Information Retrieval System Using Concept Projection Based on PDDP algorithm

Minoru SASAKI and Kenji KITA
Department of Information Science & Intelligent Systems
Faculty of Engineering, Tokushima University
Tokushima, JAPAN, 770-8506
E-mail: {sasaki, kita}@is.tokushima-u.ac.jp
Tel. +81-886-56-7496
Fax. +81-886-56-7492

Abstract

Text documents are often represented as high-dimensional and sparse vectors using words as features in a multidimensional space. These vectors require a large number of computer resources and it is difficult to capture underlying concepts referred to by the terms. In this paper, we propose to use the technique of dimensionality reduction using Concept Vectors Based on PDDP algorithm as a way of solving these problems in the vector space information retrieval model. we give experimental results of the dimensionality reduction by using this method and show that this method is an improvement over conventional vector space model.

Keywords: Information retrieval, Vector space model, Latent semantic indexing, Concept vector, Concept projection, PDDP algorithm

1 Introduction

Recently, as the World Wide Web(WWW or Web) developed rapidly, a large collection of full-text documents in electronic form is available and opportunities to get a useful piece of information are increased. On the other hand, it becomes more difficult to get useful information from such giant amount of documents. Because of this, researches such as information retrieval, information filtering and text clustering have been studied actively all over the world.

In Vector Space Model (VSM) which is a conventional information retrieval model, individual documents and queries are represented as vectors in a multidimensional space. The basic idea is to extract indexing terms from a document collection. Numbers, punctuation and stop words are removed from the indexing terms. Each document is represented as a vector of weighted term frequencies such as IDF(Inverse Document Frequency)[3]. To compare a document and a query, the similarity is performed via the similarity between two vectors (e.g. cosine similarity). In VSM, document vectors exhibit the following two properties. First, in the case of indexing large text collections, the document vectors are high-dimensional since the size of the indexing terms is typically large. Second, the document vectors are very sparse since the number of terms in one document is typically far less than the total number of indexing terms in the text collections.

When applied to such high-dimensional and sparse vectors using words as features, it takes a lot of time to estimate the similarity between two vectors and a large amount of information storage is necessary for computing resources. Hence, dimensional space reduction of VSM is a way to overcome these problems. This method is to map a vector in high dimensional space to a much lower dimensional space vector using an arbitrary matrix. To capture underlying semantics and concepts referred to by the terms, the number of dimensional reduction techniques for information retrieval have been studied in statistical pattern recognition[7][8] and matrix algebra, such as Latent Semantic Indexing (LSI) using singular value decomposition (SVD) or semi-discrete matrix decomposition (SDD) [4][6]. LSI overcomes these problems by automatically finding latent relationships in a large text collection and improved its performance of information retrieval system over conventional VSM for several text collections. However, SVD is computationally expensive for a large text collection.

As a way of solving these problems of LSI, random projection[1] is proposed to reduce dimensions of document space faster than LSI. Random projection is the technique of projecting a set of vectors to a random lower dimensional space using the inner products. Examples of this technique include the application of solving the problems in VLSI (Very Large-Scale Integrated circuit) layout[11], theoretical descriptions of properties of the matrix obtained by dimensional reduction via random projection[1][9]. Concept projection we proposed in [10] is the method of dimensionality reduction using concept vectors to derive the axes of the reduced dimensional space. To obtain the concept vector, we applied a spherical k-means algorithm[5] which computes disjoint clusters quickly for high-dimensional and sparse document sets. However, these concept vectors obtained from the spherical k-means algorithm depend on the first random partitioning.

In this paper, we propose the information retrieval system applied a new idea of the technique of concept projection using PDDP algorithm[2] for dimensional reduction of vector space information retrieval model. The PDDP algorithm, which obtain the concept vector in our algorithm, is a unsupervised hierarchical clustering algorithm. To evaluate its efficiency, we performed results of retrieval experiments on the MEDLINE test collection. Experiments show that the concept projection is faster than LSI and achieves retrieval efficiency comparable to LSI.

