \sum_{j=1}^{n} \widetilde{\mathbf{C}}_{ik} S_{kk} \widetilde{\mathbf{H}}_{jk} = \sum_{k=1}^{K} S_{kk}.$$

These can be seen as equivalent to the normalization of probabilities of Eq.(5). Therefore, $\widetilde{\mathbf{C}}_{ik} = p(w_i|z_k)$, $\widetilde{\mathbf{H}}_{jk} = p(d_j|z_k)$ and $S_{kk} = p(z_k)$. Thus $F = CH^T = \widetilde{\mathbf{C}} S \widetilde{\mathbf{H}}^T$ factorization with L_1-normalization is identical to PLSI factorization □

[1] In this column form, for clustering interpretation [2], $\mathbf{c}_k$ is the centroid for the k-th cluster, while $\mathbf{h}_k$ is the posterior probability for the k-th cluster. For hard clustering, on each row of H, set the largest element to 1 and the rest to 0.

4.1 A Probabilistic View

We can also interpret the cluster posterior obtained from matrix factorization. We can think of the rectangular input data X as a word-document matrix and perform a PLSI type probabilistic decomposition. As in Eq.(4), the joint occurrence probability $X_{ij} = P(w_i, d_j)$ can be factorized as

$$P(w_i, d_j) = \sum_k p(w_i|z_k)p(z_k)p(d_j|z_k),$$

where z_k is the latent cluster variable, and the probability factors follow the probability normalization

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

If $\sum_{ij} X_{ij} = 1$, then $\sum_{k=1}^{K} p(z_k) = 1$.

With this, the cluster posterior probability for column d_j is then

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

Translating to C, H, the equivalent probabilistic decomposition is

$$X = CH^T = (CD_C^{-1})(D_C D_H)(HD_H^{-1})^T,$$

where $D_C = \text{diag}(\mathbf{e}^T C)$ and $D_H = \text{diag}(\mathbf{e}^T H)$. Thus for standard NMF, the cluster posterior probability for column $\mathbf{x}_i$ is

$$\text{NMF:} \quad p(z_k|\mathbf{x}_i) \propto (HD_H^{-1})(D_C D_H) = (HD_C)_{ik}$$

5 An Illustration of NMF/PLSI Difference

Although NMF and PLSI optimize the same objective function as shown above, they are different computational algorithms. This fact is obvious from experiments. In all of our extensive experiments, starting with the same initial starting C_0, H_0, NMF and PLSI always converge to different solutions. Here we give an illustration. The input data matrix is

$$X = \begin{pmatrix} .048 & .042 & .047 & .024 & .029 & .026 \\ .035 & .040 & .045 & .016 & .023 & .029 \\ .031 & .019 & .031 & .040 & .045 & .042 \\ .027 & .023 & .031 & .032 & .039 & .045 \\ .047 & .043 & .035 & .026 & .021 & .019 \end{pmatrix}$$

The initial C_0, S_0, H_0 are

$$C_0 S_0 H_0^T = \begin{pmatrix} .24 & .20 \\ .02 & .27 \\ .31 & .16 \\ .07 & .26 \\ .36 & .11 \end{pmatrix} \begin{pmatrix} .34 & 0 \\ 0 & .66 \end{pmatrix} \begin{pmatrix} .18 & .19 \\ .15 & .18 \\ .15 & .21 \\ .18 & .12 \\ .18 & .14 \\ .16 & .16 \end{pmatrix}^T$$

Running the NMF algorithm, the converged solution is

$$\tilde{C}S\tilde{H}^T = \begin{pmatrix} .33 & .14 \\ .29 & .12 \\ .02 & .33 \\ .05 & .29 \\ .32 & .11 \end{pmatrix} \begin{pmatrix} .39 & 0 \\ 0 & .61 \end{pmatrix} \begin{pmatrix} .27 & .14 \\ .28 & .09 \\ .25 & .15 \\ .07 & .18 \\ .06 & .22 \\ .06 & .23 \end{pmatrix}^T$$

Running the PLSI algorithm, the converged solution is

$$CSH^T = \begin{pmatrix} .12 & .31 \\ .10 & .28 \\ .38 & .04 \\ .33 & .07 \\ .08 & .31 \end{pmatrix} \begin{pmatrix} .50 & 0 \\ 0 & .50 \end{pmatrix} \begin{pmatrix} .13 & .25 \\ .09 & .25 \\ .14 & .24 \\ .19 & .09 \\ .22 & .09 \\ .23 & .09 \end{pmatrix}^T$$

One can observe that the NMF solution differs from the PLSI solution significantly. Our example shows that starting at the same point in the multi-dimensional space, NMF and PLSI converge to *different* local minima.

