Web Document Clustering Using Threshold Selection Partitioning

Minoru Sasaki
Department of Computer and Information Sciences,
Faculty of Engineering, Ibaraki University.
4-12-1 Nakanarusawa, Hitachi, Ibaraki 316-8511, Japan
sasaki@cis.ibaraki.ac.jp

Hiroyuki Shinnou
Department of Systems Engineering,
Faculty of Engineering, Ibaraki University.
4-12-1 Nakanarusawa, Hitachi, Ibaraki 316-8511, Japan
shinnou@dse.ibaraki.ac.jp

Abstract

Clustering techniques have been applied to categorize documents on World Wide Web. In previous research, PDDP (Principal Direction Divisive Partitioning) is a well-known clustering algorithm. PDDP algorithm employs top-down and unsupervised clustering based on the principal component analysis and splits documents into two sets using a plane perpendicular to the maximum principal direction passing through the centroid of the documents. However, in case that the distribution of documents is biased, this algorithm difficult to split into two subsets accurately. In this paper, we propose a new clustering algorithm that improved the separation of the two sets considerably. We give experimental results using our clustering algorithm on the NTCIR-4 Web task.

Keywords: *Divisive Clustering, Principal Direction, Threshold Selection*

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. This causes that researches such as information retrieval, information filtering and text clustering have been studied actively all over the world.

Clustering is the unsupervised classification of data set to reduce the amount of data by categorizing or grouping similar data items together. To apply the clustering techniques, each document is usually represented as a vector of weighted term frequencies such as Text Frequency (TF) and Inverse Document Frequency (IDF) [11] [3]. For these vectors, it is necessary to calculate a similarity or distance measure that clustering algorithm defines between two vectors. From these calculations, a pair of the closest points is merged into a new single cluster. This merge process is repeated until a stopping criterion is satisfied.

Many clustering algorithms have been proposed for the data points [7] based on the vector space model [12]. Most of these conventional clustering methods can be divided into two types. The first one is non-hierarchical clustering. This type is used to separate data set into distinct clusters without defining the relationships between the clusters. There are many methods in this type such as cover-coefficient clustering algorithm[2] and k-means algorithm[5]. The another one is hierarchical clustering to builds a tree structure from data set. According to tree construction, hierarchical clustering methods can be also categorized into two types: agglomerative (bottom-up) clustering and partitional (top-down) clustering. In agglomerative clustering methods, single link method [10] begins with each data in a cluster and proceed successively by merging the closest clusters into larger ones. Partitional clustering method, on the other hand, begins with all data in a single cluster and proceed successively by splitting a cluster into two disjoint clusters. This partitioning process repeats until the user-established number of clusters is reached or all the clusters do not divide any longer.

As the method in the partitional clustering algorithm, the Principal Direction Divisive Partitioning (PDDP) Algorithm [1] is proposed recently. This method constructs a cluster hierarchy for data set and employs top-down clustering based on the principal

component analysis. Its performance is compared with other clustering methods [4] and some improvements are also implemented [14]. However, in case that the distribution of documents is biased, this algorithm difficult to split into two subsets accurately. In this paper, we propose a new clustering algorithm that improved the separation of the two sets considerably. Using our clustering algorithm, we give experimental results using this model on the NTCIR-4 Web task [1] .

2 PDDP Algorithm

We present a summary of the PDDP algorithm [1] . 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.

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

$$\mathbf{A} = \mathbf{M}_p - \mathbf{w}\mathbf{e}^T. \tag{1}$$

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

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

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 $\mathbf{C}$.

