Topic Difference Factor Extraction between Two Document Sets and its Application to Text Categorization

Takahiko Kawatani
Hewlett-Packard Labs Japan
3-8-13, Takaido-Higashi, Suginami-Ku, Tokyo, 168-0072 Japan
takahiko_kawatani@hp.com

ABSTRACT

To improve performance in text categorization, it is important to extract distinctive features for each class. This paper proposes topic difference factor analysis (TDFA) as a method to extract projection axes that reflect topic differences between two document sets. Suppose all sentence vectors that compose each document are projected onto projection axes. TDFA obtains the axes that maximize the ratio between the document sets as to the sum of squared projections by solving a generalized eigenvalue problem. The axes are called topic difference factors (TDF's). By applying TDFA to the document set that belongs to a given class and a set of documents that is misclassified as belonging to that class by an existent classifier, we can obtain features that take large values in the given class but small ones in other classes, as well as features that take large values in other classes but small ones in the given class. A classifier was constructed applying the above features to complement the kNN classifier. As the results, the micro-averaged F_1 measure for Reuters-21578 improved from 83.69 to 87.27%.

Categories & Subject Descriptors: H.3.m [**Miscellaneous**].

General Terms: Algorithms, Experimentation, Performance.

1. INTRODUCTION

Recently text categorization research has become more and more popular. According to Yang's comparative study of classifiers [1], k nearest neighbors (kNN)[1,2,3], support vector machines (SVM)[1,4], and Linear Least Squares Fit (LLSF)[5] outperform other methods proposed so far. Adaboost also achieves high performance [6]. However, because these methods have been studied in depth, increasing performance by improving individual methods seems difficult. A new approach is necessary.

Every classifier has information about document classes, and compares it with an input document. We call this information the class model. The class model is the average vector of documents belonging to the same class in Rocchio's model [7], the set of documents belonging to the same class in kNN, and a set of simple hypotheses in Adaboost. The class model needs to be precise to enable high performance. However, many classifiers, even precise class models, do not consider class-model overlapping. In most classifiers the class model of a certain class shares information with other classes. If an overlap exits in class models, unnecessary likelihood can be generated for classes an input document does not belong to. Therefore, class-model overlapping may cause misclassification. To prevent misclassification, a class model should be described using each class's distinctive information so that class model overlapping is reduced.

This paper focuses on this issue. To clarify each class's distinctive information, it is essential to extract features that reflect differences between a document set of a given class and a document set that belongs to other classes.

First, this paper proposes a method to extract projection axes that reflect topic differences between two document sets. Suppose that each document is represented as a set of sentence vectors, whose components represent frequency related values of corresponding terms, and that all sentence vectors are projected onto projection axes. The projection axes are obtained so that the ratio between the document sets as to the sum of the squared projections is maximized. By projecting the sentence vectors onto the projection axes, we can obtain the features that take large values in one document set but small ones in the other. Here, a feature denotes a linear combination of terms. Since the projection axis reflects the topic difference between the document sets, we call it a topic difference factor (TDF), and the method topic-difference-factor analysis (TDFA). By applying TDFA to text categorization, we can obtain features that take large values in a given class but small ones in other classes, and features take small values in the given class but large ones in other classes. This paper proposes a classification scheme that applies the features in a

SIGIR'02, August 11-15, 2002, Tampere, Finland.

complementary classifier to an existent one. The proposed complementary classifier corrects the likelihood scores of an input document that are assigned to each class by an existent classifier.

The rest of this paper is organized as follows. Section 2 describes how TDF's are obtained and how TDFA is interpreted. What TDF's are obtained is also illustrated for simple examples of sentence vectors. Section 3 shows how TDF's are obtained for the complementary classifier and how likelihood scores are corrected. Section 4 presents experimental results using Reuters-21578. They show that the micro averaged F_1 measure was significantly improved. Section 5 summarizes the paper.

2. TOPIC DIFFERENCE FACTOR (TDF)

2.1 Approach

We consider two document sets, $D=\{D_1,..,D_M\}$ and $T=\{T_1,..,T_N\}$. Let the k-th sentence vector in document D_m and T_n be $\boldsymbol{d}_{mk}$ ($k=1,..,K_D(m)$) and $\boldsymbol{t}_{nk}$($k=1,..,K_T(n)$), respectively. Here, let $\boldsymbol{\alpha}$ be the projection axis to be obtained. We assume $\|\boldsymbol{\alpha}\|=1$. Let P_D and P_T be the sums of squared projections of all sentence vectors in document sets D and T onto $\boldsymbol{\alpha}$, respectively. They are obtained as follows. That is,

