Text Mining using Non-Negative Matrix Factorizations

V. Paul Pauca* Farial Shahnaz† Michael W. Berry‡ Robert J. Plemmons§

January 2, 2004

Abstract

This study involves a methodology for the automatic identification of semantic features and document clusters in a heterogeneous text collection. The methodology is based upon encoding the data using low rank non-negative matrix factorization algorithms to preserve natural data non-negativity and thus avoid subtractive basis vector and encoding interactions present in techniques such as principal component analysis. Some existing non-negative matrix factorization techniques are reviewed and some new ones are proposed. Numerical experiments are reported on the use of a hybrid NMF algorithm to produce a parts-based approximation of a sparse term-by-document matrix. The resulting basis vectors and matrix projection can be used to identify underlying semantic features (topics) and document clusters of the corresponding text collection.

Keywords: text mining, non-negative matrix factorization, clustering, dimension reduction, semantic feature identification.

1 Introduction

Non-negative matrix factorization (NMF) has recently been shown to be a very useful technique in approximating high dimensional data where the data are comprised of non-negative components. In a seminal paper published in *Nature* [12], Lee and Seung proposed the idea of using NMF techniques to find a set of basis functions to represent image data where the basis functions enable the identification and classification of intrinsic "parts" that make up the object being imaged by multiple observations. They showed that NMF facilitates the analysis and classification of data from image or sensor articulation databases made up of images showing a composite object in many articulations, poses, or observation views. They also found NMF to be a useful tool in text data mining. In the past few years, several papers have discussed NMF techniques and successful applications to various databases where the data values are non-negative, e.g., [6, 8, 9, 10, 13, 14, 17].

More generally, matrix factorization techniques in data mining fall under the category of vector space methods. Very often databases of interest lead to a very high dimensional matrix representation. Low-rank factorizations not only enable the user to work with reduced dimensional models, they also often facilitate more efficient statistical classification, clustering and organization of data, and lead to faster searches and queries for patterns or trends, e.g., Berry, Drmač, and Jessup [3]. Recently, Xu et al [18] demonstrated that NMF-based indexing outperforms traditional vector space approaches to information retrieval (such as latent semantic indexing) for document clustering on a few benchmark test collections.

NMF is a vector space method to obtain a representation of data using non-negativity constraints. These constraints can lead to a parts-based representation because they allow only additive, not subtractive, combinations of the original data. This is in contrast to techniques for finding a reduced dimensional representation based on singular value decomposition-type methods such as principal component analysis (PCA) [11]. One major problem with PCA is that the basis vectors have both positive and negative components, and the data are represented as linear combinations of these vectors with positive and negative coefficients. In many applications, however, the negative components contradict physical realities. In particular, term frequencies in text mining are non-negative. In this paper, we survey some popular computational approaches (and their complexities) for NMF in the context of document clustering applications, and demonstrate the use of a *new*

[*] Department of Computer Science, Wake Forest University, Winston-Salem, NC 27109. His research was supported in part by the Air Force Office of Scientific Research under grant FA49620-03-1-0215, and by the Army Research Office under grant DAAD19-00-1-0540. Email: paucavp@wfu.edu

[†] Department of Computer Science, University of Tennessee, Knoxville, TN 37996-3450. Email: shahnaz@cs.utk.edu

[‡] Department of Computer Science, University of Tennessee, Knoxville, TN 37996-3450. Email: mberry@utk.edu

[§] Departments of Computer Science and Mathematics, Wake Forest University, Winston-Salem, NC 27109. His research was supported in part by the Air Force Office of Scientific Research under grant F49620-02-1-0107, and by the Army Research Office under grant DAAD19-00-1-0540. Email: plemmons@wfu.edu

hybrid NMF method that can enforce smoothness (or sparsity) constraints on the resulting factor matrices.

2 Non-Negative Matrix Factorization

Given an initial database expressed as an $n \times m$ matrix X, where each column is an n-dimensional non-negative vector of the original database (m vectors), the standard NMF problem is to find two new reduced-dimensional matrices W and H, in order to approximate the original matrix X by the product WH in terms of some metric. Each column of W contains a *basis vector* while each column of H contains the *weights* needed to approximate the corresponding column in X using the basis from W. The dimensions of matrices W and H are $n \times r$ and $r \times m$, respectively. Usually, the number of columns in the new (basis) matrix W is chosen so that $r \ll m$. Here, the choice of r is generally application dependent, and may also depend upon the characteristics of the particular database within the application [8].

