Algorithms of Nonlinear Document Clustering based on Fuzzy Multiset Model

Kiyotaka Mizutani
Graduate School of Systems and Information Engineering,
University of Tsukuba, Ibaraki 305-8573, Japan

Ryo Inokuchi
Graduate School of Systems and Information Engineering,
University of Tsukuba, Ibaraki 305-8573, Japan

Sadaaki Miyamoto
Department of Risk Engineering, School of Systems and Information Engineering,
University of Tsukuba, Ibaraki 305-8573, Japan

Abstract: Fuzzy multiset is applicable as a model of information retrieval because it has the mathematical structure which expresses the number and the degree of attribution of an element simultaneously. Therefore fuzzy multisets can be used also as a suitable model for document clustering. This paper aims at developing clustering algorithms based on a fuzzy multiset model for document clustering. The standard proximity measure of the cosine correlation is generalized in the multiset model and two nonlinear clustering techniques are applied to the existing clustering methods. One introduces a variable for controlling cluster volume sizes; the other one is a kernel trick used in Support Vector Machines. Moreover clustering by competitive learning is also studied. When the kernel trick has been used the classification configuration of data in a high-dimensional feature space is visualized by Self-Organizing Maps. Two numerical examples which use an artificial data and real document data are shown and effects of the proposed methods are discussed.

I. Introduction

Data clustering has frequently been discussed in relation to information retrieval models,[7, 17, 18] on the other hand fuzzy multisets provide an appropriate model of information retrieval process on the Web. Fuzzy multiset (also called fuzzy bag) has been proposed by Yager[20] and basic relations and operations of fuzzy multisets have been redefined by Miyamoto.[6, 8–10]

The theory of fuzzy multisets is a mathematical framework which can represent multiple occurrences of a subject item with degrees of relevance and it has been studied in relation to a variety of information systems including relational database. Since there are few foregoing studies of applying fuzzy multisets to information retrieval, it is thus necessary to apply this theory to establishing its mathematical model.

Clustering algorithms based on fuzzy multiset model for information retrieval have been studied by Miyamoto[11] and some algorithms (e.g., crisp c-means and fuzzy c-means) have been proposed.[12, 16]

It is well-known that these methods of clustering give linear cluster boundaries. However, in real-world situations methods of separating classes having nonlinear boundaries are more appropriate.

To obtain nonlinear cluster boundaries, two nonlinear clustering techniques are considered. One introduces a variable for controlling cluster volume sizes and the other one is a kernel trick employed in Support Vector Machines (SVM),[19] where original data are mapped into a high-dimensional feature space; data are classified linearly in the feature space but in the original data space the boundary is nonlinear.

Although clustering algorithms using the kernel trick have been studied,[3,14] there is no method of knowing how data are separated in a feature space, since we cannot observe the feature space directly. Therefore, we propose a method to visualize how data are separated in the feature space by Self-Organizing Maps (SOM)[5] using the kernel trick.

This paper is organized as follows. We first briefly review fuzzy multiset model for document clustering, introducing new multiset operations for this purpose. Second, three classes of algorithms of clustering and two methods handling nonlinearities are considered. The three algorithms are crisp c-means, fuzzy c-means and clustering by competitive learning, while the two methods handling nonlinearity are above two techniques. Moreover, we visualize how data are separated in the feature space by a kernel-based SOM.

Two numerical examples which use artificial data and real document data are shown and the effectiveness of the proposed algorithms is discussed.

II. Fuzzy Multiset Model for Document Clustering

In this section we assume that the universal set is denoted by X which is finite. We write $X = \{x_1, x_2, \ldots, x_n\}$ or $X = \{x, y, \ldots, z\}$. Scalars are denoted by $a, b, \ldots$

A. Fuzzy multisets

As Yager[20] has defined, a fuzzy multiset A of X is characterized by the count function $C_A(\cdot)$ that takes the value of a multiset of the unit interval $I = [0, 1]$. Namely,

$$C_A(x) = \{\nu, \nu', \ldots, \nu''\}, \quad \nu, \nu', \ldots, \nu'' \in I.$$

For simplicity we assume $\nu, \nu', \ldots, \nu''$ are nonzero. We write

$$A = \{\{\nu, \nu', \ldots, \nu''\}/x, \ldots\} \quad \text{or} \quad A = \{(x, \nu), (x, \nu'), \ldots, (x, \nu''), \ldots\}.$$

As a data structure, we introduce an infinite-dimensional vector:

$$C_A(x) = (\nu, \nu', \ldots, \nu'', 0, 0, \ldots).$$

Collection of such vectors is denoted by $\mathcal{V}$:

$$\mathcal{V} = \{(\nu, \nu', \ldots, \nu'', 0, 0, \ldots) : \nu, \nu', \ldots, \nu'' \in I\}.$$

A sorting operation to multisets in I is important in defining operations for fuzzy multisets. These operations denoted by S ($S : \mathcal{V} \to \mathcal{V}$) rearrange the sequence in $\mathcal{V}$ into the decreasing order:

$$S(\nu, \nu', \dots, \nu'', 0, 0, \dots) = (\mu^1, \mu^2, \dots, \mu^p, 0, 0, 0)$$

where $\mu^1 \geq \mu^2 \geq \cdots \geq \mu^p > 0$ and $\{\nu, \nu', \dots, \nu''\} = \{\mu^1, \mu^2, \dots, \mu^p\}$. Thus we can assume

$$C_A(x) = \{\mu^1, \mu^2, \dots, \mu^p, 0, 0, 0\}. \tag{1}$$

The above sorted sequence for $C_A(x)$ is called the standard form for a fuzzy multiset, as many operations are defined in terms of the standard form.[6,10]

Several, but not all, basic operations of fuzzy multisets are defined as follows. These operations of fuzzy multisets herein use a new notation of the sorting operation $S(\cdot)$. Consequently representation of the basic operations are made more compact than those in former studies.[6,8–10]

1. **inclusion:**

$$A \subseteq B \Longleftrightarrow S(C_A(x)) \leq S(C_B(x)),\ \forall x \in X.$$

2. **equality:**

$$A = B \Longleftrightarrow S(C_A(x)) = S(C_B(x)),\ \forall x \in X.$$

3. **union:**

$$C_{A \cup B}(x) = S(C_A(x)) \vee S(C_B(x)).$$

4. **intersection:**

$$C_{A \cap B}(x) = S(C_A(x) \wedge S(C_B(x)).$$

5. **sum:**

$$C_{A+B}(x) = S(S(C_A(x)) \mid S(C_B(x))).$$

6. **product:**

$$C_{A \cdot B}(x) = S(C_A(x)) \cdot S(C_B(x)).$$

7. **cardinality:**

$$|A| = \sum_{x \in X} |C_A(x)|.$$

Although the fuzzy multiset is characterized by the membership sequence, it doesn't have the vector-valued membership. When the length of two membership sequences is unequal, the zero element is added that the length of the two membership sequences becomes the same.