$$P_D = \sum_{m=1}^{M}\sum_{k=1}^{K_D(m)}(\boldsymbol{d}_{mk}{}^T\boldsymbol{\alpha})^2 = \boldsymbol{\alpha}^T\boldsymbol{S}_D\boldsymbol{\alpha}, \qquad (1)$$

$$P_T = \sum_{n=1}^{N}\sum_{k=1}^{K_T(n)}(\boldsymbol{t}_{nk}{}^T\boldsymbol{\alpha})^2 = \boldsymbol{\alpha}^T\boldsymbol{S}_T\boldsymbol{\alpha}, \qquad (2)$$

where T denotes transpose, and $\boldsymbol{S}_D$ and $\boldsymbol{S}_T$ are the matrices defined by

$$S_D = \sum_{m=1}^{M}\sum_{k=1}^{K_D(m)}\boldsymbol{d}_{mk}\boldsymbol{d}_{mk}{}^T, \qquad (3)$$

$$S_T = \sum_{n=1}^{N}\sum_{k=1}^{K_T(n)}\boldsymbol{t}_{nk}\boldsymbol{t}_{nk}{}^T. \qquad (4)$$

We call these matrices square sum matrices. The matrix divided by the number of sentences is the so-called autocorrelation matrix. Let $J(\boldsymbol{\alpha})$ be a criterion function which represents how strongly the differences between the document sets D and T are reflected on $\boldsymbol{\alpha}$. We define $J(\boldsymbol{\alpha})$ as

$$J(\boldsymbol{\alpha}) = \frac{P_D}{P_T} = \frac{\boldsymbol{\alpha}^T\boldsymbol{S}_D\boldsymbol{\alpha}}{\boldsymbol{\alpha}^T\boldsymbol{S}_T\boldsymbol{\alpha}}. \qquad (5)$$

Since the sum of squared projections of all sentence vectors onto the $\boldsymbol{\alpha}$ that maximizes $J(\boldsymbol{\alpha})$ should be large for document set D and small for T, the $\boldsymbol{\alpha}$ reflects information that appears frequently in document set D but rarely in T. In other words, it reflects distinctive information that should exist in document set D. Therefore, we call such an $\boldsymbol{\alpha}$ the positive topic difference factor (P-TDF) of document set D. Criterion function $J(\boldsymbol{\alpha})$ has the same form as that in linear discriminant analysis[8,10]. As in linear discriminant analysis, plural $\boldsymbol{\alpha}$'s are obtained as the eigenvectors of the following generalized eigenvalue problem.

$$\boldsymbol{S}_D\boldsymbol{\alpha} = \lambda\boldsymbol{S}_T\boldsymbol{\alpha}. \qquad (6)$$

Alternatively, $\boldsymbol{\alpha}$'s can be expressed as the eigenvectors of $\boldsymbol{S}_T^{-1}\boldsymbol{S}_D$.

Let $\boldsymbol{\beta}$ be another projection axis to be obtained. When $J(\boldsymbol{\beta})$ is defined as

$$J(\boldsymbol{\beta}) = \frac{P_T}{P_D} = \frac{\boldsymbol{\beta}^T\boldsymbol{S}_T\boldsymbol{\beta}}{\boldsymbol{\beta}^T\boldsymbol{S}_D\boldsymbol{\beta}}, \qquad (7)$$

information that appears frequently in document set T but rarely in D is reflected on the $\boldsymbol{\beta}$ that maximizes $J(\boldsymbol{\beta})$. The $\boldsymbol{\beta}$ becomes the P-TDF of document set T. We also call the $\boldsymbol{\beta}$ the negative topic difference factor (N-TDF) of document set D. In this case, plural $\boldsymbol{\beta}$'s are obtained as the eigenvectors of the following generalized eigenvalue problem.

$$\boldsymbol{S}_T\boldsymbol{\beta} = \lambda\boldsymbol{S}_D\boldsymbol{\beta}. \qquad (8)$$

In Eqs.(1) and (2), $\boldsymbol{d}_{mk}$ and $\boldsymbol{t}_{nk}$ might be replaced by their normalized forms, $\hat{\boldsymbol{d}}_{mk} = \boldsymbol{d}_{mk}/\|\boldsymbol{d}_{mk}\|$ and $\hat{\boldsymbol{t}}_{nk} = \boldsymbol{t}_{nk}/\|\boldsymbol{t}_{nk}\|$, respectively, to prevent influence from sentence-length variance. In this case, criterion $J(\boldsymbol{\alpha})$ or $J(\boldsymbol{\beta})$ represents the ratio between the document sets as to the sum of the squared cosine similarities of $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$ with all sentence vectors.