The usual approach to the NMF problem is to approximate X by computing a pair W and H to minimize the Frobenius norm of the difference $X - WH$. Mathematically, the problem can be stated as follows: Let $X \in R^{n \times m}$ be a data matrix of non-negative entries. Let $W \in R^{n \times r}$ and $H \in R^{r \times m}$ for some positive integer $r < m$. The objective is then to solve the optimization problem

$$\min_{W,H} \|X - WH\|_F^2, \tag{2.1}$$

subject to $W_{ij} \geq 0$ and $H_{ij} \geq 0$ for each i and j.

Of course the matrices W and H are generally not unique. Conditions resulting in uniqueness in the special case of equality, $X = WH$, have been recently studied by Donoho and Stodden [6], using cone theoretic techniques (See also Chapter 1 in Berman and Plemmons [1]). Algorithms designed to approximate X by solving the minimization problem (2.1) generally begin by initial estimates of the matrices W and H, followed by alternating iterations to improve these estimates.

Next, some existing non-negative matrix factorization techniques are reviewed and some new ones are described.

2.1 Multiplicative Method. A non-negative matrix factorization algorithm of Lee and Seung [12] is based on multiplicative update rules of W and H. We call this scheme the *multiplicative method*, and denote it by **MM**. A formal statement of the method is given next.

Algorithm for MM

1. Initialize W and H with non-negative values, and scale the columns of W to unit norm.
2. Iterate for each c, j, and i until convergence or after k iterations:
   (a) $H_{cj} \leftarrow H_{cj} \dfrac{(W^T X)_{cj}}{(W^T WH)_{cj} + eps}$
   (b) $W_{ic} \leftarrow W_{ic} \dfrac{(XH^T)_{ic}}{(WHH^T)_{ic} + eps}$
   (c) Scale the columns of W to unit norm.

Clearly the approximations W and H remain non-negative during the updates. It is generally best to update W and H "simultaneously", instead of updating each matrix fully before the other. In this case, after updating a row of H, we update the corresponding column of W. Matlab performs well with these computations. In the implementation, a small positive quantity, say the square root of the machine precision, should be added to the denominators in the approximations of W and H at each iteration step. We use a parameter $eps = 10^{-9}$ in our Matlab codes for this purpose.

It is often important to normalize the columns of X in a pre-processing step, and in the algorithm to normalize the columns of the basis matrix W at each iteration. In this case we are optimizing on a unit hypersphere, as the column vectors of W are effectively mapped to the surface of a hypersphere by the repeated normalization.

The computational complexity of Algorithm MM can be shown to be $O(rnm)$ operations per iteration. If data are added to the database then it can either be added directly to the basis matrix W along with a minor modification of H, or else if r is fixed, then further iterations can be applied starting with the current W and H as initial approximations.

Lee and Seung [13] proved that under the MM update rules the distance $\|X - WH\|_F^2$ is monotonically non-increasing. In addition it is invariant if and only if W and H are at a stationary point of the objective function (2.1). From the viewpoint of nonlinear optimization, the algorithm can be classified as a diagonally-scaled gradient descent method [8]. Lee and Seung [12] have also provided an additive algorithm. Both the multiplicative and additive algorithms are related to expectation-maximization approaches used in image processing computations such as image restoration, e.g., [16].

2.2 Sparse Encoding. Hoyer [9] has suggested a novel non-negative sparse coding scheme based on ideas from the study of neural networks, and the scheme has been applied to the decomposition of databases into independent feature subspaces by Hyvärinen and Hoyer [10]. Hoyer's method [9] has the important feature of enforcing a statistical sparsity for the weight matrix H, thus enhancing the parts-based representation of the data in W.

Mu, Plemmons and Santago [15] propose a regularization approach that, like Hoyer's method, can be used to enforce statistical sparsity of the weight matrix H. This approach uses a so-called point count regularization scheme in the computations that penalizes the *number* of nonzero entries in H, rather than $\sum_{ij} H_{ij}$, as proposed by Hoyer. Sparsity often leads to a basis representation in W that better represents parts or features on the database information in X.

2.3 A Hybrid Method. We propose a hybrid algorithm for NMF that combines some of the better features of the methods previously discussed. First, we adopt the multiplicative algorithm approach for computing an approximation to the basis matrix W at each iterative step. This computation is essentially a matrix version of the gradient descent optimization scheme. Secondly, we compute the weight matrix H using a constrained least squares (CLS) model as the metric. The purpose is to penalize non-smoothness and non-sparsity in H. This approach bears similarity to those of Hoyer and Mu, Plemmons and Santago. The CLS model is related to the least squares Tikhonov regularization technique commonly used in image restoration [16]. Our algorithm, which we denote **GD-CLS** for *gradient descent with constrained least squares*, is given next.

