Fuzzy control GA with a novel hybrid semantic similarity strategy for text clustering

Wei Song a,b,*, Jiu Zhen Liang a, Soon Cheol Park b

a School of Internet of Things Engineering, Jiangnan University, Wuxi, Jiangsu 214122, The People’s Republic of China
b School of Information & Communication Engineering, Chonbuk National University, Jeonju, Jeonbuk 561-756, Republic of Korea

Abstract

This paper proposes a fuzzy control genetic algorithm (GA) in conjunction with a novel hybrid semantic similarity measure for document clustering. Since the common clustering algorithms use vector space model (VSM) to represent document, the conceptual relationships between related terms being ignored, we use semantic similarity measures to solve this problem. In general, the semantic similarity measures can be extensively categorized into two kinds: thesaurus-based methods and corpus-based methods. However, in practice the corpus-based method is rather complicated to tackle. We propose and demonstrate a semantic space model (SSM) as the corpus-based method, where the appropriately reduced dimensions in SSM can capture the true relationship between documents in terms of concepts, rather than specific terms. Thus, the thesaurus-based method is combined with our SSM as a hybrid strategy to represent the semantic similarity measure. In GA field, the balance between the capability to converge to an optimum and the capacity to explore new solutions affects the success of search for the global optimum. We utilize a fuzzy control GA to adaptively adjust the influence between these two factors. Two textual data sets from Reuter document collection and 20-newsgroup corpus are tested in our experiments, and the results show that our fuzzy control GA combined with the hybrid semantic similarity strategy apparently outperforms the conventional GA, FCM and K-means with the traditional cosine similarity in VSM. Moreover, the superiorities of the fuzzy control GA and our hybrid semantic strategy are demonstrated by their better performance, in comparison with conventional GA with the same similarity measures.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

Owing to the overflowing of knowledge and information on the internet, it becomes impossible to manually cluster documents online. Thus automatic and high-quality partitioning data into previously unseen categories turns to be a major topic for applications such as data mining and information retrieval. Clustering [13,26,35] is a widely used unsupervised classification technique which partitions the input space into $K$ regions via some similarity measures. The partition is done in such a way that patterns within a group are more similar to each other than those belonging to different groups. There are two main kinds of clustering algorithms, i.e. partitioning and hierarchical, available in the literature. Partitioning algorithms, like K-means algorithm [4,38], attempt to create $k$ partitions in the situation that $k$ is fixed and known in advance. K-means is...
an iterative hill-climbing algorithm which suffers from the restriction of suboptimum depending on the choice of initial clustering distribution. Some, like K-means algorithm, optimize of the distance criterion either by minimizing the within cluster spread, or by maximizing the inter-cluster separation. Hierarchical algorithms, like OPTICS [1], compute a representation of the possible hierarchical cluster structure from which clusters at various solutions can be extracted. Other techniques, like graph theoretical approach [19], branch and bound procedure [18], maximum likelihood estimate technique algorithm [44], etc., perform clustering based on other criteria and have proved to be useful in some specific applications of computational intelligence. In recent decades, several types of biologically inspired clustering algorithms have been proposed for clustering. Ant clustering algorithm [20,47] projects the original data into bidimensional output grids and positions that are similar to each other in their original attributes. By doing so, the algorithm is capable of grouping items together that are similar to each other. Genetic algorithm (GA) [5,34,36,40] is a randomized search and optimization technique which can be used in complex and large landscapes and provide near-optimal solutions for its optimization problem. However, when these algorithms are applied in text clustering, most of them solely use vector space model (VSM) to represent text, that is to say, each unique word in vocabulary represents one dimension in vector space [29]. The representation by pack of words adopted in these clustering algorithms is often unsatisfactory, because VSM makes matches simply via keywords. However, the same concept can be signified by different words. Thus, the relationship between some important words which do not co-occur literally are ignored in VSM. Moreover, the inherent high dimensional space with a large number of features leads to a high cost of computational time for clustering problems. In this paper, a fuzzy control genetic algorithm is proposed, in conjunction with a hybrid semantic similarity measure, for document clustering. We use the broad-coverage taxonomy and hierarchical structure of WordNet [15,25] as thesaurus-based ontology to detect semantic relationships between documents. Moreover, a new semantic space model (SSM) is proposed and demonstrated as a corpus-based method which can appropriately depict the associative semantic relationships. Moreover, we greatly reduce the number dimensions for document representation. Considering the mutual influence between the selection pressure and the diversity of population, we recommend a fuzzy control GA which takes advantage of dynamic probabilities of crossover and mutation to realize the goals of GA, i.e. maintaining the diversity as well as sustaining the convergence capacity. As is known, the parameters of GA critically affect its success and are rather difficult to define. In our method, they are adjusted adaptively by GA itself.

