Fast Bregman Divergence NMF using Taylor Expansion and Coordinate Descent

Liangda Li, Guy Lebanon, Haesun Park
School of Computational Science and Engineering
Georgia Institute of Technology
Atlanta, GA 30032
{ldli, lebanon, hpark}@cc.gatech.edu

ABSTRACT

Non-negative matrix factorization (NMF) provides a lower rank approximation of a matrix. Due to nonnegativity imposed on the factors, it gives a latent structure that is often more physically meaningful than other lower rank approximations such as singular value decomposition (SVD). Most of the algorithms proposed in literature for NMF have been based on minimizing the Frobenius norm. This is partly due to the fact that the minimization problem based on the Frobenius norm provides much more flexibility in algebraic manipulation than other divergences. In this paper we propose a fast NMF algorithm that is applicable to general Bregman divergences. Through Taylor series expansion of the Bregman divergences, we reveal a relationship between Bregman divergences and Euclidean distance. This key relationship provides a new direction for NMF algorithms with general Bregman divergences when combined with the scalar block coordinate descent method. The proposed algorithm generalizes several recently proposed methods for computation of NMF with Bregman divergences and is computationally faster than existing alternatives. We demonstrate the effectiveness of our approach with experiments conducted on artificial as well as real world data.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing-abstracts methods

General Terms

Algorithm, Experiment, Performance

Keywords

Non-negative Matrix Factorization, Bregman Divergences, Euclidean distance, Taylor Series Expansion

1. INTRODUCTION

Non-negative matrix factorization (NMF) is a dimensionality reduction method that has attracted great attention over a decade. It approximates a matrix $\mathbf{A}$ by a product of two lower rank matrices $\mathbf{W}$ and $\mathbf{H}$ with non-negative entries minimizing the divergence between $\mathbf{A}$ and $\mathbf{WH}^T$. Using Bregman divergences, our problem is to find

$$\arg\min_{\mathbf{W}\geq 0,\mathbf{H}\geq 0} D_\phi(\mathbf{A}\|\mathbf{WH}^T),$$

where $D_\phi(\mathbf{A}\|\mathbf{WH}^T)$ denotes a Bregman divergence between $\mathbf{A}$ and $\mathbf{WH}^T$. The decomposition discovers a latent structure in the data and is useful in signal processing, collaborative filtering, clustering, and other data-mining tasks. Due to the non-negativity constraint on $\mathbf{W}$ and $\mathbf{H}$, NMF discovers a latent structure that is often more interpretable than SVD based factorizations.

Various Bregman divergences, such as Frobenius norm and KL divergence, have been used in a wide range of applications, including text clustering, signal processing, image processing, and music analysis. A general NMF algorithm for various Bregman divergences not only offers a general solution for different applications, but also enables the knowledge share across different domains. Recent years have seen a surge of interest in NMF [25, 7, 22, 20]. The earliest NMF algorithms aimed at directly optimizing the divergence, resulting in higher scale computational costs. Moreover, most algorithms have been developed for Frobenius norm(which becomes the Euclidean distance for scalars) minimization, a popular special case of the Bregman divergence.

Lee and Seung's simple multiplicative update rule [25] has been one of the most utilized method for NMF over a decade, for both Frobenius norm and KL(Kullback-Leibler) divergence. Dhillon et al. [8] extended it to solve NMF with general Bregman divergences. By using an auxiliary function, Fevotte et al. [11] discussed the updating rule under the IS divergence, and applied it to music analysis. An alternative updating rule is also provided based on the statistical interpretation of the properties of the IS divergence. Nevertheless, the above algorithms generally suffer from high computational cost and slow convergence [23].

Cichocki et al. [5] introduced an alternative algorithm with improved local updating rules, achieving high efficiency in solving NMF problems. The resulting algorithm applies to the Frobenius norm. Similar algorithms have been proposed for a few other divergences, but these algorithms show a relatively slow convergence.

Other NMF algorithms are proposed by Lin et al. [27] using projected gradient with Armijo rule to build the updating rule. Kim and Park [19, 20] proposed an NMF algorithm based on alternating non-negative least squares (ANLS) and an active set based algorithm. This algorithm was further

KDD'12, August 12–16, 2012, Beijing, China.

Table 1: Notations used in the paper

$\mathbb{R}$	real number
$\mathbb{R}_+$	nonegative real number
$\mathbb{R}^{M\times N}$	vector space of matrice of size $M \times N$
$\mathbb{R}^K$	vector space of vectors of size K
$\odot$	element-wise product or Hadamard product
$\oslash$	element-wise division
$\mathbf{a}_j$	j-th column vector of matrix $\mathbf{A}$
a_{ij}	element in the i-th row and j-th column of $\mathbf{A}$
$\|\cdot\|_2$	Euclidean distance(Frobenius norm)
D_ϕ	Bregman divergences
$\nabla^t\phi(x)$	a element-wise t-order derivative operator of ϕ at x
$\mathrm{sgn}(x)$	sign function or signum function

improved by block principal pivoting in ANLS. An extensive comparison of these algorithms appears in [22, 23]. For a survey of NMF algorithms, see [21].

In this paper, we propose a fast NMF algorithm for a general class of Bregman divergences. Using Taylor series we relate Bregman divergences to the Euclidean distance and show that the Bregman divergence optimization problem can be expressed in terms of the Euclidean distance. The discovered relationship between Bregman divergence and the Euclidean distance leads to local updating rules, and provides one efficient algorithmic framework which is applicable to NMF formulated in Bregman divergences, and solutions for a wide range of applications.

We investigate the performance of the new algorithm on both artificial and real world data sets, including Xspectra [6], AT&T Laboratory Face Image data set [1], Movielens, and Netflix. We conduct a series of experiments comparing our proposed methods to previously proposed algorithms. Our experiments show that for a wide range of Bregman divergences our new algorithm is substantially faster.

Our main contributions in this paper include

1. a new relationship connecting Bregman divergences and the Euclidean distance via Taylor series expansion, and
2. an NMF algorithm applicable for all Bregman divergences that is substantially faster.

Note that by relating all Bregman divergences to the Euclidean distance, we propose one united highly efficient algorithm applicable for all NMF formulated in any Bregman divergences. In contrast, most other existing algorithms are designed for one or only a subset of Bregman divergences.

Table 1 summarizes the notations used in this paper.

2. NMF WITH BREGMAN DIVERGENCES

In NMF, given a matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}_+^{M\times N}$, and an integer $K \leq \min(M, N)$, we are to find $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K] \in \mathbb{R}_+^{M\times K}$ whose columns represent basis vectors in a K-dimensional space, and $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K] \in \mathbb{R}_+^{N\times K}$ whose columns represent mixing proportions, such that

$$\mathbf{A} \approx \mathbf{A}' = \mathbf{W}\mathbf{H}^T. \tag{1}$$

The quality of this approximation can be measured using Bregman divergences,

$$\begin{aligned} D_\phi(\mathbf{A}\|\mathbf{A}') &= \sum_{i,j} D_\phi(a_{ij}\|a'_{ij}) \\ &= \sum_{i,j} (\phi(a_{ij}) - \phi(a'_{ij}) - \nabla\phi(a'_{ij})(a_{ij} - a'_{ij})) \end{aligned}$$

where ϕ is a univariate convex smooth function.