However, it is interesting and important to note that in this example the clustering results embedded in the solutions of NMF and PLSI are identical by an examination of H (see footnote 1): the first 3 data points (columns) belong to one cluster, and the rest 3 points belong to another cluster. This result is the same as the K-means clustering. More generally, we introduce a clustering matrix $R = (r_{ij})$, where $r_{ij} = 1$ if $\mathbf{x}_i, \mathbf{x}_j$ belong to the same cluster; $r_{ij} = 0$ otherwise. Thus the clustering results can be expressed as

$$R_{\text{NMF}} = R_{\text{PLSI}} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \tag{20}$$

6 Comparison between NMF and PLSI

In this section, we compare the clustering performance of NMF and PLSI on five real-life datasets.

6.1 Datasets

We use five datasets in our experiments, most of which are frequently used in the information retrieval research. Table 1 summarizes the characteristics of the datasets.

CSTR This dataset contains the abstracts of technical reports (TRs) published in the Computer Science Department at the University of Rochester between 1991 and 2002. The dataset has 476 abstracts, which are divided into four research areas: Natural Language Processing(NLP), Robotics/Vision, Systems, and Theory.

Datasets	# documents	# class
CSTR	476	4
WebKB	4199	4
Log	1367	9
Reuters	2900	10
WebAce	2340	20

Table 1: Dataset Descriptions.

WebKB The dataset contains webpages gathered from university computer science departments. There are about 4199 documents and they are divided into 4 categories: student, faculty, course, and project.

Log This dataset contains 1367 log text messages which are grouped into 9 categories, i.e., *configuration*, *connection*, *create*, *dependency*, *other*, *report*, *request*, *start*, and *stop*.

Reuters The Reuters-21578 Text Categorization Test Collection contains documents collected from the Reuters newswire in 1987. In our experiments, we use a subset of the data collection which includes the 10 most frequent categories among the 135 topics and has about 2900 documents.

WebAce The dataset is from WebACE project [6]. It contains 2340 documents consisting news articles from Reuters new service via the Web in October 1997. These documents are divided into 20 classes.

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

6.2 Evaluation Measures

The above document datasets are standard labeled corpora widely used in the information retrieval literature. We view the labels of the datasets as the objective knowledge on the structure of the datasets. To measure the clustering performance, we use accuracy, entropy, purity and Adjusted Rand Index (ARI) as our performance measures. We expect these measures would provide us with enough insights.

Accuracy discovers the one-to-one relationship between clusters and classes and measures the extent to which each cluster contained data points from the corresponding class. It sums up the whole matching degree between all pair class-clusters. Accuracy can be represented as:

$$Accuracy = Max(\sum_{C_k, L_m} T(C_k, L_m))/N, \tag{21}$$

where C_k is the k-th cluster, and L_m is the m-th class. $T(C_k, L_m)$ is the number of entities which belong to class m and are assigned to cluster k. Accuracy computes the maximum sum of $T(C_k, L_m)$ for all pairs of clusters and classes, and these pairs have

no overlaps. Generally, the greater the accuracy values, the better clustering performance.

Purity measures the extent to which each cluster contains data points from primarily one class. In general, larger purity values lead to better clustering solutions. The purity of a clustering solution is obtained as a weighted sum of individual cluster purity values and is given by

$$Purity = \sum_{i=1}^{K} \frac{n_i}{n} P(S_i), P(S_i) = \frac{1}{n_i} max_j(n_i^j) \tag{22}$$

where S_i is a particular cluster of size n_i, n_i^j is the number of documents of the i-th input class that were assigned to the j-th cluster, K is the number of clusters and n is the total number of points [2].

Entropy measures how classes distributed on various clusters. Generally, the smaller the entropy value, the better the clustering quality is. The entropy of the entire clustering solution is computed as:

$$Entropy = -\frac{1}{n\log_2 m} \sum_{i=1}^{K} \sum_{j=1}^{m} n_i^i \log_2 \frac{n_i^j}{n_i}, \tag{23}$$

where m is the number of original labels, K is the number of clusters. Generally, the smaller the entropy value, the better the clustering quality is. More details on the purity and entropy measures can be found in [14].

