DIMENSIONALITY REDUCTION USING NON-NEGATIVE MATRIX FACTORIZATION FOR INFORMATION RETRIEVAL

Satoru Tsuge, Masami Shishibori, Shingo Kuroiwa, Kenji Kita

Department of Information Science & Intelligent Systems
Faculty of Engineering, Tokushima University
2-1, Minami-josanjima, Tokushima, 770-8506 Japan
E-mail: {tsuge, bori, kuroiwa, kita}@is.tokushima-u.ac.jp

Abstract

The Vector Space Model (VSM) is a conventional information retrieval model, which represents a document collection by a term-by-document matrix. Since term-by-document matrices are usually high-dimensional and sparse, they are susceptible to noise and are also difficult to capture the underlying semantic structure. Additionally, the storage and processing of such matrices places great demands on computing resources. Dimensionality reduction is a way to overcome these problems. Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) are popular techniques for dimensionality reduction based on matrix decomposition, however they contain both positive and negative values in the decomposed matrices. In the work described here, we use Non-negative Matrix Factorization (NMF) for dimensionality reduction of the vector space model. Since matrices decomposed by NMF only contain non-negative values, the original data are represented by only additive, not subtractive, combinations of the basis vectors. This characteristic of parts-based representation is appealing because it reflects the intuitive notion of combining parts to form a whole. Also NMF computation is based on the simple iterative algorithm, it is therefore advantageous for applications involving large matrices. Using MEDLINE collection, we experimentally showed that NMF offers great improvement over the vector space model.

Keywords

information retrieval, vector space model, non-negative matrix factorization, dimensionality reduction

1 Introduction

With the rapid growth of online information, e.g., the World Wide Web (WWW), a large collection of full-text documents is available and opportunity for getting a useful piece of information is increased. Information retrieval is now becoming one of the most important issues for handling a large text data.

The Vector Space Model (VSM) is one of the major conventional information retrieval models that represent documents and queries by vectors in a multidimensional space[1]. Since these vectors are usually high-dimensional and sparse, they are susceptible to noise and are also difficult to capture the underlying semantic structure. Additionally, the storage and processing of such data places great demands on computing resources. Dimensionality reduction is a way to overcome these problems. Principal Component Analysis (PCA) and Singular Value Decomposition (SVD)[2] are popular techniques for dimensionality reduction based on matrix decomposition. However, the cost of their computation will be prohibitive when matrices become large.

This paper propose to apply Non-negative Matrix Factorization (NMF) [3][4] to dimensionality reduction of the document vectors in the term-by-document matrices. The NMF decomposes a non-negative matrix into two non-negative matrices. One of the decomposed matrix can be regarded as the basis vectors. The dimensionality reduction can be performed by project-

0-7803-77-2/01/$10.00

ing the document vectors onto the lower dimensional space which is formed by these basis vectors.

The NMF is distinguished from the other methods, e.g., PCA and SVD, by its non-negativity constraints. These constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations. Also, the NMF computation is based on the simple iterative algorithm, it is therefore advantageous for applications involving large matrices.

The remainder of this paper is organized as follows. In section 2, we introduce non-negative matrix factorization. In section 3, a method of dimensionally reduction with NMF is described. Section 4 shows infomation retreival results on the MEDLINE test collection and discusses these results. Finally, section 5 gives conclusions and future works.

2 Non-Negative Matrix Factorization

This section provides a brief overview of Non-negative Matrix Factorization (NMF) [3][4]. Given a non-negative $n \times m$ matrix V, the NMF finds the non-negative $n \times r$ matrix W and the non-negative $r \times m$ matrix H such that

$$V \approx WH. \tag{1}$$

The r is generally chosen to satisfy $(n+m)r < nm$, so that the product WH can be regarded as a compressed form of the data in V.

