A Similarity Measure for Text Classification and Clustering

Yung-Shen Lin, Jung-Yi Jiang, and Shie-Jue Lee, *Member, IEEE*

Abstract—Measuring the similarity between documents is an important operation in the text processing field. In this paper, a new similarity measure is proposed. To compute the similarity between two documents with respect to a feature, the proposed measure takes the following three cases into account: a) The feature appears in both documents, b) the feature appears in only one document, and c) the feature appears in none of the documents. For the first case, the similarity increases as the difference between the two involved feature values decreases. Furthermore, the contribution of the difference is normally scaled. For the second case, a fixed value is contributed to the similarity. For the last case, the feature has no contribution to the similarity. The proposed measure is extended to gauge the similarity between two sets of documents. The effectiveness of our measure is evaluated on several real-world data sets for text classification and clustering problems. The results show that the performance obtained by the proposed measure is better than that achieved by other measures.

Index Terms—Document classification, document clustering, entropy, accuracy, classifiers, clustering algorithms

1 Introduction

Text processing plays an important role in information retrieval, data mining, and web search [27], [31], [46]. In text processing, the bag-of-words model is commonly used [26], [29], [44]. A document is usually represented as a vector in which each component indicates the value of the corresponding feature in the document. The feature value can be term frequency (the number of occurrences of a term appearing in the document), relative term frequency (the ratio between the term frequency and the total number of occurrences of all the terms in the document set), or tf-idf (a combination of term frequency and inverse document frequency) [25]. Usually, the dimensionality of a document is large and the resulting vector is sparse, i.e., most of the feature values in the vector are zero. Such high-dimensionality and sparsity can be a severe challenge for similarity measure which is an important operation in text processing algorithms [5], [6], [10], [11], [23], [24], [28], [30], [33], [34], [37], [41], [42], [47], [49], [50].

A lot of measures have been proposed for computing the similarity between two vectors. The Kullback-Leibler divergence [35] is a non-symmetric measure of the difference between the probability distributions associated with the two vectors. Euclidean distance [45] is a well-known similarity metric taken from the Euclidean geometry field. Manhattan distance [45], similar to Euclidean distance and also known as the taxicab metric, is another similarity metric. The Canberra distance metric [45] is used in situations where elements in a vector are always non-negative. Cosine similarity [25] is a measure taking the cosine of the angle between two vectors. The Bray-Curtis similarity measure [40] is a city-block metric which is sensitive to outlying values. The Jaccard coefficient [21] is a statistic used for comparing the similarity of two sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets. The Hamming distance [21], [22] between two vectors is the number of positions at which the corresponding symbols are different. The extended Jaccard coefficient and the Dice coefficient [48], [49] retain the sparsity property of the cosine similarity measure while allowing discrimination of collinear vectors. An information-theoretic measure for document similarity, named IT-Sim, was proposed in [8], [39]. Chim *et al*. [11] proposed a phrase-based measure to compute the similarity based on the Suffix Tree Document (STD) model.

Similarity measures have been extensively used in text classification and clustering algorithms. The spherical *k*-means algorithm introduced by Dhillon and Modha [15] adopted the cosine similarity measure for document clustering. Zhao and Karypis [52] reported results of clustering experiments with 7 clustering algorithms and 12 different text data sets, and concluded that the objective function based on cosine similarity "leads to the best solutions irrespective of the number of clusters for most of the data sets." D'hondt *et al*. [17] adopted a cosine-based pairwise-adaptive similarity for document clustering. Zhang *et al*. [51] used cosine to calculate a correlation similarity between two projected documents in a low-dimensional semantic space and performed document clustering in the correlation similarity measure space. Kogan *et al*. [32] proposed a two step clustering procedure in which the sPDDP [14] is used to generate initial partitions in the first step and a *k*-means clustering algorithm using the Kullback-Leibler divergence is applied in the second step.

• *The authors are with the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung 80424, Taiwan. E-mail: {ccdavidlin, jungyi}@water.ee.nsysu.edu.tw; leesj@mail.ee.nsysu.edu.tw.*

Manuscript received 20 Feb. 2012; revised 24 Dec. 2012; accepted 3 Jan. 2013. Date of publication 24 Jan. 2013; date of current version 9 July 2014. Recommended for acceptance by J. Zobel.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier 10.1109/TKDE.2013.19

Dhillon *et al*. [16] proposed a divisive information-theoretic feature clustering algorithm for text classification using the Kullback-Leibler divergence. Euclidean distance is usually the default choice of similarity-based methods, e.g. *k*-NN and *k*-means algorithms [18]. Kogan *et al*. [33] combined squared Euclidean distance with relative entropy in a *k*-means like clustering algorithm. Chim *et al*. [11] performed document clustering based on the proposed phrase-based similarity measure. The extended Jaccard coefficient can be used for document data and it reduces to the Jaccard cofficient in the case of binary attributes [48], [49].

We propose a new measure for computing the similarity between two documents. Several characteristics are embedded in this measure. It is a symmetric measure. The difference between presence and absence of a feature is considered more essential than the difference between the values associated with a present feature. The similarity increases as the difference between the two values associated with a present feature decreases. Furthermore, the contribution of the difference is normally scaled. The similarity decreases when the number of presence-absence features increases. An absent feature has no contribution to the similarity. The proposed measure is extended to gauge the similarity between two sets of documents. The measure is applied in several text applications, including single-label classification, multi-label classification, *k*-means like clustering, and hierarchical agglomerative clustering, and the results obtained demonstrate the effectiveness of the proposed similarity measure.

The rest of this paper is organized as follows. Related work is briefly described in Section 2. Our proposed measure is introduced in Section 3, firstly the version between two documents, followed by the extended version between two sets of documents. Experimental results are presented in Section 4. Finally, concluding remarks are given in Section 5.

2 Related Works

Some measures which have been popularly adopted for computing the similarity between two documents are briefly presented here. Let $\mathbf{d}_1$ and $\mathbf{d}_2$ be two documents represented as vectors. The Euclidean distance [45] measure is defined as the root of square differences between the respective coordinates of $\mathbf{d}_1$ and $\mathbf{d}_2$, i.e.,

$$d_{Euc}(\mathbf{d}_1, \mathbf{d}_2) = [(\mathbf{d}_1 - \mathbf{d}_2)\cdot(\mathbf{d}_1 - \mathbf{d}_2)]^{1/2}, \quad (1)$$

where $\mathbf{A}\cdot\mathbf{B}$ denotes the inner product of the two vectors $\mathbf{A}$ and $\mathbf{B}$. Cosine similarity [25] measures the cosine of the angle between $\mathbf{d}_1$ and $\mathbf{d}_2$ as follows:

$$S_{Cos}(\mathbf{d}_1, \mathbf{d}_2) = \frac{\mathbf{d}_1\cdot\mathbf{d}_2}{(\mathbf{d}_1\cdot\mathbf{d}_1)^{1/2}(\mathbf{d}_2\cdot\mathbf{d}_2)^{1/2}}. \quad (2)$$

Pairwise-adaptive similarity [17] dynamically selects a number of features out of $\mathbf{d}_1$ and $\mathbf{d}_2$ and is defined to be

$$d_{Pair}(\mathbf{d}_1, \mathbf{d}_2) = \frac{\mathbf{d}_{1,K}\cdot\mathbf{d}_{2,K}}{(\mathbf{d}_{1,K}\cdot\mathbf{d}_{1,K})^{1/2}(\mathbf{d}_{2,K}\cdot\mathbf{d}_{2,K})^{1/2}}, \quad (3)$$

where $\mathbf{d}_{i,K}$ is a subset of $\mathbf{d}_i$, $i = 1, 2$, containing the values of the features which are the union of the K largest features appearing in $\mathbf{d}_1$ and $\mathbf{d}_2$, respectively.

The Extended Jaccard coefficient [48], [49] is an extended version of the Jaccard coefficient [21] for data processing:

$$S_{EJ}(\mathbf{d}_1, \mathbf{d}_2) = \frac{\mathbf{d}_1\cdot\mathbf{d}_2}{\mathbf{d}_1\cdot\mathbf{d}_1 + \mathbf{d}_2\cdot\mathbf{d}_2 - \mathbf{d}_1\cdot\mathbf{d}_2} \quad (4)$$

while the Dice coefficient looks similar to it and is defined as follows:

$$S_{Dic}(\mathbf{d}_1, \mathbf{d}_2) = \frac{2\mathbf{d}_1\cdot\mathbf{d}_2}{\mathbf{d}_1\cdot\mathbf{d}_1 + \mathbf{d}_2\cdot\mathbf{d}_2}. \quad (5)$$

IT-Sim, an information-theoretic measure for document similarity, was proposed in [8], [39]:

$$S_{IT}(\mathbf{d}_1, \mathbf{d}_2) = \frac{2\sum_{w_i}\min(p_{1i}, p_{2i})\log\pi(w_i)}{\sum_{w_i} p_{1i}\log\pi(w_i) + \sum_{w_i} p_{2i}\log\pi(w_i)}, \quad (6)$$

where w_i represents feature i, p_{ji} indicates the normalized value of w_i in document $\mathbf{d}_j$ for $j = 1$ or $j = 2$, and $\pi(w_i)$ is the proportion of documents in which w_i occurs.

3 Proposed Similarity Measure

Let a document $\mathbf{d}$ with m features $w_1, w_2, \ldots, w_m$ be represented as an m-dimensional vector, i.e., $\mathbf{d} = < d_1, d_2, \ldots, d_m >$. If w_i, $1 \le i \le m$, is absent in the document, then $d_i = 0$. Otherwise, $d_i > 0$. The following properties, among other ones, are preferable for a similarity measure between two documents:

1) The presence or absence of a feature is more essential than the difference between the two values associated with a present feature. Consider two features w_i and w_j and two documents $\mathbf{d}_1$ and $\mathbf{d}_2$. Suppose w_i does not appear in $\mathbf{d}_1$ but it appears in $\mathbf{d}_2$. Then w_i is considered to have no relationship with $\mathbf{d}_1$ while it has some relationship with $\mathbf{d}_2$. In this case, $\mathbf{d}_1$ and $\mathbf{d}_2$ are dissimilar in terms of w_i. If w_j appears in both $\mathbf{d}_1$ and $\mathbf{d}_2$. Then w_j has some relationship with $\mathbf{d}_1$ and $\mathbf{d}_2$ simultaneously. In this case, $\mathbf{d}_1$ and $\mathbf{d}_2$ are similar to some degree in terms of w_j. For the above two cases, it is reasonable to say that w_i carries more weight than w_j in determining the similarity degree between $\mathbf{d}_1$ and $\mathbf{d}_2$. For example, assume that w_i is absent in $\mathbf{d}_1$, i.e., $d_{1i} = 0$, but appears in $\mathbf{d}_2$, e.g., $d_{2i} = 2$, and w_j appears both in $\mathbf{d}_1$ and $\mathbf{d}_2$, e.g., $d_{1j} = 3$ and $d_{2j} = 5$. Then w_i is considered to be more essential than w_j in determining the similarity between $\mathbf{d}_1$ and $\mathbf{d}_2$, although the differences of the feature values in both cases are the same.
2) The similarity degree should increase when the difference between two non-zero values of a specific feature decreases. For example, the similarity involved with $d_{13} = 2$ and $d_{23} = 20$ should be smaller than that involved with $d_{13} = 2$ and $d_{23} = 3$.
3) The similarity degree should decrease when the number of presence-absence features increases. For a presence-absence feature of $\mathbf{d}_1$ and $\mathbf{d}_2$, $\mathbf{d}_1$ and $\mathbf{d}_2$ are dissimilar in terms of this feature as commented earlier. Therefore, as the number of presence-absence features increases, the dissimilarity between

$\mathbf{d}_1$ and $\mathbf{d}_2$ increases and thus the similarity decreases. For example, the similarity between the documents $< 1, 0, 1 >$ and $< 1, 1, 0 >$ should be smaller than that between the documents $< 1, 0, 1 >$ and $< 1, 0, 0 >$.