Bregman divergences are not symmetric in general, and the solution for $\min D_\phi(\mathbf{W}\mathbf{H}^T\|\mathbf{A})$ will be different from that of $\min D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)$. In this paper, we focus on $D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)$ which is more widely used in applications. For instance, Probabilistic Latent Semantic Analysis (PLSI) [16], a statistical technique for data analysis, optimizes the same objective function as NMF with Kullback-Leibler divergence [9]. Other examples are signal analysis using $D_{KL}(\mathbf{A}||\mathbf{W}\mathbf{H}^T)$ and music analysis using Itakura-Saito divergence $D_{IS}(\mathbf{A}||\mathbf{W}\mathbf{H}^T)$. From a theoretical perspective, maximum likelihood estimation and information theory indicate that $D_{KL}(\mathbf{A}||\mathbf{W}\mathbf{H}^T)$ is better motivated than $D_{KL}(\mathbf{W}\mathbf{H}^T||\mathbf{A})$, at least when the two arguments are probability vectors.

The choice of $\phi(x) = x^2/2$ reduces D_ϕ to the squared Frobenius norm of $\mathbf{E} = \mathbf{A} - \mathbf{W}\mathbf{H}^T$, which is the sum of squared entries of the residual matrix $\mathbf{E}$ (Notice that the Frobenius norm of a matrix can be viewed as the Euclidean distance of the corresponding vectorized matrix). Other choices for ϕ result in the non-negative Kullback-Leibler (KL) divergence or Itakura-Saito (IS) divergence. The specific choice of ϕ depends on the application. For example, the Frobenius norm has been used successfully in text clustering [3]. KL divergence is well suited for many problems in signal processing [5] while IS divergence has been shown to perform well in music recommendation [11]. For other choices of ϕ and areas where these divergences are ultilized, see [7].

In [2], Bregman divergences have been used to derive an exact characterization of the difference between the two sides of Jensen's inequality. Banerjee et al. [3] discussed a clustering algorithm for general Bregman divergences. They also showed that there exists a bijection between regular exponential families and large classes of Bregman divergences. Singh and Gordon [31] showed that methods such as NMF, Weighted SVD, pLSI et al. can be viewed in a general framework of matrix factorization with Bregman divergences. Pietra et al. [30] derived and proved convergence of iterative algorithms to minimize Bregman divergence subject to linear constraints based on auxiliary functions. Wang and Schuurmans [32] proposed a novel algorithm that extracts hidden latent structure by minimizing Bregman divergences. Lebanon [24] used Taylor series approximation to show a relationship between KL divergences and the Fisher geometry which enjoys certain axiomatic properties.

Some examples of Bregman divergences and the corresponding ϕ functions are listed in Table 2.

3. FAST ALGORITHM FOR NMF

Many existing NMF algorithms can be explained using the block coordinate descent framework [21]. Different partitions of variables $\mathbf{W}$ and $\mathbf{H}$ lead to different NMF algorithms. One natural way of partition [8, 19, 22] is the two blocks representing $\mathbf{W}$ and $\mathbf{H}$, with which the subproblems result in a nonnegativity constrained least square (NLS) problem. Another way of partition [5, 14] is $K(M + N)$ blocks where each represents a single element in $\mathbf{W}$ or $\mathbf{H}$.

Coordinate descent is also employed to solve many other problems. Wu et al. [33] came up with coordinate descent algorithm for l_1 regularized regression, Lasso. A greedy coordinate descent method was proposed in [26] to solve the Basis Pursuit problem, and can be applied to tasks such as compressed sensing and image denoising. Yun and Toh [34] proposed a block coordinate gradient descent method for

general l_1-regularized convex minimization problems. Recently, a fast coordinate descent algorithm was developed in [13] to estimate generalized linear models with convex penalties. Coordinate descent was also used for solving non-convex penalty functions, such as smoothly clipped absolute deviation (SCAD) penalty and the minimax concave penalty (MCP) in [4].

We denote the residual term in the NMF approximation (1) as $\mathbf{A}-\mathbf{W}\mathbf{H}^T = \mathbf{A}^{(k)} - \mathbf{w}_k\mathbf{h}_k^T$, where the k-residual $\mathbf{A}^{(k)}$ is define as

$$\mathbf{A}^{(k)} = \mathbf{A} - \sum_{p\neq k} \mathbf{w}_p\mathbf{h}_p^T = \mathbf{A} - \mathbf{W}\mathbf{H}^T + \mathbf{w}_k\mathbf{h}_k^T,$$

for $k = 1, \ldots, K$. In the case of $\phi(x) = \frac{1}{2}x^2$, D_ϕ leads to the squared Frobenius norm

$$\begin{aligned} D_\phi(\mathbf{A}\|\mathbf{A}') &= D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T) = \frac{1}{2}\|\mathbf{A}-\mathbf{W}\mathbf{H}^T\|_F^2 \\ &= \frac{1}{2}\|\mathbf{A}^{(k)} - \mathbf{w}_k\mathbf{h}_k^T\|_F^2 = D_\phi(\mathbf{A}^{(k)}\|\mathbf{w}_k\mathbf{h}_k^T). \end{aligned} \quad (2)$$

Let us define

$$E_t(\mathbf{A}\|\mathbf{A}') = \sum_{ij} |a_{ij} - a'_{ij}|^t, \quad t \in \{1, 2, \ldots\}$$

which is the t-th power of t-norm distance between vectorized matrices $\mathbf{A}$ and $\mathbf{A}'$. Then we have

$$\begin{aligned} E_t(\mathbf{A}\|\mathbf{A}') &= E_t(\mathbf{A}^{(k)}\|\mathbf{w}_k\mathbf{h}_k^T) \quad \text{or} \\ E_t(a_{ij}\|a'_{ij}) &= E_t(a_{ij}^{(k)}\|w_{ik}h_{jk}). \end{aligned} \quad (3)$$

Cichocki et al. [5] proposed an algorithm called Hierarchical Alternating Least Squares (`HALS`) for NMF with Frobenius norm. They designed local updating rules based on the relationship in Eqn (2), leading to a fast algorithm. Each updating step solves a sub-optimization problem with a closed form solution, and the algorithm converges much faster than the multiplicative updating rule in [25, 8]. Although similar algorithms for Alpha divergence and Beta divergence have been proposed, they aimed at minimizing $D_\phi(\mathbf{A}^{(k)}\|\mathbf{w}_k\mathbf{h}_k^T)$ instead of $D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)$, which are two different functions for most Bregman divergences other than the Frobenius norm.

In this paper, using the relationship in Eqn (3), the optimization goal changes from the approximation between the given matrix and the multiplication of two low-rank matrices to the approximation between the k-residual matrix and the multiplication of two vectors. Since elements in $\mathbf{w}_k$(or $\mathbf{h}_k$) can be computed independently, we actually focus on the approximation between $a_{ij}^{(k)}$(single element in k-residual matrix) and the multiplication of w_{ik} and h_{jk}. Based on this observation, a novel scalar coordinate descent algorithm with $k(m+n)$ scalar blocks can be designed. In the rest of this section, we will

- Derive a new relationship between general Bregman divergences and the Euclidean distance;
- Use the above relationship to replace the minimization goal from the Bregman divergences $D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)$ with an expression of $E_t(a_{ij}^{(k)}\|w_{ik}h_{jk})$;
- Design coordinate descent algorithm to optimize $E_t(a_{ij}^{(k)}\|w_{ik}h_{jk})$.