Algorithm for GD-CLS

1. Initialize W and H with non-negative values, and scale the columns of W to unit norm.
2. Iterate until convergence or after k iterations:
   (a) $W_{ic} \leftarrow W_{ic} \dfrac{(XH^T)_{ic}}{(WHH^T)_{ic} + eps}$, for c and i
   (b) Rescale the columns of W to unit norm.
   (c) Solve the constrained least squares problem:
   $$\min_{H_j}\{\|X_j - WH_j\|_2^2 + \lambda\|H_j\|_2^2\},$$
   where the subscript j denotes the j^{th} column, for $j = 1, \ldots, m$. Any negative values in H_j are set to zero. The parameter λ is a regularization value that is used to balance the reduction of the metric
   $$\|X_j - WH_j\|_2^2$$
   with enforcement of smoothness and sparsity in H.

As done in Algorithm MM, we use a small positive parameter *eps* to avoid dividing by zero or very small numbers and enhance stability in the computations for W in Step 2(a). Our numerical approach for solving the constrained least squares problem in Step 2(c) for the columns H_j of H makes use of an algorithm similar to one that we employed in [16] for regularized least squares image restoration.

3 Sample Text Collection

We have recently tested the effectiveness of these nonnegative matrix factorization techniques and report here on how the promising **GD-CLS** method performed for our text mining applications. An on-line version of the 1989 Concise Columbia Encyclopedia (or CCE) [4] was used to *mine* or discover the semantic features of the heterogeneous articles/topics comprising the collection[1]. Whereas the complete CCE contains 15,460 documents (yielding a dictionary of 29,670 important terms), we have used a subset of the collection for our Matlab-based experiments, namely the 1,063 "Letter A" articles spanning 5,117 keywords. All unimportant terms (stopwords) [2] were removed using the GTP (General Text Parser) [7] software environment. All terms (or keywords) comprising the resulting dictionary were required to occur at least twice (globally) across the collection of articles and in two or more articles.

4 Computational Results

Having parsed the first 100 "Letter A" articles of the CCE (see Section 3), we constructed a $5,117 \times 100$ term-by-document matrix X where each matrix element (x_{ij}) is the raw term frequency of keyword (or term) i in document (article) j. Although not used for this study, different term weighting functions (e.g., tf-idf, log-entropy) can be used to improve the conceptual discrimination of documents and global importance of a keyword across the collection (see [2]).

Our intent in applying the **GD-CLS** to this small collection was to assess the benefits of a *parts*-based factorization of the term-by-document matrix X for the semantic analysis of heterogeneous (multiple concepts) text. Using Matlab, we approximate the $t \times d$ (sparse) matrix X via

$$(4.2) \qquad X \simeq WH = \sum_{k=1}^{r} W_k H^k \ ,$$

[1] We note that Lee and Seung [12] used articles from the Grolier Encyclopedia to test Algorithm MM discussed in Section 2.1.

Table 1: **GD-CLS** performance on CCE (Letter A) with the data matrix X of size $5,117 \times 100$. Time is elapsed CPU time in seconds and r is set to 50.

λ	$\frac{\|X-WH\|_F^2}{\|X\|_F^2}$	nnz(H)	Time
0.1	0.578	1,758	460.05
0.01	0.219	1,923	460.65
0.001	0.171	2,132	459.50

where W and H are $t \times r$ and $r \times d$, respectively, non-negative matrices. W_k denotes the kth column of W and H^k denotes the kth row of the matrix H. Clearly, the non-negativity of W and H facilitate a parts-based representation of the matrix X whereby the basis (column) vectors of W or W_k combine to approximate the original columns (documents) of the sparse matrix X. The outer product representation of WH in Eq. (4.2) demonstrates how the rows of H or H^k essentially specify the weights (scalar multiples) of each of the basis vectors needed for each of the rank r parts of the representation. As described in [12], we can interpret the semantic feature represented by a given basis vector W_k by simply sorting (in descending order) its t elements and generating a list of the corresponding dominant terms (or keywords) for that feature. In turn, a given row of H having d elements (i.e., H^k) can be used to reveal documents sharing common basis vectors W_k, i.e., similar semantic features or meaning. The columns of H, of course, are the projections of the columns (documents) of X onto the basis spanned by the columns of W.