4) Two documents are least similar to each other if none of the features have non-zero values in both documents. Let $\mathbf{d}_1 = < d_{11}, d_{12}, \ldots, d_{1m} >$ and $\mathbf{d}_2 = < d_{21}, d_{22}, \ldots, d_{2m} >$. If

$$d_{1i}d_{2i} = 0,$$
$$d_{1i} + d_{2i} > 0$$

for $1 \leq i \leq m$, then $\mathbf{d}_1$ and $\mathbf{d}_2$ are least similar to each other. As mentioned earlier, $\mathbf{d}_1$ and $\mathbf{d}_2$ are dissimilar in terms of a presence-absence feature. Since all the features are presence-absence features, the dissimilarity reaches the extremity in this case. For example, the two documents $< x, 0, y >$ and $< 0, z, 0 >$, with x, y, and z being non-zero numbers, are least similar to each other.

5) The similarity measure should be symmetric. That is, the similarity degree between $\mathbf{d}_1$ and $\mathbf{d}_2$ should be the same as that between $\mathbf{d}_2$ and $\mathbf{d}_1$.

6) The value distribution of a feature is considered, i.e., the standard deviation of the feature is taken into account, for its contribution to the similarity between two documents. A feature with a larger spread offers more contribution to the similarity between $\mathbf{d}_1$ and $\mathbf{d}_2$.

Note that these properties are strongly related to the notion of burstiness studied in [12], to the normalization used in [7], [12], to the exploitation of positions at which the corresponding symbols are different proposed in [22], and to the deviation-based fuzzy distance adopted in [36].

The measures mentioned in Section 2 are deficient with respect to some of the above properties. For example, Euclidean does not meet properties 1, 3, 4, and 6, and Cosine, Pairwise-adaptive, Extended Jaccard, Dice, and IT-Sim do not satisfy one or more of properties 3, 4 and 6. Consider three documents $\mathbf{d}_1 = < 10, 20 >$ and $\mathbf{d}_2 = < 10, 5 >$, and $\mathbf{d}_3 = < 10, 0 >$. With Euclidean, the distance between $\mathbf{d}_1$ and $\mathbf{d}_2$ is 15 which is larger than the distance between $\mathbf{d}_2$ and $\mathbf{d}_3$, 5. This contradicts properties 1 and 3. With Cosine, the similarity between $\mathbf{d}_1$ and $\mathbf{d}_2$ is 0.8 which is lower than the similarity between $\mathbf{d}_2$ and $\mathbf{d}_3$, 0.894. This contradicts property 3.

3.1 Similarity between Two Documents

Based on the preferable properties mentioned above, we propose a similarity measure, called SMTP (Similarity Measure for Text Processing), for two documents $\mathbf{d}_1 = < d_{11}, d_{12}, \ldots, d_{1m} >$ and $\mathbf{d}_2 = < d_{21}, d_{22}, \ldots, d_{2m} >$. Define a function F as follows:

$$F(\mathbf{d}_1, \mathbf{d}_2) = \frac{\sum_{j=1}^{m} N_*(d_{1j}, d_{2j})}{\sum_{j=1}^{m} N_\cup(d_{1j}, d_{2j})} \tag{7}$$

where

$$N_*(d_{1j}, d_{2j}) = \begin{cases} 0.5\left(1 + \exp\left\{-\left(\frac{d_{1j}-d_{2j}}{\sigma_j}\right)^2\right\}\right), \\ \quad \text{if } d_{1j}d_{2j} > 0 \\ 0, \text{ if } d_{1j} = 0 \text{ and } d_{2j} = 0 \\ -\lambda, \text{ otherwise,} \end{cases} \tag{8}$$

$$N_\cup(d_{1j}, d_{2j}) = \begin{cases} 0, \text{ if } d_{1j} = 0 \text{ and } d_{2j} = 0 \\ 1, \text{ otherwise.} \end{cases} \tag{9}$$

Then our proposed similarity measure, S_{SMTP}, for $\mathbf{d}_1$ and $\mathbf{d}_2$ is

$$S_{SMTP}(\mathbf{d}_1, \mathbf{d}_2) = \frac{F(\mathbf{d}_1, \mathbf{d}_2) + \lambda}{1 + \lambda}. \tag{10}$$

The proposed measure takes into account the following three cases: a) The feature considered appears in both documents, b) the feature considered appears in only one document, and c) the feature considered appears in none of the documents. For the first case, we set a lower bound 0.5 and decrease the similarity as the difference between the feature values of the two documents increases, scaled by a Gaussian function as shown in Eq.(8) where σ_j is the standard deviation of all non-zero values for feature w_j in the training data set. For the second case, we set a negative constant $-\lambda$ disregarding the magnitude of the non-zero feature value. For the last case, the feature has no contribution to the similarity. Some remarks for this measure are discussed below.

Remark 1 (Property 1). Let $\mathbf{d}_1$ and $\mathbf{d}_2$ be two documents, and w_i be the feature being considered. Consider the following two cases: (1) $d_{1i} = a > 0$ and $d_{2i} = b > 0$, and (2) $d_{1i} = 0$ and $d_{2i} = c > 0$. Then the similarity degree for case 1 is greater than that for case 2. Note that $\sum_{j=1}^{m} N_\cup(d_{1j}, d_{2j})$ is the same for both cases, and let it be denoted by A. Also, let $\sum_{j=1, j\neq i}^{m} N_*(d_{1j}, d_{2j})$ be denoted by B.

- For case 1. We have

$$F(\mathbf{d}_1, \mathbf{d}_2) = \frac{B + 0.5\left(1 + \exp\left\{-\left(\frac{a-b}{\sigma_i}\right)^2\right\}\right)}{A}. \tag{11}$$

- For case 2. We have

$$F(\mathbf{d}_1, \mathbf{d}_2) = \frac{B - \lambda}{A}. \tag{12}$$

Since $0.5\left(1 + \exp\left\{-\left(\frac{a-b}{\sigma_i}\right)^2\right\}\right) > 0.5 > -\lambda$, Eq.(11) is obviously greater than Eq.(12).

Remark 2 (Property 2). Let $\mathbf{d}_1$ and $\mathbf{d}_2$ be two documents, and w_i be the feature being considered. Let a, b, and c be three non-zero numbers and $\|a - c\| \leq \|b - c\|$. Consider the following two cases: (1) $d_{1i} = a$ and $d_{2i} = c$, and (2) $d_{1i} = b$ and $d_{2i} = c$. Assume that the standard deviation involved holds the same value for both cases. Then the similarity degree for case 1 is greater than that for case 2. Follow the discussion in Remark 1 previously presented.

- For case 1. We have

$$F(\mathbf{d}_1, \mathbf{d}_2) = \frac{B + 0.5\left(1 + \exp\left\{-\left(\frac{a-c}{\sigma_i}\right)^2\right\}\right)}{A}. \tag{13}$$

- For case 2. We have

$$F(\mathbf{d}_1, \mathbf{d}_2) = \frac{B + 0.5\left(1 + \exp\left\{-\left(\frac{b-c}{\sigma_i}\right)^2\right\}\right)}{A}. \quad (14)$$

Since $\exp\left\{-\left(\frac{a-c}{\sigma_i}\right)^2\right\} > \exp\left\{-\left(\frac{b-c}{\sigma_i}\right)^2\right\}$, Eq.(13) is obviously greater than Eq.(14).

Remark 3 (Property 3). Let $\mathbf{d}_1$, $\mathbf{d}_2$, and $\mathbf{d}_3$ be three documents. Document $\mathbf{d}_1$ has a non-zero feature values, $\mathbf{d}_2$ has $a+b$ non-zero feature values, and $\mathbf{d}_3$ has at least $a+b$ non-zero values. Furthermore, The a non-zero feature values in $\mathbf{d}_1$ also appear in $\mathbf{d}_2$ and $\mathbf{d}_3$, and the remaining b non-zero feature values in $\mathbf{d}_2$ also appear in $\mathbf{d}_3$. Then $S_{SMTP}(\mathbf{d}_1, \mathbf{d}_2) \geq S_{SMTP}(\mathbf{d}_1, \mathbf{d}_3)$. Without loss of generality, we assume that $\mathbf{d}_1 = < x_1, \ldots, x_a, 0, \ldots, 0 >$, $\mathbf{d}_2 = < x_1, \ldots, x_a, y_1, \ldots, y_b, 0, \ldots, 0 >$, and $\mathbf{d}_3 = < x_1, \ldots, x_a, y_1, \ldots, y_b, y_{b+1}, \ldots, y_c, 0, \ldots, 0 >$, where $b \leq c$ and $x_1, \ldots, x_a, y_1, \ldots, y_c$ are non-zero values. Then we have

$$F(\mathbf{d}_1, \mathbf{d}_2) = \frac{A}{B} = \frac{a - \lambda b}{a + b}, \quad (15)$$

where $A = \sum_{j=1}^{a} N_*(x_j, x_j) + \sum_{j=1}^{b} N_*(0, y_j) + \sum_{j=a+b+1}^{m} N_*(0,0)$ and $B = \sum_{j=1}^{a} N_\cup(x_j, x_j) + \sum_{j=1}^{b} N_\cup(0, y_j) + \sum_{j=a+b+1}^{m} N_\cup(0,0)$, and

$$F(\mathbf{d}_1, \mathbf{d}_3) = \frac{C}{D} = \frac{a - \lambda c}{a + c}, \quad (16)$$

where $C = \sum_{j=1}^{a} N_*(x_j, x_j) + \sum_{j=1}^{c} N_*(0, y_j) + \sum_{j=a+c+1}^{m} N_*(0,0)$ and $D = \sum_{j=1}^{a} N_\cup(x_j, x_j) + \sum_{j=1}^{c} N_\cup(0, y_j) + \sum_{j=a+c+1}^{m} N_\cup(0,0)$. Since $a - \lambda b \geq a - \lambda c$ and $a+b \leq a+c$, Eq.(15) is greater than or equal to Eq.(16). Therefore, we conclude $S_{SMTP}(\mathbf{d}_1, \mathbf{d}_2) \geq S_{SMTP}(\mathbf{d}_1, \mathbf{d}_3)$.

Remark 4 (Property 4). For any two documents $\mathbf{d}_1$ and $\mathbf{d}_2$, $S_{SMTP}(\mathbf{d}_1, \mathbf{d}_2)$ lies between 0 and 1. Obviously, $\sum_{j=1}^{m} N_\cup(d_{1j}, d_{2j})$ has a maximum value of m when each $N_\cup(d_{1j}, d_{2j})$, $1 \leq j \leq m$, equals 1. When $\mathbf{d}_1$ and $\mathbf{d}_2$ are identical, $\sum_{j=1}^{m} N_*(d_{1j}, d_{2j})$ is evaluated to m. In this case, $F(\mathbf{d}_1, \mathbf{d}_2)$ has a maximum value of $\frac{m}{m} = 1$. On the other hand, when $d_{1j}d_{2j} = 0$ and $d_{1j} + d_{2j} > 0$ for each j, $1 \leq j \leq m$, $\sum_{j=1}^{m} N_*(d_{1j}, d_{2j})$ is evaluated to $-m\lambda$. In this case, $F(\mathbf{d}_1, \mathbf{d}_2)$ has a minimum value of $\frac{-m\lambda}{m} = -\lambda$. Therefore, $S_{SMTP}(\mathbf{d}_1, \mathbf{d}_2)$ lies between 0 and 1.

Remark 5 (Property 4). Two documents are least similar to each other if none of the features have non-zero values in both documents. Without loss of generality, we assume that $\mathbf{d}_1 = < x_1, \ldots, x_a, 0, \ldots, 0 >$ and $\mathbf{d}_2 = < 0, \ldots, 0, x_{a+1}, \ldots, x_b, 0, \ldots, 0 >$, where $x_1, \ldots, x_b$ are non-zero values. Let $A = \sum_{j=1}^{a} N_*(x_j, 0) + \sum_{j=a+1}^{b} N_*(0, x_j) + \sum_{j=b+1}^{m} N_*(0,0)$ and $B = \sum_{j=1}^{a} N_\cup(x_j, 0) + \sum_{j=a+1}^{b} N_\cup(0, x_j) + \sum_{j=b+1}^{m} N_\cup(0,0)$. Then we have

$$\begin{aligned} F(\mathbf{d}_1, \mathbf{d}_2) &= \frac{A}{B} = \frac{-a\lambda - (b-a)\lambda}{a + (b-a)} \\ &= -\lambda. \end{aligned} \quad (17)$$

By Remark 4 previously presented, we conclude that $\mathbf{d}_1$ and $\mathbf{d}_2$ are least similar to each other.