2.2 Interpretation

We consider the case of Eq.(6). Let λ_i and $\boldsymbol{\alpha}_i$ be the i-th eigenvalue and eigenvector, respectively. Since $\lambda_i=\boldsymbol{\alpha}_i^T\boldsymbol{S}_D\boldsymbol{\alpha}_i/\boldsymbol{\alpha}_i^T\boldsymbol{S}_T\boldsymbol{\alpha}_i$, λ_i represents the value of $J(\boldsymbol{\alpha}_i)$. It is known that the eigenvectors of Eq.(6) can be acquired in a two-stage procedure, and that the matrices $\boldsymbol{S}_D$ and $\boldsymbol{S}_T$ are diagonalized simultaneously[9].

Let ρ_j and $\boldsymbol{\phi}_j$ be the j-th eigenvalue and eigenvector of $\boldsymbol{S}_T$, respectively. The sum of the squared projections of all sentence vectors in document set T onto $\boldsymbol{\phi}_1$ is greater than that onto any other vector, and λ_1 represents the sum of the squared projections onto $\boldsymbol{\phi}_1$. The $\boldsymbol{\phi}_2$ gives the largest sum of the squared projections of all sentence vectors under the constraint that $\boldsymbol{\phi}_2$ is orthogonal to $\boldsymbol{\phi}_1$, and λ_2 represents the sum of the squared projections onto $\boldsymbol{\phi}_2$. The same relationship holds for the higher order eigenvectors. Suppose a sentence vector $\boldsymbol{d}_{mk}$ in document set D be mapped into vector $\boldsymbol{y}_{mk}$ in space Y. The j-th component of $\boldsymbol{y}_{mk}$ is given by

$$y_{mkj} = \boldsymbol{\phi}_j{}^T\boldsymbol{d}_{mk}/\sqrt{\rho_j}. \qquad (9)$$

The eigenvectors of the squared sum matrix of document set D in space Y correspond to $\boldsymbol{\alpha}$'s in the original space. That is, the principal components of document set D in space Y correspond to $\boldsymbol{\alpha}$'s.

On mapping to space Y, the inner product between $\boldsymbol{\phi}_j$ and $\boldsymbol{d}_{mk}$ is divided by the square root of ρ_j as shown in Eq.(9). This means that document set D in space Y is compressed in the directions of the $\boldsymbol{\phi}_j$'s with large eigenvalues, and expanded in those with small eigenvalues. For the principal components of the scaled document set D in space Y, consequently, the directions of the principal components of document set T become less dominant, and the sum of squared projections of all sentence vectors on $\boldsymbol{\alpha}$ becomes large for document set D and small for T. If $\boldsymbol{d}_{mk}$ describes the content that document set T does not include but D does, $\sum_{i=1}^{L}(\boldsymbol{d}_{mk}{}^{T}\boldsymbol{\alpha}_i)^2$ takes a large value. The value of L should be determined experimentally.

2.3 Regularization

We consider the case of Eq.(6) again. For eigenvectors to be obtained in Eq.(6), matrix $\boldsymbol{S}_T$ must be regular. When the number of sample sentence vectors is less than the dimension of the vectors, or when a certain pair of terms always co-occurs, matrix $\boldsymbol{S}_T$ is actually not regular. In these cases, matrix $\boldsymbol{S}_T$ must be regularized. Regularization can be achieved by biasing diagonal components as follows [10,12].

$$\hat{\boldsymbol{S}}_T = \boldsymbol{S}_T + \sigma^2 I, \qquad (10)$$

where σ^2 and I are a bias parameter and the identity matrix, respectively.

The meaning of Eq. (10) is as follows. Suppose that every term has a vector with the same dimension as the sentence vector and that value σ is given only to the component that corresponds to each term. Let $\boldsymbol{u}_l$ *be* the l-th term vector. Since $\sum_l \boldsymbol{u}_l\boldsymbol{u}_l{}^{T} = \sigma^2 I$, adding σ^2 to all diagonal components is equivalent to adding all term vectors to document set T. As mentioned above, the eigenvectors of $\boldsymbol{S}_T$ are the projection axes that maximize the sum of the squared projections of all sentence vectors in document set T, and the eigenvalues are the sums. Since the sum of the squared projections of all term vectors onto any normalized vector is always σ^2, in this case, the eigenvectors of $\boldsymbol{S}_T$ do not change by adding all term vectors to document set T. Since the sum of squared projections of all term vectors is added to that of all sentence vectors, however, the eigenvalues of $\boldsymbol{S}_T$ get biased by σ^2 and the criterion function also changes to

$$J(\boldsymbol{\alpha}) = P_D / (P_T + \sigma^2). \qquad (11)$$

Higher order eigenvalues of $\boldsymbol{S}_T$ usually take very small values, nearly 0. Very small eigenvalues are easily affected by noises in document set T. Since the scaling factors for document set D in space Y become very large along the directions of higher order eigenvectors, the principal component of document set D easily suffer from noises in T. Thanks to eigenvalue biasing, those scaling factors become smaller and noise influence can be reduced.