3.1 A Taylor Series Expansion of Bregman Divergences

The following proposition shows a new relationship between Bregman divergences and the Euclidean distance, which plays a key role in our fast algorithm development.

PROPOSITION 3.1.

$$D_\phi(\mathbf{A}\|\mathbf{A}') = \sum_{i,j}\sum_{t=2}^{\infty} \frac{\nabla^t\phi(a'_{ij})}{t!}(-\text{sgn}(a'_{ij}-a_{ij}))^t E_t(a_{ij}\|a'_{ij}).$$

PROOF. The Taylor expansion of $D_\phi(a_{ij}\|a'_{ij})$ leads to

$$\begin{aligned} D_\phi(a_{ij}\|a'_{ij}) &= \phi(a_{ij}) - \phi(a'_{ij}) - \nabla\phi(a'_{ij})(a_{ij}-a'_{ij}) \\ &= \nabla\phi(a'_{ij})(a_{ij}-a'_{ij}) + \sum_{t=2}^{\infty}\frac{\nabla^t\phi(a'_{ij})}{t!}(a_{ij}-a'_{ij})^t \\ &\quad - \nabla\phi(a'_{ij})(a_{ij}-a'_{ij}) \\ &= \sum_{t=2}^{\infty}\frac{\nabla^t\phi(a'_{ij})}{t!}(a_{ij}-a'_{ij})^t \\ &= \sum_{t=2}^{\infty}\frac{\nabla^t\phi(a'_{ij})}{t!}(-\text{sgn}(a'_{ij}-a_{ij}))^t E_t(a_{ij}\|a'_{ij}) \end{aligned} \quad (4)$$

where $\nabla^t\phi(a'_{ij})$ is the t-order derivative of ϕ at a'_{ij}. □

The above relationship is then employed to replace our objective function $D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)$ with an expression that involves $E_t(a_{ij}^{(k)}\|w_{ik}h_{jk})$. In the following subsection, we show how this relationship allows us to recast the problem in a form that is easier to solve and leads to a novel and efficient algorithm. Notice that this relationship is an equality rather than an approximation.

Taylor expansion has been utilized in numerical problems including a quadratic approximation of the objective or loss function. For instance, to solve a regularized log-determinant program, Hsieh et al. [18] proposed a novel algorithm which is based on Newton's method and employs a quadratic approximation. For the l_1-regularized linear least squares problem, a gradient projection method was proposed in [12] to solve the bound constrained quadratic programming reformulation. Yun. et al [34] went a step further by using quadratic approximation to solve the general l_1-regularized convex minimization problem. In each iteration, the objective is replaced by a strictly convex quadratic approximation, then block coordinate descent is used to obtain a feasible descent direction. Taylor explansion was also employed to approximate non-convex penalties, such as SCAD [10] and MCP [29].

3.2 A New Algorithm for NMF with Bregman Divergences

Based on Eqns (3) and (4) , the Bregman divergences $D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)$ can be expressed in terms $E_t(a_{ij}^{(k)}\|w_{ik}h_{jk})$ as

$$D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T) = \sum_{i,j}\sum_{t=2}^{\infty} \frac{\nabla^t\phi(a'_{ij})}{t!}(-\text{sgn}(a'_{ij}-a_{ij}))^t E_t(a_{ij}^{(k)}\|w_{ik}h_{jk}).$$

Thus, instead of calculating the partial derivatives of $D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)$ with respect to $\mathbf{W}$ and $\mathbf{H}$, we turn to the partial derivative of $E_t(a_{ij}^{(k)}\|w_{ik}h_{jk})$ with respect to smaller blocks, w_{ik} and h_{jk}. Using this and the scalar block coordinate descent framework in constrained optimization where each block consists

of a single unknown element in $\mathbf{W}$ or $\mathbf{H}$ (assuming other elements are fixed), a novel fast algorithm can be derived.

From

$$\frac{\partial}{\partial h_{jk}}\Big(\frac{\nabla^t\phi(a'_{ij})}{t!}(-\mathrm{sgn}(w_{ik}h_{jk}-a_{ij}^{(k)}))^t E_t(a_{ij}^{(k)}\|w_{ik}h_{jk})\Big)$$
$$= -w_{ik}\frac{\nabla^t\phi(a'_{ij})}{(t-1)!}(a_{ij}^{(k)}-w_{ik}h_{jk})^{t-1}+w_{ik}\frac{\nabla^{t+1}\phi(a'_{ij})}{t!}(a_{ij}^{(k)}-w_{ik}h_{jk})^t$$

and Eqn (4), we obtain

$$\begin{aligned}\frac{\partial D_\phi(a_{ij}\|a'_{ij})}{\partial h_{jk}} &= w_{ik}\nabla^2\phi(a'_{ij})(w_{ik}h_{jk}-a_{ij}^{(k)})\\ &\quad+\sum_{t=2}^{\infty}\Big(-w_{ik}\frac{\nabla^{t+1}\phi(a'_{ij})}{t!}(a_{ij}^{(k)}-w_{ik}h_{jk})^t\\ &\quad+w_{ik}\frac{\nabla^{t+1}\phi(a'_{ij})}{t!}(a_{ij}^{(k)}-w_{ik}h_{jk})^t\Big)\\ &= w_{ik}\nabla^2\phi(a'_{ij})(w_{ik}h_{jk}-a_{ij}^{(k)}).\end{aligned}$$

Summing over the matrix rows and columns, we have[1]

$$\begin{aligned}\frac{\partial D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)}{\partial h_{jk}} &= \sum_{i=1}^{M} w_{ik}\nabla^2\phi(a'_{ij})(w_{ik}h_{jk}-a_{ij}^{(k)})\\ &= [\mathbf{W}^T(\nabla^2\phi(\mathbf{W}\mathbf{H}^T)\odot(\mathbf{w}_k\mathbf{h}_k^T-\mathbf{A}^{(k)}))]_{kj}. \qquad (5)\end{aligned}$$

The solution for the scalar block h_{jk} can be obtained by solving:

$$0=\frac{\partial D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)}{\partial h_{jk}}=\sum_{i=1}^{M} w_{ik}\nabla^2\phi(a'_{ij})w_{ik}h_{jk}-\sum_{i=1}^{M} w_{ik}\nabla^2\phi(a'_{ij})a_{ij}^{(k)}$$

which leads to the element-wise updating rule:

$$h_{jk}=\frac{\sum_{i=1}^{M}\nabla^2\phi(a'_{ij})a_{ij}^{(k)}w_{ik}}{\sum_{i=1}^{M}\nabla^2\phi(a'_{ij})w_{ik}w_{ik}}. \qquad (6)$$

Similarly, we can derive an updating rule for w_{ik}.

The summary of the algorithm, which we refer to as `sBCD` (Scalar Block Coordinate Descent) is shown in Algorithm 1[2]. Note that the algorithm follows the block coordinate descent framework where each element in $\mathbf{W}$ and $\mathbf{H}$ is considered as a scalar block that we update in each step.