2 Concept Projection based on PDDP algorithm

In the case of a very high-dimensional matrix, the SVD needs a high computational requirements to

compute singular values and reduce the dimension. Therefore, it is necessary to consider a method using only the random projection to obtain the effect on the LSI. To solve this problem, we suggest to apply a concept vector representing the contents of the document to the random projection. Our algorithm "concept projection", which is the technique of projecting document vectors to a lower dimensional space, computes some concept vectors in the same dimensional space as the document vectors by clustering similar documents and applying them to derive the axes of the lower dimensional conceptual space. In this section, we briefly describe the concept projection which is applied to the vector space model.

2.1 Concept Vectors

When a set of the vectors are mapped into the vector space, the vector is divided into some groups. Such group is called cluster and the cluster consists of a set of similar documents. The concept vector is obtained by calculating a normalized centroid of the cluster and represents typical contents of the cluster.

As a simple example to obtain concept vectors, we consider to cluster normalized vectors

$$\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N \tag{1}$$

to $s (s < N)$ disjoint clusters

$$\pi_1, \pi_2, \cdots, \pi_s . \tag{2}$$

Then the centroid vector $\mathbf{m}_j$ of the vectors contained in the cluster π_j is following:

$$\mathbf{m}_j = \frac{1}{n_j} \sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i \tag{3}$$

where n_j is the number of vectors in the cluster π_j. The centroid vector $\mathbf{m}_j$ is not a unit length so that the concept vector $\mathbf{c}_j$ is obtained by divided the centroid vector by the length of it.

$$\mathbf{c}_j = \frac{\mathbf{m}_j}{\|\mathbf{m}_j\|} \tag{4}$$

2.2 PDDP Algorithm

We present a summary of the PDDP Algorithm. PDDP algorithm is unsupervised hierarchical clustering for large-sized document set with some effective features. Some general hierarchical clustering algorithms employ bottom-up clustering which constructs a cluster hierarchy from bottom to top by merging two clusters at a time. PDDP algorithm constructs a cluster hierarchy for document set and employs top-down clustering which constructs a cluster hierarchy from one cluster to which all the documents belong and the clusters are disjoint at every stage.

Figure 1 shows the PDDP algorithm. The basic of this algorithm creates a binary tree. Each node in the binary tree has the information consisted of an index of documents in the node, the centroid vector of the node's cluster the highest singular value, pointers to the left and right children nodes and a scatter value as a measure of the non-cohesiveness of a cluster. The total scatter value is defined to be the Frobenius norm of the matrix

$$\mathbf{A} = \mathbf{M}_p - \mathbf{w}\mathbf{e}^T . \tag{5}$$

Therefore the scatter value of $\mathbf{A} = (a_{ij})$ is represented as

$$\|\mathbf{A}\|_F^2 = \sum_{i,j} |a_{ij}|^2 . \tag{6}$$

The scatter value is equal to the Frobenius norm of the covariance matrix $\mathbf{C}$ as well as the sum of the eigenvalues σ_i^2 of $\mathbf{C}$.

$$\|\mathbf{A}\|_F^2 = \|\mathbf{C}\|_F = \sum_i \sigma_i^2 . \tag{7}$$

The total scatter value is used to the cluster to split next in this algorithm.

The PDDP algorithm considers n-dimensional m document vectors whose element contains a weighted numerical value and initial term-document matrix

$$\mathbf{M} = (\mathbf{d}_1, \cdots, \mathbf{d}_m) \tag{8}$$

for the document vectors as an input. As another input value, the PDDP algorithm considers a desired number of clusters c_{max}. If the value of c_{max} is set

Procedure PDDP algorithm
begin
Input $n \times m$ word–document matrix $\mathbf{M}$ and a number of clusters c_{max}
Make a single root node for binary tree
For $c = 2, 3, \cdots, c_{max}$
Select node C with the largest scatter value
Make node L and node R which are pointers to left and right children of node C
Calculate $V_C = g(\mathbf{M}_C) \equiv \mathbf{u}_C^T(\mathbf{M}_C - \mathbf{we}^T)$
For each document i in the node C
If $v_i \leq 0$, then assign document i to node L
If $v_i > 0$, then assign document i to node R
If the number of leaf nodes is c_{max} or no divisible node in the binary tree,
return the binary tree
end

Figure 1: PDDP algorithm

and c_{max} clusters are found, the PDDP algorithm stops without partitioning in the next step and returns the binary tree. If the value of c_{max} is not set, the PDDP algorithm continues until the leaf node of the binary tree contains the only one document.