Remark 6 (Property 5). For any two documents $\mathbf{d}_1$ and $\mathbf{d}_2$, $S_{SMTP}(\mathbf{d}_1, \mathbf{d}_2)$ is symmetric. The definitions of Eq.(8) and Eq.(9) are not dependent on the order of the feature values involved. Therefore, $S_{SMTP}(\mathbf{d}_1, \mathbf{d}_2) = S_{SMTP}(\mathbf{d}_2, \mathbf{d}_1)$.

Remark 7 (Property 6). Let $\mathbf{d}_1$ and $\mathbf{d}_2$ be two documents, and w_i be the feature being considered. A larger spread in w_i offers more contribution to the similarity between $\mathbf{d}_1$ and $\mathbf{d}_2$. Let a and b be two non-zero numbers, and $d_{1i} = a$ and $d_{2i} = b$. Consider two cases: (1) w_i with deviation σ_i, and (2) w_i with deviation $\sigma_i^{'}$, and $\sigma_i < \sigma_i^{'}$. Then the similarity degree with σ_i is lower than the similarity degree with $\sigma_i^{'}$. Follow the discussion in Remark 1 previously presented.

- For the case of σ_i. We have

$$F(\mathbf{d}_1, \mathbf{d}_2) = \frac{B + 0.5\left(1 + \exp\left\{-\left(\frac{a-b}{\sigma_i}\right)^2\right\}\right)}{A}. \quad (18)$$

- For the case of $\sigma_i^{'}$. We have

$$F(\mathbf{d}_1, \mathbf{d}_2) = \frac{B + 0.5\left(1 + \exp\left\{-\left(\frac{a-b}{\sigma_i^{'}}\right)^2\right\}\right)}{A}. \quad (19)$$

Since $\exp\left\{-\left(\frac{a-b}{\sigma_i}\right)^2\right\} < \exp\left\{-\left(\frac{a-b}{\sigma_i^{'}}\right)^2\right\}$, Eq.(19) is obviously greater than Eq.(18).

Remark 8. Our similarity measure is equivalent to the Jaccard coefficient [21] for documents with binary feature values by setting λ to zero. For binary feature values and $\lambda = 0$, $N_*(d_{1j}, d_{2j})$ in Eq.(8) is reduced to

- $0.5\left(1 + \exp\left\{-\left(\frac{1-1}{\sigma}\right)^2\right\}\right) = 0.5 \times 2 = 1$ if $d_{1j} = 1$ and $d_{2j} = 1$, and
- 0, if $d_{1j} = 0$ or $d_{2j} = 0$

which is identical to the Jaccard coefficient. Therefore, the Jaccard coefficient is a special case of our similarity measure.

We give an example here to illustrate how our method works. Suppose we have a document:

```
banana is sweet.
banana is good to health.
```

and another document:

```
apple is better than banana.
apple and apple pie are good to health.
```

Assume that we use word count as feature values and we consider 7 features which are apple, banana, good, health, pie, sour, and sweet, respectively. Then we have two corresponding vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ as

$$\mathbf{d}_1 = < 0, 2, 1, 1, 0, 0, 1 >, \quad \mathbf{d}_2 = < 3, 1, 1, 1, 1, 0, 0 >$$

with $m = 7$. Assume $\lambda = 1$ and $\sigma_1 = \sigma_2 = \ldots = \sigma_7 = 2$. The similarity between these two documents is

$$\begin{aligned} F(\mathbf{d}_1, \mathbf{d}_2) &= \frac{A}{1+1+1+1+1+0+1}, \\ &= -0.01843 \end{aligned}$$

where $A = (-1) + 0.5(1 + e^{-(\frac{2-1}{2})^2}) + 0.5(1 + e^{-(\frac{1-1}{2})^2}) + 0.5(1 + e^{-(\frac{1-1}{2})^2}) + (-1) + 0 + (-1)$. Therefore, $S_{SMTP} = 0.4908$.

3.2 Similarity between Two Document Sets

We extend our method to measure the similarity between two document sets. Refer to Eq.(7), it can be considered as an average score of the features occurring in at least one of the two documents. Based on this perspective, the similarity between two document sets is designed to calculate an average score of the features occurring in the two sets. Let G_1 and G_2 be two document sets containing q_1 and q_2 documents, respectively, i.e., $G_1 = \{\mathbf{d}_1^1, \mathbf{d}_2^1, \ldots, \mathbf{d}_{q_1}^1\}$ and $G_2 = \{\mathbf{d}_1^2, \mathbf{d}_2^2, \ldots, \mathbf{d}_{q_2}^2\}$ where $\mathbf{d}_j^s =< d_{j1}^s, d_{j2}^s, \ldots, d_{jm}^s >$, $s \in \{1, 2\}$, and $1 \le j \le q_1$ or $1 \le j \le q_2$. The function F between G_1 and G_2 is defined to be

$$F(G_1, G_2) = \frac{\sum_{i=1}^{q_1}\sum_{j=1}^{q_2}\sum_{k=1}^{m} N_*\left(d_{ik}^1, d_{jk}^2\right)}{\sum_{i=1}^{q_1}\sum_{j=1}^{q_2}\sum_{k=1}^{m} N_\cup\left(d_{ik}^1, d_{jk}^2\right)} \tag{20}$$

$$= \frac{\sum_{k=1}^{m}\sum_{i=1}^{q_1}\sum_{j=1}^{q_2} N_*\left(d_{ik}^1, d_{jk}^2\right)}{\sum_{k=1}^{m}\sum_{i=1}^{q_1}\sum_{j=1}^{q_2} N_\cup\left(d_{ik}^1, d_{jk}^2\right)} \tag{21}$$

and the similarity measure, S_{SMTP}, for G_1 and G_2 is

$$S_{SMTP}(G_1, G_2) = \frac{F(G_1, G_2) + \lambda}{1 + \lambda}. \tag{22}$$

Consider a specific feature w_k, we have

$$\begin{aligned}
&\sum_{i=1}^{q_1}\sum_{j=1}^{q_2} N_*\left(d_{ik}^1, d_{jk}^2\right) \\
&= \sum_{d_{ik}^1>0}\sum_{d_{jk}^2>0} N_*\left(d_{ik}^1, d_{jk}^2\right) + \sum_{d_{ik}^1=0}\sum_{d_{jk}^2>0} N_*\left(d_{ik}^1, d_{jk}^2\right) + \\
&\quad \sum_{d_{ik}^1>0}\sum_{d_{jk}^2=0} N_*\left(d_{ik}^1, d_{jk}^2\right) + \sum_{d_{ik}^1=0}\sum_{d_{jk}^2=0} N_*\left(d_{ik}^1, d_{jk}^2\right) \\
&= \sum_{d_{ik}^1>0}\sum_{d_{jk}^2>0}\left[0.5\left(1 + \exp\left\{-\left(\frac{d_{ik}^1 - d_{jk}^2}{\sigma_k}\right)^2\right\}\right)\right] \\
&\quad +(-\lambda) \times \left(q_1 - \sum_{i=1}^{q_1} sgn\left(d_{ik}^1\right)\right)\sum_{j=1}^{q_2} sgn\left(d_{jk}^2\right) \\
&\quad +(-\lambda) \times \left(\sum_{i=1}^{q_1} sgn\left(d_{ik}^1\right)\right)\left(q_2 - \sum_{j=1}^{q_2} sgn\left(d_{jk}^2\right)\right),
\end{aligned} \tag{23}$$

where

$$sgn(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

Similarly, we have

$$\begin{aligned}
&\sum_{i=1}^{q_1}\sum_{j=1}^{q_2} N_\cup\left(d_{ik}^1, d_{jk}^2\right) \\
&= \sum_{i=1}^{q_1} sgn\left(d_{ik}^1\right)\sum_{j=1}^{q_2} sgn\left(d_{jk}^2\right) + \left(q_1 - \sum_{i=1}^{q_1} sgn\left(d_{ik}^1\right)\right) \times \\
&\quad \sum_{j=1}^{q_2} sgn\left(d_{jk}^2\right) + \sum_{i=1}^{q_1} sgn\left(d_{ik}^1\right)\left(q_2 - \sum_{j=1}^{q_2} sgn\left(d_{jk}^2\right)\right).
\end{aligned} \tag{25}$$

Let δ_k^s, $s \in \{1, 2\}$, be the number of documents in G_s with non-zero values in w_k, i.e.,

$$\delta_k^s = \sum_{i=1}^{q_s} sgn\left(d_{ik}^s\right). \tag{26}$$

Then Eq.(23) and Eq.(25) can be expressed as

$$\begin{aligned}
&\sum_{i=1}^{q_1}\sum_{j=1}^{q_2} N_*\left(d_{ik}^1, d_{jk}^2\right) \\
&= \sum_{d_{ik}^1>0}\sum_{d_{jk}^2>0}\left[0.5\left(1 + \exp\left\{-\left(\frac{d_{ik}^1 - d_{jk}^2}{\sigma_k}\right)^2\right\}\right)\right] \\
&\quad +(-\lambda) \times (q_1 - \delta_k^1)\delta_k^2 + (-\lambda) \times \delta_k^1\left(q_2 - \delta_k^2\right),
\end{aligned} \tag{27}$$

$$\begin{aligned}
&\sum_{i=1}^{q_1}\sum_{j=1}^{q_2} N_\cup\left(d_{ik}^1, d_{jk}^2\right) \\
&= \delta_k^1\delta_k^2 + \left(q_1 - \delta_k^1\right)\delta_k^2 + \delta_k^1\left(q_2 - \delta_k^2\right).
\end{aligned} \tag{28}$$

The computation for the first term of Eq.(27) is very costly. We provide an approximation to reduce the complexity involved in the computation. The idea is to use the means instead of individual values. Let v_k^1 and v_k^2 be the statistical means of the non-zero feature values in w_k in G_1 and G_2, respectively, i.e.,

$$v_k^s = \frac{\sum_{i=1}^{q_s} d_{ik}^s sgn\left(d_{ik}^s\right)}{\delta_k^s} \tag{29}$$

for $s = 1, 2$. We approximate the first term of Eq.(27) by

$$\begin{aligned}
&\sum_{d_{ik}^1>0}\sum_{d_{jk}^2>0}\left[0.5\left(1 + \exp\left\{-\left(\frac{d_{ik}^1 - d_{jk}^2}{\sigma_k}\right)^2\right\}\right)\right] \\
&\approx \sum_{d_{ik}^1>0}\sum_{d_{jk}^2>0}\left[0.5\left(1 + \exp\left\{-\left(\frac{v_k^1 - v_k^2}{\sigma_k}\right)^2\right\}\right)\right] \\
&= 0.5\left(1 + \exp\left\{-\left(\frac{v_k^1 - v_k^2}{\sigma_k}\right)^2\right\}\right)\left(\delta_k^1\right)\left(\delta_k^2\right).
\end{aligned} \tag{30}$$

In this way, $\sum_{i=1}^{q_1}\sum_{j=1}^{q_2} N_*\left(d_{ik}^1, d_{jk}^2\right)$, and therefore $F(G_1, G_2)$, can be computed efficiently.

Remark 1. The formula Eq.(20) used is essentially an average of similarity values between individual documents. Note that $\sum_{k=1}^{m} N_*\left(d_{ik}^1, d_{jk}^2\right)$ of the numerator calculates the similarity between two documents coming from G_1 and G_2, respectively. The numerator is the total sum of these similarity values. The denominator serves the purpose of normalization. With the approximation, a set can be only represented by its center. It is not needed to store the information of the member patterns in a set. Similarity computation with a set is only done with its center instead of with all its member patterns. In general, if the patterns can be nicely clustered, i.e., the patterns of a cluster are close to each other, then the members of the cluster can be nicely represented by the center of the

cluster and the quality of the approximation is good. For document classification and clustering problems, this is usually the case.

Remark 2. If Eq.(23) is used, then obviously the computation of $F(G_1, G_2)$ between G_1 and G_2 is $O(q_1 q_2 m)$ in time complexity. If Eq.(30) is used instead, the means v_k^1 and v_k^2 are computed in $O(q_1 m)$ and $O(q_2 m)$, respectively. Therefore, the time complexity of computing $F(G_1, G_2)$ is $\max(O(q_1 m), O(q_2 m))$.