An alternative approach (see [5] where this approach is used for summarizing video) to identifying document clusters of similar meaning would be to construct a $d \times d$ similarity matrix $S = \tilde{X}^T\tilde{X}$, where $\tilde{X}$ is the matrix X whose columns have been scaled (normalized) to unit length[2]. In this case, each element of S (say s_{ij}) would reflect the cosine of document (column) vectors i and j in a simple vector space representation of the original term-by-document matrix X. Document content is then modeled using a parts-based representation of the non-negative matrix S to automatically reveal document clusters or sets having similar cosine scores. As a caveat, of course, *high* cosine relevancy measures do not always guarantee *high* semantic similarity among documents (see [2, 3]).

Tables 1 and 2 illustrate the performance of the **GD-CLS** method for computing the NMF of the 5,117 × 100 matrix X and 100 × 100 similarity matrix S when $r = 50$ basis vectors are generated. Table 1 is measuring a summarization task while Table 2 is measuring a clustering task. In both tables, $nnz(H)$ denotes the number of nonzeros in the 50×100 matrix H and all execution times reflect elapsed CPU times obtained using Matlab version 6.5 on a Sun Microsystems SunBlade 1000 workstation[3]. Table 2 also illustrates the computational advantages of document clustering using the NMF of the matrix $S = \tilde{X}^T\tilde{X}$ versus the NMF of the matrix $n \times m$ matrix X followed by the necessary $m(m-1)/2$ pairwise cosine calculations involving the rows of H.

Table 2: **GD-CLS** performance on CCE (Letter A) using $S = \tilde{X}^T\tilde{X}$. Time is elapsed CPU time in seconds and r is set to 50.

λ	$\frac{\|S-WH\|_F^2}{\|S\|_F^2}$	nnz(H)	Time (Cosines[a])
0.1	0.343	2,198	14.73 (459.20)
0.01	0.300	2,158	15.93 (459.66)
0.001	0.301	2,071	17.73 (458.73)

[a] Time for computing W and H from X using **GD-CLS** followed by cosine calculations involving the rows of H.

Notice that as λ increases, the (normalized) Frobenius norm of the error increases (more so with X than S). However, we note that the sparsity of H decreases for the NMF of X with larger λ. This reflects the enforcement of smoothness (or sparsity) for larger choices of the regularization parameter λ (see Section 2.3). For the NMF of S, we have observed no significant effects on the error $\|S - WH\|_F^2/\|S\|_F^2$ or sparsity of H with different choices of λ, which is quite promising in terms of robustness of this approach.

Five semantic features (columns of W) obtained from the three different NMF models represented by Table 1 include: chemistry, Greek mythology, the Persian Gulf, philosophy, and the Civil War. One anomaly was observed in the basis vector W_{16} obtained for the NMF(X) with $\lambda = 0.01$ where two different concepts were clearly represented by this vector (medicine and the Civil War). Overall, we noticed no significant difference in the interpretability of the basis vectors W_k with different choices of the regularization parameter λ.

For interpretation purposes, the articles (by title) can be identified by the nonzeros of each H^k which would be present in the kth *part* of the approximation to X in Eq. (4.2). Each part (or span of W_k) can be used to classify the documents so the sparsity of H greatly affects the diversity of topics with which any particular semantic feature can be associated. As an alternative to the document clustering of NMF(X), one can list the articles identified by the rows of H

[2] Scaling each column X_k by $\|X_k\|_2$ suffices here.

[3] 500 MHz UltraSPARC-IIe processor with a 256KB L2 cache, 512MB DRAM and 20GB internal disk.

(i.e., H^k) when computing the NMF(S). For the most part, the same five semantic features (topics) mentioned above are identifiable with each NMF model specified by Table 2. These preliminary results suggest that many of the *parts* produced by the NMF(S) can be *noisy* in the sense that diverse topics are shown to span the same semantic features (basis vectors W_k). Further tests similar to those described in [18] (especially with larger text collections) are needed to better understand the accuracy of document clustering produced by our hybrid NMF approach applied to the similarity matrix S.

5 Concluding Remarks

Several other methods have been proposed for non-negative matrix factorization. (See [14] for a recent survey). Studies and comparisons of various algorithms for non-negative factorizations have been given by Guillamet and Vitria [8] and by Liu and Yi [14]. Each approach has advantages and disadvantages, but in general the algorithms examined in these papers have been found to be effective on certain applications, and so the method of choice is often application dependent [8].