Example: Suppose $X = \{a, b, c\}$ and

$$A = \{(0.7, 0.4, 0.1)/a, (0.5, 0.3)/b, (0.9, 0.2)/c\},$$
$$B = \{(0.6, 0.5)/a, (0.7)/b, (0.6)/c\}$$

are fuzzy multisets of X. Then we have

$$A \cup B = \{(0.7, 0.5, 0.1)/a, (0.7, 0.3)/b, (0.9, 0.2)/c\},$$
$$A \cap B = \{(0.6, 0.4)/a, (0.5)/b, (0.6)/c\},$$
$$A + B = \{(0.7, 0.6, 0.5, 0.4, 0.1)/a, (0.7, 0.5, 0.3)/b, (0.9, 0.6, 0.2)/c\},$$
$$A \cdot B = \{(0.42, 0.2)/a, (0.35)/b, (0.54)/c\}.$$

B. Fuzzy multiset space

We have assumed $C_A(x)$ is finite at the beginning. However extension to infinite fuzzy multisets is straightforward: $C_A(x) = (\mu^1, \dots, \mu^p, \dots)$ in which we can arrow infinite nonzero elements. A reasonable assumption to this sequence is $\mu^j \to 0$ $(j \to +\infty)$.

Metric spaces are defined on the collection of fuzzy multisets of X. Let

$$C_A(x) = (\mu_A^1, \dots, \mu_A^p, \dots), \quad C_B(x) = (\mu_B^1, \dots, \mu_B^p, \dots).$$

Then we can define

$$d_1(A, B) = \sum_{x \in X} \sum_{j=1}^{\infty} |\mu_A^j - \mu_B^j|,$$

which is an ℓ_1 metric. Moreover we can also define

$$d_2(A, B) = \sqrt{\sum_{x \in X} \sum_{j=1}^{\infty} |\mu_A^j - \mu_B^j|^2},$$

as the ℓ_2 type metric. Moreover a scalar product $\langle A, B \rangle$ using the algebraic product is introduced in the latter space:

$$\langle A, B \rangle = \sum_{x \in X} C_{A \cdot B}(x).$$

Then we have

$$d_2(A, B)^2 = \langle A, A \rangle + \langle B, B \rangle - 2\langle A, B \rangle.$$

The $d_2(A, B)$ metric naturally induces a norm:

$$\|A\| = d_2(A, \emptyset).$$

It is not difficult to see the metric space with d_1 is extended to a Banaach space and that with $\langle A, B \rangle$ a Hilbert space, since it is straightforward to define $cnst \cdot A$: multiplication of A by a constant $cnst$. These metric are useful in discussing fuzzy multiset model for data clustering.

III. Nonlinear Clustering Methods

It is well-known that the methods of crisp c-means and fuzzy c-means clustering provide linear cluster boundaries. However, it is also known that real-world examples require methods of separating classes having nonlinear boundaries.

In the following we distinguish two kinds of nonlinearities. First kind is called "*mild nonlinearity*" and the second is called "*strong nonlinearity*". A typical mild nonlinearity occurs as follows. Let us suppose two spherical clusters of objects are given. A cluster is large while the other is small. An algorithm of c-means clustering can separate the two clusters, but if the two clusters are close enough, a part of the larger cluster is misclassified into the smaller cluster, since the boundary by the clustering algorithm is the Voronoi boundary of the two regions with the two cluster centers. Such a nonlinearity can be handled by using an additional variable for controlling cluster volume sizes[15] which will be discussed below.

There are, however, nonlinearities that cannot be handled in this way. For handling such strong nonlinearities, we employ a kernel trick in SVM.[19]

A. Preliminaries

The objects to be clustered herein may be documents, or they can be items of information on the web. We call them documents or simply objects; they are denoted by $X = \{x_1, \ldots, x_n\}$. A document x_k is identified with the corresponding fuzzy multiset, namely, the same symbol x_k is used for both the document and the fuzzy multiset. We use the distance d_2 which is regarded as a Hilbert space with the scalar product $\langle x_k, x_\ell \rangle$.

The proximity measure used for clustering is a generalization of the cosine correlation in the fuzzy multiset space:[12,16]

$$s(x_k, x_\ell) = \frac{\langle x_k, x_\ell \rangle}{\|x_k\| \|x_\ell\|}. \tag{2}$$

1. Crisp c-means

The method of crisp c-means[1,2] has been known to be the best known technique of data clustering whereby a set of data is classified using mutual distance or correlation between a pair objects. The method of crisp c-means is alternative minimization of an objective function.

We consider the following objective function. Here, the subscript *ccm* indicates the objective function of crisp c-means:

$$J_{ccm}(U, V) = -\sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik} s(x_k, v_i).$$

$U = (u_{ik})$ is the membership matrix: u_{ik} is the degree by which x_k belongs to cluster i and it takes value of 0 or 1; the matrix U should be subject to the constraint

$$M_c = \{\, (u_{ik}) \ : u_{ik} \in \{0,1\},\ \sum_i u_{ik} = 1,\ \forall k\ ;\ u_{jk} \geq 0,\ \forall j,k\ \}.$$

$V = (v_1, \ldots, v_c)$; v_i is the center of cluster i, $1 \leq i \leq c$. Notice that v_i is a fuzzy multiset.

The iterative solutions for the clustering are obtained by alternative minimization of this function: $\min_{U \in M_c} J_{ccm}(U,V)$ while fixing V to be the last minimizing element, and $\min_V J_{ccm}(U,V)$ while fixing U to be the last minimizing element.

The optimal solutions of crisp c-means are calculated as follows.

$$u_{ik} = 1 \Longleftrightarrow i = \arg\max_{1 \leq j \leq c} s(x_k, v_j),$$
$$u_{ik} = 0 \Longleftrightarrow i \neq \arg\max_{1 \leq j \leq c} s(x_k, v_j),$$

and

$$v_i = \frac{\sum_{k=1}^{n} u_{ik} x_k}{\| \sum_{k=1}^{n} u_{ik} x_k \|}.$$

2. *Fuzzy c-means*

We use the entropy-based objective function which has been proposed by Miyamoto and Mukaidono[13] instead of the standard objective function by Bezdek[1] for fuzzy c-means clustering. Here, the subscript $efcm$ indicates the objective function of entropy-based fuzzy c-means:

$$J_{efcm}(U,V) = -\sum_{i=1}^{c}\sum_{k=1}^{n}[u_{ik} s(x_k, v_i) - \lambda^{-1} u_{ik} \log u_{ik}]$$

where λ is a positive constant. The constraint of the matrix U of fuzzy c-means is

$$M_f = \{\, (u_{ik}) \ : u_{ik} \in [0,1]\ \sum_i u_{ik} = 1,\ \forall k\ ;\ u_{jk} \geq 0,\ \forall j,k\ \}.$$

The iterative solutions for the clustering are obtained by alternative minimization of the objective function as well as crisp c-means.