Let's give an example here. Suppose a training data set contains several document subsets, and G_1 and G_2 are two of them. G_1 consists of three documents:

$$\mathbf{d}_1^1 = < 3.6, 2.4, 0.2, 0.0, 0.0 >,$$
$$\mathbf{d}_2^1 = < 3.5, 2.5, 0.0, 0.0, 0.0 >,$$
$$\mathbf{d}_3^1 = < 3.4, 2.6, 0.1, 0.0, 0.0 >,$$

and G_2 consists of another three documents:

$$\mathbf{d}_1^2 = < 0.1, 0.0, 4.0, 1.9, 0.0 >,$$
$$\mathbf{d}_2^2 = < 0.1, 0.0, 3.9, 2.0, 0.0 >,$$
$$\mathbf{d}_3^2 = < 0.1, 0.0, 4.1, 2.1, 0.0 > .$$

Note that $q_1 = 3$ and $q_2 = 3$. Also, there are five features and each document is a 5-dimensional vector. We have $\delta_1^1 = 3$, $\delta_2^1 = 3$, $\delta_3^1 = 2$, $\delta_4^1 = 0$, $\delta_5^1 = 0$, $\delta_1^2 = 3$, $\delta_2^2 = 0$, $\delta_3^2 = 3$, $\delta_4^2 = 3$, and $\delta_5^2 = 0$. Assume $\lambda = 0.1$ and $\sigma_1 = \sigma_2 = \ldots = \sigma_5 = 2$. Then

- For feature w_1, $k = 1$:

$$\begin{aligned}
&\sum_{i=1}^{q_1}\sum_{j=1}^{q_2} N_*\left(d_{i1}^1, d_{j1}^2\right) \\
&= \sum_{d_{i1}^1>0}\sum_{d_{j1}^2>0}\left[0.5\left(1+\exp\left\{-\left(\frac{d_{i1}^1 - d_{j1}^2}{\sigma_1}\right)^2\right\}\right)\right] \\
&\quad +(-\lambda)\times\left(q_1-\delta_1^1\right)\delta_1^2 + (-\lambda)\times\delta_1^1\left(q_2-\delta_1^2\right) \\
&= 4.7522, \qquad (31)
\end{aligned}$$

- For feature w_3, $k = 3$:

$$\begin{aligned}
&\sum_{i=1}^{q_1}\sum_{j=1}^{q_2} N_*\left(d_{i3}^1, d_{j3}^2\right) \\
&= \sum_{d_{i3}^1>0}\sum_{d_{j3}^2>0}\left[0.5\left(1+\exp\left\{-\left(\frac{d_{i3}^1 - d_{j3}^2}{\sigma_3}\right)^2\right\}\right)\right] \\
&\quad +(-\lambda)\times(q_1-\delta_3^1)\delta_3^2 + (-\lambda)\times\delta_3^1(q_2-\delta_3^2) \\
&= 2.7749, \qquad (32)
\end{aligned}$$

Therefore, according to Eq.(21), we have

$$\begin{aligned}
F(G_1, G_2) &= \frac{4.7522 - 0.9 + 2.7749 - 0.9 + 0}{9+9+9+9+0} \\
&= 0.1591 \qquad (33)
\end{aligned}$$

and

$$\begin{aligned}
S_{SMTP}(G_1, G_2) &= \frac{F(G_1, G_2) + \lambda}{1+\lambda} \\
&= 0.2355. \qquad (34)
\end{aligned}$$

Next, we take approximations to Eq.(31) and Eq.(32) by Eq.(30). Note that $v_1^1 = 3.5$, $v_1^2 = 0.1$, $v_3^1 = 0.1$, and $v_3^2 = 4.0$. We have

$$\begin{aligned}
&\sum_{d_{i1}^1>0}\sum_{d_{j1}^2>0}\left[0.5\left(1+\exp\left\{-\left(\frac{d_{i1}^1 - d_{j1}^2}{\sigma_1}\right)^2\right\}\right)\right] \\
&\approx 0.5\left(1+\exp\left\{-\left(\frac{v_1^1 - v_1^2}{\sigma_1}\right)^2\right\}\right)\left(\delta_1^1\right)\left(\delta_1^2\right) \qquad (35) \\
&= 4.7502, \\
&\sum_{d_{i3}^1>0}\sum_{d_{j3}^2>0}\left[0.5\left(1+\exp\left\{-\left(\frac{d_{i3}^1 - d_{j3}^2}{\sigma_3}\right)^2\right\}\right)\right] \\
&\approx 0.5\left(1+\exp\left\{-\left(\frac{v_3^1 - v_3^2}{\sigma_3}\right)^2\right\}\right)\left(\delta_3^1\right)\left(\delta_3^2\right) \qquad (36) \\
&= 3.0672,
\end{aligned}$$

and

$$\sum_{i=1}^{q_1}\sum_{j=1}^{q_2} N_*\left(d_{i1}^1, d_{j1}^2\right) = 4.7502,$$
$$\sum_{i=1}^{q_1}\sum_{j=1}^{q_2} N_*\left(d_{i3}^1, d_{j3}^2\right) = 2.7672.$$

Therefore, we have

$$\begin{aligned}
F(G_1, G_2) &= \frac{4.7502 - 0.9 + 2.7672 - 0.9 + 0}{9+9+9+9+0} \\
&= 0.1588
\end{aligned}$$

and

$$S_{SMTP}(G_1, G_2) = 0.2352 \qquad (37)$$

which is very close to Eq.(34).

Remark 3. If $q_1 = q_2 = 1$, Eq.(21) reduces to Eq.(7). If either $q_1 = 1$ or $q_2 = 1$, Eq.(21) can be used to measure the similarity between a document and a document set.

Remark 4. The approximation proposed above can be applied easily to the Euclidean distance measure. Suppose the Euclidean distance between two sets G_1 and G_2 is defined as

$$\begin{aligned}
d_{Euc}(G_1, G_2) &= \frac{\sum_{i=1}^{q_1}\sum_{j=1}^{q_2}\sum_{k=1}^{m}\left(d_{ik}^1 - d_{jk}^2\right)^2}{q_1 q_2} \qquad (38) \\
&= \frac{\sum_{k=1}^{m}\sum_{i=1}^{q_1}\sum_{j=1}^{q_2}\left(d_{ik}^1 - d_{jk}^2\right)^2}{q_1 q_2}. \qquad (39)
\end{aligned}$$

Then, by following the same reasoning line adopted in the derivation of Eq.(23) and Eq.(30), $\sum_{i=1}^{q_1}\sum_{j=1}^{q_2}\left(d_{ik}^1 - d_{jk}^2\right)^2$ can

be approximated as

$$\begin{aligned}
&\sum_{i=1}^{q_1}\sum_{j=1}^{q_2}\left(d_{ik}^1 - d_{jk}^2\right)^2 \\
&= \sum_{d_{ik}^1>0}\sum_{d_{jk}^2>0}\left(d_{ik}^1 - d_{jk}^2\right)^2 + \sum_{d_{ik}^1=0}\sum_{d_{jk}^2>0}\left(d_{ik}^1 - d_{jk}^2\right)^2 + \\
&\quad \sum_{d_{ik}^1>0}\sum_{d_{jk}^2=0}\left(d_{ik}^1 - d_{jk}^2\right)^2 + \sum_{d_{ik}^1=0}\sum_{d_{jk}^2=0}\left(d_{ik}^1 - d_{jk}^2\right)^2 \\
&\approx \left(v_k^1 - v_k^2\right)^2\delta_k^1\delta_k^2 + \left(v_k^2\right)^2\left(q_1 - \delta_k^1\right)\delta_k^2 + \left(v_k^1\right)^2\delta_k^1\left(q_2 - \delta_k^2\right).
\end{aligned}$$

However, it is not so quite obvious to apply the proposed approximation to Cosine, Pairwise-adaptive, Extended Jaccard, and IT-Sim.

4 Experimental Results

In this section, we investigate the effectiveness of our proposed similarity measure SMTP. The investigation is done by applying our measure in several text applications, including k-NN based single-label classification (SL-kNN) [18], k-NN based multi-label classification (ML-kNN) [50], k-means clustering (k-means) [9], and hierarchical agglomerative clustering (HAC) [19]. We also compare the performance of SMTP with that of other five measures, Euclidean [45], Cosine [25], Extended Jaccard (EJ) [48], [49], Pairwise-adaptive (Pairwise) [17], and IT-Sim [8], [39] described in Section 2. Note that the percentage of features taken into account for the Pairwise-adaptive measure is set to be 100%. For the Pairwise-adaptive measure, K is determined by the product of the minimum number of non-zero features in the two documents and the percentage of features taken into account. For example, suppose we have two documents $\mathbf{d}_1 =< 0, 3, 0, 4, 2 >$ and $\mathbf{d}_2 =< 0, 2, 1, 0, 0 >$. The minimum number of non-zero features in these two documents is 2. Then we take $2\times100\% = 2$ largest features from $\mathbf{d}_1$ and $\mathbf{d}_2$, respectively. The features from $\mathbf{d}_1$ are feature 2 and feature 4, while the features from $\mathbf{d}_2$ are feature 2 and feature 3. The union of these features contains feature 2, feature 3, and feature 4. Therefore, $K = 3$, and $\mathbf{d}_{1,K} =< 3, 0, 4 >$ and $\mathbf{d}_{2,K} =< 2, 1, 0 >$. In this case, the results obtained by Pairwise-adaptive and Cosine are different. In the following, we use a computer with AMD FX(tm)-4100 Quad-Core Processor 3.6GHz, 8GB of RAM to conduct the experiments. The programming language used is MATLAB7.0.

4.1 Applications

A brief description for the four applications are given below.

1) SL-kNN. k-NN [18] is one of the most popular methods for single-label classification in which a document can belong to only one category. It classifies an unseen document by comparing it to its k nearest neighbors in a specified training set. Given a document $\mathbf{d}$, let D_k, with corresponding label set L_k, be a set containing the k most similar documents to $\mathbf{d}$. Then $\mathbf{d}$ is classified to class c which appears most frequently in L_k. A random choice is made when a tie occurs.
2) ML-kNN. ML-kNN [50] is an adaptation of k-NN for multi-label classification in which a document can belong to more than one category. An unseen document is labeled based on its k nearest neighbors using the maximum a posteriori estimate. For a document $\mathbf{d}$, let D_k, with corresponding label set L_k, be a set containing the k most similar documents to $\mathbf{d}$. If the probability that $\mathbf{d}$ belongs to class c given L_k is greater than the probability that $\mathbf{d}$ does not belong to class c given L_k, then $\mathbf{d}$ is classified to class c.
3) k-means. k-means [9] is one of the most popular methods which produce a single clustering. It requires the number of clusters, k, to be specified in advance. Initially, k clusters are specified. Then each document in the document set is re-assigned based on the similarity between the document and the k clusters. Then the k clusters are updated. Then all the documents in the document set are re-assigned. This process is iterated until the k clusters stay unchanged.
4) HAC. HAC [19] produces a sequence of clusterings of decreasing number of clusters at each step. The first clustering contains as many clusters as the number of documents in the document set, i.e., each cluster contains one distinct document. Then the second clustering is produced by merging two most similar clusters into one. This process continues until the final clustering is obtained, which contains only one cluster consisting of all the documents in the document set.

TABLE 1
Description of the Data Sets Used in the Experiments

Data Set	# of training documents	# of testing documents	# of features	# of classes
WebKB	2803	1396	7786	4
Reuters-8	5485	2189	17745	8
RCV1	15000	15000	47236	101

4.2 Data Sets

Three data sets, named WebKB [2], Reuters-8 [1], and RCV1 [38], respectively, are used in the experiments presented below. Table 1 shows some important characteristics of the three data sets.

Each data set is briefly described below.

1) WebKB. The documents in the WebKB data set are webpages collected by the World Wide Knowledge Base (Web→Kb) project of the CMU text learning group [13], [43]. The documents were manually classified into several different classes. The data set can be obtained from [2]. The documents of this data set were not predesignated as training or testing patterns. We divide them randomly into training and testing subsets. Among the 4199 documents, 2803 are randomly selected for training and the rest, 1396, are for testing. Table 2 shows the distribution of the documents in each class randomly selected

TABLE 2
Distribution of Documents per Class in WebKB

Class	# of training documents	# of testing documents	Subtotal of documents
Project	336	168	504
Course	620	310	930
Faculty	750	374	1124
Student	1097	544	1641
Total	2803	1396	4199

for training and testing, respectively. The number of features involved is 7786.