The algorithm above is expressed in a general form for all Bregman divergences. Replacing $\phi(x)$ with the corresponding expression provides the specific algorithm for each specific Bregman divergence. Interestingly, for squared Frobenius norm, the updating rule is precisely the same as `HALS` algorithm proposed in [5]. Some specific updating rules are listed in Table 2.

The following rearrangements of expressions show an interesting relationship between `sBCD` and two other NMF algorithms, `Multiplcative Updating` and `Gradient Descent` methods. According to Eqns (5) and (6), we have

$$\begin{aligned}h_{jk} &= [\frac{\sum_{i=1}^{M}\nabla^2\phi(a'_{ij})(a_{ij}-a'_{ij}+w_{ik}h_{jk})w_{ik}}{\sum_{i=1}^{M}\nabla^2\phi(a'_{ij})w_{ik}w_{ik}}]_+\\ &= [h_{jk}+\frac{[\mathbf{W}^T(\nabla^2\phi(\mathbf{W}\mathbf{H}^T)\odot(\mathbf{A}-\mathbf{W}\mathbf{H}^T))]_{kj}}{[(\mathbf{W}\odot\mathbf{W})^T\nabla^2\phi(\mathbf{W}\mathbf{H}^T)]_{kj}}]_+\\ &= [h_{jk}+\frac{1}{[(\mathbf{W}\odot\mathbf{W})^T\nabla^2\phi(\mathbf{W}\mathbf{H}^T)]_{kj}}(-\frac{\partial D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)}{\partial h_{kj}})]_+\end{aligned}$$

[1] definition of $\odot$ can be found in Table 1.

[2] $[x]_+=\max\{x,0\}$.

Algorithm 1 `sBCD` Algorithm

1: Given $\mathbf{A}\in\mathbb{R}^{M\times N}$, a reduced dimension K, and function ϕ for a Bregman divergence, initialize values for $\mathbf{W}$ and $\mathbf{H}$.
2: $\mathbf{A}'=\mathbf{W}\mathbf{H}^T$
3: $\mathbf{E}=\mathbf{A}-\mathbf{A}'$
4: **repeat**
5: $\quad\mathbf{B}=\nabla^2\phi(\mathbf{A}')$
6: $\quad$**for** $k=1,2,\ldots,K$ **do**
7: $\quad\quad\mathbf{A}^{(k)}=\mathbf{E}+\mathbf{w}_k\mathbf{h}_k^T$
8: $\quad\quad$**for** $j=1,2,\ldots,N$ **do**
9: $\quad\quad\quad h_{jk}=[\frac{\sum_{i=1}^{M}b_{ij}a_{ij}^{(k)}w_{ik}}{\sum_{i=1}^{M}b_{ij}w_{ik}w_{ik}}]_+$
10: $\quad\quad$**end for**
11: $\quad\quad$**for** $i=1,2,\ldots,M$ **do**
12: $\quad\quad\quad w_{ik}=[\frac{\sum_{j=1}^{N}b_{ij}a_{ij}^{(k)}h_{jk}}{\sum_{j=1}^{N}b_{ij}h_{jk}h_{jk}}]_+$
13: $\quad\quad$**end for**
14: $\quad\quad\mathbf{E}=\mathbf{A}^{(k)}-\mathbf{w}_k\mathbf{h}_k^T$
15: $\quad$**end for**
16: $\quad\mathbf{A}'=\mathbf{W}\mathbf{H}^T$
17: **until** stopping criterion is reached

where we can view $-\frac{\partial D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)}{\partial h_{jk}}$ as the gradient direction, $\frac{1}{[(\mathbf{W}\odot\mathbf{W})^T\nabla^2\phi(\mathbf{W}\mathbf{H}^T)]_{kj}}$ as the step size. Notice that a hard constraint is enforced to ensure h_{jk} to be nonnegative.

On the other hand, multiplicative updating rule proposed in [25] can be written as:

$$\begin{aligned}h_{jk} &= h_{jk}\frac{[\mathbf{W}^T(\nabla^2(\mathbf{W}\mathbf{H}^T)\odot\mathbf{A})]_{kj}}{[\mathbf{W}^T(\nabla^2(\mathbf{W}\mathbf{H}^T)\odot\mathbf{W}\mathbf{H}^T)]_{kj}}\\ &= h_{jk}+\frac{h_{jk}}{[\mathbf{W}^T(\nabla^2\phi(\mathbf{W}\mathbf{H}^T)\odot\mathbf{W}\mathbf{H}^T)]_{kj}}(-\frac{\partial D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)}{\partial h_{jk}}).\end{aligned}$$

A gradient descent algorithm for NMF is also proposed in [25], which uses a fixed step size. With the above rearrangements of expressions, we can see that the difference between `sBCD`, `Multiplicative Updating` and `Gradient Descent` is the step sizes only. The advantage of `sBCD` and `Multiplicative Updating` over `Gradient Descent` is that they choose step sizes according to the result of previous iteration. Further comparsion of step sizes of `sBCD` and `Multiplicative Updating` shows that

$$\frac{1}{[(\mathbf{W}\odot\mathbf{W})^T\nabla^2\phi(\mathbf{W}\mathbf{H}^T)]_{kj}}\geq\frac{h_{jk}}{[\mathbf{W}^T(\nabla^2\phi(\mathbf{W}\mathbf{H}^T)\odot\mathbf{W}\mathbf{H}^T)]_{kj}}.$$

The above equation illustrates that `Multiplicative Updating` uses a conservative step size in order to keep the update result nonnegative, while a longer step is used by `sBCD` to make each updating more efficient.

3.3 Fast NMF algorithm with Sparsity Constraints

An important variation of NMF is the NMF subject to sparsity constraints [19] on one or both factors. For imposing sparsity on $\mathbf{H}$, the objective function is replaced with the following penalized version:[3]

$$L(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)=D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)+\alpha\sum_{k=1}^{K}\|\mathbf{h}_k\|_1,$$

[3] Notice in implementation, regularization term $\|\mathbf{W}\|_F$ is added to prevent it from growing too large.

Table 2: Updating rules for specific Bregman Divergences

Description	Function $\phi(x)$	$\nabla^2\phi(x)$	$D_\phi(a\|a')$	updating rule
Frobenius norm	$\frac{x^2}{2}$	1	$(a-a')^2/2$	$h_{jk}=\frac{\sum_{i=1}^M a_{ij}^{(k)}w_{ik}}{\sum_{i=1}^M w_{ik}w_{ik}}$
KL-divergence	$x\log x$	$1/x$	$a\log\frac{a}{a'}-a+a'$	$h_{jk}=\frac{\sum_{i=1}^M a_{ij}^{(k)}w_{ik}/a'_{ij}}{\sum_{i=1}^M w_{ik}w_{ik}/a'_{ij}}$
Itakura-Saito divergence	$-\log x$	$1/x^2$	$\frac{a}{a'}-\log\frac{a}{a'}$	$h_{jk}=\frac{\sum_{i=1}^M a_{ij}^{(k)}w_{ik}/a'^2_{ij}}{\sum_{i=1}^M w_{ik}w_{ik}/a'^2_{ij}}$
Beta divergence	$\frac{1}{\beta(\beta+1)}(x^{\beta+1}-(\beta+1)x+\beta)$	$x^{\beta-1}$	$\frac{1}{\beta(\beta+1)}(a^{\beta+1}-a'^{\beta+1}-(\beta+1)a'^{\beta}(a-a'))$	$h_{jk}=\frac{\sum_{i=1}^M a'^{\beta-1}_{ij}a_{ij}^{(k)}w_{ik}}{\sum_{i=1}^M a'^{\beta-1}_{ij}w_{ik}w_{ik}}$

where α is regularization paramter. Although $\|\cdot\|_1$ is not differentiable in general, in NMF it is differentiable in the specific domain due to the condition that $\mathbf{H}$ is non-negative.

