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
{lidl, 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 and is computationally faster than existing alternatives. 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 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.

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

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

where $D_\phi(A \| WH^T)$ denotes a Bregman divergence between $A$ and $WH^T$. 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 $W$ and $H$, 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

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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD ’12, August 12–16, 2012, Beijing, China.
Copyright 2012 ACM 978-1-4503-1462-6/12/08 ...$15.00.
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 any 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 unified 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. 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.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{R}$</td>
<td>real number</td>
</tr>
<tr>
<td>$\mathbb{R}_{\geq 0}$</td>
<td>nonnegative real number</td>
</tr>
<tr>
<td>$\mathbb{R}^{M \times N}$</td>
<td>vector space of matrices of size $M \times N$</td>
</tr>
<tr>
<td>$\mathbb{R}^K$</td>
<td>vector space of vectors of size $K$</td>
</tr>
<tr>
<td>$\odot$</td>
<td>element-wise product or Hadamard product</td>
</tr>
<tr>
<td>$\circ$</td>
<td>element-wise division</td>
</tr>
<tr>
<td>$a_{ij}$</td>
<td>$j$-th column vector of matrix $A$</td>
</tr>
<tr>
<td>$a_{i}$</td>
<td>element in the $i$-th row and $j$-th column of $A$</td>
</tr>
<tr>
<td>$| \cdot |_2$</td>
<td>Euclidean distance (Frobenius norm)</td>
</tr>
<tr>
<td>$D_\phi$</td>
<td>Bregman divergences</td>
</tr>
<tr>
<td>$\nabla \phi(x)$</td>
<td>a element-wise 1-order derivative operator of $\phi$ at $x$</td>
</tr>
<tr>
<td>$\text{sign}(x)$</td>
<td>sign function or signum function</td>
</tr>
</tbody>
</table>

2. **NMF WITH BREGMAN DIVERGENCES**

In NMF, given a matrix $A = [a_{ij}] \in \mathbb{R}_{+}^{M \times N}$, and an integer $K \leq \min(M, N)$, we are to find $W = [w_1, w_2, \ldots, w_K] \in \mathbb{R}_{+}^{M \times K}$ whose columns represent basis vectors in a $K$-dimensional space, and $H = [h_1, h_2, \ldots, h_K] \in \mathbb{R}_{+}^{N \times K}$ whose columns represent mixing proportions, such that

$$A \approx A' = WH^T.$$  \hspace{1cm} (1)

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

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

where $\phi$ is a univariate convex smooth function.

Bregman divergences are not symmetric in general, and the solution for $\min D_\phi(WH^T|A)$ will be different from that of $\min D_\phi(A|WH^T)$. In this paper, we focus on $D_\phi(A|WH^T)$ 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 $D_{KL}(A|WH^T)$ and music analysis using Itakura-Saito divergence $D_{IS}(A|WH^T)$. From a theoretical perspective, maximum likelihood estimation and information theory indicate that $D_{KL}(A|WH^T)$ is better motivated than $D_{KL}(WH^T|A)$, at least when the two arguments are probability vectors.

The choice of $\phi(x) = x^2/2$ reduces $D_\phi$ to the squared Frobenius norm of $E = A - WH^T$, which is the sum of squared entries of the residual matrix $E$. Notice that the Frobenius norm of a matrix can be viewed as the Euclidean distance of the corresponding vectorized matrix. Other choices for $\phi$ result in the non-negative Kullback-Leibler (KL) divergence or Itakura-Saito (IS) divergence. The specific choice of $\phi$ 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 $\phi$ and areas where these divergences are utilized, 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 $\phi$ 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 $W$ and $H$ lead to different NMF algorithms. One natural way of partition [8, 19, 22] is the two blocks representing $W$ and $H$, 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 $W$ or $H$.