In the process of iteration of the PDDP algorithm, a partition of p documents is considered to split the partition. the $n \times p$ term-document matrix

$$\mathbf{M}_p = (\mathbf{d}_1, \cdots, \mathbf{d}_p) \tag{9}$$

is sub-matrix of the initial matrix $\mathbf{M}$ consisting of some selection of p columns of $\mathbf{M}$. The principal directions of the matrix $\mathbf{M}_p$ are the eigenvectors of the covariance matrix $\mathbf{C}$ of the matrix $\mathbf{M}_p$. The covariance matrix $\mathbf{C}$ is

$$\mathbf{C} = (\mathbf{M}_p - \mathbf{we}^T)(\mathbf{M}_p - \mathbf{we}^T)^T, \tag{10}$$

where

$$\mathbf{w} = \mathbf{M}_p\mathbf{e}/p \tag{11}$$

is the mean of the document $d_1, \cdots, d_p$. Each document vector is projected onto the leading eigenvector which is represented as the principal component and principal direction of $\mathbf{C}$.

The i-th document vector $\mathbf{d}_i$ is projected onto the leading eigenvector $\mathbf{u}$ as follows:

$$v_i = \mathbf{u}^T(\mathbf{d}_i - \mathbf{w}) \quad (1 \leq i \leq p), \tag{12}$$

where $v_1, \cdots, v_p$ is used to determine the splitting for the cluster $\mathbf{M}_p$. The document $\mathbf{d}_i$ is classified according to the corresponding v_i's sign. If the value v_i of the document i is not more than 0, the document i is classified into the left child. If the value v_i is more than 0, the document i is classified into the right child.

2.3 Concept Projection

As a preparation, the k concept vectors $\mathbf{r}_1, \cdots, \mathbf{r}_k$ are provided for the concept projection. The concept vector is obtained by calculating a normalized centroid of the cluster and represents typical contents of the cluster. To project a n-dimensional vector $\mathbf{u}$ to lower k-dimensional space using the concept vectors, this algorithm computes a k-dimensional vector $\mathbf{u}'$ which equals to the inner products:

$$u'_1 = \mathbf{r}_1 \cdot \mathbf{u}, \cdots, u'_k = \mathbf{r}_k \cdot \mathbf{u}. \tag{13}$$

Therefore, k-dimensional vector $\mathbf{u}$ is given by

$$\mathbf{u}' = (u'_1, \cdots, u'_k). \tag{14}$$

These calculations of the projection can be written in matrix notation to a simple computation task. The matrix $\mathbf{R}$ is a $n \times k$ matrix such that the i-th column of $\mathbf{R}$ corresponds to u'_i. Then, the projection to k-dimensional space is given by

$$\mathbf{u}' = \mathbf{R}^T \mathbf{u}. \tag{15}$$

If the matrix $\mathbf{R}$ is an arbitrary orthonormal matrix, this projection has the property of approximately preserving the distances between vectors[1][9].

3 Experiments

3.1 Efficiency of the Concept Vectors using PDDP Algorithm

In this section, we describe our vector space information retrieval model using the concept projection and experimentally evaluate the efficiency of the model using the MEDLINE collection. The MEDLINE collection consists of 1033 documents from medical journals and 30 queries and relevancy judgments of the queries. We first preprocessed all the documents in the MEDLINE collection to remove all the stop words using a stop list of 439 common English words such as "a" or "about". We also remove words occurring in only one document after the stop word elimination. The remaining words were stemmed using the Porter algorithm and 4329 index terms are obtained as a result of the preprocessing.