The optimal solutions of fuzzy c-means can be obtained by the Lagrange multiplier method; they are given by

$$u_{ik} = \frac{\exp(\lambda s(x_k, v_i))}{\sum_{j=1}^{c} \exp(\lambda s(x_k, v_j))}, \qquad v_i = \frac{\sum_{k=1}^{n} u_{ik} x_k}{\| \sum_{k=1}^{n} u_{ik} x_k \|}.$$

The metrics on fuzzy multiset spaces are discussed and algorithms for calculating cluster centers have been considered by Miyamoto.[11] It has been proved there that the centers are well-defined fuzzy multisets. It should also be noted, however, that nonlinearities discussed below have not been considered yet.

B. Variable for controlling cluster volume

We introduce a variable for controlling cluster volume sizes into the above objective functions for *mild nonlinearity*.[15,16]

1. *Fuzzy c-means with variable for controlling cluster volume*

We consider the next objective function with which a variable is added to the entropy term:

$$J_{efcma}(U, V, \alpha) = -\sum_{i=1}^{c} \sum_{k=1}^{n} [u_{ik} s(x_k, v_i) - \lambda^{-1} u_{ik} \log u_{ik}/\alpha_i] \tag{3}$$

where $\alpha = (\alpha_1, \ldots, \alpha_c)$ is c-dimensional variable for controlling cluster volume sizes, which subjects to the constraint:

$$\mathcal{A} = \{\, \alpha \;:\; \sum_i \alpha_i = 1 \;;\; \alpha_i \geq 0,\; i = 1, \cdots, c \,\}.$$

The following algorithm **FCMA** is fuzzy c-means clustering algorithm in which the variable for controlling cluster volume sizes.

Algorithm FCMA.

FCMA0. Set initial values of $(\bar{U}, \bar{V}, \bar{\alpha})$.

FCMA1. Solve $\min_{U \in M_f} J(U, \bar{V}, \bar{\alpha})$ and let the optimal solution be new $\bar{U}$.

FCMA2. Solve $\min_{V} J(\bar{U}, V, \bar{\alpha})$ and let the optimal solution be new $\bar{V}$.

FCMA3. Solve $\min_{\alpha \in \mathcal{A}} J(\bar{U}, \bar{V}, \alpha)$ and let the optimal solution be new $\bar{\alpha}$.

FCMA4. If the solution is convergent, stop; else go to step **FCMA1**.

End of FCMA.

The optimal solutions of **FCMA** can be obtained by the Lagrange multiplier method; they are given by

$$u_{ik} = \frac{\alpha_i \exp(\lambda s(x_k, v_i))}{\sum_{j=1}^{c} \alpha_j \exp(\lambda s(x_k, v_j))}, \tag{4}$$

$$v_i = \frac{\sum_{k=1}^{n} u_{ik} x_k}{\| \sum_{k=1}^{n} u_{ik} x_k \|}, \tag{5}$$

$$\alpha_i = \frac{1}{n} \sum_{k=1}^{n} u_{ik}. \tag{6}$$

Notice that all operations are properly defined on the fuzzy multiset space.

2. Crisp c-means with variable for controlling cluster volume

A crisp c-means clustering algorithm can easily be derived by modifying J_{efcma}. We consider the next objective function:

$$J_{ccma}(U, V, \alpha) = -\sum_{i=1}^{c} \sum_{k=1}^{n} [u_{ik} s(x_k, v_i) + \lambda^{-1} u_{ik} \log \alpha_i]. \tag{7}$$

This objective function is obtained from eliminating term $u_{ik} \log u_{ik}$ from J_{efcma}. We see that $\min_{U \in M_c} J_{ccma}$ leads to a crisp solution, since it is linear with respect to u_{ik}. The following algorithm **CCMA** is a crisp c-means clustering algorithm in which the variable for controlling cluster volume sizes.

Algorithm CCMA.

CCMA0. Set initial values of $(\bar{U}, \bar{V}, \bar{\alpha})$.

CCMA1. Solve $\min_{U \in M_c} J_{ccma}(U, \bar{V}, \bar{\alpha})$ and let the optimal solution be new $\bar{U}$.

CCMA2. Solve $\min_{V} J_{ccma}(\bar{U}, V, \bar{\alpha})$ and let the optimal solution be new $\bar{V}$.

CCMA3. Solve $\min_{\alpha \in \mathcal{A}} J(\bar{U}, \bar{V}, \alpha)$ and let the optimal solution be new $\bar{\alpha}$.

CCMA4. If the solution is convergent, stop; else go to step **CCMA1**.

End of CCMA.

The optimal solution of **CCMA1** is calculated by

$$u_{ik} = 1 \Longleftrightarrow i = \arg \max_{1 \le j \le c} s(x_k, v_j), \tag{8}$$

$$u_{ik} = 0 \Longleftrightarrow i \neq \arg \max_{1 \le j \le c} s(x_k, v_j). \tag{9}$$

The optimal solutions of $\bar{V}$ and $\bar{\alpha}$ are calculated by (5) and (6).

C. Kernel trick

Kernel trick is a well-known technique in SVM[19] by which nonlinear classification is effectively realized. Crisp c-means clustering algorithm using kernels has been proposed by Girolami[3] and fuzzy c-means using kernels has been proposed by Miyamoto.[14] Here we consider kernels in this model in order to handle nonlinear clustering (which is called *strong nonlinearity*).

The kernel trick implies that we consider a mapping of an object x_k into another high-dimensional feature space $\Phi(x)$. The mapping $\Phi(\cdot)$ is not explicitly known but the scalar product $\langle \Phi(x), \Phi(y) \rangle$ is given as a kernel function $K(x, y)$. Here we consider the RBF kernel which is most frequently used:

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle = \exp(-C \ \|x - y\|^2).$$

1. *Kernel-based fuzzy c-means with variable for controlling cluster volume*

We consider the next objective function which is entropy-based objective function with variable for controlling cluster volume sizes:[16]

$$J_{kefcma}(U, W, \alpha) = -\sum_{i=1}^{c} \sum_{k=1}^{n} [u_{ik} s(\Phi(x_k), w_i) - \lambda^{-1} u_{ik} \log u_{ik}/\alpha_i] \tag{10}$$

where $W = (w_1, \ldots, w_c)$ is cluster centers in a high-dimensional feature space.