The Rand Index is defined as the number of pairs of objects that are both located in the same cluster and the same class, or both in different clusters and different classes, divided by the total number of objects [12]. Adjusted Rand Index (ARI) which adjusts Rand Index is set between 0 and 1 [5]:

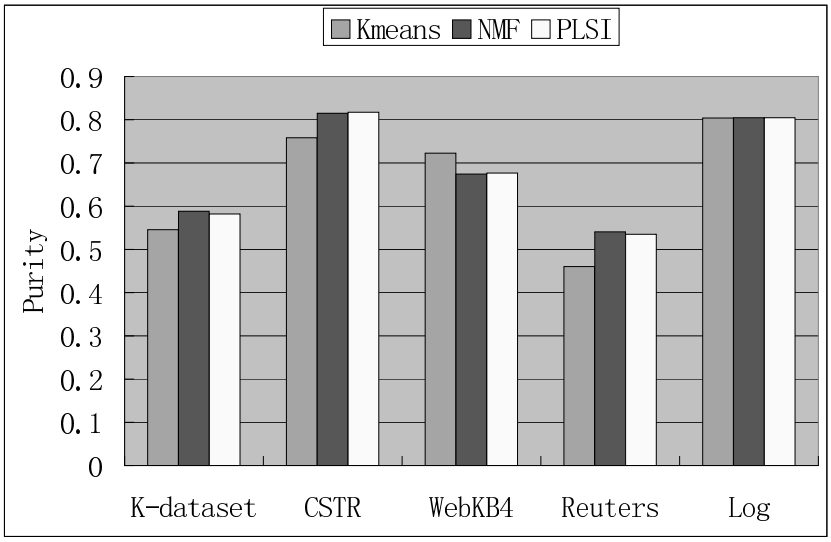
$$ARI = \frac{a - \frac{bc}{n(n-1)/2}}{(1/2)(b+c) - \frac{bc}{n(n-1)/2}}, \tag{24}$$

where $a = \sum_{i,j} \frac{V_{ij}(V_{ij}-1)}{2}$, $b = \sum_i \frac{V_i(V_i-1)}{2}$, $c = \sum_j \frac{V^j(V^j-1)}{2}$, V_{ij} is the number of objects that are in both of class i and cluster j, V_i is the number of objects in the class i, and V^j is the number of objects in cluster j. The higher the Adjusted Rand Index, the more resemblance between the clustering results and the labels.

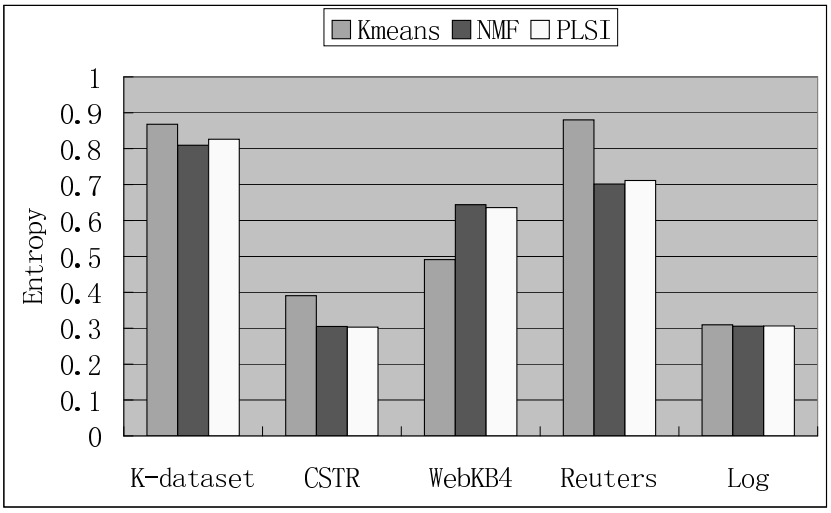
6.3 Performance Comparison

For each of the five datasets we first run K-means clustering. This serves as a comparison and also initialization. From the K-means solution, H_0 is constructed from the cluster assignments and C_0 is simple the cluster centroids (see footnote 1). The H_0 obtained this way is discrete (0 and 1) and is very sparse (mostly zeroes). This is generally poor for multiplicative updating algorithms. Thus we smooth H_0 by adding

[2] $P(S_i)$ is also called the individual cluster purity.