The corresponding `sBCD` updating rule is

$$h_{jk}=\frac{\alpha+\sum_{i=1}^M\nabla^2\phi(a'_{ij})a_{ij}^{(k)}w_{ik}}{\sum_{i=1}^M\nabla^2\phi(a'_{ij})w_{ik}w_{ik}},$$

$$w_{ik}=\frac{\sum_{j=1}^N\nabla^2\phi(a'_{ij})a_{ij}^{(k)}h_{jk}}{\sum_{i=1}^N\nabla^2\phi(a'_{ij})h_{jk}h_{jk}}.$$

Sparsity on $\mathbf{W}$ can be imposed in an analogous way.

4. EXPERIMENTAL RESULTS

4.1 Data Set and Performance Evaluation

The experiments are conducted on artificial and real world data sets. The randomly generated data sets have problem sizes $(M,N,K)=(2000,1000,30)$, $(2000,1000,60)$, and $(3000,2000,30)$. The initial matrices for $\mathbf{W}$ and $\mathbf{H}$ were generated with uniform random values in [0.5, 1.5]. To remove sampling noise we average results using 5 different initial values. Our first real world data set follows the Xspectra setup in [6]. A matrix of size 1000×10 is formed by using ten noisy mixtures of five smooth sources. Mixed signals are corrupted by additive Guassian noise. For this data set, the ground-truth factors are provided, enabling the testing of algorithms' accuracy in recovering factors. A larger real world data set is the AT&T face image data set [1]. This data set contains 400 facial images (10 images of each of 40 different people) with a single facial image containing 92×112 pixels in 8-bit grey level. The resulting data matrix is of size 10304×400. Movielens data set with rating matrix of size 71567×65133 is also used, The largest data set is Netflix with a sparse rating matrix of size 480189×17770.

We employ two types of metrics, Bregman divergences $D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)$ and Signal to Interference Ratio(`SIR`) [5].

SIR is a commonly used metric in signal processing. We employ it here to judge how well the computed factors $\mathbf{W}$ and $\mathbf{H}$ match the ground-truth. Denoting the ground-truth as $\hat{\mathbf{W}}$ and $\hat{\mathbf{H}}$, the values of SIR for $\mathbf{W}$ is calculated as:

$$\mathrm{SIR}(\mathbf{W},\hat{\mathbf{W}})=\frac{10}{K}\sum_{k=1}^K\log(\frac{\|\hat{\mathbf{w}}_k\|_2^2}{\|\mathbf{w}_k-\hat{\mathbf{w}}_k\|_2^2})$$

where $\mathbf{w}_k$ and $\hat{\mathbf{w}}_k$ are normalized to have unit L_2 norm. SIR for $\mathbf{H}$ is computed analogously.

4.2 Performance with Various Bregman Divergences

In our experiments we use four Bregman divergences: Frobenius, KL, IS and Beta ($\beta=2$) divergences. Our algorithm is compared to the following three methods for NMF using specific subset of Bregman divergences:

`Conjugate gradient(CG):` This approach is based on the alternating nonnegative least squares (ANLS) framework, and solves the nonnegative least square subproblems efficiently by conjugate gradient method [15] for the numerical solution of particular systems of linear equations.

`BlockPivot:` This algorithm [22] is also based on the alternating nonnegative least squares (ANLS) framework and solves the nonnegative least square subproblems efficiently by using an active set like method called block principal pivoting.

`GCD/CCD:` A coordinate descent algorithm called Greedy Coordinate Descent (`GCD`) described in [17] to solve the NMF with Frobenius norm. It takes a greedy step of maximum decrease in objective function, and select important variables to update more often.

A Cyclic Coordinate Descent (`CCD`) algorithm is also proposed for the NMF with KL divergence. It differs from `GCD` by that the number of update for each variable is exactly the same, thus may conduct some unnecessary updates on unimportant variables. Newton's method is employed to solve each one-variable sub-problem.

and the following methods which are designed for NMF formulating using general Bregman divergences:

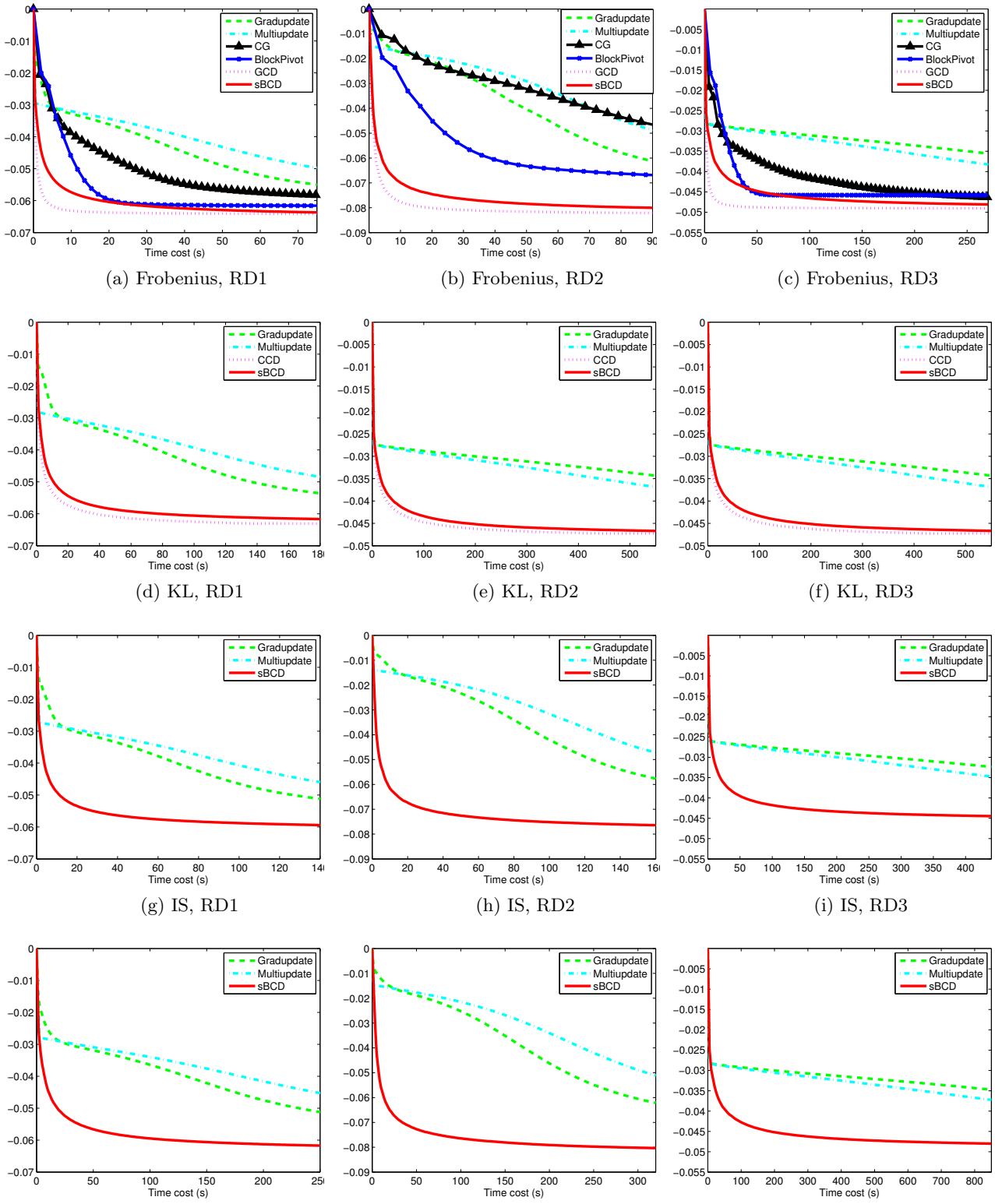
`Multiplicative updating:` This approach uses the multiplicative updating rule in [25, 8].