When each document is represented by a vector, the elements of a document vector d are assigned two-part values $d_{ij} = L_{ij} \times G_i$. In the experiments, the factor L_{ij} is a local weight that reflects the weight of term i within document j and the factor G_i is a global weight that reflects the overall value of term i as an indexing term for the entire document collection as follows:

$$L_{ij} = \begin{cases} 1 + \log f_{ij} & (f_{ij} > 0) \\ 0 & (f_{ij} = 0) \end{cases} \tag{16}$$

$$G_i = 1 + \sum_{j=1}^{n} \frac{\frac{f_{ij}}{F_i} \log \frac{f_{ij}}{F_i}}{\log n} \tag{17}$$

where n is the number of documents in the collection, f_{ij} is the frequency of the i-th term in the j-th document, and F_i is the frequency of the i-th term throughout the entire document collection.

From the document vectors, the fixed numbers of concept vectors are obtained by using the PDDP algorithm and the dimension of each of document vectors and query vectors is reduced by the concept projection. Similarity is calculated by inner product between the reduced vectors and the top 50 documents of the similarity are outputed as a retrieval result. Figure 2 shows the recall-precision curve of this experiment. In this figure, 'PDDP' is a retrieval result for the information retrieval model reduced to 88 dimensions by using the concept projection based on the PDDP algorithm. 'VSM' is a conventional information retrieval model, 'LSI100' is an information retrieval model of rank-100 approximation by LSI and 'k-means' is an information retrieval model of rank-500 approximation by using the spherical k-means algorithm.

The result of this experiment shows that the information retrieval model using the concept projection based on the PDDP algorithm improves the retrieval performance vastly in comparison with the conventional information retrieval model and the spherical k-means algorithm. The concept projection based on the PDDP algorithm has the nearly equal performance to the information retrieval model using LSI.

3.2 Learning Speed of the Concept Projection

In this section, we describe results which measured the processing time to construct a information retrieval model and processing time when it is required for retrieval for one query. In our experiments, we use Sun's UNIX workstation with Ultra Sparc (clocked at 250 MHz) in CPU and 640 MB main memory. In the experiment of an information retrieval model of rank-500 approximation by using the spherical k-means algorithm, processing time to construct an information retrieval model is required for about 2 minutes as shown in Table 1. In