The equation (1) can be rewritten column by columns as

$$\boldsymbol{v} \approx W\boldsymbol{h}, \tag{2}$$

where $\boldsymbol{v}$ and $\boldsymbol{h}$ are the corresponding columns of V and H. Each data vector $\boldsymbol{v}$ is approximated by a linear combination of the columns of W, weighted by the components of $\boldsymbol{h}$. Therefore, W can be regarded as containing a basis that is optimized for the linear approximation of the data in V. Since relatively few basis vectors are used to represent many data vectors, good approximation can only be achieved if the basis vectors discover structure that is latent in the data. This idea is similar to Latent Semantic Indexing (LSI)[2][5]. By projecting document vectors onto new space with lower dimensions using these basis vectors, the NMF may achieve better performance than SVD.

The NMF does not allow negative entries in the matrix W and H. These constraints lead to a parts-based representation because they allow only additive, not subtractive, combination. This characteristic of parts-based representation is appealing because it reflects the intuitive notion of combining parts to form a whole.

2.1 NMF computation

Here, we introduce two algorithms based on iterative estimation of W and H[3][4]. At each iteration of the algorithms, the new value of W or H is found by multiplying the current value by some factor that depends on the quality of the approximation in equation (1). Repeated iteration of the update rules is guaranteed to converge to a locally optimal matrix factorization.

First, we introduce the update rules given in the next equations,

$$\overline{H}_{ij} = H_{ij}\frac{(W^TV)_{ij}}{(W^TWH)_{ij}}, \tag{3}$$

$$\overline{W}_{ij} = W_{ij}\frac{(VH^T)_{ij}}{(WHH^T)_{ij}}. \tag{4}$$

Repeated iteration of these update rules converges to a local minimum of the objective function:

$$F = \sum_i \sum_j (V_{ij} - (WH)_{ij})^2. \tag{5}$$

This objective function is defined as the square of the Euclidean distance between V and WH for a measure. We call these update rules *update rule 1.*

Next, we introduce the update rules which maximize the following objective function:

$$F = \sum_i \sum_j (V_{ij}\log((WH)_{ij}) - (WH)_{ij}) \tag{6}$$

This objective function is defined as the Kullback-Leibler divergence for a measure. The decomposed matrices W and H are updated as follows:

$$\overline{H}_{ij} = H_{ij}\sum_k W_{kj}\frac{V_{kj}}{(WH)_{kj}}, \tag{7}$$

$$\hat{W}_{ij} = W_{ij}\sum_k \frac{V_{ik}}{(WH)_{ik}}H_{jk},$$

$$\overline{W}_{ij} = \frac{\hat{W}_{ij}}{\sum_k \hat{W}_{kj}}. \tag{8}$$

We call these update rules *update rule 2.*

3 Dimensionality Reduction Using NMF

We now describe our vector space information retrieval model which incorporates NMF-based dimensionality reduction. To apply this NMF to dimensionality reduction of a term-by-document matrix in the information retrieval, we attempt to regard the term-by-document matrix as the data matrix V of the NMF. The following is a resume of the information retrieval by using NMF-based dimensionality reduction:

1. Extract indexing terms from the entire document collection using an appropriate stop list and stemming algorithm. Let we have n indexing terms and m documents.
2. Create m document vectors $d_1, d_2, \ldots, d_m$, where d_j is the i-th component of document vector, a_{ij} is defined $a_{ij} = L_{ij} \times G_i$. Here, L_{ij} is the local weighting for the i-th term in document d_j, and G_i is the global weighting for the i-th term.
3. Apply the non-negative matrix factorization to the term-by-document matrix. The basis vectors W are computed by this process.
4. Project the document vectors onto new r-dimensional space. The columns of W form the axes of this space.
5. Using the same transformation, map a query vector into the r-dimensional space.
6. Calculate the similarity between transformed document vectors and a query vector.

4 Information Retrieval Experiments

4.1 Conditions

In experiments, we used the MEDLINE collection. This collection consists of thirty queries and 1,033 documents. The average number of relevant documents for each query was 23.2. We first preprocessed documents to eliminate non-content-bearing stopwords using a stop list of 439 common English words. Terms occurring in only one document were also removed. The remaining terms were then stemmed using the Porter algorithm[6]. The preprocessing step resulted in 4328 indexing terms.