2.4 Example

This section illustrates what TDF's are obtained for simple examples. Suppose that four 5-dimensional sentence vectors are given in document set D and T as shown in Table 1. The n-th component in the vectors is supposed to represent the existence of term n. Let D-m or T-m represent the m-th sentence vector in document set D or T. The differences and similarities between the document sets D and T are as follows.

- Term 5 does not occur in document set *T,* but it occurs in D-1.
- In document set D, term 2 co-occurs with term 3 in D-3, and term 1 with term 4 in D-4.
- In document set T, term 1 co-occurs with term 3 in T-3 and term 2 with term 4 in T-4.
- In both document sets, term 1 co-occurs with term 2 in D-1 and T-1, and term 3 with term 4 in D-2 and T-2.

The TDF's were obtained by setting σ^2 at 0.1. Table 2 shows eigenvalue λ_n (n=1,..5) and eigenvector $\boldsymbol{\alpha}_n$=(α_{n1},.., α_{n5}) (n=1,..5) of Eq.(6) as P-TDF's of document set D. Table 3 shows eigenvalues μ_n (n=1,..5) and eigenvector $\boldsymbol{\beta}_n$=(β_{n1},.., β_{n5}) (n=1,..5) of Eq.(8) as N-TDF's of document set D. In Table 2 and 3, $\boldsymbol{\alpha}_n$ or $\boldsymbol{\beta}_n$ is normalized so that $\boldsymbol{\alpha}_n{}^{T}\boldsymbol{S}_D\boldsymbol{\alpha}_n$=$\lambda_n$ or $\boldsymbol{\beta}_n{}^{T}\boldsymbol{S}_T\boldsymbol{\beta}_n$=$\mu_n$. Consequently, the dynamic range of the squared inner products between sentence vectors and TDF get larger for lower order PDF. This normalization means weighting according to the degree of topic differences reflected on each eigenvector. This normalization was applied in the experiments. Table 4 shows the projections of each sentence vector onto both the first and the second eigenvector.

From these results the following can be noted.

(1) For the first eigenvector, $\boldsymbol{\alpha}_1$, shown in Table 2, both α_{11} and α_{14} take negative values, and both α_{12} and α_{13} positive values. This shows that $\boldsymbol{\alpha}_1$ reflects the term co-occurrence between 1 and 4 and the one between 2 and 3 in document set D , so the projections of D-3 and D-4 onto $\boldsymbol{\alpha}_1$ have large absolute values with different signs, and the projections of any other sentence vectors take zero, as shown in Table 4 .

Table 1. Sentence vectors comprising document set D and T.

#	D	T
1	11001	11000
2	00110	00110
3	01100	10100
4	10010	01010

Table 2. P-TDF's of document set D.

n	λ_n	α_{n1}	α_{n2}	α_{n3}	α_{n4}	α_{n5}
1	20.00	-1.58	1.58	1.58	-1.58	0.00
2	10.74	0.11	0.11	-0.04	-0.04	3.05
3	0.97	-0.01	-0.01	-0.40	-0.40	0.01
4	0.22	-0.41	-0.41	0.14	0.14	0.84
5	0.00	0.35	-0.35	0.35	-0.35	0.00

Table 3. N-TDF's of document set D.

n	μ_n	β_{n1}	β_{n2}	β_{n3}	β_{n4}	β_{n5}
1	20.00	1.58	-1.58	1.58	-1.58	0.00
2	2.78	-0.72	-0.72	0.23	0.23	1.31
3	0.97	0.01	0.01	-0.40	-0.40	-0.01
4	0.00	-0.25	0.25	0.25	-0.25	0.65
5	0.00	0.24	-0.24	-0.24	0.24	0.70

Table 4. Projections of each document.

text	$\boldsymbol{\alpha}_1$	$\boldsymbol{\alpha}_2$	$\boldsymbol{\beta}_1$	$\boldsymbol{\beta}_2$
D-1	0.00	3.27	0.00	-0.13
D-2	0.00	-0.07	0.00	0.46
D-3	3.16	0.08	0.00	-0.49
D-4	-3.16	0.08	0.00	-0.49
T-1	0.00	0.22	0.00	-1.44
T-2	0.00	-0.07	0.00	0.46
T-3	0.00	0.08	3.16	-0.49
T-4	0.00	0.08	-3.16	-0.49

(2) For the second eigenvector $\boldsymbol{\alpha}_2$ shown in Table 2, only α_{25} takes a large value. This shows that $\boldsymbol{\alpha}_2$ reflects the occurrence of term 5 in document set D, so only D-1 takes a large projection value as shown in Table 4.