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 $A - WH^T = A^{(k)} - w_k h_k^T$, where the $k$-residual $A^{(k)}$ is define as

$$A^{(k)} = A - \sum_{p \neq k} w_p h_p^T = A - WH^T + w_k h_k^T,$$

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

$$D_\phi(A\|A') = D_\phi(A\|WH^T) = \frac{1}{2} \| A - WH^T \|_F^2 = \frac{1}{2} \| A^{(k)} - w_k h_k^T \|_F^2 = D_\phi(A^{(k)}\|w_k h_k^T).$$

Let us define

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

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

$$E_t(A\|A') = E_t(A^{(k)}\|w_k h_k^T) \quad \text{or} \quad E_t(a_{ij}\|a_{ij}') = E_t(a_{ij}^{(k)}\|w_k h_{jk}).$$

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 $D_\phi(A^{(k)}\|w_k h_k^T)$ instead of $D_\phi(A\|WH^T)$, 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 $w_k$ or $h_k$ 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 $D_\phi(A\|WH^T)$ 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.**

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

**Proof.** The Taylor expansion of $D_\phi(a_{ij}\|a_{ij}')$ leads to

$$D_\phi(a_{ij}\|a_{ij}') = \phi(a_{ij}) - \phi(a_{ij}') - \nabla \phi(a_{ij}') (a_{ij} - a_{ij}') + \sum_{t=2}^{\infty} \frac{\nabla^t \phi(a_{ij}')}{t!} (a_{ij} - a_{ij}')^t$$

$$= \sum_{t=2}^{\infty} \frac{\nabla^t \phi(a_{ij}')}{t!} (a_{ij} - a_{ij}')^t$$

$$= \sum_{t=2}^{\infty} \frac{\nabla^t \phi(a_{ij}')}{t!} (-\text{sgn}(a_{ij}' - a_{ij}))^t E_t(a_{ij}\|a_{ij}').$$

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

The above relationship is then employed to replace our objective function $D_\phi(A\|WH^T)$ 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 expansion 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 $D_\phi(A\|WH^T)$ can be expressed in terms $E_t(a_{ij}^{(k)}\|w_{ik} h_{jk})$ as

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

Thus, instead of calculating the partial derivatives of $D_\phi(A\|WH^T)$ with respect to $W$ and $H$, 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 $W$ or $H$ (assuming other elements are fixed), a novel fast algorithm can be derived. From

$$\frac{\partial}{\partial h_{jk}} \left( \frac{\nabla^T \phi(a_j)}{t!} (-\text{sgn}(w_{ik} h_{jk} - a_j^{(k)}))^t \text{E}(a_j^{(k)} || w_{ik} h_{jk}) \right)$$

$$= -w_{ik} \frac{\nabla^T \phi(a_j^{(k)})}{(t-1)!} (a_j^{(k)} - w_{ik} h_{jk})^{t-1} + w_{ik} \frac{\nabla^T+1 \phi(a_j^{(k)})}{t!} (a_j^{(k)} - w_{ik} h_{jk})^t,$$

and Eqn (4), we obtain

$$\frac{\partial D_\phi(a_j || a_j^{(k)})}{\partial h_{jk}} = w_{ik} \nabla^2 \phi(a_j^{(k)})(w_{ik} h_{jk} - a_j^{(k)})$$

$$+ \sum_{t=2}^{\infty} \left( -w_{ik} \frac{\nabla^T+1 \phi(a_j^{(k)})}{t!} (a_j^{(k)} - w_{ik} h_{jk})^t \right)$$

$$+ w_{ik} \frac{\nabla^T+1 \phi(a_j^{(k)})}{t!} (a_j^{(k)} - w_{ik} h_{jk})^t.$$

Summing over the matrix rows and columns, we have

$$\frac{\partial D_\phi(A || \mathbf{WH}^T)}{\partial h_{jk}} = \sum_{i=1}^{M} w_{ik} \nabla^2 \phi(a_j^{(k)})(w_{ik} h_{jk} - a_j^{(k)})$$

$$= \left[ \mathbf{W}^T (\nabla^2 \phi(\mathbf{WH}^T)) \odot (w_{ik} h_{jk} - A^{(k)}) \right]_{kj}.$$ (5)