$$\|\mathbf{A}\|_F^2 = \|\mathbf{C}\|_F = \sum_i \sigma_i^2. \tag{3}$$

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

$$\mathbf{M} = (\mathbf{d}_1, \cdots, \mathbf{d}_m) \tag{4}$$

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 and c_{max}

[1] http://research.nii.ac.jp/ntcweb/index.html

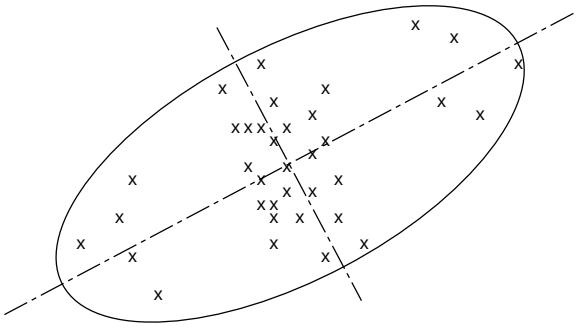

Figure 1. data set aggregated near the centroid of cluster

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

$$\mathbf{M}_p = (\mathbf{d}_1, \cdots, \mathbf{d}_p) \tag{5}$$

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

$$\mathbf{C} = (\mathbf{M}_p - \mathbf{w}\mathbf{e}^T)(\mathbf{M}_p - \mathbf{w}\mathbf{e}^T)^T, \tag{6}$$

where

$$\mathbf{w} = \mathbf{M}_p\mathbf{e}/p \tag{7}$$

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:

$$v_i = \mathbf{u}^T(\mathbf{d}_i - \mathbf{w}) \quad (1 \le i \le p), \tag{8}$$

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.

Procedure Threshold Selection Partitioning Algorithm
begin
Input $n \times m$ word–document matrix $\mathbf{M}$ and a number of clusters
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 $V_c = g(M_C) \equiv u_C^T(M_C - we^T)$
Calculate the threshold k to partition the target cluster
For each document i in the node C
If $v_i \leq k$, then assign document i to node L
If $v_i > k$, 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 3. Threshold Selection Partitioning Algorithm

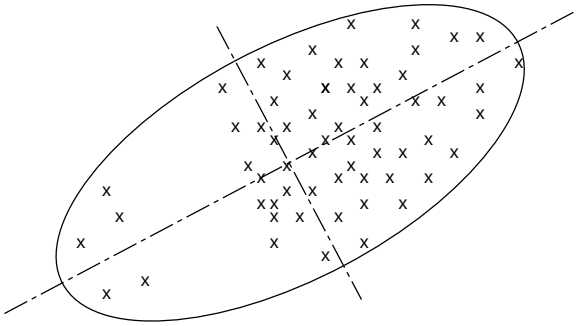

Figure 2. biased data set in a cluster

3 Threshold Selection Partitioning Algorithm

PDDP algorithm employs top-down and unsupervised clustering based on the principal component analysis and splits documents into two sets using a plane perpendicular to the maximum principal direction passing through the centroid of the documents.

However, in case that the distribution of documents is biased, the PDDP algorithm is difficult to split into two clusters accurately. This causes that the cluster is partitioned by the plane perpendicular to the maximum principal direction passing through the centroid of the cluster. For example, we consider that there are data points aggregated near the centroid of cluster displayed in figure 1. In this case, although the cluster consists of some clusters, documents with similar contents are partitioned into discrete clusters. Thus, it is possible to obtain two clusters with similar contents. For another example, we consider that there are data points biased in a cluster displayed in figure 2. The cluster consists two subset: a smaller data set and a larger data set in the good portion of the cluster. In this case, it is difficult to partition the cluster into two clusters as well as the former example.

To solve this problem in the PDDP algorithm, we propose a new clustering algorithm that improved the separation of the two sets considerably. Our algorithm employs a hierarchical clustering as well as the PDDP algorithm so that constructs a cluster hierarchy for data set. Our algorithm also employs partitional clustering method so that the algorithm begins with all data in a single cluster and proceed successively by splitting a cluster into two disjoint clusters.

Figure 3 shows our Threshold Selection Partitioning (TSP) algorithm. In the process of iteration of the TSP 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 calculate the threshold k to partition the target cluster and 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 k, the document i is classified into the left child. If the value v_i is more than k, the document i is classified into the right child.

3.1 Threshold Selection Method

For the principal vector obtained by TSP algorithm, Threshold value k is calculated by using the distribution of data points in the cluster to divide the cluster into two clusters. When the threshold value k is obtained, the cluster is partitioned by the plane perpendicular to the maximum principal direction passing through the threshold k. To calculate the optimum threshold value k, we apply the threshold selection method based on the pattern recognition [9] . This method is often used for digital image processing techniques such as processing binarization of grayscale image to binary image and extracting a target object image from a image.