We have demonstrated how a hybrid algorithm for computing the NMF can be used effectively in text classification. The proposed **GD-CLS** algorithm can be used to compute a parts-based approximation $X \simeq WH$ of a sparse term-by-document matrix X in which the quality of approximation (error reduction) can be enhanced by an enforcement of smoothness and sparsity in H. Further work is needed in exploring the effects of different term weighting schemes (for X) on the quality of the basis vectors W_k and document clustering using the parts-based representation of the similarity matrix S. Designing a robust C++ implementation of **GD-CLS** within the GTP software environment [7] will facilitate a more comprehensive study of the benefits of NMF for text mining applications.

References

[1] A. Berman and R. Plemmons. *Non-Negative Matrices in the Mathematical Sciences*, SIAM Press Classics Series, Philadelphia, 1994.

[2] M. Berry and M. Browne. *Understanding Search Engines: Mathematical Modeling and Text Retrieval*, SIAM, Philadelphia, 1999.

[3] M. Berry, Z. Drmač, and E. Jessup. "Matrices, Vector Spaces, and Information Retrieval", *SIAM Review*, Vol. 41, pp. 335-362, 1999.

[4] *Concise Columbia Encyclopedia.* Columbia University Press, New York, Second Edition, 1989.

[5] M. Cooper and J. Foote, "Summarizing Video using Non-Negative Similarity Matrix Factorization", *Proc. IEEE Workshop on Multimedia Signal Processing* St. Thomas, US Virgin Islands, 2002.

[6] D. Donoho and V. Stodden. "When does Non-Negative Matrix Factorization Give a Correct Decomposition into Parts?", preprint, Department of Statistics, Stanford University, 2003.

[7] J.T. Giles, L. Wo, and M.W. Berry. "GTP (General Text Parser) Software for Text Mining", in *Statistical Data Mining and Knowledge Discovery*, H. Bozdogan (Ed.), CRC Press, Boca Raton, (2003), pp. 455-471.

[8] D. Guillamet and J. Vitria. "Determining a Suitable Metric when Using Non-Negative Matrix Factorization", *16th International Conference on Pattern Recognition (ICPR'02)*, Vol. 2, Quebec City, QC, Canada, 2002.

[9] P. Hoyer. "Non-Negative Sparse Coding", *Neural Networks for Signal Processing XII (Proc. IEEE Workshop on Neural Networks for Signal Processing)*, Martigny, Switzerland, 2002.

[10] A. Hyvärinen and P. Hoyer. "Emergence of Phase and Shift Invariant Features by Decomposition of Natural Images into Independent Feature Subspaces", *Neural Computation*, Vol. 12, pp. 1705-1720, 2000.

[11] I. Jolliffe. *Principle Component Analysis*, 2nd Ed., Springer Series in Statistics, Springer-Verlag, New York, 2002.

[12] D. Lee and H. Seung. "Learning the Parts of Objects by Non-Negative Matrix Factorization", *Nature*, Vol. 401, pp. 788-791, 1999.

[13] D. Lee and H. Seung. "Algorithms for Non-Negative Matrix Factorization", *Advances in Neural Processing*, 2000.

[14] W. Liu and J. Yi. "Existing and New Algorithms for Non-negative Matrix Factorization", preprint, Computer Sciences Dept., UT Austin, 2003.

[15] Z. Mu, R. Plemmons and P. Santago. "Iterative Ultrasonic Signal and Image Deconvolution for Estimating the Complex Medium Response", preprint, submitted to *IEEE Transactions on Ultrasonics and Frequency Control*, 2003.

[16] S. Prasad, T. Torgersen, V. Pauca, R. Plemmons, and J. van der Gracht. "Restoring Images with Space Variant Blur via Pupil Phase Engineering", Optics in Info. Systems, Special Issue on Comp. Imaging, SPIE Int. Tech. Group Newsletter, Vol. 14, No. 2, pp. 4-5, 2003.

[17] S. Wild, J. Curry and A. Dougherty. "Motivating Non-Negative Matrix Factorizations", Proceedings of the Eighth SIAM Conference on Applied Linear Algebra, Williamsburg, VA, July 15-19, 2003. See http://www.siam.org/meetings/la03/proceedings/.

[18] W. Xu, X. Liu, and Y. Gong. "Document-Clustering based on Non-Negative Matrix Factorization", Proceedings of SIGIR'03, July 28 - August 1, Toronto, CA, pp. 267-273, 2003.