2) Reuters-8. Reuters-21578 ModeApt*è* Split Text Categorization Test Collection [3] contains thousands of documents collected from Reuters newswire in 1987. The most widely used version is Reuters-21578 ModeApt*è*, which contains 90 categories and 12902 documents. We use the 8 most frequent ones of the 90 categories and all the documents with less than or more than one topic are removed. The resulting data set is named Reuters-8 in which about 71% (5485/7674) of the documents were predesignated for training and the other documents, about 29% (2189/7674), were predesignated for testing. Table 3 shows the distribution of the documents in each class for training and testing. The data set can be obtained from [1]. The number of features involved is 17745.
3) RCV1. The RCV1 data set consists of 804414 news stories produced by Reuters from 20 Aug 1996 to 19 Aug 1997. There are 47236 features and 101 categories involved in this data set. We use 5 subsets of topics in LYRL2004 split defined in Lewis *et al*. [38]. The data set we use contains 30000 documents, of which 15000 were predesignated for training and the rest were predesignated for testing. The 5 subsets are arbitrarily named Subset1∼Subset5, as shown in Table 4. Each subset has 3000 training patterns and 3000 testing patterns. In Table 4, PMC denotes the percentage of documents belonging to more than one category and ANL denotes the average number of categories for each document. For example, in Subset1, 96.57% of the training documents belong to more than one category and each training document is assigned to 3.176 categories on average.

TABLE 3
Distribution of Documents per Class in Reuters-8

Class	# of training documents	# of testing documents	Subtotal of documents
acq	1596	696	2292
crude	253	121	374
earn	2840	1083	3923
grain	41	10	51
interest	190	81	271
money-fx	206	87	293
ship	108	36	144
trade	251	75	326
Total	5485	2189	7674

TABLE 4
Characteristics of the Five Subsets in RCV1

	Training		Testing	
RCV1	PMC(%)	ANL	PMC(%)	ANL
Subset1	96.57	3.176	96.83	3.277
Subset2	96.83	3.230	97.27	3.230
Subset3	96.90	3.187	96.70	3.243
Subset4	96.93	3.164	96.73	3.251
Subset5	97.17	3.207	96.43	3.215

4.3 Classification

For WebKB, the randomly selected training documents are used for training/validation and the testing documents are used for testing. For Reuters-8 and RCV1, the predesignated training data are used for training/validation and the predesignated testing data are used for testing. Note that the data for training/validation are separate from the data for testing in each case.

4.3.1 Single-Label Document Classification

In this experiment, we compare the performance of our measure and the others in single-label document classification. The performance is evaluated by the classification accuracy [46], *AC*, which compares the predicted label of each document with that provided by the document corpus:

$$AC = \frac{\sum_{i=1}^{n} E\left(c_i, c_i'\right)}{n}, \tag{40}$$

where n is the number of testing documents, and c_i and c_i' are the target label and the predicted label, respectively, of the ith document. $E(c_i, c_i') = 1$ if $c_i = c_i'$, and $E(c_i, c_i') = 0$ otherwise.

The single-label classifier SL-kNN is used with different similarity measures in this experiment. Fig. 1 shows the classification accuracy obtained by SL-kNN with our measure using ten different λ settings, i.e., $\lambda = 0.0001$, 0.001, 0.01, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, and 1.0, on the training/validation data of WebKB and Reuters-8, respectively, represented in word-count weighting. As shown in the figure, high λ values are better. In classification, individual patterns are compared one to one. For two patterns of the same category being compared, the case that one feature appears in one pattern but does not appear in the other pattern occurs less often than the case that one feature appears in both patterns. With a high value of λ, the former case is allowed to contribute as significantly to the similarity as the latter case. Therefore, high λ values are better for classification. We use the setting $\lambda = 1.0$ for SMTP to compare with other measures on testing accuracies. Note that the testing data are totally separate from the training/validation data. Table 5 shows the classification accuracies obtained by SL-kNN with different measures using different k values, i.e., $k = 1 \sim 15$, on the testing data of WebKB represented in word-count weighting. As can be seen from the table, SMTP performs better than the others in all cases. For example, when $k = 3$ SMTP achieves 0.8338 in accuracy, while Euclidean only gets 0.6705, EJ gets 0.7407, Cosine gets 0.7328, Pairwise gets 0.7249, and IT-Sim gets 0.8059, respectively. Table 6 shows the classification accuracies obtained by SL-kNN with different

TABLE 5
Classification Accuracies by SL-*k*NN with Different Measures on Testing Data of WebKB in Word-Count

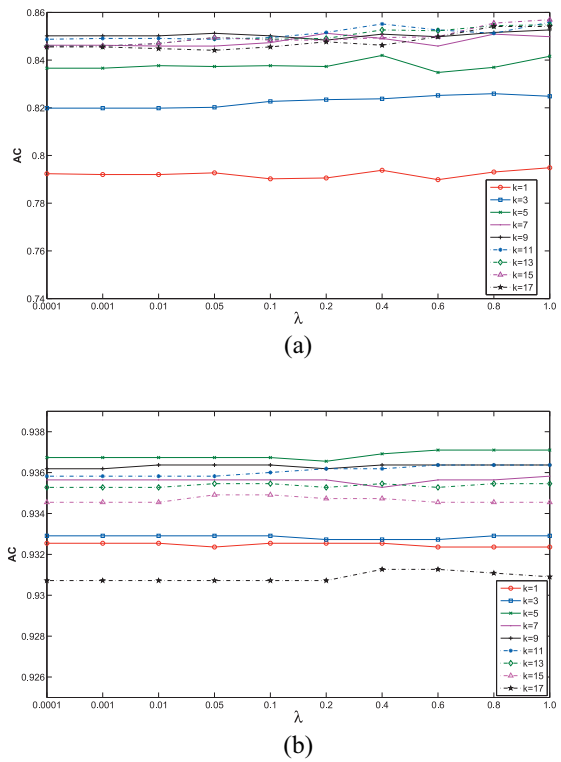
	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$	$k=11$	$k=13$	$k=15$
Euclidean	0.6777	0.6705	0.6884	0.6691	0.6605	0.6605	0.6605	0.6454
EJ	0.7056	0.7407	0.7564	0.7715	0.7715	0.7744	0.7708	0.7736
Cosine	0.6970	0.7328	0.7507	0.7593	0.7615	0.7701	0.7658	0.7622
Pairwise	0.6683	0.7249	0.7436	0.7521	0.7393	0.7486	0.7378	0.7393
IT-Sim	0.7636	0.8059	0.8266	0.8374	0.8324	0.8338	0.8360	0.8381
SMTP	**0.7837**	**0.8338**	**0.8467**	**0.8553**	**0.8553**	**0.8503**	**0.8517**	**0.8603**

TABLE 6
Classification Accuracies by SL-*k*NN with Different Measures on Testing Data of Reuters-8 in Word-Count

	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$	$k=11$	$k=13$	$k=15$
Euclidean	0.8799	0.8908	0.8872	0.8785	0.8821	0.8831	0.8799	0.8762
EJ	0.9137	0.9319	0.9315	0.9274	0.9246	0.9287	0.9296	0.9296
Cosine	0.9137	0.9265	0.9319	0.9287	0.9274	0.9278	0.9246	0.9296
Pairwise	0.9013	0.9191	0.9242	0.9223	0.9233	0.9214	0.9159	0.9159
IT-Sim	0.8963	0.9159	0.9333	0.9434	0.9411	0.9392	0.9402	0.9406
SMTP	**0.9338**	**0.9411**	**0.9420**	**0.9447**	**0.9461**	**0.9466**	**0.9434**	**0.9443**

measures using different k values, i.e., $k = 1 \sim 15$, on the testing data of Reuters-8 represented in word-count weighting. Again, SMTP performs better than the others in all cases.

Next, we work with the data sets in different formats. Table 7 shows the classification accuracies obtained by SL-*k*NN with different measures using different k values, i.e., $k = 1 \sim 15$, on the testing data of WebKB represented in tf-idf weighting. Table 8 shows the classification accuracies obtained by SL-*k*NN with different measures using different k values, i.e., $k = 1 \sim 15$, on the testing data of Reuters-8 represented in tf-idf weighting. From these two tables, we can see SMTP performs better than the others in all cases for WebKB and Reuters-8 in tf-idf weighting. Note that the accuracies shown in Tables 7 and 8 are very close to their counterparts listed in Tables 5 and 6. This shows that both tf-idf and word-count perform equally well in this application. Comparisons of Euclidean, EJ, and Cosine on the testing data of WebKB and Reuters-8 in z-score weighting are shown in Tables 9 and 10, respectively. Note that z-score adjusts the attributes to have zero means. As a result, zero and negative attribute values are resulted from the adjustment. In Pairwise, IT-Sim, and SMTP, zero values with an attribute indicate that this attribute does not appear in the corresponding documents. Therefore, these measures are not included in the comparisons with z-score. The accuracies shown in Tables 9 and 10 are much lower than their counterparts listed in Tables 5–8. Therefore, z-score is inferior to tf-idf and word-count in this application.



(a) (b)

Fig. 1. Classification accuracies by SL-*k*NN on training/validation data with SMTP using different λ values. (a) WebKB. (b) Reuters-8.

A comparison of SMTP with the other methods in terms of efficiency is shown in Table 11. In this table, the time is listed in seconds and is the average of k being 1, 3, ..., 15 each of which requires nearly identical time consumption. The data sets involved are represented in tf-idf weighting. Euclidean needs to do one inner product in each computation, while EJ and Cosine need to do three different inner products. Therefore, EJ and Cosine are about 3 times slower than Euclidean. Pairwise needs to do three inner products in each computation, but the length of the vectors can be shorter. For WebKB and Reuters-8, Pairwise is about 2.5 times slower than Euclidean. IT-Sim is about 6-10 times slower than Euclidean, while SMTP is about 2 times slower than Euclidean.

4.3.2 *Multi-Label Document Classification*

In this experiment, we compare the performance of our measure and the others in multi-label document classification. The performance is evaluated by microaveraged breakeven point (BEP) and microaveraged F1 (F1) [46]

TABLE 7
Classification Accuracies by SL-*k*NN with Different Measures on Testing Data of WebKB in tf-idf

	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$	$k=11$	$k=13$	$k=15$
Euclidean	0.4957	0.4570	0.4656	0.4527	0.4448	0.4341	0.4277	0.4226
EJ	0.6562	0.6812	0.7106	0.7235	0.7314	0.7450	0.7450	0.7586
Cosine	0.6476	0.6762	0.7077	0.7292	0.7199	0.7371	0.7436	0.7615
Pairwise	0.6332	0.6648	0.6927	0.7034	0.7120	0.7206	0.7206	0.7342
IT-Sim	0.7271	0.7672	0.7958	0.8109	0.8188	0.8181	0.8245	0.8281
SMTP	**0.7880**	**0.8238**	**0.8374**	**0.8438**	**0.8410**	**0.8467**	**0.8438**	**0.8453**

TABLE 8
Classification Accuracies by SL-*k*NN with Different Measures on Testing Data of Reuters-8 in tf-idf

	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$	$k=11$	$k=13$	$k=15$
Euclidean	0.7072	0.7282	0.7190	0.6976	0.6916	0.6834	0.6807	0.6738
EJ	0.7963	0.8191	0.8488	0.8607	0.8712	0.8789	0.8785	0.8808
Cosine	0.7921	0.8145	0.8415	0.8497	0.8547	0.8666	0.8698	0.8712
Pairwise	0.7775	0.8004	0.8273	0.8365	0.8406	0.8579	0.8598	0.8593
IT-Sim	0.7921	0.8342	0.8671	0.8803	0.8913	0.8981	0.9009	0.9041
SMTP	**0.9374**	**0.9383**	**0.9470**	**0.9466**	**0.9484**	**0.9488**	**0.9470**	**0.9475**

TABLE 9
Classification Accuracies by SL-*k*NN with Different Measures on Testing Data of WebKB in z-score

	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$	$k=11$	$k=13$	$k=15$
Euclidean	0.5595	0.5408	0.5372	0.5007	0.4685	0.4384	0.4234	0.4119
EJ	0.5731	0.5989	0.6454	**0.6769**	**0.6941**	**0.7020**	**0.7049**	**0.7077**
Cosine	**0.5860**	**0.6032**	**0.6468**	0.6719	0.6898	0.6941	0.6913	0.6870

TABLE 10
Classification Accuracies by SL-*k*NN with Different Measures on Testing Data of Reuters-8 in z-score

	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$	$k=11$	$k=13$	$k=15$
Euclidean	**0.6359**	0.6364	0.6035	0.5879	0.5674	0.5573	0.5473	0.5418
EJ	0.6067	0.6661	0.7177	**0.7661**	**0.7917**	**0.8168**	**0.8287**	**0.8360**
Cosine	0.6080	**0.6734**	**0.7273**	0.7615	0.7848	0.7967	0.8209	0.8237

defined as follows:

$$MicroP = \frac{\Sigma_{i=1}^{p} TP_i}{\Sigma_{i=1}^{p}(TP_i + FP_i)}, \quad (41)$$

$$MicroR = \frac{\Sigma_{i=1}^{p} TP_i}{\Sigma_{i=1}^{p}(TP_i + FN_i)}, \quad (42)$$

$$F1 = \frac{2 \times MicroP \times MicroR}{MicroP + MicroR}, \quad (43)$$

$$BEP = \frac{MicroP + MicroR}{2}, \quad (44)$$

where p is the number of categories. TP_i (true positives wrt c_i) is the number of c_i testing documents that are correctly classified to c_i. FP_i (false positives wrt c_i) is the number of non-c_i testing documents that are incorrectly classified to c_i. FN_i (false negatives wrt c_i) is the number of c_i testing documents that are incorrectly classified to non-c_i.