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

$$0 = \frac{\partial D_\phi(A || \mathbf{WH}^T)}{\partial h_{jk}} = \sum_{i=1}^{M} w_{ik} \nabla^2 \phi(a_j^{(k)})(w_{ik} h_{jk} - a_j^{(k)})$$

$$- \sum_{i=1}^{M} w_{ik} \nabla^2 \phi(a_j^{(k)})(w_{ik} h_{jk}).$$

which leads to the element-wise updating rule:

$$h_{jk} = \sum_{i=1}^{M} w_{ik} \nabla^2 \phi(a_j^{(k)}) a_j^{(k)} w_{ik}.$$

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

Note that the algorithm follows the block coordinate descent framework where each element in $W$ and $H$ 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 other NMF algorithms, Multiplicative Updating and Gradient Descent methods. According to Eqns (5) and (6), we have

$$h_{jk} = \left[ \frac{\sum_{i=1}^{M} w_{ik} \nabla^2 \phi(a_j^{(k)}) (a_j^{(k)} - w_{ik} h_{jk}) w_{ik}}{\sum_{i=1}^{M} \nabla^2 \phi(a_j^{(k)}) w_{ik}} \right]_+$$

$$= h_{jk} + \left[ \frac{[\mathbf{W}^T (\nabla^2 \phi(\mathbf{WH}^T)) \odot (A - \mathbf{WH}^T)]_{kj}}{[\mathbf{W}^T (\nabla^2 \phi(\mathbf{WH}^T)) \odot \mathbf{WH}^T]_{kj}} \right]_+$$

$$= h_{jk} + \left[ \frac{1}{[\mathbf{W}^T (\nabla^2 \phi(\mathbf{WH}^T))]_{kj}} (\frac{\partial D_\phi(A || \mathbf{WH}^T)}{\partial h_{kj}}) \right]_+$$


Algorithm 1 sBCD Algorithm

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

where we can view $\frac{\partial D_\phi(A || \mathbf{WH}^T)}{\partial h_{kj}}$ as the gradient direction, $\frac{1}{[\mathbf{W}^T (\nabla^2 \phi(\mathbf{WH}^T))]_{kj}}$ 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:

$$h_{jk} = h_{jk} + \frac{[\mathbf{W}^T (\nabla^2 (\mathbf{WH}^T) \odot A)]_{kj}}{[\mathbf{W}^T (\nabla^2 (\mathbf{WH}^T) \odot \mathbf{WH}^T)]_{kj}}$$

$$= h_{jk} + \frac{h_{jk}}{[\mathbf{W}^T (\nabla^2 \phi(\mathbf{WH}^T) \odot \mathbf{WH}^T)]_{kj}} (\frac{\partial D_\phi(A || \mathbf{WH}^T)}{\partial h_{kj}})$$

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 comparison of step sizes of sBCD and Multiplicative Updating shows that

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

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 $H$, the objective function is replaced with the following penalized version:

$$L(A || \mathbf{WH}^T) = D_\phi(A || \mathbf{WH}^T) + \alpha \sum_{k=1}^{K} \|h_{k}\|_1.$$ 

\footnote{Notice in implementation, regularization term $\|W\|_F$ is added to prevent it from growing too large.}
Table 2: Updating rules for specific Bregman Divergences

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

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

The corresponding $\text{sBCD}$ updating rule is

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

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

Sparsity on $W$ 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 $W$ and $H$ 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 $1000 \times 10$ 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 pixels in 8-bit grey level. The resulting data matrix is of size $10304 \times 400.$ Movielens data set with rating matrix of size $71567 \times 65133$ is also used. The largest data set is Netflix with a sparse rating matrix of size $480180 \times 17770.$

We employ two types of metrics, Bregman divergences $D_{\phi}(A|WH^T)$ 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 $W$ and $H$ match the ground-truth. Denoting the ground-truth as $W$ and $H$, the values of SIR for $W$ is calculated as:

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

where $w_k$ and $\hat{w}_k$ are normalized to have unit $L_2$ norm. SIR for $H$ 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 specific 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 $D_{\phi}(A|WH^T).$ 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.
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 $\log_{10} \frac{D_{\phi}(A \parallel WH^T)}{D_{\phi}(A \parallel W_0H_0^T)}$, where $W_0$ and $H_0$ are initial values for $W$ and $H$.