As a simple example to calculate the threshold value k in the figure 4, we consider the problem to divide a set of N data into two clusters using the threshold k. The set of data is represented as follows:

$$\{d_1, d_2, \cdots, d_i, \cdots, d_N\} \ \ (1 \leq i \leq N). \tag{13}$$

We consider that a set of data is divided into two clusters using the threshold value k: C_1 and C_2. If the number of data in the cluster C_1 is n_1 and the number of data in the cluster C_2 is n_2, the probabilities ω_1 and ω_2 of the cluster C_1 and C_2 are represented respectively as follows:

$$\omega_1 = P(C_1) = \frac{n_1}{N}, \tag{14}$$

$$\omega_2 = P(C_2) = \frac{n_2}{N} = 1 - \omega_1. \tag{15}$$

If the point at the intersection of principal vector and the perpendicular line which the data d_i drops is defined as h_i, expectation values μ_1 and μ_2 of each clusters are represented as follows:

$$\begin{aligned} \mu_1 &= \sum_{d_i \in C_1} h_i P(d_i|C_1) \\ &= \frac{\sum_{d_i \in C_1} h_i}{n_i}, \end{aligned} \tag{16}$$

$$\begin{aligned} \mu_2 &= \sum_{d_i \in C_2} h_i P(d_i|C_2) \\ &= \frac{\sum_{d_i \in C_2} h_i}{n_i}. \end{aligned} \tag{17}$$

Moreover, variances σ_1 and σ_2 can be also represented as follows:

$$\begin{aligned} \sigma_1 &= \sum_{d_i \in C_1} (h_i - \mu_1)^2 P(d_i|C_1) \\ &= \frac{\sum_{d_i \in C_1} (h_i - \mu_1)^2}{n_i}, \end{aligned} \tag{18}$$

$$\begin{aligned} \sigma_2 &= \sum_{d_i \in C_2} (h_i - \mu_2)^2 P(d_i|C_2) \\ &= \frac{\sum_{d_i \in C_2} (h_i - \mu_2)^2}{n_i}. \end{aligned} \tag{19}$$

From these formulas we obtain an average variance of these clusters σ_W^2, an average variance between these clusters σ_B^2 and an average of all data σ_T^2.

$$\sigma_W = \omega_1 \sigma_1^2 + \omega_2 \sigma_2^2 \tag{20}$$

$$\begin{aligned} \sigma_B &= \omega_1(\mu_1 - \mu_T)^2 + \omega_2(\mu_2 - \mu_T)^2 \\ &= \omega_1 \omega_2 (\mu_1 - \mu_2)^2 \end{aligned} \tag{21}$$

$$\sigma_T^2 = \sigma_W^2 + \sigma_B^2 \tag{22}$$

Evaluation measure of the threshold k calculated by the target data set is defined as

$$\eta = \frac{\sigma_B^2}{\sigma_T^2}. \tag{23}$$

In this formula, the average of all data σ_T^2 is a constant value independent of the threshold k. Thus the optimum threshold k^* can be calculated by maximizing the average variance between these clusters σ_B^2 as follows:

$$\sigma_B^2(k^*) = \max_{L \leq k \leq M} \sigma_B^2(k) \tag{24}$$

4 Experiment

In this section, we experimentally evaluate the efficiency of our clustering system with the TSP algorithm using a Web test collection in the NTCIR-4 Web task.

4.1 Test Collection

In the NTCIR-4 Web task, to evaluate an efficiency of clustering results, we apply the test collection "NW100G-01" [6] . This collection "NW100G-01" has 100GB web documents crawled in 2001 in '.jp' domain. In the task A (information retrieval task), the top 200 web documents in retrieval results for given 47 queries are extracted from this test collection. The obtained 200 web documents are the target data for the task D (topic classification task).