(3) Similarly, the term co-occurrence between 1 and 3 and that between 2 and 4 in document set T are reflected on $\boldsymbol{\beta}_1$ shown in Table 3. Only projections of T-3 and T-4 take non-zero values.

(4) The difference between D-1 and T-1 is reflected on $\boldsymbol{\beta}_2$, as shown in Table 3. The projection of T-1 takes a large absolute value.

(5) In Table 2 and 3, the eigenvectors of the 3rd order or higher are not effective as TDF's because the eigenvalues of those orders are small.

These observations confirm that not only the term occurrence difference between the document sets but also the term co-occurrence differences are reflected on TDF's. In this paper, a document is represented as a set of sentence vectors so that the term co-occurrence difference is reflected precisely on the TDF's.

3. APPLICATION TO TEXT CATEGORIZATION

3.1 Approach

As described in section 1, this paper does not aim at developing a classifier that only uses TDF's, but at a classification scheme that combines an existent classifier with its complementary one which uses TDF's. The reasons are as follows.

(1) We define a set of documents belonging to class l as document set D and a set of documents belonging to classes other than l as document set T. If we construct a classifier that only uses TDF's, we have to apply TDFA to the document sets. In this case, document set T must include all documents that are not included in D. As a result the number of documents in document set T would probably be much larger than that in D, and document set T would probably include many documents irrelevant to class l. In such a case, it is doubtful whether subtle differences between class l and documents easily confused with those in class l can be reflected exactly on the TDF's. To reflect such subtleties , only the documents that are easily confused should be included in document set T. Those documents can be obtained by using the results of an existent classifier.

(2) It is considered easier to achieve high performance by combining TDF-based classifier with an existent high performance classifier than by constructing a classifier using TDF's only.

Based on these considerations, a classification scheme was constructed such that the complementary classifier corrects likelihood scores obtained by an existent classifier. The existent classifier acts as the main classifier. If an input document has features that should appear in a given class, a gain is added to the likelihood score of the class in the complementary classifier, and if an input document has features that should not appear in a given class, a penalty is imposed.

3.2 Complementary Classifier

The TDF's of class l is obtained as follows. First, all training documents are classified by the main classifier and

the likelihood scores of each class are obtained. Let γ be a threshold given for each class. The training documents that have a likelihood score of class *l* larger than γ are selected. Among the selected documents, the documents belonging to class *l* are added to document set *D* and the documents belonging to other classes to *T*. Every document in document set *T* is the one that has been misclassified or nearly misclassified as belonging to class *l*. We call such a document a competing document of class *l*. Document sets *D* and *T* are thus defined and used for solving Eq.(6) and (8). The P-TDF's of class *l* are given by eigenvector $\{\boldsymbol{\alpha}_n\}(n=1,..,L_G)$ of Eq.(6) and N-TDF's by eigenvector $\{\boldsymbol{\beta}_n\}(n=1,..,L_P)$ of Eq.(8).

Let $g(X)$ and $p(X)$ be a gain and a penalty of input document X for class *l*, respectively. Suppose document *X* is composed of a sentence vector set $\{\boldsymbol{x}_1,.., \boldsymbol{x}_K\}$. $g(X)$ and $p(X)$ can be obtained as follows.

$$g(X) = \sum_{i=1}^{L_G} \sum_{k=1}^{K} (\boldsymbol{x}_k{}^T \boldsymbol{\alpha}_i)^2 , \qquad (12)$$

$$p(X) = \sum_{i=1}^{L_P} \sum_{k=1}^{K} (\boldsymbol{x}_k{}^T \boldsymbol{\beta}_i)^2 , \qquad (13)$$

where L_G and L_P are parameters whose values should optimally be determined by experiments. Let $Lik(X)$ be the likelihood score of document *X* for class *l* in the main classifier and $Lik_C(X)$ the corrected likelihood score. $Lik_C(X)$ can be given as follows.

$$Lik_C(X) = Lik(X) + a\, g(X) - b\, p(X), \qquad (14)$$

where *a* and *b* are positive parameters, which should be determined by experiments. $Lik_C(X)$ is computed for documents that have $Lik(X)$ larger than γ. A document with $Lik(X)$ smaller than or equal to γ is not judged as belonging to class *l* under any conditions. If $Lik_C(X)$ exceeds another threshold δ, the input document *X* is judged as belonging to class *l*.

When Eqs.(12) or (13) are used as a gain or a penalty, $g(X)$ or $p(X)$ tend to take a large value for long documents. To reduce the influence of document-length variance, normalizing $g(X)$ or $p(X)$ by the number of sentences in document X might be effective. When sentence vectors are normalized to reduce influence from sentence-length variance in the TDF calculation, the normalized sentence vectors in document X should be used in Eqs.(12) and (13).

