Nonnegative Matrix Factorization via Rank-One Downdate

Michael Biggs MBIGGS@ALUMNI.UWATERLOO.CA
Ali Ghodsi AGHODSIB@UWATERLOO.CA
Stephen Vavasis VAVASIS@UWATERLOO.CA
University of Waterloo, Waterloo, ON N2L 3G1

Abstract

Nonnegative matrix factorization (NMF) was popularized as a tool for data mining by Lee and Seung in 1999. NMF attempts to approximate a matrix with nonnegative entries by a product of two low-rank matrices, also with nonnegative entries. We propose an algorithm called rank-one downdate (R1D) for computing an NMF that is partly motivated by the singular value decomposition. This algorithm computes the dominant singular values and vectors of adaptively determined submatrices of a matrix. On each iteration, R1D extracts a rank-one submatrix from the original matrix according to an objective function. We establish a theoretical result that maximizing this objective function corresponds to correctly classifying articles in a nearly separable corpus. We also provide computational experiments showing the success of this method in identifying features in realistic datasets. The method is also much faster than other NMF routines.

1. Nonnegative Matrix Factorization

Several problems in information retrieval can be posed as low-rank matrix approximation. The seminal paper by Deerwester et al. (1990) on latent semantic indexing (LSI) showed that approximating a term-document matrix describing a corpus of articles via the SVD led to powerful query and classification techniques. A drawback of LSI is that the low-rank factors in general will have both positive and negative entries, and there is no obvious statistical interpretation of the negative entries. This led Lee and Seung (1999) among others to propose *nonnegative matrix factorization* (NMF), that is, approximation of a matrix $A \in \mathbf{R}^{m\times n}$ as a product of two factors WH^T, where $W \in \mathbf{R}^{m\times k}$, $H \in \mathbf{R}^{n\times k}$, both have nonnegative entries, and $k \leq \min(m, n)$. Lee and Seung showed intriguing results with a corpus of images. In a related work, Hofmann (1999) showed the application of NMF to text retrieval. Nonnegative matrix factorization has its roots in work of Gregory and Pullman (1983), Paatero and Tapper (1994) and Cohen and Rothblum (1993).

Since the problem is NP-hard (Vavasis, 2007), it is not surprising that no algorithm is known to solve NMF to optimality. Heuristic algorithms proposed for NMF have generally been based on incrementally improving the objective $\|A - WH^T\|$ in some norm using local moves. A particularly sophisticated example of local search is due, e.g., to Kim and Park (2007). A drawback of local search is that it is sensitive to initialization and it is also sometimes difficult to establish convergence.

We propose an NMF method based on greedy rank-one downdating that we call R1D. R1D is partly motived by Jordan's algorithm for computing the SVD, which is described in Section 2. Unlike local search methods, greedy methods do not require an initial guess. In Section 3, we compare our algorithm to Jordan's SVD algorithm, which is the archetypal greedy downdating procedure. Previous work on greedy downdating algorithms for NMF is the subject of Section 4. In Section 5, we present the main theoretical result of this paper, which states that in a certain model of text due to Papadimitriou et al. (2000), optimizing our objective function means correctly identifying a topic in a text corpus; and Section 6 discusses the complexity of this problem. We then turn to computational experiments: in Section 7, we present results for R1D on image datasets, and in Section 8, we present results on text.

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

2. Algorithm and Objective Function

Rank-one downdate (R1D) is based on the simple observation that the leading singular vectors of a nonnegative matrix are nonnegative. This is a consequence of the Perron-Frobenius theorem (Golub & Van Loan, 1996). Based on this observation, it is trivial to compute a rank-one NMF. This idea can be extended to approximate a higher order NMF. Suppose we compute the rank-one NMF and then subtract it from the original matrix. The original matrix will not be nonnegative any more but all negative entries can be forced to be zero or positive and the procedure can be repeated.

An improvement on this idea takes only a submatrix of the original matrix and applies the Perron-Frobenius theorem. The point is that taking the whole matrix will in some sense average the features, whereas a submatrix can pick out particular features. A second reason to take a submatrix is that a correctly chosen submatrix may be very close to having a rank of one, so the step of forcing the residuals to be zero will not introduce significant inaccuracy (since they will already be close to zero).

The outer loop of the R1D algorithm may be described as follows.