`Gradient descent:` This algorithm [25] calculates the first derivative of divergence based objective function, and uses a fixed step size in each iteration.

`sBCD:` Our proposed approach. The code is available at `http://www.cc.gatech.edu/grads/l/lli86/sbcd.zip`.

Table 3 shows above approaches' applicability to different divergences. For each specific divergence, not all listed approaches are compared since some of them may be not applicable. Therefore, for each case we conduct a separate series of experiments.

Figure 1 and 3 compare the performance of our approach with other algorithms measured by $D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)$. In general, `Multiplicative updating` converges relatively slow compared to others, but often find a good solution. For real world data, `Gradient descent` performs poorly. `Multiplicative updating` performs better than `Gradient descent`.



(a) Frobenius, RD1 (b) Frobenius, RD2 (c) Frobenius, RD3

(d) KL, RD1 (e) KL, RD2 (f) KL, RD3

(g) IS, RD1 (h) IS, RD2 (i) IS, RD3

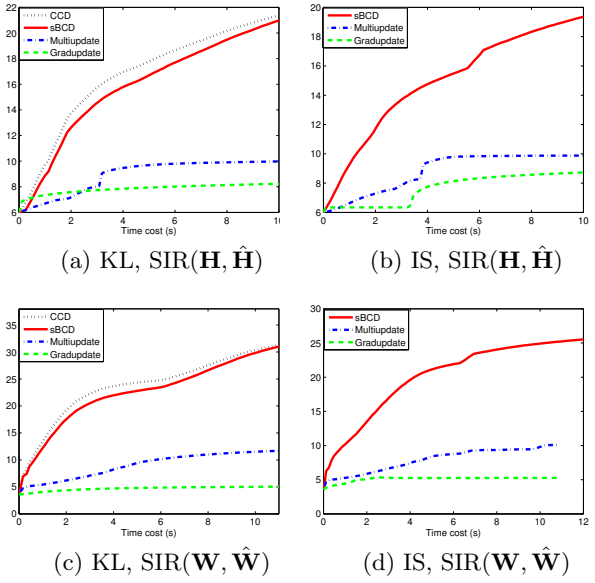
(j) Beta, RD1 (k) Beta, RD2 (l) Beta, RD3

Figure 1: Performance of NMF with various Bregman Divergences

`CG`: dash triangle line, `BlockPivot`: dash cross line, `Multiplicative updating`: dash dot line, `Gradient descent`: dash line, `sBCD`: solid line. y axis: relative residual value $log_{10}\frac{D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)}{D_\phi(\mathbf{A}\|\mathbf{W}_0\mathbf{H}_0^T)}$, where $\mathbf{W}_0$ and $\mathbf{H}_0$ are initial values for $\mathbf{W}$ and $\mathbf{H}$. (M, N, K) values are `RD1`: $(2000, 1000, 30)$, `RD2`: $(2000, 1000, 60)$, `RD2`: $(3000, 2000, 30)$.

Table 3: Applicability of Compared Methods

	Frobenius	KL	IS	Beta
CG	√	×	×	×
BlockPivot	√	×	×	×
GCD	√	×	×	×
CCD	×	√	×	×
Multiupdate	√	√	√	√
Gradupdate	√	√	√	√
sBCD	√	√	√	√



(a) KL, SIR($\mathbf{H}, \hat{\mathbf{H}}$) (b) IS, SIR($\mathbf{H}, \hat{\mathbf{H}}$)

(c) KL, SIR($\mathbf{W}, \hat{\mathbf{W}}$) (d) IS, SIR($\mathbf{W}, \hat{\mathbf{W}}$)

Figure 2: Performance of NMF with various Bregman Divergences

This experiment is conducted on `5 smooth` data set where the problem size is $(M, N, K) = (1000, 10, 5)$. In the figures of the first row, the y axis measures SIR($\mathbf{W}, \hat{\mathbf{W}}$); in the figures of the second row, the y axis measures SIR($\mathbf{H}, \hat{\mathbf{H}}$).

For Frobenius norm, `CG` can reach better solutions than the above two, but requires huge computational cost. `Block-Pivot` performs better than `CG`, significantly reducing the computational cost by using block pvioting scheme to speed up the process of finding optimal solution. Our approach performs better than all others expect `GCD`. For KL divergence, `CCD` is only slightly better than `sBCD`. The difference is much smaller than the difference between `GCD` and `sBCD` under Frobenius norm. Although our approach does not outperform `GCD` and `CCD`, we must notice that `GCD` targets at solving NMF with Frobenius norm only, while `CCD` targets at KL divergence only. On the other hand, our approach provides a general solution to NMF with Bregman divergences.

For IS and Beta divergences, our approach performs better than all other compared approaches, in both artificial and real world data. Its convergence behavior and the solution to the factorization is the best among all the approaches.

When handling large real world data sets, we notice that the performance curve of the above algorithms was not necessarily so smooth. However, the advantage of `sBCD` and `CCD` over `Multiplicative updating` and `Gradient descent` is still very significant, especially at the first several iterations. The more sparse $\mathbf{A}$ is, the greater improvement `sBCD` can obtain over the others.

Table 4: Performance for various reduced ranks

The table shows the time and iteration numbers needed for convergences under both IS and Beta divergence. The input matrix is of size 2000×1500 and the convergence criterion $\frac{\Delta D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)}{D_\phi(\mathbf{A}\|\mathbf{W}_0\mathbf{H}_0^T)} = 10^{-0.04}$ is set to measure the relative decrease in residual values, where $\Delta D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)$ is the absolute difference of $D_\phi(\mathbf{A}\|\mathbf{W}\mathbf{H}^T)$ between two consecutive iterations.