There were two different types of term weighting: global weighting and local weighting. Local weighting was functioned to determine how many times each term appears in the entire collection. The d_{ij}, i-th element of the document vectors d_j was given by

$$d_{ij} = L_{ij} \cdot G_i, \tag{9}$$

where, L_{ij} is the weight for term i in the document d_j, G_j is the global weight for term i. As a term weighting scheme, we used log-entroy[7]:

Local weight:

$$L_{ij} = log(1 + f_{ij}), \tag{10}$$

Global weight:

$$G_i = 1 + \sum_j \frac{p_{ij} \cdot log(p_{ij})}{log(n)}, \tag{11}$$

where n is the number of documents in the collection, and f_{ij} indicates the frequency of the i-th term in the j-th document, and $p_{ij} = \frac{f_{ij}}{\sum_j f_{ij}}$.

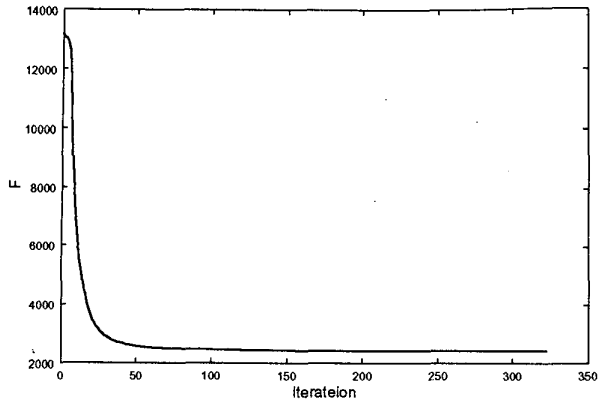
We used the random values from 0.0 to 1.0 for the initial values of decomposed matrices W and H

For the retrieval evaluation, we measured the non-interpolated average precision, which refers to an average of precision at various points of recall using the top fifty documents retrieved[6][8]. This score was calculated by using "trec_eval" program[9].

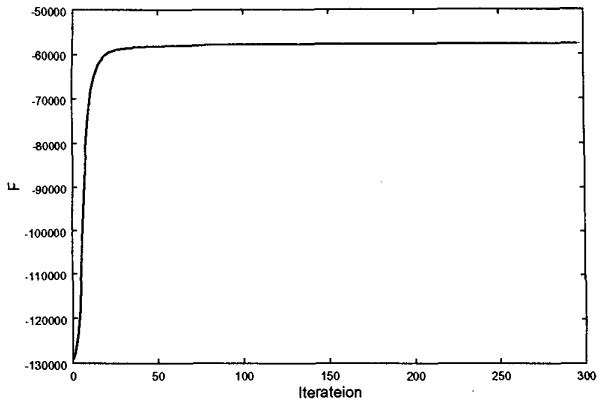
4.2 Experimental results

Figure 1 (a) shows that the cost of objective function F given by equation (5) as a function of number of iterations under the condition that the number of the reduced dimensions r was fixed to sixty hundred. Also, Figure 1 (b) shows the cost of objective function given by equation (6) as a function of number of iterations under same conditions. We could see from these figure that the cost of objective function F of the both update rules converged after only 20 iterations.

Next, Figure 2 shows the average precision as a function of reduced dimensions under the condition that the number of iterations was twenty, in which the