Algorithm 1 R1D

```
input  $A \in \mathbf{R}^{m\times n}$, $k > 0$
output  $W \in \mathbf{R}^{m\times k}$, $H \in \mathbf{R}^{n\times k}$
for $\mu = 1$ to $k$ do
    $[M, N, \mathbf{u}, \mathbf{v}, \sigma] =$ ApproxRankOneSubmatrix($A$)
    $W(M, \mu) = \mathbf{u}(M)$
    $H(N, \mu) = \sigma\mathbf{v}(N)$
    $A(M, N) = 0$
end for
```

Here, M is a subset of $\{1, \ldots, m\}$, N is a subset of $\{1, \ldots, n\}$, $\mathbf{u} \in \mathbf{R}^m$, $\mathbf{v} \in \mathbf{R}^n$ and $\sigma \in \mathbf{R}$, and $\mathbf{u}, \mathbf{v}$ are both unit vectors. The function `ApproxRankOneSubmatrix` selects these five values so that the submatrix of A indexed by rows M and N is approximately rank one, and in particular, is approximately equal to $\mathbf{u}(M)\sigma\mathbf{v}^T(N)$. We follow Matlab subscripting conventions, so that $A(M, N)$ denotes this particular submatrix.

This outer loop for R1D may be called "greedy rank-one downdating" since it greedily tries to fill the columns of W and H from left to right by finding good rank-one submatrices of A and subtracting them from A. The classical greedy rank-one downdating algorithm is Jordan's algorithm for the SVD, described in Section 3. Related work on greedy rank-one downdating for NMF is the topic of Section 4.

The subroutine `ApproxRankOneSubmatrix`, presented later in this section, is a heuristic routine to maximize the following objective function:

$$f(M, N, \mathbf{u}, \sigma, \mathbf{v}) = \|A(M, N)\|_F^2 - \gamma\|A(M, N) - \mathbf{u}\sigma\mathbf{v}^T\|_F^2 \tag{1}$$

Here, γ is a penalty parameter. The Frobenius norm of an $m \times n$ matrix B, denoted $\|B\|_F$ is defined to be $\sqrt{B(1,1)^2 + B(1,2)^2 + \cdots + B(m,n)^2}$. The rationale for (1) is as follows: the first term in (1) expresses the objective that $A(M, N)$ should be large, while the second term penalizes departure of $A(M, N)$ from being a rank-one matrix.

Since the optimal $\mathbf{u}, \sigma, \mathbf{v}$ come from the SVD (once M, N are fixed), the above objective function can be rewritten just in terms of M and N as

$$\begin{aligned} f(M, N) &= \sum_{i=1}^{p} \sigma_i(A(M,N))^2 - \gamma \sum_{i=2}^{p} \sigma_i(A(M,N))^2 \\ &= \sigma_1(A(M,N))^2 \\ &\qquad - (\gamma - 1) \cdot \big(\sigma_2(A(M,N))^2 \\ &\qquad + \cdots + \sigma_p(A(M,N))^2\big), \end{aligned} \tag{2}$$

where $p = \min(|M|, |N|)$. The penalty parameter γ should be greater than 1 so that the presence of low-rank contributions is penalized rather than rewarded.

We conjecture that maximizing (1) is NP-hard (see Section 6), so we instead propose a heuristic routine for optimizing it. The procedure alternates improving M, N, $\mathbf{u}$, σ and $\mathbf{v}$ cyclically. First, observe that if M, N are already known, then the optimal choice of $\mathbf{u}$, σ, $\mathbf{v}$ can be found with the SVD. For fixed $(\mathbf{v}, N)$, the objective function (1) is separable by rows of the matrix. In particular, the contribution of row $i \in M$ is

$$\|A(i, N)\|^2 - \gamma\|A(i, N) - \beta_i\mathbf{v}^T\|^2,$$

where $\beta_i = u_i\sigma$. Note that β_i may be undefined if $i \notin M$. Nonetheless, given $\mathbf{v}$, the optimal β_i (i.e., the choice that minimizes $\|A(i, N) - u_i\mathbf{v}^T\|$) is easy to compute: it is $A(i, N)\mathbf{v}$, the solution to a simple least-squares minimization. Thus, we conclude that putting column i into index set M is favorable for the overall objective function provided that $f_i > 0$, where

$$f_i = \|A(i, N)\|^2 - \gamma\|A(i, N) - A(i, N)\mathbf{v}\mathbf{v}^T\|^2.$$

The formula for f_i can be simplified as follows:

$$\begin{aligned} f_i &= A(i,N)A(i,N)^T - \gamma(A(i,N) \\ &\qquad - A(i,N)\mathbf{v}\mathbf{v}^T)(A(i,N) - A(i,N)\mathbf{v}\mathbf{v}^T)^T \\ &= -(\gamma - 1)A(i,N)A(i,N)^T + \gamma(A(i,N)\mathbf{v})^2. \end{aligned}$$