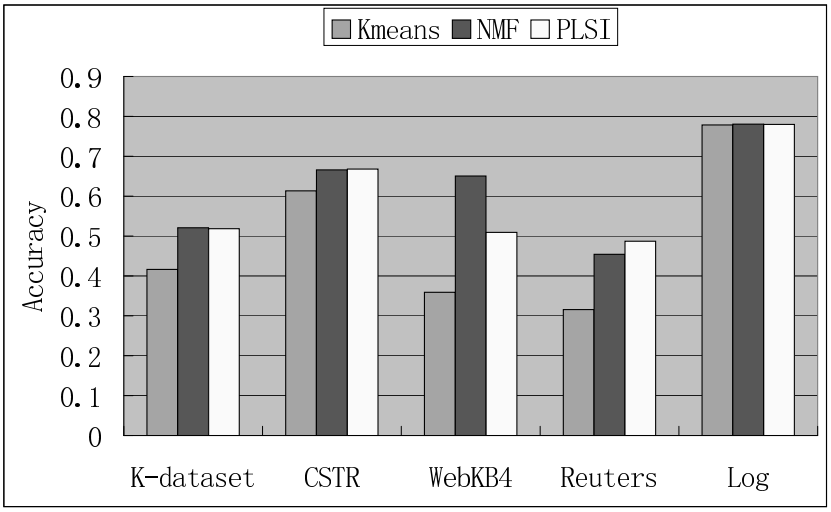


(a) Purity

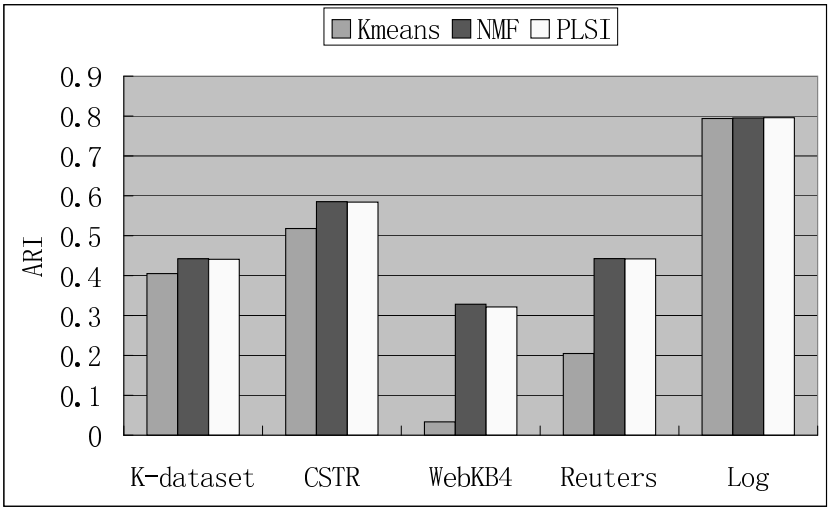


(b) Entropy

Figure 1: Performance Comparison of NMF and PLSI: I

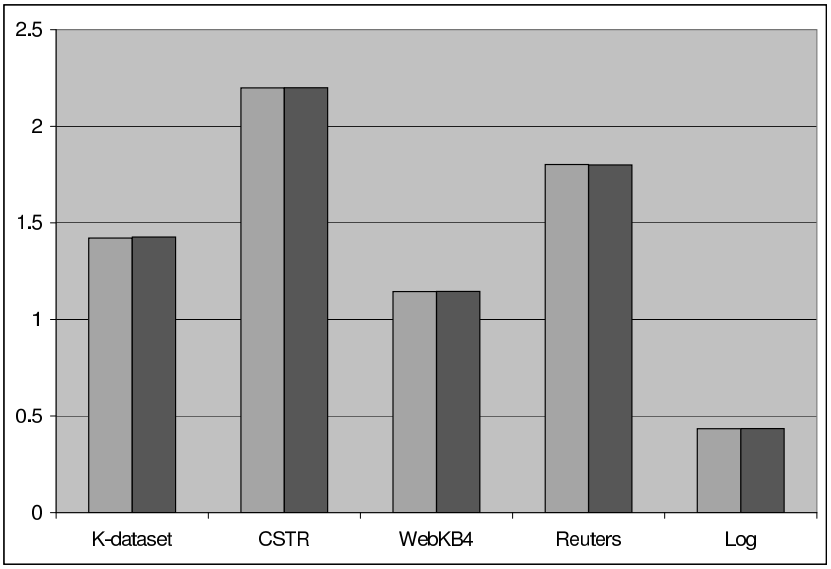


(a) Accuracy

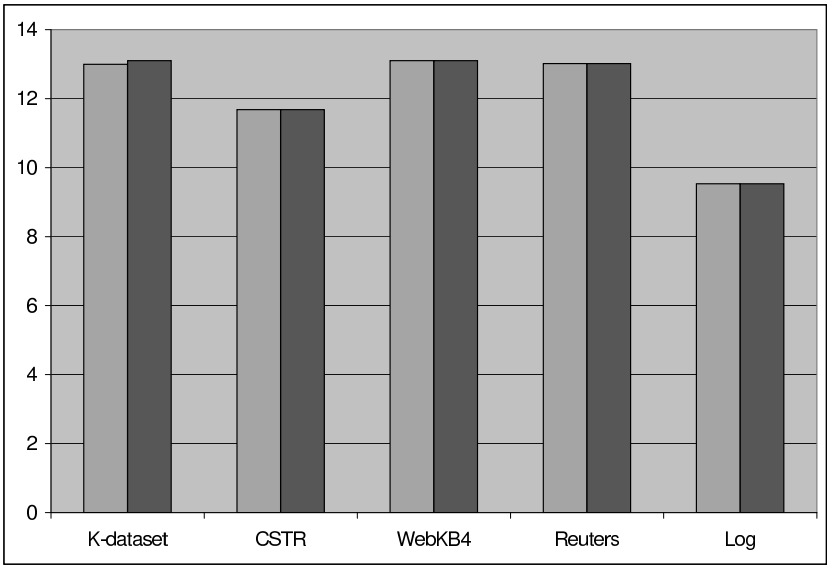


(b) Adjust Rand Index

Figure 2: Performance Comparison of NMF and PLSI: II



(a) J_{NMF}



(b) J_{PLSI}

Figure 3: Comparison of NMF and PLSI on J_{NMF} in Eq.(2) and $-J_{PLSI}$ in Eq.(3).

a small constant [3] to every element of H_0. We then do necessary normalization on C_0, H_0. Starting from this smoothed K-means solution, we run NMF or PLSI. From the NMF or PLSI solution, we harden the posterior H (see footnote 2) to obtain a discrete H (containing 0 and 1). From here, the performance measures are computed. We typically run 10 runs and obtain the average.

The clustering solutions of NMF and PLSI are compared based on accuracy, entropy, purity, and ARI as shown in Figure 1 and Figure 2. And the NMF objective function J_{NMF} in Eq.(2) and the negative PLSI objective function $-J_{PLSI}$ in Eq.(3) are compared in Figure 3. From these cluster assignment figures, we observe that NMF and PLSI lead to similar clustering results and objective function residues. For example, as shown in Figure 1(a), in terms of purity value, the differences between the clustering solutions obtained by NMF and PLSI are less than 0.02 in all the datasets. We can observe similar behavior for other performance measures as well [4].