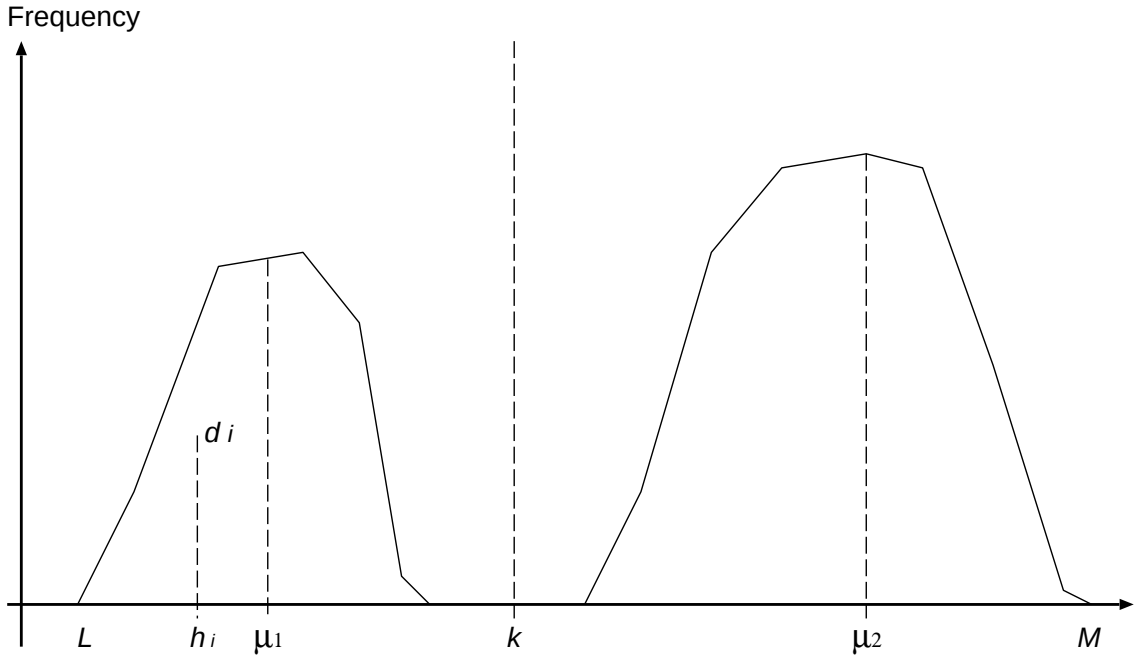

Figure 4. distribution of the frequencies projected to the principal vector

4.2 Term Weighting Method

In our experiment, each document is represented as a vector for the 200 web documents retrieved by one query. To make a document vector, first, we remove HTML tags from each document. By this preprocessing, all the web documents are converted from HTML files to plain text files. However, It is possible that words or phrases that links with other sites are contained in HTML tags such as link pages and HTML documents using Javascript. When the text contains no content words by removing HTML tags, we add words that links with other sites to the text by hand. For the obtained plain text, morphological analysis is performed automatically using Chasen to extract noun words. By this noun extraction, these noun words are defined as index words.

When each document is represented by a vector, the elements of a document vector d are assigned two-part values

$$d_{ij} = L_{ij} \times G_i. \tag{25}$$

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 [3] as follows:

$$L_{ij} = \begin{cases} 1 + \log f_{ij} & (f_{ij} > 0) \\ 0 & (f_{ij} = 0) \end{cases} \tag{26}$$

$$G_i = 1 + \sum_{j=1}^{n} \frac{\frac{f_{ij}}{F_i} \log \frac{f_{ij}}{F_i}}{\log n} \tag{27}$$

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.

4.3 Clustering

By this preprocessing described in previous subsection, a term-document matrix is obtained for the 200 Web documents. Then these web documents can be categorized by the TSP algorithm. In this clustering, TSP algorithm requires users to specify the number of clusters. However, in our experiment, we execute the TSP algorithm repeatedly by making a null leaf node. Calculating the threshold value k to divide a cluster in the TSP algorithm, The valley of the distribution does not exist in the cluster. That is why the partitioning plane is located at the end of the target cluster.