If we rescale by $\gamma - 1$ (which does not affect the acceptance criterion), and we define new penalty parameters $\bar{\gamma} := \gamma/(\gamma-1)$, then we see that row i is accepted provided that

$$\bar{\gamma}(A(i,N)\mathbf{v})^2 - A(i,N)A(i,N)^T > 0.$$

A similar analysis applies to the columns, and leads to the conclusion that, given values for M and $\mathbf{u}$, column j should be accepted provided that

$$\bar{\gamma}(\mathbf{u}^T A(M,j))^2 - A(M,j)^T A(M,j) > 0.$$

The next issue is the choice of a starting guess for $M, N, \mathbf{u}, \sigma, \mathbf{v}$. The algorithm should be initialized with a starting guess that has a positive score, or else the rules for discarding rows and columns could conceivably discard all rows or columns. To put this more strongly, in order to improve the score of a converged solution, it seems sensible to select a starting guess with a high score. For this reason, R1D uses a single column of A as its starting guess, and in particular, the column of A with the greatest norm. (A single row may also be chosen.) It then chooses $\mathbf{u}$ to be the normalization of this column. This column is exactly rank one, so for the correct values of σ and $\mathbf{v}$ the first penalty term of (1) is zero. We have derived the following algorithm for the subroutine `ApproxRankOneSubmatrix` occuring in statement ⟨2⟩ in R1D.

Algorithm 2 ApproxRankOneSubmatrix

```
input $A \in \mathbf{R}^{m\times n}$ ,  parameter $\bar{\gamma} > 1$
output $M \subset \{1,\ldots,m\}$, $N \subset \{1,\ldots,n\}$,
       $\mathbf{u} \in \mathbf{R}^m$, $\mathbf{v} \in \mathbf{R}^n$, $\sigma \in \mathbf{R}$
 1: Select $j_0 \in \{1,\ldots,n\}$ to maximize $\|A(:,j_0)\|$
 2: $M = \{1,\ldots,m\}$
 3: $N = \{j_0\}$
 4: $\sigma = \|A(:,j_0)\|$
 5: $\mathbf{u} = A(:,j_0)/\sigma$
 6: repeat
 7:     Let $\bar{\mathbf{v}} = A(M,:)^T\mathbf{u}(M)$
 8:     $N = \{j : \bar{\gamma}\bar{v}(j)^2 - \|A(M,j)\|^2 > 0\}$
 9:     $\mathbf{v}(N) = \bar{\mathbf{v}}(N)/\|\bar{\mathbf{v}}(N)\|$
10:     Let $\bar{\mathbf{u}} = A(:,N)\mathbf{v}(N)$
11:     $M = \{i : \bar{\gamma}\bar{u}(i)^2 - \|A(i,N)\|^2 > 0\}$
12:     $\sigma = \|\mathbf{u}(M)\|$
13:     $\mathbf{u}(M) = \bar{\mathbf{u}}(M)/\sigma$
14: until stagnation in $M, N, \mathbf{u}, \sigma, \mathbf{v}$
```

The 'Repeat' loop is guaranteed to make progress because each iteration increases the value of the objective function. On the other hand, there does not seem to be any easy way to derive a useful prior upper bound on its number of iterations. In practice, it proceeds quite quickly, usually converging in 10–15 iterations. But to guarantee fast termination, monotonicity can be forced on M and N by requiring M to shrink and N to grow. In other words, statement ⟨8⟩ can be replaced by

$$N = N \cup \{j : \bar{\gamma}\bar{v}(j)^2 - \|A(M,j)\|^2 > 0\},$$

and statement ⟨11⟩ by

$$M = M - \{i : \bar{\gamma}\bar{u}(i)^2 - \|A(i,N)\|^2 \le 0\}.$$

Our experiments indicate that this change does not have a major impact on the performance of R1D.

Another possible modification to the algorithm is as follows: we modify the objective function by adding a second penalty term $-\rho|M|\cdot|N|$ to (1) where $\rho > 0$ is a parameter. The purpose of this term is to penalize very low-norm rows or columns from being inserted into $A(M,N)$ since they are probably noisy. For data with larger norm, the first term of (1) should dominate this penalty. Notice that this penalty term is also separable so it is easy to implement: the formula in ⟨8⟩ is changed to $\bar{\gamma}\bar{v}(j)^2 - \|A(M,j)\|^2 - \bar{\rho}|M| > 0$ while the formula in ⟨11⟩ becomes $\bar{\gamma}\bar{u}(i)^2 - \|A(i,N)\|^2 - \bar{\rho}|N| > 0$, where $\bar{\rho} = \rho/(\gamma-1)$. A good value for $\bar{\rho}$ is to set it so that in the initial starting point, the third penalty term is a small fraction (say $\bar{\eta} = 1/20$) of the other terms. This leads to the following definition for ρ:

$$\rho = \bar{\eta}(\bar{\gamma}-1)\sigma^2/m,$$

which may be computed immediately after ⟨4⟩.

Greedy rank-one downdating appears to be much faster than other NMF algorithms. Generating each column of W and H requires approximately 20 matrix-vector multiplications; these multiplications are always at least as sparse as the original data. There is no iterative improvement phase. It can also be much faster than the SVD, especially for sparse data.