6.4 Agreements Between NMF and PLSI

However, the closeness of NMF and PLSI on these performance measures merely indicates that the *level* of agreement between the NMF clustering solution and the known class label information is close to the *level* of agreement between PLSI and known class labels, and how *approximate* that solutions of NMF and PLSI are to the original matrix.

To understanding the difference between NMF and PLSI, we compare NMF and PLSI solutions *directly*: We measure the number of differences in clustering of data pairs using the clustering matrix R in Eq.(20). To normalize the difference so that datasets of different sizes can be compared with each other, we measure the relative difference:

$$\delta = ||R_{\text{NMF}} - R_{\text{PLSI}}||_F \Big/ \sqrt{||R_{\text{NMF}}||_F^2/2 + ||R_{\text{PLSI}}||_F^2/2}$$

The computed results, the average of 10 different runs, are listed in line A of Table 2. The results show that the differences between NMF and PLSI are quite substantial for WebKB (24%), and ranges between 1% to 8% in general cases.

	WebAce	CSTR	WebKB	Reuters	Log
A	0.083	0.072	0.239	0.070	0.010
B	0.029	0.025	0.056	0.051	0.010
C	0.022	0.013	0.052	0.040	0.012

Table 2: Disagreements between NMF and PLSI. All 3 type experiments begin with the same smoothed K-means. (A) Smoothed K-means to NMF. Smoothed K-means to PLSI. (B) Smoothed K-means to NMF to PLSI. (C) Smoothed K-means to PLSI to NMF.