4. EXPERIMENTS

4.1 Experimental Conditions

The experimental conditions in this paper follow Yang's experiments [1] and Reuters-21578 was used. According to ModApte Split, the documents were selected which belong to classes that have at least one document in the training set and the test set. This resulted in 87 classes, a training set of 7770 documents, and a test set of 3019 documents.

For the training and the test data, sentence segmentation, lemmatization, replacement of uppercases by lowercases, replacement of all digits by "0", removal of "-", "/" and all punctuation, stop-word removal, and term selection were conducted as a preprocess. Sentence segmentation was needed since a document is represented as a set of sentence vectors in this paper. Ordinary documents were segmented by finding periods. In the corpus, however, there are many table-like documents where terms are spaced by the same interval without any punctuation. For such documents one line was regarded as one sentence. For the term selection, 2500 terms were selected based on χ^2 statistics[11].

To represent sentences as vectors, each term was weighted based on *tf-idf*. The weight of the *i*-th term, w_i, was determined as follows.

$$w_i = (1 + \log f_i) \log(N_D / n_i) , \qquad (15)$$

where f_i, n_i, and N_D represent the frequency of an *i*-th term in a given sentence, the number of documents including an *i*-th term and the total number of documents, respectively.

4.2 Experimental Methods

As the main classifier, kNN classifier [1] was used ,which enables high performance by using a simple algorithm. For a given input document, the classifier calculates similarity scores with each training document and then, finds k nearest neighbors out of the training documents. As the similarity measure, the cosine similarity was adopted, as it is commonly done. The value of k was set at 45 in accordance with Yang's experiment [1]. The similarity scores with the documents belonging to the same class out of all k documents are summed up as the likelihood score of that class. Thus, the likelihood scores are obtained for the classes that the documents of k nearest neighbors belong to. The input document is judged as belonging to a class if the likelihood score of that class is larger than a threshold. The threshold is determined in advance for each class so that it maximizes the evaluation score.

To evaluate the proposed method, the standard recall, precision, and F_1 measure were used. Recall (r) is defined as the ratio of correct assignments by the classifier divided

Table 5. Performance comparison for Reuters-21578.

classifier	*Recall*	*Precision*	F_1
SVM	81.20	91.37	85.99
kNN	83.39	88.07	85.67
LSF	85.07	84.89	84.98
Nnet	78.42	87.85	82.87
NB	76.88	82.45	79.56
kNN(this paper)	81.57	85.93	83.69

by the total number of correct assignments. Precision (p) is defined as the ratio of correct assignments by the classifier divided by the total number of classifier's assignments. The F_1 measure is defined as F_1=2rp/(r+p). Table 5 shows the micro-averaged evaluation scores reported by Yang [1] for several well-known methods including the kNN classifier and those of the kNN classifier in this paper. In Table 5, Nnet and NB stand for neural network and naive Bayes, respectively. The F_1 measure of the kNN result in this paper is 83.69%, which is inferior to that in Yang's report by 2%.

When L_G and L_P in Eqs.(12) and (13), and a and b in Eq.(14) were determined, the following problem occurred. Although these parameters should be determined using training data, TDF's obtained by using training data were tuned to the training data. If the parameters are determined by using such TDF's, the parameters would be doubly tuned to the training data. Evaluating test data using the parameters obtained in such a way is obviously inappropriate. To remedy this problem, cross validation was also conducted. The test data were divided into N blocks, and the parameters were determined by using N-1 blocks as the secondary training data and the remaining block was used as true test data. After running experiments N times by rotating the blocks, the results for each true test data were summed up as the results of total test data. Since the summed up results were not tuned to the test data, the results can be compared with other method's results.

The parameters were obtained as follows. The a and b in Eq.(14) were obtained by applying linear discriminant analysis. Threshold δ was obtained for each class to maximize the F_1 measure. The linear discriminant analysis and threshold determination were conducted for every combination of L_G with L_P, each of which was restricted to be lower than 15. The L_G and L_P were determined by selecting the combination which gave the best results. The linear discriminant analysis was conducted between a document set and a competing document set of each class. The competing documents were selected by using threshold γ that was used in the TDF-calculation stage. For the analysis, a 3-dimensional vector, whose components were given by $Lik(X)$, $g(X)$ and $p(X)$, was generated for each document. As a result of the linear discriminant analysis, the weights of $Lik(X)$, $g(X)$, and $p(X)$, which optimally separate a document set from a competing document set of each class, were obtained. By dividing the weights of $g(X)$ and $p(X)$ by that of $Lik(X)$, a and b in Eq.(14) were determined.

Furthermore, the experiments confirmed that combining sentence vector normalization with normalizing Eqs.(12) and (13) by the number of sentences in an input document was effective. The experimental results shown in the next paragraph are the best ones obtained by varying the parameters including γ and σ^2.