		IS			Beta		
	K	`Grad`	`Multi`	`sBCD`	`Grad`	`Multi`	`sBCD`
time	5	466.97	353.43	150.61	1109.56	803.84	346.51
(s)	10	632.48	525.91	200.17	1498.8	1198.62	461.98
	20	1082.11	905.42	236.77	2392.99	2009.69	550.89
	30	1318.35	1089.61	310.01	3241.07	2550.13	711.39
	40	2032.49	1634.5	325.22	4337.59	3838.38	829.43
	60	2474.16	2196.3	393.08	5103.28	4715.99	973.78
	80	3293.67	2769.9	446.3	7157.01	5591.19	1042.86
iter	5	171.1	85.74	30.4	174.3	91.2	26.1
	10	268.4	147.4	41.1	277	180.4	42.4
	20	320	221.6	45	378	213.1	46.6
	30	435.4	338.2	50.1	536	381.9	56.3
	40	588.3	416.8	53.3	620	466.5	63
	60	718.1	595.7	62.8	840.9	686.4	72.4
	80	891.5	784.6	71.7	1057.7	865.3	81.3

Figure 2 compares the performance of our approach with the other algorithms and the performance is measured by `SIR`. It illustrates that `sBCD` and `CCD` can recover the ground-truth factors $\hat{\mathbf{W}}$ and $\hat{\mathbf{H}}$ much better than `Multiplicative updating` and `Gradient descent`. Note that $\hat{\mathbf{H}}$ can be recovered much better than $\hat{\mathbf{W}}$. This may due to the fact that $\hat{\mathbf{H}}$ is given signal matrix while $\hat{\mathbf{W}}$ is generated randomly.

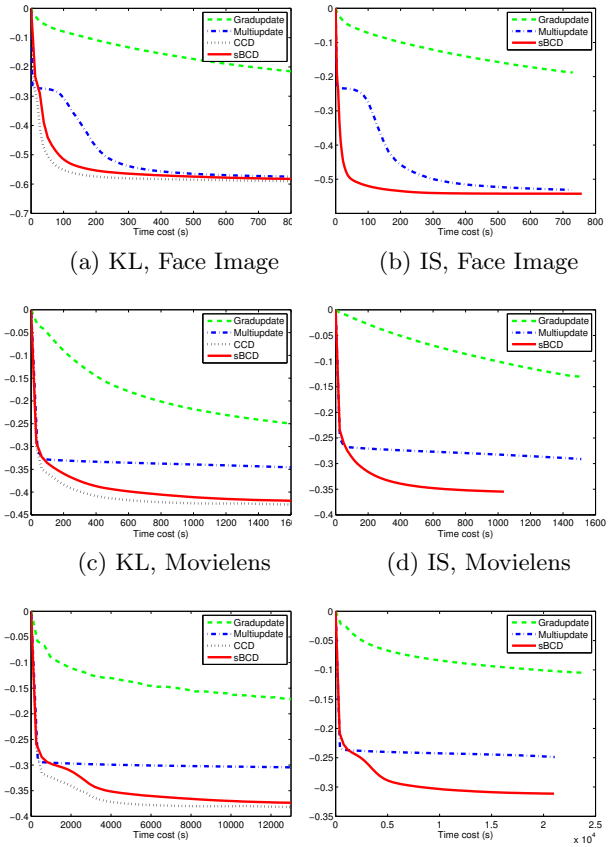
The following experiments evaluate how the variation of K influences the performance. From Table 4 we can see that our algorithm has a significant advantage in convergence behavior over the other two approaches. The advantage becomes greater with the increase of K. For `sBCD`, the number of iterations needed for convergence increases very slowly, while there is a sharp increase in the case of `Multiplicative updating` and `Gradient descent`.

4.3 Other Variations

In `sBCD`, the computation of the term $\nabla^2\phi(a'_{ij})$ may be costly, since a'_{ij} varies in each iteration. To address this issue, we also consider the following two alternative updating rules which can, in some cases, provide additional computational savings. Performance comparison of `sBCD` and those two variations is shown in Figure 4.

`sBCD-AL-A`	`sBCD-AL-B`
$h_{jk} = \frac{\sum_{i=1}^{M} \nabla^2\phi(a_{ij}) a_{ij}^{(k)} w_{ik}}{\sum_{i=1}^{M} \nabla^2\phi(a_{ij}) w_{ik} w_{ik}}$	$h_{jk} = \frac{\sum_{i=1}^{M} a_{ij}^{(k)} w_{ik}}{\sum_{i=1}^{M} w_{ik} w_{ik}}$

As shown in Figure 4, in most cases, `sBCD-AL-A` gives the slowest convergence and the worst solution. `sBCD` converges faster and obtains a better solution than `sBCD-AL-B`. `sBCD` also performs better consistently. The nature of $\mathbf{A}$ and the ϕ may decide how well the alternative updating rules approximate `sBCD`. For example, in IS divergence, `sBCD` gains an impressive advantage over the other two algorithms. However, in a few cases(especially the signal data), the difference among the three algorithms is not significant. This indicates that the two alternative algorithms may be more suitable for certain real world applications due to their simplicity in implementation and relatively lower storage cost.



(a) KL, Face Image (b) IS, Face Image

(c) KL, Movielens (d) IS, Movielens

(e) KL, Netflix (f) IS, Netflix

Figure 3: Performance of NMF with various Bregman Divergences on Large Scale Data

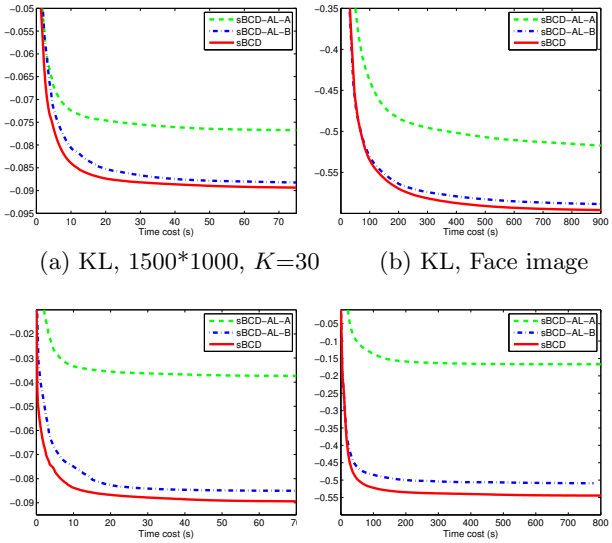
The y axis measures the logarithm of the relative residual value. (M, N, K) values are Face Image: $(10304, 400, 20)$, Movielens: $(71567, 65133, 20)$, Netflix: $(480189, 17770, 20)$. Matrices in Movielens and Netflix are very sparse.

For NMF with additional constraints, due to lack of space we only report here the result for the KL-divergence case when imposing sparsity constraint on $\mathbf{H}$ only. Figure 5 shows that with even with constraints added, the relative trends of our three proposed algorithms remain the same.

Here we choose text summarization as the application task, and conduct experiments to explore how different divergences can affect the topic generation. In this application, a document-text matrix is first built to describe the corpus. The matrix is then factorized by our proposed NMF algorithms to analyze the topic distribution over the corpus. Finally, the obtained document-topic matrix is used as features in model training for text summarization. We expect stronger features obtained from this NMF process.