[3] In our experiments, we choose the constant to be 0.2 as it generally leads to good performance.

[4] One thing we need to point out is that, in terms of accuracy, NMF and PLSI have a large difference of about 0.2 on *WebKB* dataset. This is because *WebKB* contain a lot of confusing webpages that can be assigned to one or more clusters and the accuracy measure takes into account the entire distribution of the documents in a particular cluster and not just the largest class as in the computation of the purity.

Function J_{NMF} defines a surface in the multi-dimensional space. Because this global objective function is not a convex function, there are in general a very large number of local minima in the high p-dimensional space. Our experimental results suggest that starting with same initial smoothed K-means solution, NMF and PLSI converge to different local minima. In many cases, NMF and PLSI converge to *nearby* local minima as they have similar clustering performance and objective functions; In other cases they converge to *not-so-nearby* local minima.

6.5 Word Occurrence Probability

In order to get a better understanding, we further compare the word occurrence probabilities of two corresponding clusters obtained from NMF and PLSI solutions for every dataset. In Figure 4, Figure 5, Figure 6, Figure 7, and Figure 8, the X axis represents the top ten words with the largest probabilities $P(w_i \mid z_k)$ picked from each cluster k obtained from PLSI, and these probabilities are illustrated as the declining solid lines along the Y axis. The dotted lines present the "probabilities" $C_{k'}(w_i)$ of the same words in the corresponding cluster k' from NMF. The solid lines and the dotted lines with the same symbols (e.g., $\circ$, $\diamond$, and $\times$) are from the corresponding clusters of PLSI and NMF solutions. The legend shown in Figure 4 gives a clearer explanation of curves, where C_i presents the probabilities of top 10 words in the cluster i from PLSI, and K_i shows the corresponding probabilities of the same 10 words in the corresponding cluster i from NMF. In each figure, we picked four pairs of clusters for each dataset. Observe that every pair of curves are almost in parallel to each other with very similar slopes. In Figure 5, Figure 4, Figure 6, Figure 7, and Figure 8, the solid lines are the occurrence probabilities of the top 10 words in a cluster obtained from PLSI solution. The dotted lines with the same symbols as those on the solid lines represent the corresponding occurrence probabilities in the corresponding cluster from NMF solutions. In summary, we observe that NMF and PLSI have similar word clustering results.

7 A Hybrid NMF-PLSI Algorithm

We have seen that NMF and PLSI optimize the same objective function, but their different detailed algorithms converge to different local minima. An interesting question arises. Starting from a local minimum of NMF, could we jump out the local minimum by running the PLSI algorithm? Strictly speaking, if an algorithm makes an infinitesimal step, it will not jump out of a local minimum (we ignore the situation that the minimum could be saddle points). But PLSI algorithm is a finite-step algorithm, so it is possible to jump out of a local minimum reached by NMF. Vice versa, NMF is also a finite-step algorithm.

Interestingly, experiments indicate that we can jump out of local minima this way. The results are shown in Table 2 Lines B & C. In Line B, we start from the K-means solution with smoothing and converge to a local minimum using NMF. Starting from the same local minimum, we run PLSI till convergence. The solution changed and the difference is given in Line B. This change indicates that we jump out of the local minimum. The changes in the solutions are smaller than Line A, as expected. In Line