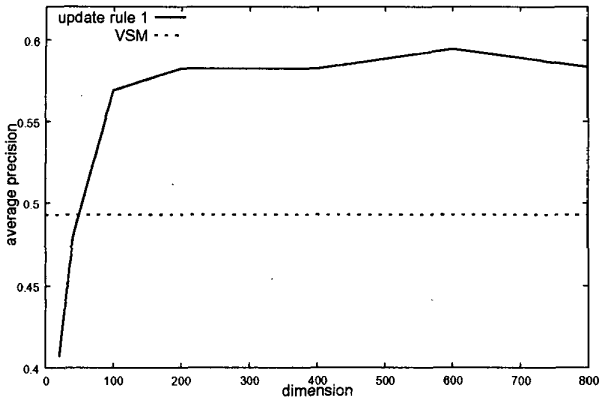


(a) Update rule 1

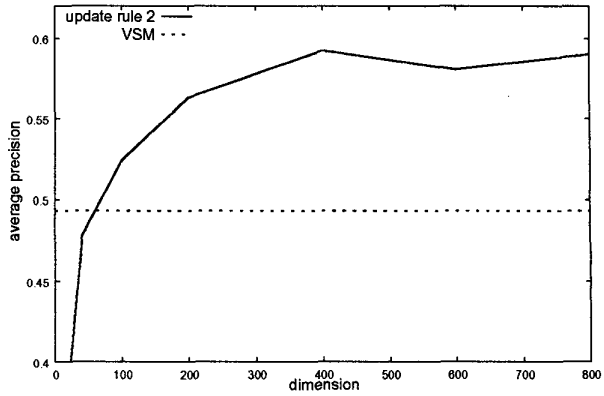


(b) Update rule 2

Figure 1: Cost of objective function as a function of number of iterations



(a) Update rule 1



(b) Update rule 2

Figure 2: Average precision as a function of dimensions

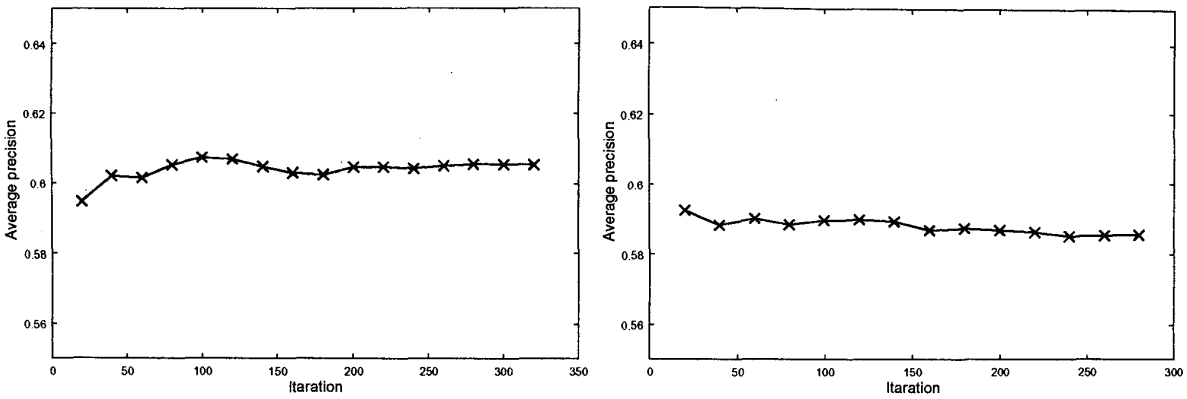
cost of the objective function converged. For comparison, the average precision of VSM is also shown in this figure. The dimensions of VSM equaled the number of indexing terms, 4328.

We could see from this figure that the each NMF which had more than 100 dimensions improved the performances compared with VSM. The best performance of the each NMF was achieved at 600 and 400 dimensions using update rule 1 and update rule 2, respectively. These performances were comparable to SVD with same dimensions. As a result, we could consider that the number of the semantic structure in this data was about 500.

We also could see from this figure that the average precision significantly degraded where the dimensions was less than 100. This result implied that only 100 basis vectors were not enough to capture the semantic structure in this data.