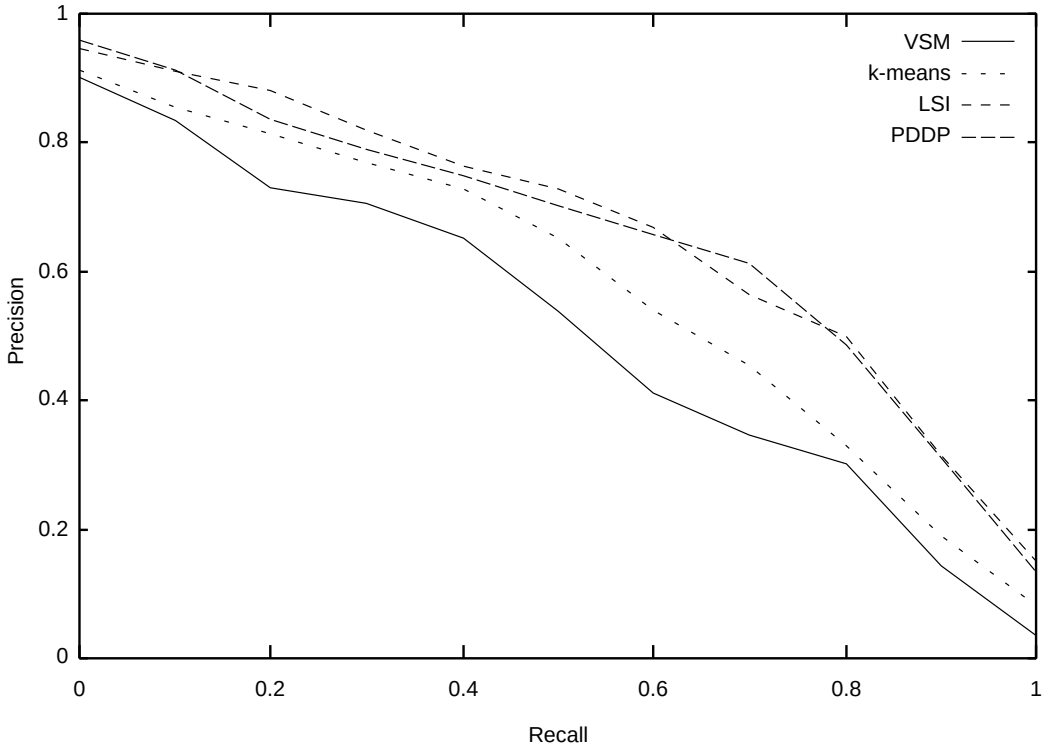

Figure 2: Recall-Precision curve for comparison between models

the experiment of the concept projection based on the PDDP algorithm, documents are first clustered using this algorithm as preparation. As the result which clustered without specifying the number of clusters, 88 clusters are obtained. Dimensionality reduction based on the concept projection is performed with the number of dimension equal to the number of these clusters, so that processing time to construct an information retrieval model is required for about 3 minutes. In the experiment of LSI using SVD, we use Lanczos method which is highest speed of any other methods in SVDPACK in order to calculate SVD of a matrix. As the result of the experiment using an information retrieval model of rank-500 approximation based on LSI, processing time to construct an information retrieval model is required for about 24 minutes. Judging from these results, the concept projection can construct an information retrieval model faster than LSI.

In this processing time to construct a information retrieval model, The size of main memory affects the processing time to compute SVD of a matrix. When an experiment is conducted using very large scale set of documents, there needs very much quantity of swap space in which a numerical values are stored so that it is possible for the processing time to become vary slow. However, the UNIX workstation used in our experiment has 640 MB main memory so that it is considered that there is almost no influence of memory size.

4 Conclusion

In this paper, we propose to use the technique of concept projection based on PDDP algorithm for dimensionality reduction of vector space information retrieval model. we give experimental results of the dimensionality reduction by using this method and show that this method is an improvement over conventional vector space model. In comparison

Table 1: Speed Comparisons

Method	Dimension	Learning Speed	Retrieval Speed
k-means	500	about 2 min.	4 sec.
PDDP	88	about 3 min.	4 sec.
LSI	500	about 24 min.	4 sec.

with the LSI, we found that the concept projection has the nearly equal retrieval performance. Further work could involve analyzing the other clustering algorithm to obtain the better concept vectors.

References

[1] R. I. Arriaga and S. Vempala. An algorithmic theory of learning: Robust concepts and random projection. In *Proceedings of the 40th Foundations of Computer Science*, pages 616–623, 1999.

[2] D. L. Boley. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4):325–344, 1998.

[3] E. Chicholm and T. G. Kolda. New term weighting formulas for the vector space method in information retrieval. Technical report, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 1998.

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

[5] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. Technical report, IBM Almaden Research Center, 1999.

[6] S. T. Dumais. Using lsi for information filtering: Trec-3 experiments. In *D. Harman (Ed.), Overiew of The Third Text REtrieval Conference (TREC3) National Institute of Standards and Technology Special Publication*, pages 219–230, 1995.

[7] C. Faloutsos and K. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 163–174, 1995.

[8] T. G. Kolda and D. P. O'Leary. A semidiscrete matrix decomposition for latent semantic indexing in information retrieval. In *Proceedings of ACM Transaction on Information Systems*, volume 16, pages 322–346, 1998.

[9] C. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the 17th ACM Symposium on the Principles of Database Systems*, pages 159–168, 1998.

[10] M. Sasaki and K. Kita. Dimensionality reduction of vector space information retrieval model based on random projection. *Journal of Natural Language Processing*, 8(1):1–19, 2001.

[11] S. Vempala. Random projection: A new approach to vlsi layout. In *Proceedings of the 39th Foundations of Computer Science, Palo Alto*, pages 389–395, 1998.