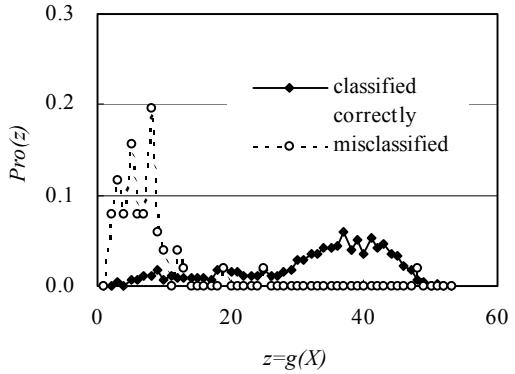

Figure 1. Probability density distribution of $g(x)$ for correctly classified and misclassified documents by kNN as belonging to class "earn".

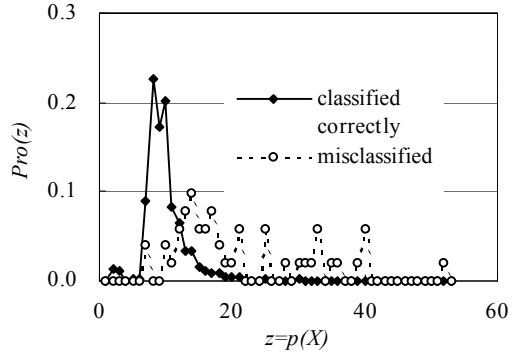

Figure 2. Probability density distribution of $p(x)$ for correctly classified and misclassified documents by kNN as belonging to class "earn".

4.3 Experimental Results

Figure 1 shows the probability density distributions of gain $g(X)$ for test documents correctly classified and misclassified by kNN as belonging to class "earn". In Fig.1, the horizontal axis shows $z=g(X)$ and the vertical axis the probability density $Prob(z_k)$, which is derived as follows.

$$Prob(z_k) = n(z_k) / \sum_k n(z_k) \qquad (16)$$

where $n(z_k)$ represents the number of documents whose $g(X)$ take value z_k. Similarly, Fig. 2 shows the probability density distributions of penalty $p(X)$. The number of documents correctly classified as belonging to class "earn" is 1077 and the number of the misclassified documents 51. Both L_G and L_P in Eqs.(12) and (13) were set at 5. Out of all documents, the misclassified documents are the most difficult ones to separate from the documents belonging to class "earn". As

shown in Figs.1 and 2, however, the misclassified documents are well separated though slight overlaps occurred.

The effectiveness of introducing σ^2 in Eq.(11) was investigated when cross validation was conducted. The σ^2 was set at the average of the diagonal components of $\boldsymbol{S}_T$ multiplied by τ. Figure 3 shows the relationship between τ and the F_1 measure when a likelihood score was corrected only by $g(X)$. The N was set at 20. As the figure shows, the F_1 measure peaks at τ=2.0. As mentioned in **2.3**, $\boldsymbol{\alpha}$ tends to be easily affected by noises in document set T if σ^2 is small. This is the reason why the F_1 measure is small for τ<2.0. If σ^2 is too large, on the other hand, P_T in Eq.(11) is not necessarily small because σ^2 becomes dominant in the denominator. Therefore, P_D /P_T may be small. This is the reason why the F_1 measure is small for τ>2.0.

Table 6 shows the classification results for training data and test data before the likelihood correction. Table 7 illustrates the classification results after the likelihood correction without cross validation. It shows the following three cases: (a) evaluate trainig data using parameters trained by training data, (b) evaluate all test data using parameters trained by all test data, (c) evaluate all test data using parameters trained by training data. When parameters were trained using training data, the F_1 measure of training data significantly increased. The F_1 measure, nevertheless, remains less than 95%. This implies that class boundaries are not clear between many classes. The F_1 measures of all test data using parameters trained by all test data, 89.33%, is quite different from that by training data, 85.87%. The former seems to be due to overtuning to test data, and the latter due to overtuning to training data. These F_1 measures do not represent true performance for test data.

Table 8 shows the classification results based on cross validation for N=2, 5, 10, and 20. Table 8 shows the F_1 measures when the likelihood was corrected by either only $g(X)$ or $p(X)$, and the F_1 measures, precision, and recall when the likelihood was corrected by both $g(X)$ and $p(X)$. These measures are micro-averaged ones. From this table, the following can be noted.

- Likelihood correction by $g(X)$ or $p(X)$ effectively improves performance.
- Likelihood correction by $g(X)$ is more effective than that by $p(X)$. This fact shows that, in each class, it is easier to extract features that should occur than features that should not occur. This is because a competing document set of each class is composed of documents belonging to various classes and because P-TDF's of each competing document set(N-TDF's of each class) are not clear in comparison with P-TDF's of each class.