$(M, N, K)$ values are RD1: (2000, 1000, 30), RD2: (2000, 1000, 60), RD3: (3000, 2000, 30).
Table 3: Applicability of Compared Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Frobenius</th>
<th>KL</th>
<th>IS</th>
<th>Beta</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>BlockPivot</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>GCD</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>CCD</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Multipdate</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Gradupdate</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>sBCD</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 4: Performance for various reduced ranks

<table>
<thead>
<tr>
<th>K</th>
<th>IS Grad</th>
<th>Multi sBCD</th>
<th>Grad</th>
<th>Multi sBCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>466.97</td>
<td>354.43</td>
<td>150.61</td>
<td>346.51</td>
</tr>
<tr>
<td>10</td>
<td>632.48</td>
<td>525.91</td>
<td>200.17</td>
<td>461.98</td>
</tr>
<tr>
<td>20</td>
<td>1092.11</td>
<td>903.42</td>
<td>236.77</td>
<td>550.89</td>
</tr>
<tr>
<td>30</td>
<td>1318.35</td>
<td>1089.61</td>
<td>310.01</td>
<td>711.39</td>
</tr>
<tr>
<td>40</td>
<td>2032.49</td>
<td>1634.5</td>
<td>325.22</td>
<td>829.43</td>
</tr>
<tr>
<td>60</td>
<td>2474.16</td>
<td>2196.3</td>
<td>393.08</td>
<td>973.78</td>
</tr>
<tr>
<td>80</td>
<td>3293.67</td>
<td>2769.9</td>
<td>446.33</td>
<td>1042.86</td>
</tr>
</tbody>
</table>

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(\(\mathbf{H}, \mathbf{H}\)) in the figures of the second row, the y axis measures SIR(\(\mathbf{W}, \mathbf{W}\)).

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 pivoting 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 perform GCD and CCD, we must notice that CCD 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 A is, the greater improvement sBCD can obtain over the others.

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 A and the \(\phi\) 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.
employ the ROUGE metric employed in the experiments. Structural SVM algorithm is each single document. A 10-fold cross validation process is by manually extracting a certain number of sentences from pairs. The respective ground-truth summaries are generated of experiments. It contains around 147 summary-document stronger features obtained from this NMF process.

tures in model training for text summarization. W e expect

Finally , the obtained document-topic matrix is used as fea-
corpus. The matrix is then factorized by our proposed NMF
tation, a document-text matrix is first built to describe the

Figure 3: Performance of NMF with various Breg-
mancan contribute to a better summarization model.
also illustrate that stronger features extracted from NMF
Here we choose text summarization as the application task, and conduct experiments to explore how different diver-
gences and the Euclidean distance is new. In addition to the effectiveness of our approach.
The relationship that we derive between Bregman diver-
gences, such as K-means, SVM.

The DUC2001 data set is used for evaluation in this series leading to our updating rule, this connection may be used
The DUC2001 data set is used for evaluation in this series
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 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 diver-
gences 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 of [28].

5. CONCLUSIONS AND FUTURE WORK

Table 5 shows that when divergence is KL, a best summa-
rization 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.

In this paper, a novel fast algorithm named sBCD is pro-
solved 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 diver-
gences 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

<table>
<thead>
<tr>
<th>Divergence</th>
<th>Frobenius</th>
<th>KL</th>
<th>IS</th>
<th>Beta</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUGE-1-R</td>
<td>0.38215</td>
<td>0.38432</td>
<td>0.57931</td>
<td>0.58726</td>
</tr>
<tr>
<td>ROUGE-1-P</td>
<td>0.45734</td>
<td>0.46157</td>
<td>0.45432</td>
<td>0.45391</td>
</tr>
<tr>
<td>ROUGE-1-F</td>
<td>0.51001</td>
<td>0.52043</td>
<td>0.50043</td>
<td>0.49422</td>
</tr>
<tr>
<td>ROUGE-2-R</td>
<td>0.41423</td>
<td>0.45231</td>
<td>0.43921</td>
<td>0.43811</td>
</tr>
<tr>
<td>ROUGE-2-P</td>
<td>0.35342</td>
<td>0.36031</td>
<td>0.34232</td>
<td>0.34412</td>
</tr>
<tr>
<td>ROUGE-2-F</td>
<td>0.38634</td>
<td>0.39872</td>
<td>0.38123</td>
<td>0.38092</td>
</tr>
<tr>
<td>ROUGE-W-R</td>
<td>0.24107</td>
<td>0.25352</td>
<td>0.23726</td>
<td>0.23581</td>
</tr>
<tr>
<td>ROUGE-W-P</td>
<td>0.32131</td>
<td>0.33342</td>
<td>0.30523</td>
<td>0.30343</td>
</tr>
<tr>
<td>ROUGE-W-F</td>
<td>0.27132</td>
<td>0.28023</td>
<td>0.26932</td>
<td>0.26808</td>
</tr>
</tbody>
</table>

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