3. Relationship to the SVD

The classical rank-one greedy downdating algorithm is Jordan's algorithm for computing the singular value decomposition (SVD) (Stewart, 1993). Recall that the SVD takes as input an $m \times n$ matrix A and returns three factors U, Σ, V such that $U \in \mathbf{R}^{m\times k}$ and U has orthonormal columns (i.e., $U^TU = I$), $\Sigma \in \mathbf{R}^{k\times k}$ and is diagonal with nonnegative diagonal entries, and $V \in \mathbf{R}^{n\times k}$ also with orthonormal columns, such that $U\Sigma V^T$ is the optimal rank-k approximation to A in either the 2-norm or Frobenius norm. (Recall that the 2-norm of an $m \times n$ matrix B, denoted $\|B\|_2$, is

Algorithm 3 JordanSVD

```
input $A \in \mathbf{R}^{m\times n}$ and $k \le \min(m,n)$
output $U, \Sigma, V$ as above.
for $\mu = 1$ to $k$ do
    Select a random nonzero $\bar{\mathbf{u}} \in \mathbf{R}^m$
    $\sigma = \|\bar{\mathbf{u}}\|$
    $\mathbf{u} = \bar{\mathbf{u}}/\sigma$
    repeat {power method}
        $\bar{\mathbf{v}} = A^T\mathbf{u}$
        $\mathbf{v} = \bar{\mathbf{v}}/\|\bar{\mathbf{v}}\|$
        $\bar{\mathbf{u}} = A\mathbf{v}$
        $\sigma = \|\bar{\mathbf{u}}\|$
        $\mathbf{u} = \bar{\mathbf{u}}/\sigma$
    until stagnation in $\mathbf{u}, \sigma, \mathbf{v}$
    $A = A - \mathbf{u}\sigma\mathbf{v}^T$
    $U(:,\mu) = \mathbf{u}$
    $V(:,\mu) = \mathbf{v}$
    $\Sigma(\mu,\mu) = \sigma$
end for
```

defined to be $\sqrt{\lambda_{\max}(B^TB)}$, where $\lambda_{\max}$ denotes the maximum eigenvalue.)

Thus, we see that R1D is quite similar to the SVD. The principal difference is that R1D tries to find a submatrix indexed by $M \times N$ at the same time that it tries to identify the optimal $\mathbf{u}$ and $\mathbf{v}$. Hence, the formulas for $\mathbf{u}$ and $\mathbf{v}$ occurring in $\langle 9\rangle$ and $\langle 13\rangle$ of subroutine `ApproxRankOneSubmatrix`, which were presented earlier as solutions to a least-squares problem, may also be regarded as steps in a power method. In particular, this means that if M and N are fixed, then the inner repeat loop of this subroutine will indeed converge to the dominant singular triple of $A(M,N)$.

As mentioned earlier, a shortcoming of the SVD is that its factors contain both positive and negative numbers. It has another subtler shortcoming when used for clustering which is as follows: because the SVD always operates on the entire matrix, it can return a singular vector that averages the results from two nearly disjoint topics in a corpus (see Biggs et al. (2008) for an example). R1D avoids this pitfall by seeking a submatrix that is approximately rank-one as it applies the power method.

4. Related Work

As mentioned in the introduction, most algorithms proposed in the literature are based on forming an initial W and H and then improving them by local search on an objective function. The objective function usually includes a term of the form $\|A - WH^T\|$ in some norm, and may include other terms.

A few previous works follow an approach similar to ours, namely, greedy subtraction of rank-one matrices. This includes the work of Bergmann et al. (2003), who identify the rank-one matrix to subtract as the fixed point of an iterative process. Asgarian and Greiner (2006) find the dominant singular pair and then truncate it. Gillis (2006) finds a rank-one understimator and subtracts that. Boutsidis and Gallopoulos (2007) consider the use of a greedy algorithm for initializing other algorithm and make the following interesting observation: The nonnegative part of a rank-one matrix has rank at most 2.

The main innovation herein is the idea that the search for the rank-one submatrix should itself be an optimization subproblem. This observation allows us to compare one candidate submatrix to another. (Gillis also phrases his subproblem as optimization, although his optimization problem does not explicitly seek submatrices like ours.) A second innovation is our analysis in Section 5 showing that if the subproblem were solved optimally, then R1D would be able to accurately find the topics in the model of ϵ-separable corpora (Papadimitriou et al., 2000).