TABLE 11
Efficiency of SL-*k*NN on Testing Data with Different Measures

WebKB					
Euclidean	EJ	Cosine	Pairwise	IT-Sim	SMTP
946	3757	2889	2045	5348	1683
Reuters-8					
Euclidean	EJ	Cosine	Pairwise	IT-Sim	SMTP
942	2341	2231	2369	9669	1734

The multi-label classifier ML-*k*NN is used with different similarity measures in this experiment. As described in Section 4.3.1, we obtain the λ value according to the training/validation data. The setting $\lambda = 1.0$ is adopted for SMTP to compare with the other measures on testing accuracies. Note that the testing data used are totally separate from the training/validation data. Table 12 and Fig. 2(a) show the $F1$ values obtained by ML-*k*NN with different measures using different k values, i.e., $k = 8 \sim 12$, on the testing data of RCV1 in tf-idf weighting. In Table 12, the expression of $x \pm y$ in an entry denotes the statistical mean x and standard deviation y associated with the entry. Obviously, SMTP performs better than others in all cases. Table 13 and Fig. 2(b) show the *BEP* values obtained by ML-*k*NN with different measures using different k values, i.e., $k = 8 \sim 12$, on the testing data of RCV1 in tf-idf weighting. Again, they show the superiority of SMTP over the other measures. A comparison of SMTP with the other methods in terms of efficiency is shown in Table 14. We can see that SMTP runs as fast as Euclidean in this case. Note that RCV1 has a very high dimension and many attributes appear in one document but do not appear in other documents. Therefore, a lot of operations can be saved with SMTP.

Comparisons of Euclidean, EJ, and Cosine on the testing data of RCV1 in z-score weighting are shown in Table 15.

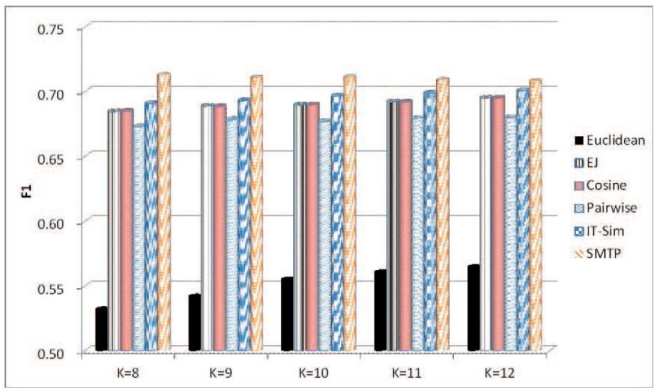
TABLE 12
*F*1 values by ML-*k*NN with Different Measures on Testing Data of RCV1 in tf-idf

	Euclidean	EJ	Cosine	Pairwise	IT-Sim	SMTP
$k = 8$	0.5320±0.0422	0.6847±0.0061	0.6850±0.0064	0.6729±0.0073	0.6910±0.0086	**0.7130**±0.0051
$k = 9$	0.5418±0.0324	0.6889±0.0070	0.6886±0.0072	0.6784±0.0076	0.6932±0.0068	**0.7111**±0.0056
$k = 10$	0.5551±0.0308	0.6898±0.0087	0.6900±0.0090	0.6771±0.0073	0.6965±0.0075	**0.7114**±0.0062
$k = 11$	0.5606±0.0256	0.6923± 0.0093	0.6920±0.0093	0.6796±0.0057	0.6990±0.0074	**0.7092**±0.0056
$k = 12$	0.5646±0.0331	0.6951±0.0060	0.6951±0.0057	0.6801±0.0067	0.7009±0.0038	**0.7083**±0.0070

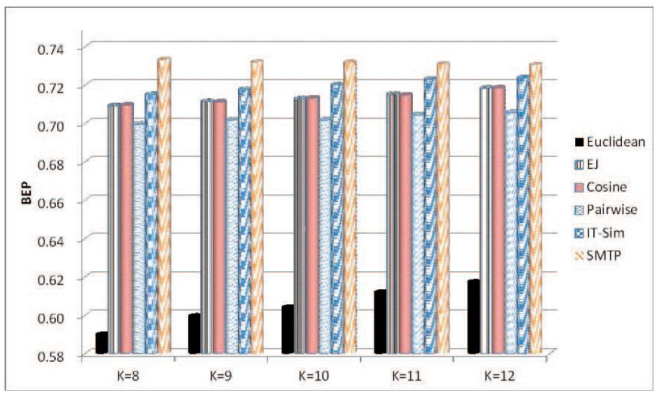
TABLE 13
BEP Values, by ML-*k*NN, with Different Measures on Testing Data of RCV1 in tf-idf

	Euclidean	EJ	Cosine	Pairwise	IT-Sim	SMTP
$k = 8$	0.5900± 0.0338	0.7093± 0.0031	0.7095±0.0030	0.6996±0.0053	0.7150±0.0040	**0.7334**±0.0022
$k = 9$	0.5997±0.0311	0.7114±0.0053	0.7112±0.0053	0.7019±0.0031	0.7176±0.0042	**0.7320**±0.0040
$k = 10$	0.6042±0.0208	0.7128±0.0044	0.7132±0.0051	0.7018±0.0024	0.7201±0.0037	**0.7320**±0.0043
$k = 11$	0.6121±0.0208	0.7153±0.0046	0.7149±0.0049	0.7046±0.0026	0.7230±0.0044	**0.7311**±0.0030
$k = 12$	0.6174±0.0204	0.7184±0.0029	0.7185±0.0027	0.7056±0.0037	0.7238±0.0028	**0.7307**±0.0036

For simplicity, we omit standard deviations in Table 15. From this table, Tables 12, and 13, we can see that z-score is inferior to tf-idf in this application.



(a)



(b)

Fig. 2. Classification performance comparison by ML-*k*NN with different measures on testing data of RCV1 in tf-idf. (a) *F*1. (b) *BEP*.

TABLE 14
Efficiency of ML-*k*NN with Different Measures on Testing Data of RCV1 in tf-idf

Euclidean	EJ	Cosine	Pairwise	IT-Sim	SMTP
1767	6579	5288	1846	17870	**1754**

4.4 Clustering

For a document corpus with p classes and n documents, we remove the class labels. Then we randomly selected one-third of the documents for training/validation and the remaining for testing. Note that the data for training/validation are separate from the data for testing.

4.4.1 *k*-Means Based Document Clustering

In this experiment, we compare the performance of our measure and the others employed in the *k*-means based clustering method. The accuracy, *AC*, and entropy, *En*, are adopted to gauge the clustering performance [10], [52] as follows:

$$AC = \frac{\sum_{i=1}^{k} Most_i}{n}, \tag{45}$$

$$En = \frac{\sum_{i=1}^{k} n_i (\sum_{j=1}^{p} -\frac{n_i^j}{n_i} \log \frac{n_i^j}{n_i})}{(\log p) n}, \tag{46}$$

where $Most_i$ is the majority number of documents having identical class labels in the ith cluster, n_i is the number of documents in the ith cluster, and n_i^j is the number of documents with label j in the ith cluster. In Eq.(46), $\sum_{j=1}^{p} -\frac{n_i^j}{n_i} \log \frac{n_i^j}{n_i}$ of the numerator calculates the randomness

TABLE 15
Classification Performance Comparison by ML-*k*NN with Different Measures on Testing Data of RCV1 in z-score

	$F1$				
	$k = 8$	$k = 9$	$k = 10$	$k = 11$	$k = 12$
Euclidean	0.3040	0.2918	0.2727	0.2875	0.2672
EJ	**0.5599**	0.5606	**0.5690**	**0.5822**	0.5762
Cosine	0.5513	**0.5630**	0.5660	0.5767	**0.5768**
	BEP				
	$k = 8$	$k = 9$	$k = 10$	$k = 11$	$k = 12$
Euclidean	0.3899	0.3860	0.3854	0.3974	0.4007
EJ	**0.6085**	**0.6118**	**0.6206**	**0.6261**	**0.6300**
Cosine	0.6073	0.6117	0.6160	0.6219	0.6276

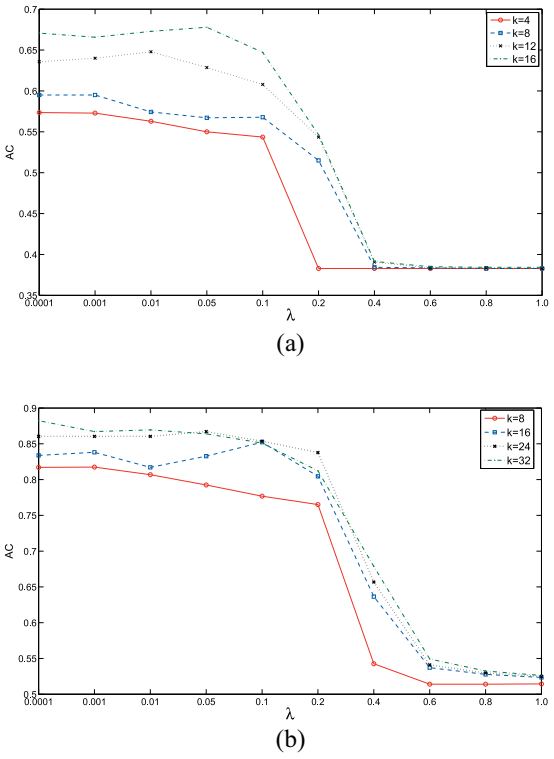


Fig. 3. Clustering accuracies by *k*-means on training/validation data with SMTP using different λ values. (a) WebKB. (b) Reuters-8.

of each cluster. The numerator gives the total sum of randomness contributed by all the clusters. The denominator purports to normalize the *En* value with maximum being 1. In general, a good clustering has a high value in *AC* and a low value in *En*. Fig. 3 shows the clustering accuracy obtained by *k*-means with our measure using ten different λ settings, i.e., λ = 0.0001, 0.001, 0.01, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, and 1.0, on the training/validation data of WebKB and Reuters-8, respectively, in word-count weighting. As shown in the figure, low λ values are better. In clustering, an individual pattern is compared with the centers of clusters. Each center is the aggregate of all its member patterns. For such one-to-many comparisons, the case that one feature appears in a pattern but does not appear in a center (or vice versa) occurs more often than the case that one feature appears in both. With a small value of λ, the latter case is allowed to contribute as significantly to the similarity as the former case. Therefore, low λ values are better for clustering. We use the setting λ = 0.0001 for SMTP to compare with other measures on testing accuracies. Note that the testing data are totally separate from the training/validation data. Tables 16 and 17 show the

TABLE 16
AC Values by *k*-means with Different Measures on Testing Data of WebKB in Word-Count