The DUC2001 data set is used for evaluation in this series of experiments. It contains around 147 summary-document pairs. The respective ground-truth summaries are generated by manually extracting a certain number of sentences from each single document. A 10-fold cross validation process is employed in the experiments. Structural SVM algorithm is used for model training and prediction. For evaluation, we employ the ROUGE metric[4] [28].

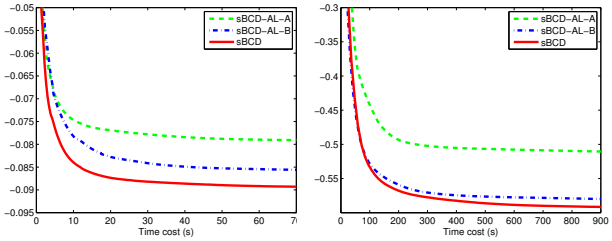
[4]For details, see http://berouge.com/default.aspx.



(a) KL, 1500*1000, K=30 (b) KL, Face image

(c) IS, 1500*1000, K=30 (d) IS, Face image

Figure 4: Comparison of Our Three Proposed Algorithms



(a) KL, 1500*1000, K=30 (b) KL, Face image

Figure 5: NMF with Additional Constraints

Table 5 shows that when divergence is KL, a best summarization model can be trained. Using Frobenius norm also leads to a comparable result. When divergence is IS or Beta, the summary prediction is relatively inaccurate. Thus, we can conclude that, in text topic analysis, using the NMF with Frobenius norm and KL divergence is more suitable than IS divergence and Beta divergence. The above results also illustrate that stronger features extracted from NMF contribute to a better summarization model.

5. CONCLUSIONS AND FUTURE WORK

In this paper, a novel fast algorithm named `sBCD` is proposed to solve the NMF with Bregman divergences. The algorithm is designed by deriving an equivalent optimization problem involving the Euclidean distance. A local updating rule is obtained by setting the gradient of the new objective function to zero with respect to each element of the two matrix factors. Experimental results demonstrate the effectiveness of our approach.

The relationship that we derive between Bregman divergences and the Euclidean distance is new. In addition to leading to our updating rule, this connection may be used in other data mining algorithms based on Bregman divergences, such as K-means, SVM.

Table 5: Performance of Text Summarization Using NMF with Different Divergences

Divergence	Frobenius	KL	IS	Beta
ROUGE-1-R	0.58215	0.58342	0.57931	0.57826
ROUGE-1-P	0.45734	0.46137	0.45432	0.45391
ROUGE-1-F	0.51001	0.52043	0.50043	0.49422
ROUGE-2-R	0.44123	0.45231	0.43921	0.43491
ROUGE-2-P	0.35342	0.36031	0.34232	0.34412
ROUGE-2-F	0.38634	0.39872	0.38123	0.38092
ROUGE-W-R	0.24107	0.24532	0.23726	0.23581
ROUGE-W-P	0.32313	0.33342	0.30523	0.30343
ROUGE-W-F	0.27132	0.28023	0.26932	0.26808

6. ACKNOWLEDGMENTS

We are very grateful to the editor and anonymous reviewers for valuable comments and suggestions based on which we were able to improve the manuscript substantially. The work of the first and third authors were supported in part by the National Science Foundation grants CCF-0732318 and CCF-0808863.

7. REFERENCES

[1] http://www.cl.cam.ac.uk/research/dtg/attarchive /facedatabase.html.

[2] A. Banerjee. Optimal bregman prediction and jensenạ́s equality. In *In Proc. International Symposium on Information Theory (ISIT)*, page 2004, 2004.

[3] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with bregman divergences. *J. Mach. Learn. Res.*, 6:1705–1749, December 2005.

[4] P. Breheny and J. Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *Annals of Applied Statistics*, 5(1):232–253, 2011.

[5] A. Cichocki and A.-H. Phan. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE Transactions on Fundamentals of Electronics*, 92:708–721, 2009.

[6] A. Cichocki and R. Zdunek. Nmflab for signal and image processing. In *tech. rep, Laboratory for Advanced Brain Signal Processing*, Saitama, Japan, 2006. BSI, RIKEN.

[7] A. Cichocki, R. Zdunek, and S. A. A.-H. Phan. Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation. New York, USA, 2009. Wiley.

[8] I. S. Dhillon and S. Sra. Generalized nonnegative matrix approximations with bregman divergences. In *Neural Information Proc. Systems*, pages 283–290, 2005.

[9] C. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Comput. Stat. Data Anal.*, 52:3913–3927, April 2008.

[10] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456).

[11] C. Fevotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural Comput.*, 21:793–830, March 2009.

[12] M. Figueiredo, R. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE J. of Selected Topics in Signal Proc*, 1:586–598, 2007.

[13] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 1 2010.

[14] N. Gillis and F. Glineur. Accelerated multiplicative updates and hierarchical als algorithms for nonnegative matrix factorization. *Neural Comput.*, 24(4):1085–1105, 4 2012.

[15] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6), 1952.

[16] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99*, pages 50–57, New York, NY, USA, 1999. ACM.

[17] C.-J. Hsieh and I. S. Dhillon. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 1064–1072, New York, NY, USA, 2011. ACM.

[18] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar. Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in Neural Information Processing Systems 24*, pages 2330–2338, 2011.

[19] H. Kim and H. Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23:1495–1502, June 2007.

[20] H. Kim and H. Park. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM J. Matrix Anal. Appl.*, 30:713–730, July 2008.

[21] J. Kim, Y. He, and H. Park. Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework. Under review.

[22] J. Kim and H. Park. Toward faster nonnegative matrix factorization: A new algorithm and comparisons. *IEEE International Conference on Data Mining*, 0:353–362, 2008.

[23] J. Kim and H. Park. Fast nonnegative matrix factorization: An active-set-like method and comparisons. In *SIAM Journal on Scientific Computing*, 2011.

[24] G. Lebanon. Axiomatic geometry of conditional models. *Information Theory, IEEE Transactions*, 51:1283–1294, April 2005.

[25] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562. MIT Press, 2000.

[26] Y. Li and S. Osher. Coordinate descent optimization for l1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Probl. Imaging*, 3(3).

[27] C.-J. Lin. Projected gradient methods for non-negative matrix factorization. *Neural Computation*, 19:2756–2779, October 2007.

[28] C. Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *NAACL*, pages 71–78, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[29] R. Mazumder, J. Friedman, and T. Hastie. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 106(495).

[30] S. D. Pietra, V. D. Pietra, and J. Lafferty. Duality and auxiliary functions for bregman distances. Technical report, School of Computer Science, Carnegie Mellon University, 2002.

[31] A. P. Singh and G. J. Gordon. A unified view of matrix factorization models. In *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II*, ECML PKDD '08, pages 358–373, Berlin, Heidelberg, 2008. Springer-Verlag.

[32] S. Wang and D. Schuurmans. Learning continuous latent variable models with bregman divergences. In *In Proc. IEEE International Conference on Algorithmic Learning Theory*, page 2004, 2003.

[33] T. Wu and K. Lange. Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, 2(1):224–244, 2008.

[34] S. Yun and K.-C. Toh. A coordinate gradient descent method for l1-regularized convex minimization. *Computational Optimization and Applications*, 48(2).