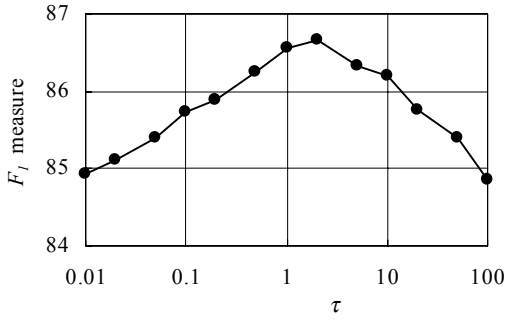

Figure 3. Relationship between F_1 measure and regularization parameter.

Table 6. Performance before liklihood correction.

evaluation data	*Recall*	*Precision*	F_1
training data	90.69	89.02	89.85
test data	81.57	85.93	83.69

Table 7. Performance after liklihood correction without cross validation.

	evaluation data	parameter training dsta	*Recall*	*Precision*	F_1
a	training data	training data	94.15	95.06	94.60
b	test data	test data	85.44	93.59	89.33
c	test data	training data	84.74	87.02	85.87

Table8. Performance after liklihood correction with cross validation .

		N			
$Lik_C(X)$	measure	2	5	10	20
Lik(X)+ag(X)	F_1	86.44	86.70	86.67	86.66
Lik(X)-bp(X)	F_1	84.55	85.04	85.09	85.18
Lik(X)+ag(X) -bp(X)	F_1	86.64	86.95	87.10	87.27
	Precision	90.03	90.59	91.03	91.28
	Recall	83.49	83.60	83.49	83.60

- The F_1 measure improved from 83.69% obtained by kNN classifier alone to 87.27% when the likelihood was corrected by both $g(X)$ and $p(X)$. The score is significantly better than the scores of the other classifiers shown in Table 5.

The averages of L_G and L_P were 3.0 and 1.7, respectively, when N was set at 20. When the dimension of sentence vectors was individually determined for each class by assigning only terms appearing in each class and its

competing document set to the vector components, the processing time to obtain 15 eigenvectors of Eqs.(6) or (8) for all classes was about 60 minutes using a workstation with a 120-Mhz clock. Thus, the computational cost of the proposed method is not high.

5. Summary

The objective of this paper is to extract distinctive features for each document class and to apply them to text categorization. This paper proposes Topic Difference Factor Analysis as a method to extract difference factors between two document sets by obtaining projection axes which maximize the ratio between the document sets as to the sum of squared projections of all sentence vectors. Tests confirmed that not only term occurrence difference between document sets but also term co-occurrence difference are reflected on the topic difference factors. By applying this method to document classification, we can obtain the features that should occur in a given class and the features that should not occur in the class. This paper proposes a classification scheme that uses the features in a complementary classifier to correct the likelihood of an input to each class of an existent classifier. By applying the kNN classifier as the existent one, improved the micro-averaged F_1 measure for Reuters-21578 from 83.69 to 87.27%.

These experimental results prove that topic difference factors have been successfully extracted. This paper, however, has not shown what the topic difference factors are for each class. Interpreting topic difference factors is an important remaining problem. TDFA has many possible applications, and the interpretation will be important for new applications.

REFERENCES

[1] Y. Yang and X. Liu. Re-examination of Text Categorization. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99),* pp. 43-49, 1999.

[2] B. Masand, G. Linoff and D. Walts. Classifying News Stories Using Memory Based Reasoning. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'92),* pp. 59-64, 1992.

[3] Y. Yang. Expert network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94),* pp. 13-22, 1994.

[4] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *European Conference on Machine Learning (ECML),* 1998.

[5] Y. Yang and C.G. Chute. An Example-based Mapping method for Text Categorization. *ACM Transaction on Information Systems (TOIS)*, 12(3), pp. 252-277, 1994.

[6] R. E. Schapire and Y. Singer. BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*, 39, pp. 135-168, 2000.

[7] J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The Smart Retrieval System-Experiments in Automatic Document Processing*, pp. 313-323, Prentice-Hall, 1971.

[8] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*, John Wiley & Sons Inc., 1973.

[9] K. Fukunaga. *Introduction to Statistical Pattern Recognition (Second Edition)*, Academic Press Inc., 1990.

[10] G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*, John Wiley & Sons, Inc., 1992.

[11] Y. Yang and J. P. Pederson. Feature Selection in Statistical Learning for Text Categorization. In *The Fourteenth International Conference on Machine Learning,* pp. 412-420, 1997.

[12] J. H. Friedman. Regularized Discriminant Analysis. *J. Amer. Statist. Assoc.* **84**, pp.165-175, 1989..