The solutions in the **FCMA** alternative minimization algorithm are

$$u_{ik} = \frac{\alpha_i \exp(\lambda s(\Phi(x_k), w_i))}{\sum_{j=1}^{c} \alpha_j \exp(\lambda s(\Phi(x_k), w_j))}, \tag{11}$$

$$w_i = \frac{\sum_{k=1}^{n} u_{ik} \Phi(x_k)}{\| \sum_{k=1}^{n} u_{ik} \Phi(x_k) \|}, \tag{12}$$

$$\alpha_i = \frac{1}{n} \sum_{k=1}^{n} u_{ik}, \tag{13}$$

in which (13) is the same as (6). However, solutions of (11) and (12) can not directly be obtained, since an explicit form of $\Phi(x_k)$ is unavailable.

This problem is solved by eliminating W by substituting (12) into (11):

$$s(\Phi(x_k), w_i) = \frac{\langle \Phi(x_k), w_i \rangle}{\|\Phi(x_k)\| \|w_i\|} = \frac{\sum_\ell u_{i\ell} K_{k\ell}}{\sqrt{K_{kk} \sum_j \sum_\ell u_{ij} u_{i\ell} K_{j\ell}}} \tag{14}$$

where $K_{j\ell} = K(x_j, x_\ell)$. Thus, by repeating (13) and (11), we obtain an iterative solution for u_{ik} and α_i. Notice that (14) should be used in calculating u_{ik} by (11).

2. Kernel-based crisp c-means with variable for controlling cluster volume

We consider the next objective function which is objective function with variable for controlling cluster volume sizes:

$$J_{kccma}(U, W, \alpha) = -\sum_{i=1}^{c}\sum_{k=1}^{n}[u_{ik}s(\Phi(x_k), w_i) + \lambda^{-1}u_{ik}\log\alpha_i] \qquad (15)$$

where $W = (w_1, \ldots, w_c)$ is cluster centers in a high-dimensional feature space. The algorithm with the kernel uses (13),(14) and (8), (9).

D. Clustering by competitive learning

Clustering by competitive learning is also a standard technique of unsupervised classification.[2] A basic clustering algorithm by competitive learning **CCL** is as follows. This algorithm can directly be applied to the document clustering.

Algorithm CCL.

CCL1. Randomly select cluster centers $v_i, i = 1, \ldots, c$. Normalize x_k:

$$x_k \leftarrow x_k/\|x_k\|,\ k = 1, \ldots, n.$$

Set $t = 0$ and randomly select an object x_k. Repeat **CCL2** and **CCL3** until the solution is convergent.

CCL2. Allocate x_k to the cluster i:

$$i = \arg\max_{1\le j\le c}\langle x_k, v_j\rangle.$$

CCL3. Update the cluster center:

$$v_i \leftarrow v_i + \eta(t)\ x_k, \quad \text{and}$$
$$v_i \leftarrow v_i/\|v_i\|.$$

Let $t \leftarrow t + 1$.

End of CCL.

The parameter $\eta(t)$ is a learning rate coefficient and satisfies

$$\sum_{t=1}^{\infty}\eta(t) = \infty,\ \sum_{t=1}^{\infty}\eta^2(t) < \infty,\ t = 1, 2, \cdots.$$

For example, $\eta(t) = Const/t$ satisfies these conditions and decreases with the time t.

To handle the strong nonlinearity, we consider the use of kernels in this algorithm. We use $\Phi(x_k)$ instead of x_k; hence the normalization is

$$y_k \leftarrow \Phi(x_k)/\|\Phi(x_k)\|$$

and the allocation rule is

$$i = \arg \max_{1 \leq j \leq c} \langle y_k, v_j \rangle.$$

Since we do not use $\Phi(x_k)$ explicitly, $\langle y_k, v_i \rangle$ has to be represented by the kernel.

Let

$$p_{ik}(x_k, i; t) = \langle y_k, v_i \rangle$$

be the value of the scalar product at the time t. From the updating equations

$$v_i \leftarrow v_i + \eta(t) y_k, \quad v_i \leftarrow v_i / \|v_i\|,$$

we note

$$p_{ik}(x_k, i; t+1) = \langle\ y_k, \frac{v_i + \eta(t) y_\ell}{\|v_i + \eta(t) y_\ell\|}\ \rangle.$$

Put $V_i(t) = \|v_i\|$ and note that

$$\langle y_k, y_\ell \rangle = \frac{K_{k\ell}}{\sqrt{K_{kk} K_{\ell\ell}}}$$

where $K_{k\ell} = K(x_k, x_\ell)$. We then have

$$V_i^2(t+1) = V_i^2(t) + 2\eta(t) p_{i\ell}(x_\ell, i; t) + \eta^2(t) K_{\ell\ell}, \tag{16}$$

$$p_{ik}(x_k, i; t+1) = \frac{p_{ik}(x_k, i; t) + \eta(t) \dfrac{K_{k\ell}}{\sqrt{K_{kk} K_{\ell\ell}}}}{V_i(t+1)}. \tag{17}$$

These equations are used instead of the algorithm **CCL**. Allocation of an object x_k to a cluster should use maximum of $p_{ik}(x_k, i; t), i = 1, \ldots, c$.

The following algorithm **K-CCL** is clustering by competitive learning algorithm in which the kernel function is used.

Algorithm K-CCL.

K-CCL1. Initialize $p_{ik}, i = 1, \ldots, c, k = 1, \ldots, n$. Set $t = 0$.
Repeat **K-CCL2** and **K-CCL3** until the solution is convergent.

K-CCL2. Allocate x_k to the cluster i:

$$i = \arg \max_{1 \leq j \leq c} p_{jk}(x_k, j; t).$$

K-CCL3. Update $p_{ik}(x_k, i; t+1)$ by using (16) and (17).
Let $t \leftarrow t + 1$.

End of K-CCL.

IV. Visualization of data in a feature space by self-organizing maps

Although clustering algorithms using the kernel trick have been studied,[3, 12, 14, 16] there is no method of knowing how data are separated in the feature space, since we cannot observe the feature space directly. We therefore consider visualization of the data configuration in the feature space by Self-Organizing Maps (SOM) using the kernel trick.

A. Self-organizing maps (SOM)

SOM[5] is a method of unsupervised learning whereby spatial relations of objects (patterns) are represented. The hexagonal array of the second layer is used in this paper. Given an input pattern, reference vector m_i that has the maximum inner product to the input vector is the winner m_ℓ. The nodes in the neighborhood N_c of m_ℓ are all modified. The radius of N_c is reduced with the time. The algorithm of SOM is similar to **CCL** except that the modification in the neighborhood is used. Notice that in the following algorithm **SOM**, K shows the number of nodes, and not the number of clusters.