	$k=4$	$k=8$	$k=12$	$k=16$
Euclidean	0.4313	0.4715	0.4820	0.4790
EJ	0.3948	0.4022	0.4197	0.4645
Cosine	0.5585	0.5925	0.6124	0.6088
Pairwise	0.5641	0.5733	0.6030	0.5952
IT-Sim	**0.6777**	**0.6882**	**0.6889**	**0.7284**
SMTP	0.5933	0.6147	0.6831	0.7214

TABLE 17
En Values by *k*-means with Different Measures on Testing Data of WebKB in Word-Count

	$k=4$	$k=8$	$k=12$	$k=16$
Euclidean	0.9142	0.8733	0.8550	0.8396
EJ	0.9281	0.9052	0.8895	0.8225
Cosine	0.7140	0.6821	0.6732	0.6657
Pairwise	0.7216	0.6981	0.6854	0.6886
IT-Sim	**0.5960**	**0.5483**	**0.5367**	**0.5091**
SMTP	0.6770	0.6343	0.5552	0.5139

TABLE 18
AC Values by *k*-means with Different Measures on Testing Data of Reuters-8 in tf-idf

	$k=8$	$k=16$	$k=24$	$k=32$
Euclidean	0.5724	0.6288	0.7415	0.7682
EJ	0.8014	0.8529	0.8505	0.8536
Cosine	0.8294	0.8368	0.8611	0.8223
Pairwise	0.8196	0.8584	0.8692	0.8728
IT-Sim	**0.8450**	0.8698	0.8760	0.8747
SMTP	0.7906	**0.8702**	**0.8796**	**0.8964**

TABLE 19
En Values by *k*-means with Different Measures on Testing Data of Reuters-8 in tf-idf

	$k=8$	$k=16$	$k=24$	$k=32$
Euclidean	0.5252	0.4596	0.3559	0.2701
EJ	0.2531	0.1962	0.1896	0.1944
Cosine	0.2348	0.2054	0.1810	0.2114
Pairwise	0.2562	0.1922	0.1804	0.1752
IT-Sim	**0.2081**	**0.1740**	**0.1672**	0.1704
SMTP	0.3136	0.2072	0.1842	**0.1640**

AC and *En* values, respectively, obtained by *k*-means with $k = 4 \sim 16$ on the testing data of WebKB in word-count weighting. As can be seen from these tables, IT-Sim offers the best performance in *AC* and *En* for WebKB, and SMTP is the runner-up. Tables 18 and 19 show the *AC* and *En* values, respectively, obtained by *k*-means with $k = 8 \sim 32$ on the testing data of Reuters-8 in tf-idf weighting. For Reuters-8, SMTP offers the best performance in *AC* except for $k = 8$. Comparisons of Euclidean, EJ, and Cosine on the testing data of Reuters-8 in z-score weighting are shown in Table 20. From this table, Tables 18, and 19, we can see that z-score is inferior to tf-idf in this application.

A comparison of SMTP with the other methods in terms of efficiency is shown in Table 21. IT-Sim is very good

TABLE 20
Classification Performance Comparison by *k*-means with Different Measures on Testing Data of Reuters-8 in z-score

AC				
	$k=8$	$k=16$	$k=24$	$k=32$
Euclidean	0.5117	0.5139	0.5161	0.5241
EJ	0.6073	0.6107	0.6039	0.5952
Cosine	**0.6395**	**0.6649**	**0.6495**	**0.6421**
En				
	$k=8$	$k=16$	$k=24$	$k=32$
Euclidean	0.6423	0.6389	0.6351	0.6209
EJ	0.4698	0.4632	0.4578	0.4608
Cosine	**0.4411**	**0.4273**	**0.4348**	**0.4392**

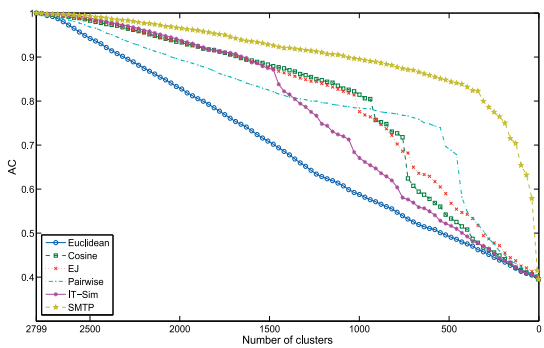
TABLE 21
Efficiency of *k*-means on Testing Data in tf-idf with Different Measures

WebKB ($k = 16$)					
Euclidean	EJ	Cosine	Pairwise	IT-Sim	SMTP
75	128	106	1103	684	1084
Reuters-8 ($k = 16$)					
Euclidean	EJ	Cosine	Pairwise	IT-Sim	SMTP
258	327	305	4036	2320	1631

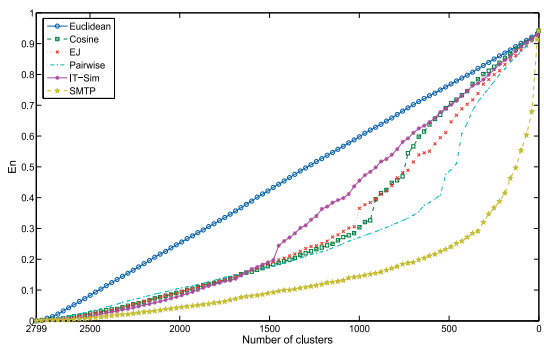
for WebKB. It achieves the best performance in *AC* and *En*, and runs fast. Cosine performs as well as Pairwise, but runs much faster. SMTP performs well, but runs slow. For Reuters-8, IT-Sim runs slow although it performs well. SMTP performs equally well as Pairwise and IT-Sim, but runs faster. EJ and Cosine not only perform well but also run very fast in this case.

4.4.2 Hierarchical Agglomerative Document Clustering

In this experiment, we compare the performance of our measure and the others employed in HAC. Fig. 4 shows the *AC* and *En* values, respectively, obtained by HAC on the testing data of WebKB in tf-idf weighting. Fig. 5 shows the *AC* and *En* values, respectively, obtained by HAC on the testing data of Reuters-8 in tf-idf weighting. As shown in these figures, SMTP performs significantly better in *AC* and *En* than the others. A comparison of SMTP with the other methods in terms of efficiency is shown in Table 22. Euclidean, EJ, and Cosine can run at least three times faster

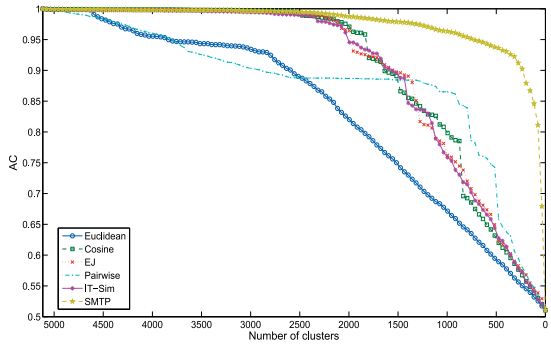


(a)

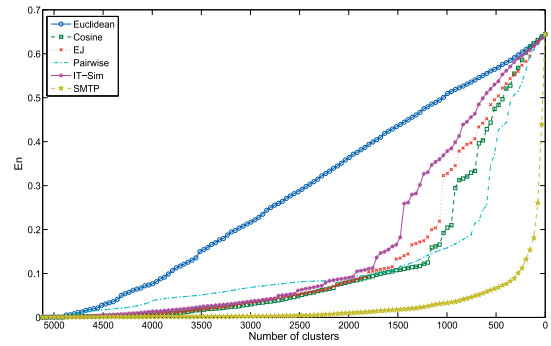


(b)

Fig. 4. Clustering performance by HAC on testing data of WebKB in tf-idf. (a) *AC*. (b) *En*.



(a)



(b)

Fig. 5. Clustering performance by HAC on testing data of Reuters-8 in tf-idf. (a) *AC*. (b) *En*.

than the other three measures, but they cannot perform well in *AC* and *En*. SMTP not only runs faster but also performs better in *AC* and *En* than Pairwise and IT-Sim for both WebKB and Reuters-8.

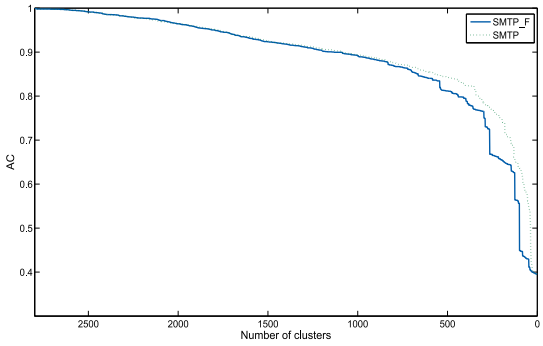
The results shown in Figs. 4 and 5 were based on Eq.(30) which, as mentioned, offers an approximation to Eq.(27). To see the effect of the approximation, we use Eq.(27) and redo the clustering work for WebKB and Reuters-8. The results are shown in Figs. 6 and 7, respectively. Note that the measure applying Eq.(27) is denoted as SMTP-F. As can be seen from these figure, the difference in *AC* and *En* between SMTP and SMTP-F is very small and can be

TABLE 22
Efficiency of HAC on Testing Data in tf-idf with Different Measures

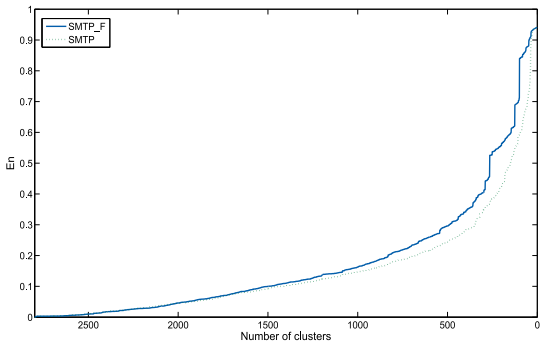
WebKB					
Euclidean	EJ	Cosine	Pairwise	IT-Sim	SMTP
681	960	990	5099	3154	2430
Reuters-8					
Euclidean	EJ	Cosine	Pairwise	IT-Sim	SMTP
2196	2572	2724	23035	11683	7239

TABLE 23
Clustering Efficiency Comparison between SMTP and SMTP-F on Testing Data of WebKB and Reuters-8 in tf-idf

	SMTP-F	SMTP
WebKB	**2430**	83221
Reuters-8	**7240**	131438

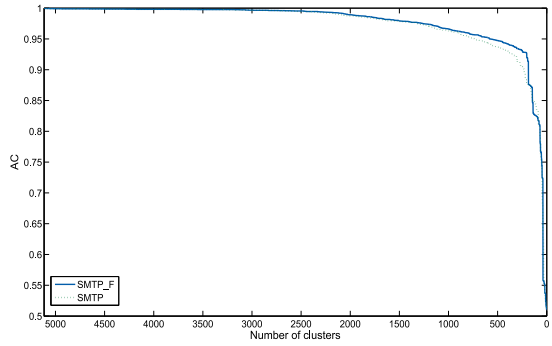


(a)

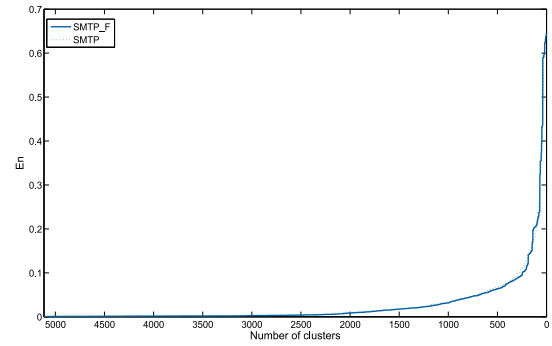


(b)

Fig. 6. Clustering performance by HAC with SMTP and SMTP-F on testing data of WebKB in tf-idf. (a) *AC*. (b) *En*.



(a)



(b)

Fig. 7. Clustering performance by HAC with SMTP and SMTP-F on testing data of Reuters-8 in tf-idf. (a) *AC*. (b) *En*.

negligible. However, the difference in computation time is very significant, as shown in Table 23 in terms of seconds. Obviously, SMTP runs much faster than SMTP-F.