5. Behavior of this objective function on a nearly separable corpus

In this section, we establish the main theoretical result of the paper, namely, that the objective function given by (1) is able to correctly identify a topic in a nearly separable corpus. We define our *text model* as follows. There is a universe of *terms* numbered $1,\ldots,m$. There is also a set of *topics* numbered $1,\ldots,t$. Topic k, for $k = 1,\ldots,t$, is a probability distribution over the terms. Let $P(i,k)$ denote the probability of term i occurring in topic k. Thus, P is a singly stochastic matrix, i.e., it has nonnegative entries with column sums exactly 1. We assume also that there is a probability distribution over topics; say the probability of topic k is τ_k, for $k = 1,\ldots,t$. The text model is thus specified by P and $\tau_1,\ldots,\tau_t$. We use the Zipf distribution as the model of document length. In particular, there is a number L such that all documents have length less than L, and the probability that a document of length l occurs is

$$\frac{1/l}{1 + 1/2 + \cdots + 1/(L-1)}.$$

We have checked that the Zipf model is a good fit for several common datasets.

A *document* is generated from this text model as follows. First, topic k is chosen at random according to the probability distribution $\{\tau_1,\ldots,\tau_t\}$. Then, a length l is chosen at random from $\{1,\ldots,L-1\}$ according to the Zipf distribution. Finally, the document itself is chosen at random by selecting l terms independently according to the probability distribution $P(:,k)$. A *corpus* is a set of n documents chosen independently using this text model. Its *term-document matrix* is the $m \times n$ matrix A such that $A(i,j)$ is the frequency of term i in document j.

We further assume that the text model is *ϵ-separable*, meaning that each topic k is associated with a set of terms $T_k \subset \{1,\ldots,m\}$, that $T_1,\ldots,T_t$ are mutually disjoint, and that $P(i,k) \le \epsilon$ for $i \notin T_k$, i.e., the probability that a document on topic k will use a term outside of T_k is small. Let $P_{\min} = \min\{P(i,k) : i \in T_k, k=1,\ldots,t\}$. Without loss of generality, $P_{\min} > 0$ since any row $i \in T_k$ such that $P(i,k) = 0$ may be removed from T_k without affecting the validity of the model. Parameter ϵ must satisfy an inequality mentioned below. This corpus model is quite similar to that of Papadimitriou et al. (2000). One difference is in the the document length model. Our model also relaxes several assumptions of Papadimitriou et al.

Our main theorem is that the objective function given by (1) correctly finds documents associated with a particular topic in a corpus.

Theorem 1. *Let $(P,(\tau_1,\ldots,\tau_t))$ specify a text model, and let $\alpha > 0$ be chosen arbitrarily. Assume $\epsilon > 0$ is chosen smaller than a function $\epsilon(P_{\min}, m, t, \alpha)$ (see Biggs et al. (2008) for this function). Suppose that the text-model is ϵ-separable with respect to $T_1,\ldots,T_t$, the subsets of terms defining the topics. Let A be the term-document matrix of a corpus of n documents drawn from this model when the document-length parameter is L.*

Choose $\gamma = 4$ in (1). Then with probability tending to 1 as $n \to \infty$ and $L \to \infty$, the optimizing pair (M,N) of (1) satisfies the following. Let $D_1,\ldots,D_t$ be the partitioning of the columns of A according to topics. There exists a topic $k \in \{1,\ldots,t\}$ such that $A(M,N)$ and $A(T_k,D_k)$ are nearly coincident in the following sense.

$$\sum_{(i,j)\in(M\times N)\triangle(T_k\times D_k)} A(i,j)^2 \le \alpha \sum_{(i,j)\in M\times N} A(i,j)^2.$$

Here, $X \triangle Y$ denotes the set-theoretic symmetric difference $(X-Y)\cup(Y-X)$. The proof of this theorem is lengthy and appears in Biggs et al. (2008). It relies on Chernoff-Hoeffding estimates and perturbation results for singular vectors such as Theorem 8.6.5 of Golub and Van Loan (1996).

6. On the complexity of maximizing $f(M,N)$

In this section, we observe that the problem of globally maximizing (2) is NP-hard at least in the case that γ is treated as an input parameter. This observation explains why R1D settles for a heuristic maximization of (2) rather than exact maximization. First, observe that the maximum biclique (MBC) problem is NP-hard as proved by Peeters (2003). We show that the MBC problem can be transformed to an instance of (2).

Let us recall the definition of the MBC problem. The input is a bipartite graph G. The problem is to find an (m,n)-complete bipartite subgraph K (sometimes called a *biclique*) of G such that mn is maximized, i.e., the number of edges of K is maximized.

Suppose we are given G, an instance of the maximum biclique problem. Let A be the left-right adjacency matrix of G, that is, if $G=(U,V,E)$ where $U\cup V$ is the bipartition of the node set, then A has $|U|$ rows and $|V|$ columns, and $A(i,j)=1$ if $(i,j)\in E$ for $i \in U$ and $j \in V$, else $A(i,j)=0$.