Algorithm SOM.

SOM1. Generate initial nodes $m_i(1), i = 1, \cdots, K$. Normalize x_k:

$$x_k \leftarrow x_k/\|x_k\|,\ k = 1, \ldots, n.$$

Set $t = 0$ and randomly select an object x_k. Repeat **SOM2** and **SOM3** until the solution is convergent.

SOM2. Let

$$\ell = \arg \max_{1 \le i \le K} \langle x_k, m_i \rangle.$$

SOM3. Update all nodes i in the neighborhood N_c of the winner m_ℓ.

$$m_i \leftarrow m_i + \eta(t)\ x_k, \quad \text{and}$$
$$m_i \leftarrow m_i/\|m_i\|.$$

Let $t \leftarrow t + 1$.

End of SOM.

B. Kernel-based SOM

Formula of **SOM** using a kernel function can be derived in a similar manner to the kernel-based CCL. Instead of the measure $\langle x_k, m_i \rangle$, the next measure in a high-dimensional feature space is considered:

$$p_{ik}(x_k, i; t) = \langle y_k, m_i \rangle,$$

where

$$y_k \leftarrow \Phi(x_k)/\|\Phi(x_k)\|.$$

From the updating equations

$$m_i \leftarrow m_i + \eta(t)y_k, \quad m_i \leftarrow m_i/\|m_i\|.$$

Since we do not use $\Phi(x_k)$ explicitly, $\langle y_k, m_i\rangle$ has to be represented by the kernel. Here,

$$p_{ik}(x_k, i; t+1) = \langle\ y_k, \frac{m_i + \eta(t)y_j}{\|m_i + \eta(t)y_j\|}\ \rangle.$$

Put $M_i(t) = \|m_i\|$ and note that

$$\langle y_j, y_k\rangle = \frac{K_{jk}}{\sqrt{K_{jj}K_{kk}}}$$

where $K_{jk} = K(x_j, x_k)$. We then have

$$M_i^2(t+1) = M_i^2(t) + 2\eta(t)p_{ij}(x_j, i; t) + \eta^2(t)K_{jj}, \tag{18}$$

$$p_{ik}(x_k, i; t+1) = \frac{p_{ik}(x_k, i; t) + \eta(t)\dfrac{K_{jk}}{\sqrt{K_{jj}K_{kk}}}}{M_i(t+1)} \tag{19}$$

which is the updating formula for $p_{ik}(x_k, i; t)$. These equations are used instead of the algorithm **SOM**. Notice that the value of m_i is unnecessary, only the position of m_i in the grid is used.

The following algorithm **K-SOM** is a Self-Organizing Maps algorithm in which the kernel function is used.

Algorithm K-SOM.

K-SOM1. Initialize $p_{ik}, i = 1, \ldots, K, k = 1, \ldots, n$. Set $t = 0$.
Repeat **K-SOM2** and **K-SOM3** until the solution is convergent.

K-SOM2. Allocate x_k to the node ℓ:

$$\ell = \arg\max_{1\le i\le K} p_{ik}(x_k, i; t).$$

K-SOM3. Update $p_{ik}(x_k, i; t+1)$ for all nodes i in the neighborhood N_c of the node ℓ by using (18) and (19).
Let $t \leftarrow t + 1$.

End of K-SOM.

V. Numerical examples

Two examples were tested, first of which is an illustrative and capability of the kernel-based algorithm to handle nonlinearity was tested; the second is based on real document data on which result by different algorithms were compared.

A. An artificial data

An artificially generated 60 fuzzy multisets were analyzed. These fuzzy multisets consist of three elements; Figure 1 shows a three dimensional plot of those fuzzy multisets.

Results by the algorithm **CCM** (crisp c-means) and **FCM** (entropy-based fuzzy c-means) are shown in Figure 2, and results by the algorithm **CCMA** and **FCMA** are shown in Figure 3, while results by four algorithms with the RBF kernel are shown in Figure 4. Two clusters are represented by $\square$ and $+$ in these figures. Figure 2 and Figure 3 show that the original data were not adequately divided into the two classes. It is moreover obvious that the methods without the kernel cannot separate the ring around the ball and the ball within the ring. In contrast, Figure 4 shows the effectiveness of the kernel-based method for having nonlinear boundary between clusters. Unlike the original data space, the two classes can be linearly separated in the high-dimensional feature space, although we cannot observe the feature space directly. Therefore, the structure of the data in the feature space is visualized by SOM.

Figure 5 shows the result of applying the algorithm **K-SOM** to this example. In this figure $\square$ and $+$ are nodes nearest to the corresponding objects. This result shows that the ring around the ball and the ball within the ring were separated almost linearly in the feature space and those fuzzy multisets were mapped from the high-dimensional feature space into the two dimensional space by **K-SOM**. Furthermore, the black area shows the class of the ring around the ball, while the white area shows the class of the ball within the ring. Namely, the boundary of black and white (gray-zone) indicates the cluster boundary of the two classes in the kernel-based SOM. Thus visualization of the objects in the high-dimensional feature space is successful in this figure.

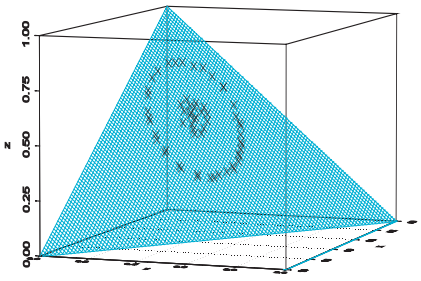

Figure 1. An artificial data set which consists of 60 fuzzy multisets.

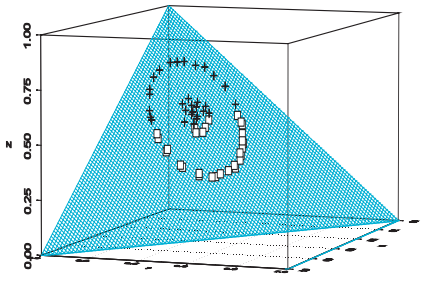

Figure 2. Classification result by algorithms **CCM** and **FCM** ($\lambda = 18$).

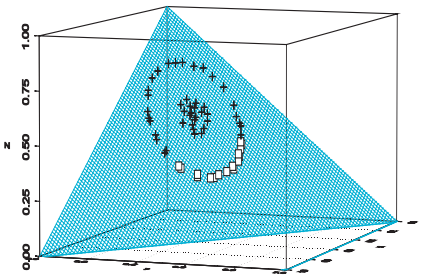

Figure 3. Classification result by algorithms **CCMA** and **FCMA** ($\lambda = 18$).