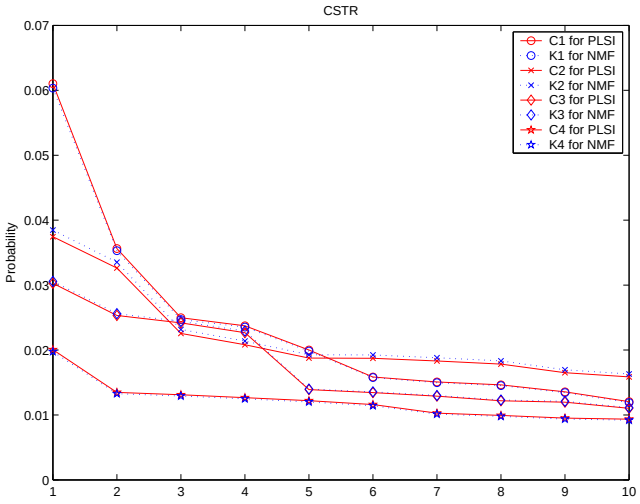


Figure 4: CSTR Dataset

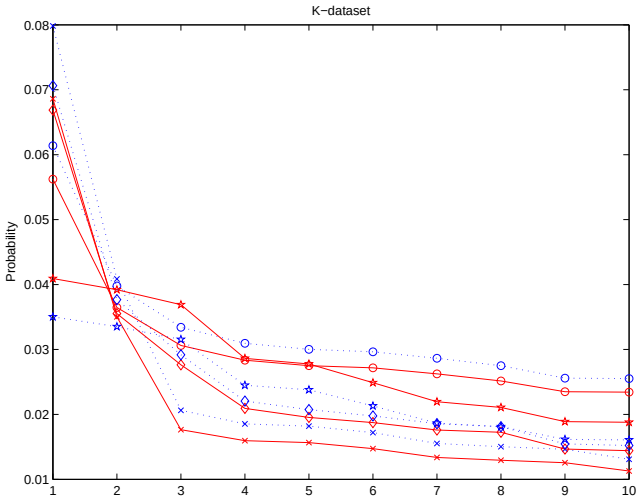


Figure 5: WebACE Dataset

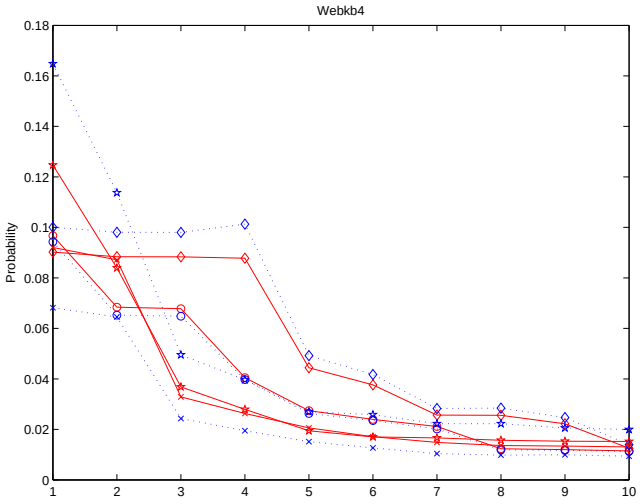


Figure 6: WebKB dataset.

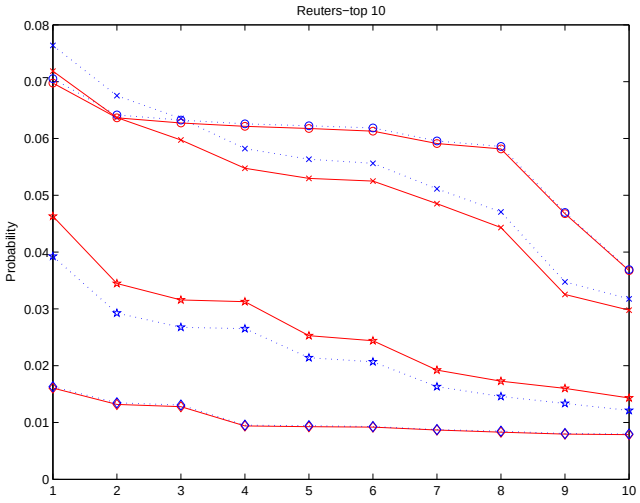


Figure 7: Reuter Dataset

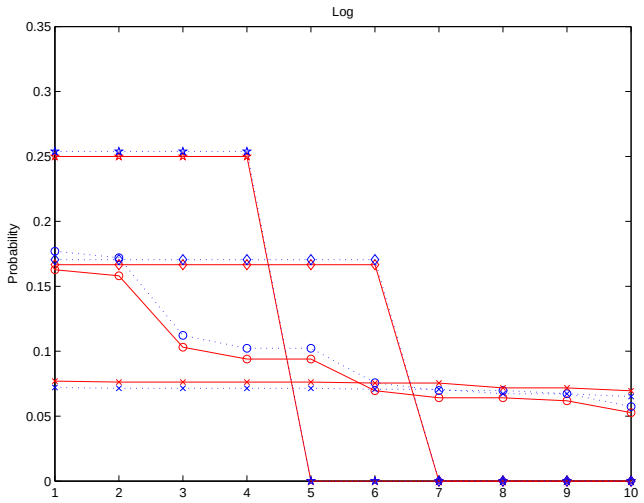


Figure 8: Log Dataset

C, we start from the K-means solution with smoothing and then run PLSI to converge to a local minimum; we then jump out of this local minimum by running NMF. The difference of the solutions is given in Line C. The changes in the solutions are smaller than line A, as expected. The changes are also smaller than line B, indicating the local minimum reached by PLSI is perhaps slightly deeper than the local minima reached by NMF.