5 Conclusion

We have presented a novel similarity measure between two documents. Several desirable properties are embedded in this measure. For example, the similarity measure is symmetric. The presence or absence of a feature is considered more essential than the difference between the values associated with a present feature. The similarity degree increases when the number of presence-absence feature pairs decreases. Two documents are least similar to each other if none of the features have non-zero values in both documents. Besides, it is desirable to consider the value distribution of a feature for its contribution to the similarity between two documents. The proposed scheme has also been extended to measure the similarity between two sets of documents. To improve the efficiency, we have provided an approximation to reduce the complexity involved in the computation. We have investigated the effectiveness of our proposed measure by applying it in *k*-NN based single-label classification, *k*-NN based multi-label classification, *k*-means clustering, and hierarchical agglomerative clustering (HAC) on several real-world data sets. The results have shown that the performance obtained by the proposed measure is better than that achieved by other measures.

The λ values used in the experiments are obtained through tuning based on the training/validation data which are separate from the testing data. In general, a high value of λ, e.g., 0.6$\sim$1.0, can be set for applications where many features appear commonly in the documents being compared, and a small value of λ, e.g., 0.01$\sim$0.0001, can be set for applications where many features appear in one document but not in other documents. In this work, we are focusing on the performance resulted from the application of different similarity measures in different classification/clustering algorithms. The algorithms and the data sets adopted are intended to be popular and easily accessible for anyone interested in this research area. However, it would be of greater value evaluating the performance of the measures on larger test-beds, e.g., DMOZ [4]. Also, this work mainly focuses on textural features. It would be interesting to investigate the effectiveness and efficacy of our proposed model in the scenarios that involve non-textual features and objects. Besides, as can be seen from the experimental results, the usefulness of a similarity measure could depend on (1) application domains, e.g., text or image, (2) feature formats, e.g., word count or tf-idf, and (3) classification/clustering algorithms. It would be a very interesting topic to examine how certain similarity measures behave in different classification/clustering tasks, such as that done in Fang *et al*. [20].

Acknowledgments

The authors are grateful to the anonymous reviewers for their comments, which were very helpful in improving the quality and presentation of the paper. This work was

supported in part by the National Science Council under the grants NSC-99-2221-E-110-064-MY3 and NSC99-2622-E-110-007-CC3, and in part by the "Aim for the Top University Plan" of the National Sun Yat-Sen University and Ministry of Education.

REFERENCES

[1] [Online]. Available: http://web.ist.utl.pt/~acardoso/datasets/

[2] [Online]. Available: http://www.cs.technion.ac.il/~ronb/thesis.html

[3] [Online]. Available: http://www.daviddlewis.com/resources/testcollections/reuters21578/

[4] [Online]. Available: http://www.dmoz.org/

[5] P. K. Agarwal and C. M. Procopiuc, "Exact and approximation algorithms for clustering," in *Proc. 9th Annu. SODA*, Philadelphia, PA, USA, 1998, pp. 658–667.

[6] D. W. Aha, "Lazy learning: Special issue editorial," *Artif. Intell. Rev.*, vol. 11, no. 1–5, pp. 7–10, 1997.

[7] G. Amati and C. J. V. Rijsbergen, "Probabilistic models of information retrieval based on measuring the divergence from randomness," *ACM Trans. Inform. Syst.*, vol. 20, no. 4, pp. 357–389, 2002.

[8] J. A. Aslam and M. Frost, "An information-theoretic measure for document similarity," in *Proc. 26th SIGIR*, Toronto, ON, Canada, 2003, pp. 449–450.

[9] G. H. Ball and D. J. Hall, "A clustering technique for summarizing multivariate data," *Behav. Sci.*, vol. 12, no. 2, pp. 153–155, 1967.

[10] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.

[11] H. Chim and X. Deng, "Efficient phrase-based document similarity for clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 9, pp. 1217–1229, Sept. 2008.

[12] S. Clinchant and E. Gaussier, "Information-based models for ad hoc IR," in *Proc. 33rd SIGIR*, Geneva, Switzerland, 2010, pp. 234–241.

[13] M. Craven *et al.*, "Learning to extract symbolic knowledge form the world wide web," in *Proc. 15th Nat. Conf. Artif. Intell.*, Menlo Park, CA, USA, 1998.

[14] I. S. Dhillon, J. Kogan, and C. Nicholas, "Feature selection and document clustering," in *A Comprehensive Survey of Text Mining*, M. W. Berry, Ed. Heidelberg, Germany: Springer, 2003.

[15] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Mach. Learn.*, vol. 42, no. 1, pp. 143–175, 2001.

[16] I. S. Dhillon, S. Mallela, and R. Kumar, "A divisive infromation-theoretic feature clustering algorithm for text classification," *J. Mach. Learn. Res.*, vol. 3, pp. 1265–1287, Mar. 2003.

[17] J. D'hondt, J. Vertommen, P.-A. Verhaegen, D. Cattrysse, and J. R. Duflou, "Pairwise-adaptive dissimilarity measure for document clustering," *Inf. Sci.*, vol. 180, no. 12, pp. 2341–2358, 2010.

[18] R. O. Duda, P. E. Hart, and D. J. Stork, *Pattern Recognition.* New York, NY, USA: Wiley, 2001.

[19] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Sci.*, vol. 95, no. 25, pp. 14863–14868, 1998.

[20] H. Fang, T. Tao, and C. Zhai, "A formal study of heuristic retrieval constraints," in *Proc. 27th SIGIR*, Sheffield, South Yorkshire, U.K., 2004, pp. 49–56.

[21] C. G. González, W. Bonventi, Jr., and A. L. V. Rodrigues, "Density of closed balls in real-valued and autometrized boolean spaces for clustering applications," in *Proc. 19th Brazilian Symp. Artif. Intell.*, Savador, Brazil, 2008, pp. 8–22.

[22] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, 1950.

[23] K. M. Hammouda and M. S. Kamel, "Efficient phrase-based document indexing for web document clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 10, pp. 1279–1296, Oct. 2004.

[24] K. M. Hammouda and M. S. Kamel, "Hierarchically distributed peer-to-peer document clustering and cluster summarization," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 5, pp. 681–698, May 2009.

[25] J. Han and M. Kamber, *Data Mining: Concepts and Techniques,* 2nd ed. San Francisco, CA, USA: Morgan Kaufmann; Boston, MA, USA: Elsevier, 2006.

[26] T. Joachims, "A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization," in *Proc. 14th Int. Conf. Mach. Learn.*, San Francisco, CA, USA, 1997, pp. 143–151.

[27] T. Joachims and F. Sebastiani, "Guest editors' introduction to the special issue on automated text categorization," *J. Intell. Inform. Syst.*, vol. 18, no. 2/3, pp. 103–105, 2002.

[28] T. Kanungo *et al.*, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.

[29] H. Kim, P. Howland, and H. Park, "Dimension reduction in text classification with support vector machines," *J. Mach. Learn. Res.*, vol. 6, pp. 37–53, Jan. 2005.

[30] S.-B. Kim, K.-S. Han, H.-C. Rim, and S. H. Myaeng, "Some effective techniques for naive bayes text classification," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 11, pp. 1457–1466, Nov. 2006.

[31] K. Knight, "Mining online text," *Commun. ACM*, vol. 42, no. 11, pp. 58–61, 1999.

[32] J. Kogan, C. Nicholas, and V. Volkovich, "Text mining with information-theoretic clustering," *Comput. Sci. Eng.*, vol. 5, no. 6, pp. 52–59, 2003.

[33] J. Kogan, M. Teboulle, and C. K. Nicholas, "Data driven similarity measures for *k*-means like clustering algorithms," *Inform. Retrieval*, vol. 8, no. 2, pp. 331–349, 2005.

[34] S. Kolliopoulos and S. Rao, "A nearly linear-time approximation scheme for the Euclidean *k*-median problem," in *Proc. 7th Annu. ESA*, Prague, Czech Republic, 1999, pp. 362–371.

[35] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annu. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[36] S.-J. Lee and C.-S. Ouyang, "A neuro-fuzzy system modeling with self-constructing rule generation and hybrid SVD-based learning," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 3, pp. 341–353, Jun. 2003.

[37] V. Lertnattee and T. Theeramunkong, "Multidimensional text classification for drug information," *IEEE Trans. Inform. Technol. Biomed.*, vol. 8, no. 3 pp. 306–312, Sept. 2004.

[38] D. D. Lewis, Y. Yang, T. Rose, and F. Li, "RCV1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, Apr. 2004.

[39] D. Lin, "An information-theoretic definition of similarity," in *Proc. 15th Int. Conf. Mach. Learn.*, San Francisco, CA, USA, 1998.

[40] M. G. Michie, "Use of the bray-curtis similarity measure in cluster analysis of foraminiferal data," *Math. Geol.*, vol. 14, no. 6, pp. 661–667, 1982.

[41] T. Mitchell, *Machine Learning.* Boston, MA, USA: McGraw-Hill, 1997.

[42] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Mach. Learn.*, vol. 39, no. 2/3, pp. 103–134, 2000.

[43] K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell, "Learning to classify text from labeled and unlabeled documents," in *Proc. 15th Nat. Conf. Artif. Intell.*, Menlo Park, CA, USA, 1998.

[44] G. Salton and M. J. McGill, *Introduction to Modern Retrieval*. London, U.K.: McGraw-Hill, 1983.

[45] T. W. Schoenharl and G. Madey, "Evaluation of measurement techniques for the validation of agent-based simulations against streaming data," in *Proc. ICCS*, Kraków, Poland, 2008.

[46] F. Sebastiani, "Machine learning in automated text categorization," *ACM CSUR*, vol. 34, no. 1, pp. 1–47, 2002.

[47] C. Silva, U. Lotric, B. Ribeiro, and A. Dobnikar, "Distributed text classification with an ensemble kernel-based learning approach," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 3, pp. 287–297, May 2010.

[48] A. Strehl and J. Ghosh, "Value-based customer grouping from large retail data-sets," in *Proc. SPIE*, vol. 4057. Orlando, FL, USA, Apr. 2000, pp. 33–42.

[49] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining.* Boston, MA, USA: Addision-Wesley, 2006.

[50] M. L. Zhang and Z. H. Zhou, "ML-*k*NN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007.

[51] T. Zhang, Y. Y. Tang, B. Fang, and Y. Xiang, "Document clustering in correlation similarity measure space," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 6, pp. 1002–1013, Jun. 2012.
[52] Y. Zhao and G. Karypis, "Comparison of agglomerative and partitional document clustering algorithms," in *Proc. Workshop Clustering High Dimensional Data Its Appl. 2nd SIAM ICDM*, 2002, pp. 83–93.

Yung-Shen Lin received the B.S. degree in survey engineering from National Defense University, Taiwan, in 1986 and the M.S. degree in information engineering from I-SHOU University, Taiwan, in 2004. He is currently pursuing the Ph.D. degree in electrical engineering at the National Sun Yat-Sen University, Taiwan. His current research interests include machine learning and data mining.

Jung-Yi Jiang received the B.S. degree from I-SHOU University, Taiwan, in 2002, and the M.S and Ph.D. degrees in electrical engineering in 2004 and 2011, respectively, from the National Sun Yat-Sen University, Taiwan. He is currently a Post-Doctorate at the Center for Research of E-life Digital Technology, National Cheng Kung University, Taiwan. His current research interests include machine learning, data mining, and information retrieval.

Shie-Jue Lee received the B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1977 and 1979, respectively, and the Ph.D. degree in computer science from the University of North Carolina, Chapel Hill, NC, USA, in 1990. Dr. Lee joined the faculty of the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan, in 1983, and has been a Professor with the department since 1994. His current research interests include artificial intelligence, machine learning, data mining, information retrieval, and soft computing. Prof. Lee was the recipient of the Best Paper Award in several international conferences. He has also been awarded the Distinguished Teachers Award of the Ministry of Education, in 1993, the Distinguished Research Award in 1998, the Distinguished Teaching Award in 1993 and 2008, and the Distinguished Mentor Award in 2008, all from National Sun Yat-Sen University. He served as the Program Chair for several international conferences. He was the Director of the Southern Telecommunications Research Center, National Science Council, from 1998 to 1999, the Chair of the Department of Electrical Engineering from 2000 to 2003, and the Deputy Dean of the Academic Affairs from 2008 to 2011. He is now the Director of the NSYSU-III Research Center and the Vice President of Library and Information Services, National Sun Yat-Sen University. He is a member of the IEEE.