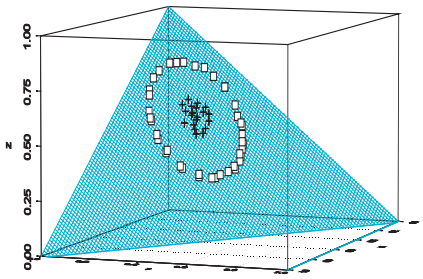

Figure 4. Classification result by four algorithms with the RBF kernel ($\lambda = 10, C = 80$).

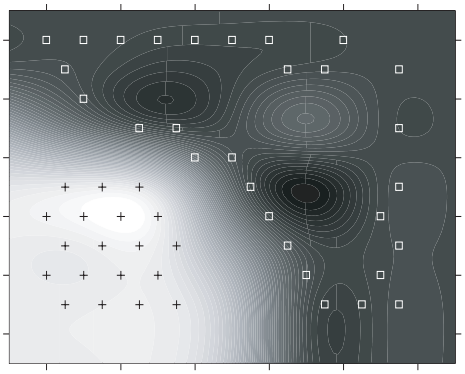

Figure 5. Classification result by the algorithm **K-SOM** ($\eta = 0.05, C = 20.0$).

B. Real document data

We have done document clustering about 30 documents which had been extracted from documents in article database for fuzzy theory and systems on Japan.[4]

The documents are represented by $D = \{d_1, \ldots, d_{30}\}$ and Table 3 shows those document titles. Three terms of t_1=NEURAL NETWORK, t_2=FUZZY, t_3=IMAGE have been used for clustering of these documents.

The rules for determining the membership values are as follows.

- If the term t is in the **title**, its membership is 1.0.
- If the term t is in the **keyword**, its membership is 0.6.
- If the term t is in the **abstract**, its membership is 0.2.
- Else, its membership is 0.

Table 1. Number of misclassified documents.

algorithm	*with the RBF kernel*	*without the RBF kernel*
FCM	0	4
CCM	0	4
FCMA	0	3
CCMA	0	4
CCL	0	4

Table 2 shows fuzzy multisets of 30 documents for which the membership has been given based on the above rules. $d_1 \sim d_{12}$ are concerning '*neural network*' and $d_{13} \sim d_{30}$ are concerning '*image processing*'. **FCM, CCM, FCMA, CCMA** and **CCL** with and without the kernel function have been applied. The parameters are $c = 2$, $\lambda = 6$ and $cnst = 0.6$ in the RBF kernel.

Table 1 shows the number of misclassified document by ten clustering methods. It is seen that the kernel-based method is effective for the real document data.

Figure 6 and Figure 7 show the memberships for a cluster. The horizontal axis is the number of document and the vertical axis is the membership value. A square "□" shows a document concerning '*neural network*', while a cross "+" shows concerning '*image processing*'. Hence this cluster implies documents of neural network. There are three documents (three "+" symbols) misclassified in Figure 6, while there is no misclassified documents in Figure 7.

Notice that, though it isn't shown in the figure here, document 31 is not classified to the class of image processing by other algorithms except for **FCMA**, i.e., this document is misclassified document in other algorithms except for **FCMA**.

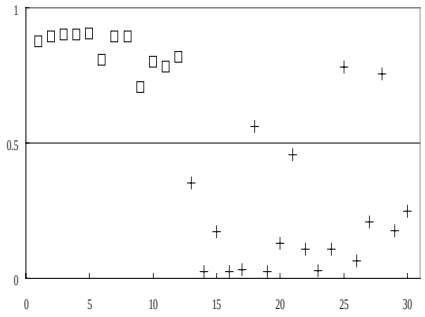

Figure 6. Classification result by the algorithm **FCMA** ($\lambda = 6.0$).

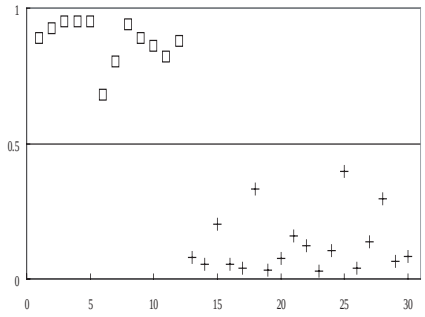

Figure 7. Classification result by the algorithm **K-FCMA** ($\lambda = 6.0, C = 0.6$).

Figure 8 and Figure 9 show the results of applying algorithms **SOM** without and with the kernel function. A number from 1 to 30 shows the number of document, i.e., $1 \sim 12$ with under bar show a document concerning '*neural network*' and $13 \sim 30$ without under bar show a document concerning '*image processing*'. Furthermore, the white area implies the class of '*neural network*', while the black area shows the class of '*image processing*'.

In Figure 8, misclassified documents or documents with high memberships in the class of image processing by the algorithm without the kernel are located on the gray-zone or near the white area. In contrast, two classes of documents are well-separated in Figure 9. Namely, documents concerning neural network are located on white area and documents concerning image processing are located on black area.

Moreover, these results are consistent with the fact that near documents in the SOM representation correspond to similar documents, e.g., the documents 10 and 11 are written by the same author, and these documents are located near in both **SOM** and **K-SOM**; the documents 25 and 28 are written by the same author and they are near in the two figures.

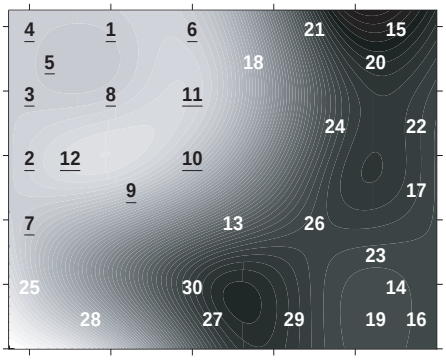

Figure 8. Classification result by the algorithm **SOM** ($\eta = 0.5$).

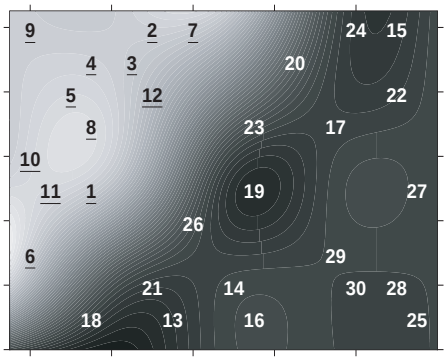

Figure 9. Classification result by the algorithm **K-SOM** ($\eta = 0.04, C = 0.01$).

VI. Conclusion

New frameworks and models are necessary in order to improve current information systems and technologies. For this purpose we have proposed new document clustering methods based on a fuzzy multiset model.