Consider maximizing (2) for this choice of A. We require the following preliminary lemmas whose proofs are omitted.

Lemma 2. *Let A be a matrix that has either of the following as a submatrix:*

$$U_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ or } U_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}. \qquad (3)$$

Then $\sigma_2(A) > 0.618$.

This lemma leads to the following lemma.

Lemma 3. *Suppose all entries of $A \in \mathbf{R}^{m\times n}$ are either 0 or 1, and suppose and at least one entry is 1. Suppose M,N are the optimal solution for maximizing $f(M,N)$ given by (2). Suppose also that the parameter γ is chosen to be $2.7mn+1$ or larger. Then the optimal choice of M,N must yield a matrix $A(M,N)$ of all 1's, possibly augmented with some rows or columns that are entirely zeros.*

Now consider the main claim, namely, that optimize (M,N) of the objective function for this A corresponds to the max biclique. If $A(M,N)$ includes a row or column entirely of zeros, then this row or column may be dropped without affecting the value of the objective function (2). Hence it follows from the lemma that without loss of generality that the optimizer (M,N) of (2) indexes a matrix of all 1's. In that case, $\sigma_1(A(M,N)) = \sqrt{|M|\cdot|N|}$ while $\sigma_2(A(M,N)) =$

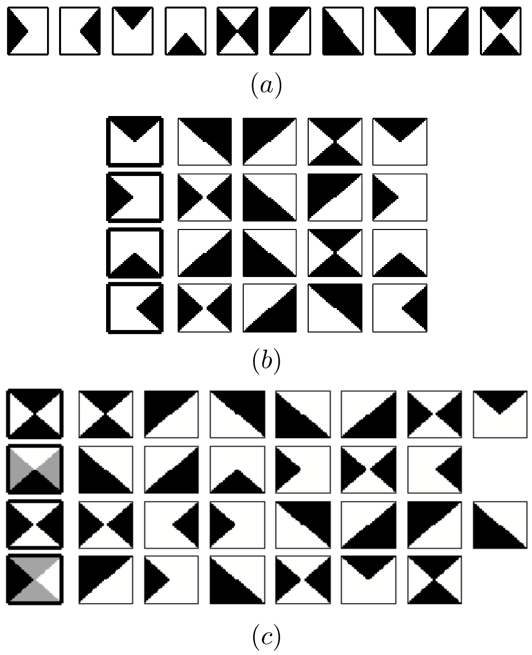

Figure 1. A binary image dataset is depicted in (a); white indicates zeros. The result of R1D on this dataset is shown in (b), and LSI in (c).

$\cdots = \sigma_p(A(M,N)) = 0$ (where $p = \min(|M|,|N|)$), and hence $f(M,N) = |M| \cdot |N|$. Thus, the value of the objective function corresponds exactly to the number of edges in the biclique. This completes the proof that biclique is reducible in polynomial time to maximizing (2).

We note that Gillis (2006) also uses the result of Peeters for a similar purpose, namely, to show that the subproblem arising in his NMF algorithm is also NP-hard.

The NP-hardness result in this section requires that γ be an input parameter. We conjecture that (2) is NP-hard even when γ is fixed (say $\gamma = 4$ as used herein).

7. Image dataset test cases

We first demonstrate the performance of R1D on a simple binary image dataset, depicted in Figure 1 (*a*). Each of the ten dataset images is composed of one or two "basis" triangles. The results of R1D (with parameter $\bar{\gamma} = 4$) and LSI on this dataset are shown in Figure 1 (*b*) and (*c*), respectively, and the interpretation is as follows. The leftmost column illustrates the four leading columns of W, which are the learned features. For each of these, the images on the right are the dataset images with the largest entries in the corresponding column of H; they should be closely associated with the feature on the left.