4.4 Evaluation Method

The NTCIR-4 topical classification task evaluates the techniques for clustering the highly ranked retrieval results efficiently. For each query, the top 200 web documents retrieved by information retrieval task are categorized into some clusters. The obtained clusters are sorted according to the number of the relevant Web documents for the query to obtain the sorted list of the documents. From this sorted list, the list of the top 20 Web documents is extracted and is used to evaluate the efficiency of the system.

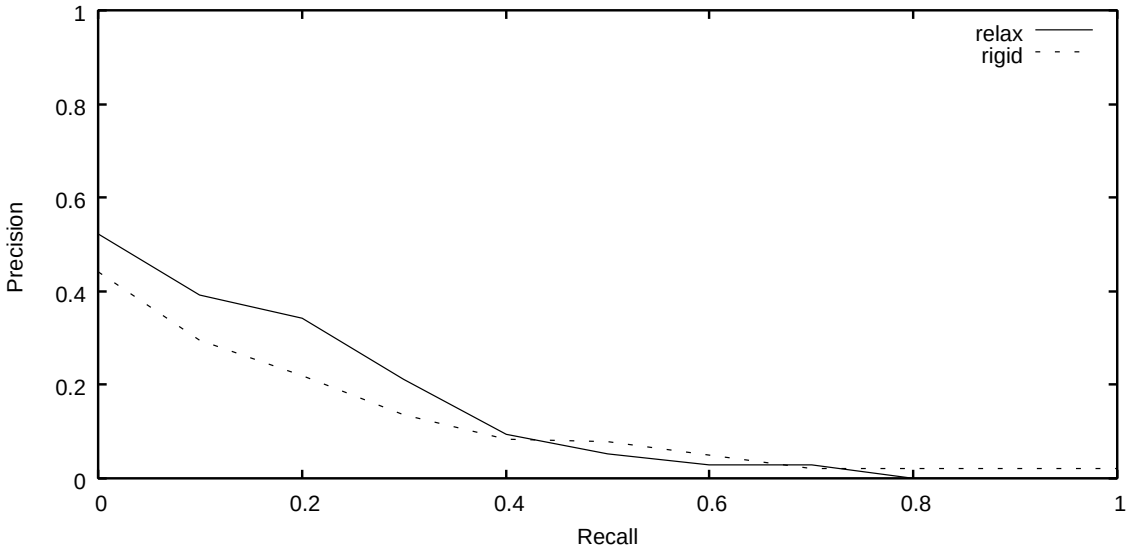

Figure 5. Recall-Precision Curve

To judge the relevant documents, the NTCIR-4 Web task defines four level relevance as follows:

- S : Highly Relevant
- A : fairly Relevant
- B : partially Relevant
- C : Irrelevant

The number of the relevant Web documents are calculated from the obtained the top 20 documents. To evaluate the performance of our system, we use the evaluation metrics to measure performance in terms of recall and precision [8] [13].

$$\text{Recall} = \frac{\text{Number of relevant docs. outputted}}{\text{Total number of docs. outputted}}, \quad (28)$$

$$\text{Precision} = \frac{\text{Number of relevant docs. outputted}}{\text{Total number of relevant docs.}}. \quad (29)$$

These evaluation metrics of text clustering system are possible to use recall or precision individually. In this experiments, evaluation of the ranked output system results in a 11-pointed recall-precision curve generally, with points plotted that represent precision at various recall percentages. Typically, as average performance over a large set of queries, Average precision at each standard recall level across all queries is computed.

4.5 Experimental Results

Figure 5 shows results of our experiment of the clustering system using the TSP algorithm. In this figure, the full curve "relax" shows the average recall-precision curve in case that the relevant documents are considered as the documents labeled 'S', 'A' and 'B' from the four level relevance. The broken curve "rigid" shows the average recall-precision curve in case that the relevant documents are considered as the documents labeled 'S' and 'A'. As seen in this graph, efficiency of our system is lower than expected results because we improve only the partitioning method in the PDDP algorithm. However, we are not able to evaluate the efficiency using the TSP algorithm so that it is necessary to compare the efficiency with the other clustering algorithms such as the PDDP algorithm. These evaluation is the most important issue in the future.