In this paper, fuzzy multiset operations have been redefined using the sorting operator S. When compared with the previous formulation, the present representation is more compact and sorting operation is explicit.

Nonlinearities in unsupervised automatic classification have moreover been dealt with using additional variable for controlling cluster volume sizes and

employing the kernel trick in SVM. Moreover, we have visualized how data are separated in the feature space by a kernel-based SOM. Numerical examples show that the effectiveness of the kernel-based method to both an illustrative example and real document data.

Although we have used only simple examples to show the effectiveness of the proposal method easily, we are developing an information retrieval system where our theory is used. To design our proposal method on an actual web, we will consider as follows. In the second example, membership value was given according to the place (title, keyword and abstract) where a keyword appeared in the document. When applied to information on the web, a simple way to give membership values is to use tag information (<title>, <h1>, <meta>, etc.) in the HTML document. For example, membership value 1 should be given if a keyword is contained in the part of <title>, and 0.6 should be given if a keyword is contained in the part of <h1>, etc. Document information on the web can thus be expressed by using a fuzzy multiset, and our clustering algorithms can be applied.

Moreover, the nonlinear clustering algorithms proposed here can be applied to such huge information as well as the traditional linear clustering algorithms (e.g., hard and fuzzy c-means etc.). Notice that the computational complexity of the kernel-based hard/fuzzy c-means is $\mathcal{O}(n^3)$. On the other hand, even when the kernel function is used, the computational complexity is $\mathcal{O}(n)$ in clustering by competitive learning. Namely, it is possible to classify by the same computational complexity as traditional linear clustering algorithms, even when huge documents are classified.

Future studies include simplification of the model, reduction of computation, and comparison of these methods on large sets of document data. Furthermore, an application to information retrieval including information sources on the web should be studied using the present model and the way of evaluating the usefulness of the proposal method.

References

1. J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum, New York, 1981.
2. R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification, 2nd Ed.*, Wiley, New York, 2001.
3. M. Girolami, Mercer kernel-based clustering in feature space, *IEEE Trans. on Neural Networks*, Vol.13, No.2, 2002.
4. "Japan Society for Fuzzy Theory and Intelligent Informatics", URL address: http://www.j-soft.org/index-e.html.
5. T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, Heidelberg, 1995.
6. Z.-Q. Liu, S. Miyamoto (Eds.), *Soft Computing and Human-Centered Machines*, Springer, Tokyo, 2000.

7. S. Miyamoto, *Fuzzy Sets in Information Retrieval and Cluster Analysis*, Kluwer Academic Publishers, Dordrecht, 1990.

8. S. Miyamoto, Basic operations of fuzzy multisets, *J. of Japan Society for Fuzzy Theory and Systems*, Vol.8, No.4, pp. 639–645, 1996 (in Japanese).

9. S. Miyamoto, Fuzzy multisets with infinite collections of memberships, *Proc. of the 7th International Fuzzy Systems Association World Congress (IFSA'97)*, June 25–29, 1997, Prague, Czech Republic, Vol.1, pp. 61–66, 1997.

10. S. Miyamoto, Fuzzy multiset and their generalizations, in C.S.Calude *et al.*, eds., *Multiset Processing*, Lecture Notes in Computer Science, LNCS 2235, Springer, Berlin, pp. 225–235, 2001.

11. S. Miyamoto, Information clustering based on fuzzy multisets, *Information Processing and Management*, Vol.39, pp. 195–213, 2003.

12. S. Miyamoto, K. Mizutani, Fuzzy multiset space and c-means clustering using kernels with application to information retrieval, *Proc. of the 10th International Fuzzy Systems Association World Congress (IFSA'03)*, June 29–july 2, 2003, Istanbul, Turkey, T.Bilgic *et al.*, eds.: LNAI 2751, pp. 387–395, 2003.

13. S. Miyamoto, M. Mukaidono, Fuzzy c-means as a regulation and maximum entropy approach, *Proc. of the 7th International Fuzzy Systems Association World Congress (IFSA'97)*, June 25–29, 1997, Prague, Czech Republic, Vol.2, pp.86–92, 1997.

14. S. Miyamoto, D. Suizu, Fuzzy c-means clustering using kernel functions in support vector machines, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol.7, No.1, 2003.

15. S. Miyamoto, K. Umayahara, Fuzzy c-means with variable for cluster sizes, *Proc. of the 16th Fuzzy System Symposium*, Sept. 6–8, 2000, Akita, pp. 537–538, 2000 (in Japanese).

16. K. Mizutani, S. Miyamoto, Fuzzy multiset model for information retrieval and clustering using a kernel function, *Proc. of the 14th International Symposium(ISMIS'03)*, Oct.28-31, 2003, Maebashi, Japan, N.Zhong *et al.*, eds.: LNAI 2871, pp. 417–421, 2003.

17. C.J.van Rijsbergen, *Information Retrieval*, Butter Worths, London, 1979.

18. G. Salton, M.J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

19. V. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

20. R.R. Yager, On the theory of bags, *Int. J. General Systems*, Vol.13, pp. 23–37, 1986.

Appendix

Table 2. Fuzzy multisets for 30 documents.