Based on the ability of NMF for jumping out of local minima of PLSI and vice versa, we propose a hybrid algorithm that alternatively runs NMF and PLSI, with the goal of successively jumping out local minima and therefore converging to a better minimum. The hybrid algorithm consists of 2 steps (1) K-means and smooth. (2) Iterate until converge: (2a) Run NMF to converge. and (2b) Run PLSI to converge. We run the hybrid algorithm on all five datasets and the results are listed in Table 3. We observe that: (1) NMF and PLSI always improve upon K-means. (2) Hybrid always improves upon NMF and PLSI; the clustering accuracy improvements are especially obvious on the first three out of five datasets.

The hybrid method will converge. Each step of PLSI and NMF will lower the objective value and thus the process is monotonic. Since the objective has a lower bound, this guarantees the convergence. However, we are not sure about the rate of convergence at this stage of the research. Note that the hybrid process is carried out near a local minima of the objective function landscape, i.e., (A) either near the place where some of the gradient of the model parameters (conditional probabilities) is zero (B) or the positive conditional probabilities reach the boundary of feasibility region is zero. For this reason, the rate of convergence will likely to be linear, instead of quadratic. Experiments show typically the number of iterations is around 10 .

	Reuters	WebKB	CSTR	WebAce	Log
A	0.316	0.410	0.617	0.416	0.775
B	0.454	0.619	0.666	0.520	0.778
C	0.487	0.510	0.668	0.519	0.779
D	0.521	0.644	0.878	0.523	0.781

Table 3: Clustering Accuracy. (A) K-means. (B) NMF-only. (C) PLSI-only. (D) Hybrid.

8 Summary

In this paper, we study the relationships between NMF (with I-divergence objective) and PLSI in the clustering framework; in particular, we show that i) both NMF and PLSI have similar data representations; ii) both NMF and PLSI minimize the same objective function; and iii) NMF with L_1 normalization is identical to PLSI factorization. These three relationships establish the equivalence between NMF and PLSI in the clustering framework. Based on this analysis, we propose a hybrid algorithm which alternatively runs NMF and PLSI. Extensive experiments on 5 datasets show the significant improvement of the hybrid method over PLSI or NMF.

Acknowledgment. Chris Ding is partially supported by the US Dept of Energy, Office of Science and a University of Texas STARS Award. Tao Li is partially supported by a 2005 IBM Faculty Award, a 2005 IBM Shared University Research (SUR) Award, and the NSF grant IIS-0546280. Wei Peng is supported by a Florida International University Presidential Graduate Fellowship.

References

[1] D. Blei, A. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:9931022, 2003.

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

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

[4] Eric Gaussier and Cyril Goutte. Relation between plsa and nmf and implications. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 601–602, New York, NY, USA, 2005. ACM Press.

[5] Milligan GW and Cooper MC. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivar Behav Res*, 21:846–850, 1986.

[6] Eui-Hong Han, Daniel Boley, Maria Gini, Robert Gross, Kyle Hastings, George Karypis, Vipin Kumar, Bamshad Mobasher, and Jerome Moore. WebACE: A web agent for document categorization and exploration. In *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*. ACM Press, 1998.

[7] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 289–296, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

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

[9] D.D. Lee and H. S. Seung. Algorithms for non-negatvie matrix factorization. In T. G. Dietterich and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. The MIT Press, 2001.

[10] Tao Li. A general model for clustering binary data. In *KDD*, pages 188–197, 2005.

[11] V. P. Pauca, F. Shahnaz, M.W. Berry, and R.J. Plemmons. Text mining using non-negative matrix factorization. In *Proc. SIAM Int'l conf on Data Mining (SDM 2004)*, pages 452–456, 2004.

[12] Rand WM. Objective criteria for the evaluation of clustering methods. *J Am Stat Assoc*, 66:846–850, 1971.

[13] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proc. ACM Conf. Research development in IR(SIRGIR)*, pages 267–273, 2003.

[14] Ying Zhao and George Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.