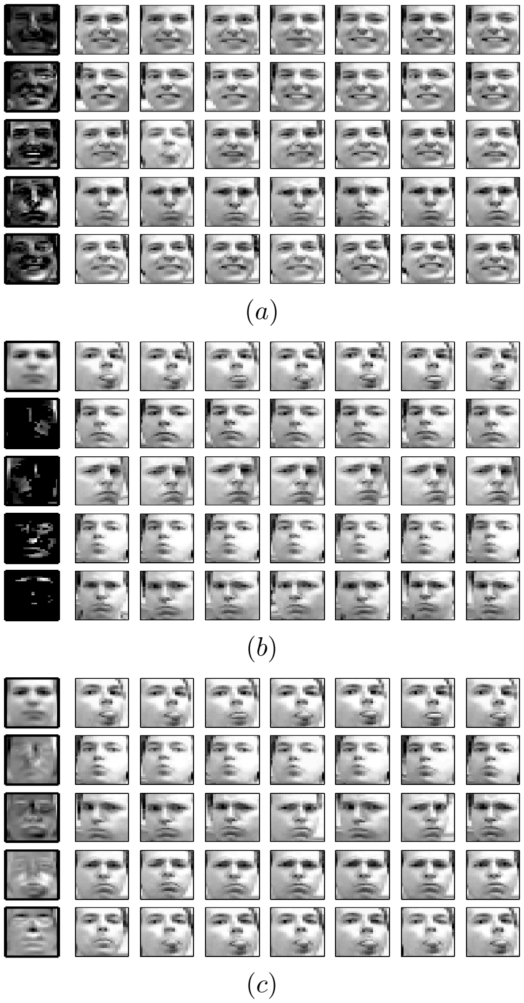

Figure 2. Three algorithms applied to the Frey face dataset (black indicates zeros): (a) NMF with divergence criterion, (b) our R1D algorithm for NMF, and (c) LSI

R1D discovered the four triangles as a basis, and to each it associated exactly the dataset images which contain the appropriate triangle. Alternatively, the LSI factorization is not as interpretable.

We have also compared results against NMFDIV from nmfpack (Hoyer, 2000; Hoyer, 2004). NMFDIV requires k, the number of basis vectors to compute, as an input parameter which globally affects the factors W and H. If k is correctly set to 4, NMFDIV is able to compute the same correct result as R1D. Otherwise, some or all of the basis vectors will appear incorrect, including the first ones. R1D and LSI will each compute the same leading columns regardless of k, and on this dataset they will not compute more than 4 columns; all subsequent columns of W and H will be

Table 1. The amount of sparsity in the NMF computed by R1D ($\bar{\gamma} = 2$) on the Frey face dataset. It is presented as the percentage of zero values in the first few columns of W and H.

Column	% zeros in W	% zeros in H
1	0.00	0.00
2	0.82	0.69
3	0.69	0.68
4	0.82	0.88
5	0.94	0.73

zero.

Figure 2 conducts a similar experiment on the Frey face dataset, which consists of 1965 registered face images of size 28×20. Again, the leading columns of W present the "eigenfaces" or "features" discovered in the dataset, and the corresponding column of H selects dataset images that are classified as carrying the feature most prominently. R1D seems to be the most successful at finding features and classifying images; in each case, the column of W shows a particular highlight that distinguishes some images in the dataset from others. NMFDIV appears to be slightly inferior to R1D, while LSI is noticeably worse.

In this experiment, the algorithms computed 30 basis vectors of the NMF. NMFDIV was allowed 500 iterations which took 727 seconds; in contrast, LSI required 20 seconds and R1D took 47 seconds.

Additionally, R1D is effective at finding a sparse factorization. Table 1 demonstrates the sparsity in the first few columns of W and H. The first column of W and H is fully dense, because the data matrix appears to be approximately rank-one; its first singular value is dominant. Apart from this, the other columns of the NMF are sparse, and the sparsity can be controlled by the $\bar{\gamma}$ parameter (here we have used $\bar{\gamma} = 2$). Alternatively, both NMFDIV and LSI perform a dense factorization with very few values near zero in any column.

8. Text dataset test cases

In Tables 2 and 3 we illustrate LSI versus R1D (with parameter $\bar{\gamma} = 4$) on the TDT Pilot Study (TDT Study, 1997). The columns of each table are the leading columns of W, with the leading terms per column displayed. The LSI results show that the topics are not properly separated and terms from different topics recur or are mixed. The columns in the R1D table are clearly identifiable topics, and the terms in each columns are all correctly associated with the given topics.

Table 2. Topics found by LSI on the TDT Pilot Study corpus (tf-idf normalization).

Topic 1	Topic 2	Topic 3	Topic 4
SIMPSON	ISRAEL	ISRAEL	BOSNIAN
PRESIDENT	ISRAELI	ISRAELI	SERBS
CLINTON	BOSNIAN	PALESTINIAN	SERB
POLICE	PEACE	GAZA	SARAJEVO
HOUSE	SERBS	ARAFAT	BOSNIA
ISRAEL	BOSNIA	PLO	NATO
BOSNIAN	SERB	JERUSALEM	SIMPSON
HAITI	SARAJEVO	PEACE	BIHAC
UNITED	PALESTINIAN	PALESTINIANS	AIR
GOVERNMENT	NATO	SIMPSON	TROOPS

Table 3. Topics found by R1D on the TDT Pilot Study corpus (tf-idf normalization). Note that all words in a column do in fact refer to the same news event.