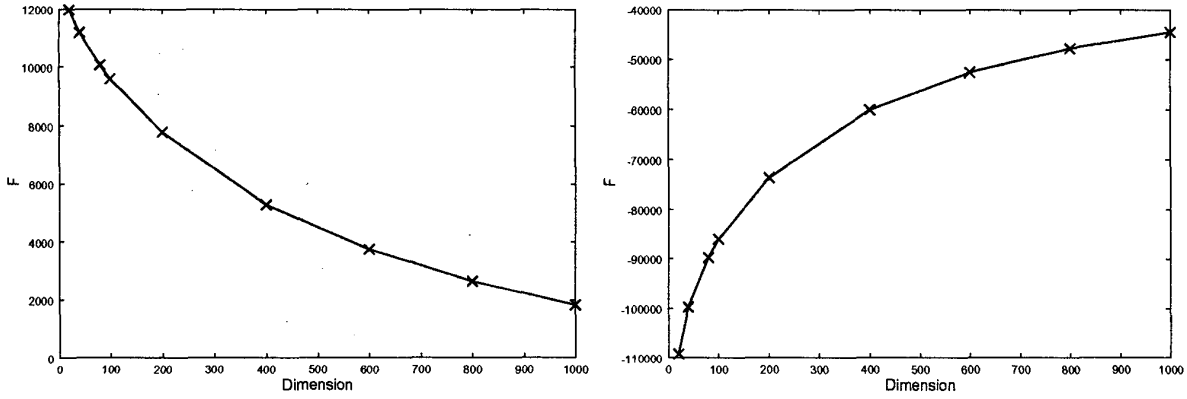
4.3 Discussions

In this section, we will discuss the experimental results described in Section **4.2**. First, we consider the relationship between the average precision and the number of iterations. Because the NMF is based on iterative updates of W and H, the similarity between V and WH depends on the number of iterations. Therefore, the number of iterations can be considered to affect the retrieval performance.



(a) Update rule 1 (b) Update rule 2

Figure 3: Average precision as a function of iterations



(a) Update rule 1 (b) Update rule 2

Figure 4: Cost of objective function as a function of dimensions

Figure 3 shows the average precision as a function of number of iterations. The reduced dimensions were fixed to 600 for update rule 1 and 400 for update rule 2. These dimensions gave the best performance in previous experiments (see Section **4.2**). In Figure 3, the left figure (a) and the right figure (b) indicate the average precision using update rule 1 and update rule 2, respectively.

We could see from these figures that the average precision almost remained same nevertheless the number of iterations increased. When the number of iterations were more than 20, these average precision curves were close to the cost curves which were shown in Figure 1. As a result, we could consider that the strong dependence existed between the average precision and the cost of objective function. Therefore, the cost of objective function could be considered to use for the restriction of iteration.

Next, we investigate the relationships between the cost of the objective function and the number of dimensions. Figure 4 shows the cost of the objective function as a function of dimensions under the condition that the number of iterations was fixed to 20.

We could see from these figures that the matrix product WH was close to the original data matrix V in proportion to increase the dimension. However, in Figure 2, we showed that the average precision could not be improved nevertheless the dimension increased.

As a result, the dependence was not observed between the cost of the objective function and the average precision when the number of dimensions were more that 300.

5 Conclusions

We have proposed a method for dimensionality reduction of the vector space information retrieval model using the Non-negative Matrix Factorization (NMF). The NMF decomposes a non-negative matrix, i.e., term-by-document matrix, into two non-negative matrices. One of the decomposed matrix can be regarded as the basis vectors. The dimensionality reduction can be performed by projecting the document vectors onto the lower dimensional space which is formed by these basis vectors.

Experimental results on the MEDLINE test collection showed the proposed method gave a better performance than the conventional vector space model. Therefore, we can conclude that the basis vectors, i.e., the columns of the decomposed matrix W, could discover the semantic structure that is latent in the data. We are now planning to analyze the basis vector in order to discover what kinds of semantic structure exist in them.

References

[1] G. Salton and J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill Book Company, 1983.

[2] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[3] D. Lee and H. Seung. Algorithms for non-negative matrix factorization. *NIPS 2000*, 2000.

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

[5] M. Berry, S. Dumais, and G. O'Brien. Using linear algebra for intelligent information retrieval. *Journal of the American Society for Information Science. SIAM Review*, 37((4)):573–595, 1995.

[6] I. Witten, A. Moffat, and T. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images.* Van Nostrand Reinhold, New York, 1994.

[7] E. Chisholm and T. Kolda. New term weighting formulas for the vector space method in information retrieval. *Technical Memorandum ORNL-13756*, 1999.

[8] D. Lewis. Evaluating text categorization. *Proc. of Speech and Natural Language Workshop*, pages 312–318, 1991.

[9] TREC homepage. http://trec.nist.gov/.