In this experiment, as well as the vector space model based on the information retrieval, we use only noun words as elements of document vector. But, almost HTML files in the test collection contain HTML tags and these tags have an important information such as link pages and page title and so on. It is necessary to embed these HTML tags information in vector space or define an effective term weighting scheme for the HTML tags.

5 Conclusion

In this paper, we propose a new clustering algorithm that improved the separation of the two sets considerably and names the TSP algorithm. Using the Web document clustering system based on the TSP algorithm, we give experimental results using this model on the NTCIR-4 Web task. In this experiment, efficiency of our system is lower than expected results because we improve only the partitioning method in the PDDP algorithm. However, we are not able to evaluate the efficiency using the TSP algorithm so that it is nec-

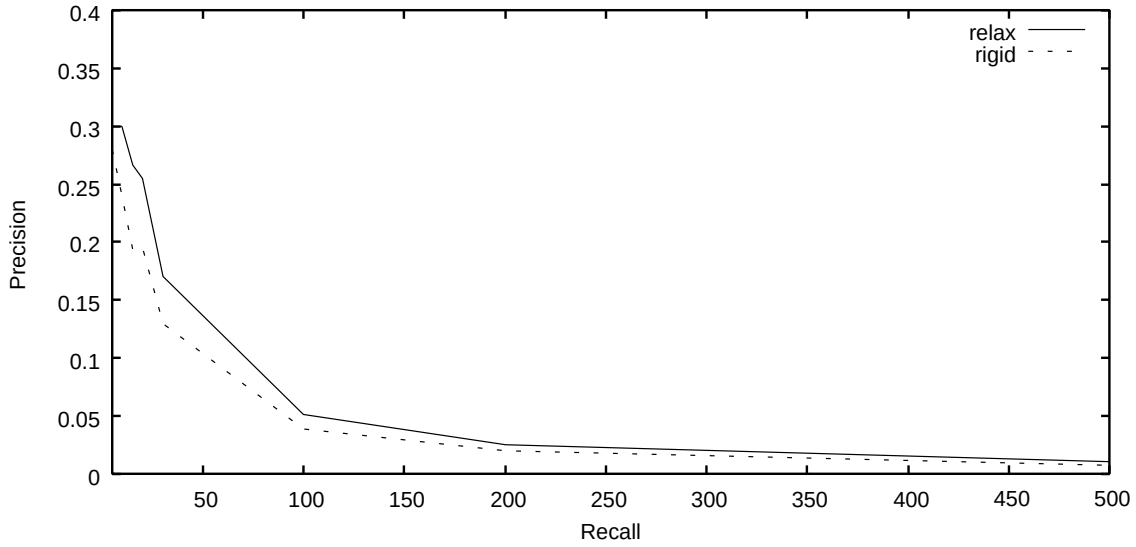

Figure 6. Documents-Precision Curve

essary to compare the efficiency with the other clustering algorithms such as the PDDP algorithm. These evaluation is the most important issue in the future. Moreover, almost HTML files in the test collection contain HTML tags and these tags have an important information such as link pages and page title and so on. It is necessary to embed these HTML tags information in vector space or define an effective term weighting scheme for the HTML tags.

References

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

[2] F. Can and E. A. Ozkarahan. Concepts and effectiveness of the cover-coefficient-based clustering methodology for text databases. *ACM Trans. Database Syst.*, 15(4):483–517, 1990.

[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] I. Dhillon, J. Kogan, and C. Nicholas. *Feature Selection and Document Clustering*. (M. Berry ed.) Springer-Verlag, New York, 2003.

[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] K. Eguchi, K. Oyama, E. Ishida, N. Kando, and K. Kuriyama. An evaluation of the web retrieval task at the third ntcir workshop. *SIGIR Forum*, 38(1):39–45, 2004.

[7] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

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

[9] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, Mar. 1979.

[10] E. Rasmussen. Clustering algorithms. *Information retrieval: data structures and algorithms*, pages 419–442, 1992.

[11] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Journal of Information Processing and Management*, 24(5):513–523, 1988.

[12] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

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

[14] S. Xu and J. Zhang. Clustering web document sets with different closeness. Technical report, Department of Computer Science, University of Kentucky, 2004.