Topic 1	Topic 2	Topic 3	Topic 4
SIMPSON	MASTERS	KOREA	DENG
JUDGE	PAIRINGS	KOREAN	XIAOPING
ITO	AUGUSTA	NORTH	RONG
JURY	AMATEUR	KIM	PARAMOUNT
DEFENSE	TOURNAMENT	PYONGYANG	CHINA
TRIAL	ROUND	SEOUL	HEALTH
ANGELES	GOLF	SUNG	CHINESE
LOS	NOTED	NUCLEAR	KONG
PROSECUTION	PLAYERS	SOUTH	HONG
CASE	GEORGIA	COMMUNIST	DAUGHTERS

NMFDIV (and the other implementations of NMF in nmfpack) were not run on this dataset because they would exhaust all of the computer's memory. As noted earlier, R1D on text datasets is able to efficiently work with sparse matrices throughout its operation. R1D was able to compute 80 basis vectors of the TDT corpus in 171 seconds, whereas LSI required 269 seconds.

9. Conclusions

We have proposed an algorithm called R1D for nonnegative matrix factorization. It is based on greedy rank-one downdating according to an objective function, which is heuristically maximized. We have shown that the objective function is well suited for identifying topics in the ϵ-separable text model. Finally, we have shown that the algorithm performs well in practice.

This work raises several interesting open questions. First, the ϵ-separable text model seems rather too simple to describe real text, so it would be interesting to see if the results generalize to more realistic models.

A second arising question asks whether a result like Theorem 1 will hold for the R1D algorithm. In other words, if the heuristic subroutine `ApproxRankOneSubmatrix` is applied to an ϵ-separable corpus, does it successfully identify a topic? Here is an example of a difficulty. Suppose $n \to \infty$ much faster than L. In this case, the document j with the highest norm will be the one in which l_j is very close to L and in which one entry $A(i, j)$ is very close to L while the rest are mostly zeros. This is because the maximizer of $\|\mathbf{x}\|_2$ subject to the constraint that $\|\mathbf{x}\|_1 = C$ occurs when one entry of $\mathbf{x}$ is equal to C and the rest are zero. It is likely that at least one instance of such a document will occur regardless of the matrix $P(\cdot, \cdot)$ if n is sufficiently large. This document will then act as the seed for expanding M and N, but it may not be similar to any topic. This scenario can perhaps be prevented by a more intelligent selection of a starting vector for `ApproxRankOneSubmatrix.`

References

Asgarian, N., & Greiner, R. (2006). Using rank-1 biclusters to classify microarray data. Department of Computing Science, University of Alberta, Edmonton, AB, Canada.

Bergmann, S., Ihmels, J., & Barkai, N. (2003). Iterative signature algorithm for the analysis of large-scale gene expression data. *Physical Review E*, *67*, 031902.

Biggs, M., Ghodsi, A., & Vavasis, S. (2008). Nonnegative matrix factorization via rank-one downdate. Available online at http://www.arxiv.org/abs/0805.0120.

Boutsidis, C., & Gallopoulos, E. (2007). SVD based initialization: A head start for nonnegative matrix factorization. In press.

Cohen, J., & Rothblum, U. (1993). Nonnegative ranks, decompositions and factorizations of nonnegative matrices. *Linear Algebra and its Applications*, *190*, 149–168.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*, 391–407.

Gillis, N. (2006). Approximation et sous-approximation de matrices par factorisation positive: algorithmes, complexité et applications. Master's thesis, Université Catholique de Louvain, Louvain-la-Neuve, Belgium. In French.

Golub, G. H., & Van Loan, C. F. (1996). *Matrix computations, 3rd edition.* Baltimore: Johns Hopkins University Press.

Gregory, D. A., & Pullman, N. J. (1983). Semiring rank: Boolean rank and nonnegative matrix rank. *J. Combin. Inform. System Sci*, *3*, 223–233.

Hofmann, T. (1999). Probabilistic latent semantic analysis. *UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden, July 30-August 1, 1999* (pp. 289–296). Morgan Kaufmann.

Hoyer, P. (2000). nmfpack - matlab code for nmf. http://http://www.hiit.fi/node/70.

Hoyer, P. (2004). Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, *5*, 1457–1469.

Kim, H., & Park, H. (2007). Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. Bioinformatics (to appear).

Lee, D., & Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788–791.

Paatero, P., & Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, *5*, 111–126.

Papadimitriou, C., Raghavan, P., Tamaki, H., & Vempala, S. (2000). Latent semantic indexing: A probabilistic analysis. *J. Comput. Syst. Sci.*, *61*, 217–235.

Peeters, R. (2003). The maximum edge biclique problem is NP-complete. *Discrete Applied Mathematics*, *131*, 651–654.

Stewart, G. W. (1993). On the early history of the singular value decomposition. *SIAM Review*, *35*, 551–566.

TDT Study (1997). Topic detection and tracking pilot study. http://projects.ldc.upenn.edu/TDT/.

Vavasis, S. (2007). On the complexity of nonnegative matrix factorization. arxiv.org, 0708.4149.