$x_1 =$	$\{ (0.6, 0.2, 0.0)/t_1 , (1.0, 0.6, 0.2)/t_2 , (0.0, 0.0, 0.0)/t_3 \}$
$x_2 =$	$\{ (1.0, 0.6, 0.2)/t_1 , (1.0, 0.2, 0.0)/t_2 , (0.0, 0.0, 0.0)/t_3 \}$
$x_3 =$	$\{ (1.0, 0.6, 0.2)/t_1 , (1.0, 0.6, 0.0)/t_2 , (0.0, 0.0, 0.0)/t_3 \}$
$x_4 =$	$\{ (1.0, 0.6, 0.0)/t_1 , (1.0, 0.6, 0.2)/t_2 , (0.0, 0.0, 0.0)/t_3 \}$
$x_5 =$	$\{ (1.0, 0.6, 0.2)/t_1 , (1.0, 0.6, 0.2)/t_2 , (0.0, 0.0, 0.0)/t_3 \}$
$x_6 =$	$\{ (1.0, 0.6, 0.2)/t_1 , (0.6, 0.0, 0.0)/t_2 , (0.0, 0.0, 0.0)/t_3 \}$
$x_7 =$	$\{ (1.0, 0.2, 0.0)/t_1 , (1.0, 0.6, 0.2)/t_2 , (0.0, 0.0, 0.0)/t_3 \}$
$x_8 =$	$\{ (1.0, 0.2, 0.0)/t_1 , (0.6, 0.2, 0.0)/t_2 , (0.0, 0.0, 0.0)/t_3 \}$
$x_9 =$	$\{ (1.0, 0.6, 0.2)/t_1 , (1.0, 0.6, 0.2)/t_2 , (0.6, 0.0, 0.0)/t_3 \}$
$x_{10} =$	$\{ (0.6, 0.2, 0.0)/t_1 , (1.0, 0.6, 0.2)/t_2 , (0.2, 0.0, 0.0)/t_3 \}$
$x_{11} =$	$\{ (0.6, 0.0, 0.0)/t_1 , (1.0, 0.6, 0.2)/t_2 , (0.2, 0.0, 0.0)/t_3 \}$
$x_{12} =$	$\{ (1.0, 0.2, 0.0)/t_1 , (1.0, 0.2, 0.0)/t_2 , (0.2, 0.0, 0.0)/t_3 \}$
$x_{13} =$	$\{ (0.2, 0.0, 0.0)/t_1 , (0.2, 0.0, 0.0)/t_2 , (0.2, 0.0, 0.0)/t_3 \}$
$x_{14} =$	$\{ (0.0, 0.0, 0.0)/t_1 , (0.0, 0.0, 0.0)/t_2 , (0.2, 0.0, 0.0)/t_3 \}$
$x_{15} =$	$\{ (0.0, 0.0, 0.0)/t_1 , (1.0, 0.6, 0.2)/t_2 , (1.0, 0.2, 0.0)/t_3 \}$
$x_{16} =$	$\{ (0.0, 0.0, 0.0)/t_1 , (0.0, 0.0, 0.0)/t_2 , (0.2, 0.0, 0.0)/t_3 \}$
$x_{17} =$	$\{ (0.0, 0.0, 0.0)/t_1 , (0.2, 0.0, 0.0)/t_2 , (1.0, 0.6, 0.2)/t_3 \}$
$x_{18} =$	$\{ (0.0, 0.0, 0.0)/t_1 , (1.0, 0.2, 0.0)/t_2 , (0.2, 0.0, 0.0)/t_3 \}$
$x_{19} =$	$\{ (0.0, 0.0, 0.0)/t_1 , (0.0, 0.0, 0.0)/t_2 , (1.0, 0.0, 0.0)/t_3 \}$
$x_{20} =$	$\{ (0.0, 0.0, 0.0)/t_1 , (0.6, 0.2, 0.0)/t_2 , (0.6, 0.2, 0.0)/t_3 \}$
$x_{21} =$	$\{ (0.0, 0.0, 0.0)/t_1 , (0.6, 0.2, 0.0)/t_2 , (0.2, 0.0, 0.0)/t_3 \}$
$x_{22} =$	$\{ (0.0, 0.0, 0.0)/t_1 , (1.0, 0.2, 0.0)/t_2 , (1.0, 0.6, 0.2)/t_3 \}$
$x_{23} =$	$\{ (0.0, 0.0, 0.0)/t_1 , (0.2, 0.0, 0.0)/t_2 , (1.0, 0.2, 0.0)/t_3 \}$
$x_{24} =$	$\{ (0.0, 0.0, 0.0)/t_1 , (1.0, 0.0, 0.0)/t_2 , (1.0, 0.2, 0.0)/t_3 \}$
$x_{25} =$	$\{ (1.0, 0.2, 0.0)/t_1 , (0.2, 0.0, 0.0)/t_2 , (0.2, 0.0, 0.0)/t_3 \}$
$x_{26} =$	$\{ (0.2, 0.0, 0.0)/t_1 , (0.2, 0.0, 0.0)/t_2 , (0.6, 0.2, 0.0)/t_3 \}$
$x_{27} =$	$\{ (1.0, 0.2, 0.0)/t_1 , (0.0, 0.0, 0.0)/t_2 , (1.0, 0.2, 0.0)/t_3 \}$
$x_{28} =$	$\{ (1.0, 0.2, 0.0)/t_1 , (0.0, 0.0, 0.0)/t_2 , (0.2, 0.0, 0.0)/t_3 \}$
$x_{29} =$	$\{ (0.6, 0.0, 0.0)/t_1 , (0.0, 0.0, 0.0)/t_2 , (0.6, 0.2, 0.0)/t_3 \}$
$x_{30} =$	$\{ (0.6, 0.2, 0.0)/t_1 , (0.0, 0.0, 0.0)/t_2 , (0.6, 0.0, 0.0)/t_3 \}$

Table 3. 30 document titles for document clustering.

d_1 :	Structural planning support system by neuro-fuzzy network.
d_2 :	Knowledge acquisition for integrity assessment of RC bridge deck based on neural network.
d_3 :	Sales promotion planning support system by neural networks.
d_4 :	Solution method for constructing learning groups by neural network.
d_5 :	Automated generation of fuzzy relation knowledge-base by neural networks.
d_6 :	On error back propagation algorithm incorporating competitive networks.
d_7 :	Acquisition method of fuzzy control rules by neural network with dynamic creative function of hidden units.
d_8 :	A system for forecasting out-off slope collapse using standard data base and simple rainfall factors based on neural network in heavy rain.
d_9 :	Pattern recognition by neural networks and fuzzy inference.
d_{10} :	Fractal and chaos by recurrent fuzzy models.
d_{11} :	Computer tomography by neuro-fuzzy inversion.
d_{12} :	Behavior planning of the autonomous decentralized robots based on fuzzy and neural network.
d_{13} :	Linguistic expression of picture using inference model of human emotions.
d_{14} :	A study on characteristics of faces as a human interface.
d_{15} :	Kansei evaluation model developed by self-organizing maps.
d_{16} :	Human sensory perception oriented fuzzy image processing in color copy system.
d_{17} :	Extraction of distinctive knowledge and its application to car-name image recognition.
d_{18} :	Fuzzy modeling of color quality.
d_{19} :	Tracking control of an unmanned helicopter using visual image information.
d_{20} :	Automatic parts pick-up at conveyer site based on the discriminate methods.
d_{21} :	Peeling robot by fuzzy logic.
d_{22} :	Fuzzy optical flow and its application to medical image.
d_{23} :	Intelligent navigation for an unmanned helicopter using GPS and image information.
d_{24} :	Finding of cerebral aneurysms from MRA images with estimating normal arteries aided by fuzzy logic.
d_{25} :	Construction of a dialog system with emotions for elderly persons by neural networks.
d_{26} :	An optical method for processing information depicted by the picture.
d_{27} :	Emotion extraction from facial expression by parallel sand glass type neural networks.
d_{28} :	Approach to emotion oriented intelligent system by parallel sand glass type neural networks and emotion generating calculations.
d_{29} :	Displaying an impression on picture formed by many figures.
d_{30} :	Dynamic recalling system on knowledge for representing impression.