The following parts of this paper are organized as below: Section 2 describes how to calculate semantic similarity via thesaurus-based method. In Section 3 a semantic space model (SSM) is proposed and demonstrated, in conjunction with the thesaurus-based measure as a hybrid strategy for semantic similarity calculation. The details of fuzzy control genetic algorithm based on the semantic similarity measure for document clustering are given in Section 4. Experiment results and analysis are given in Section 5. Conclusions are given in Section 6.

2. Semantic similarity based on ontology

Semantic similarity is a generic method adopted in the fields of information retrieval (IR) and artificial intelligence (AI). Semantic similarity between two words is often represented by the similarity between the concepts associated with the two words. Thus, it can overcome the limitations of the simple cosine measure based on the bag-of-words representation of documents in VSM. Several semantic similarity methods have been developed in the specific applications of computational intelligence [21,31]. In general, these measures can be categorized into two kinds: thesaurus-based methods and corpus-based methods.

2.1. WordNet: A widely used thesaurus

WordNet is a widely used semantic dictionary developed under the direction of Miller [25]. It organizes words, i.e. nouns, verbs, adjectives and adverbs, by synonym sets, named synsets, each representing a distinct concept. Synsets are interlinked via conceptual-semantic and lexical relations [22]. The version utilized in this paper is WordNet 2.0, which consists of 109,377 synsets and 144,684 words.

2.2. Thesaurus-based semantic similarity calculation

In this paper we adopt the thesaurus-based method given by Li et al. [21]. In their study two factors for calculating the similarity between two concepts are taken into account: (1) The shortest path length between the two concepts and (2) the depth of the subsumer in the hierarchical structure. That is to say, given two concepts \(c_1\) and \(c_2\), the semantic similarity is defined as:

\[
sim(c_1, c_2) = f_1(l) \cdot f_2(h)
\]

where \(l\) is the shortest path length connecting concept \(c_1\) and \(c_2\). \(h\) is the depth of subsumer in the hierarchical structure. Suppose the respective effects of \(l\) and \(h\) on the similarity are independent from each other, therefore, the similarity function is comprised of two independent functions of \(f_1\) and \(f_2\).

Suppose a word has multiple meanings, therefore, different paths may exist and the minimum length of the path connecting the two concepts is a direct method to calculate the similarity. It is intuitive that the similarity between two concepts
would nonlinearly decrease as the shortest path connected them increases. Therefore, it would be reasonable to expect that the similarity decreases at an exponential rate as the shortest path increases, and $f_1$ is defined by:

$$f_1(l) = e^{-ax}$$  \hspace{1cm} (2)$$

where $a$ is a real constant between 0 and 1. From (2) we can see that when the path length decreases to zero, the similarity would monotonically increase toward 1. While the path length increases infinitely, the similarity should monotonically decrease to 0. However, only the shortest path for semantic similarity calculation may be not so accurate, the shortest path length method must be revised by adding more information from the hierarchical semantic structure of WordNet. It is intuitive that concepts at higher levels of the hierarchy have more general information, while concepts at lower levels have more concrete semantics. Thus, the depth of concept in the hierarchy should be taken into account. The depth $h$ of the subsumer is derived by calculating the shortest length of links from the subsumer to the root concept of the ontology. According to this observation, the depth function to similarity is defined by:

$$f_2(h) = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}$$  \hspace{1cm} (3)$$

where $\beta > 0$ is a smoothing factor. Also, $f_2$ can be considered as an extension of Shepard’s law [39], which claims that exponential-decay functions are a universal law of stimulus generalization for psychological science. We have achieved the semantic similarity between two concepts based on the thesaurus method by far. However, the common corpus-based (or information-based) method [32] is a rather difficult to tackle. We cannot easily access it solely from the semantic nets. In next section the SSM is proposed to simulate VSM.

3. Semantic similarity calculation based on SSM

In this part we propose and demonstrate a semantic space model (SSM) whose whole dimensions precisely simulate the original vector space model via cosine and Euclidean distance similarity calculation, and the appropriately reduced space can hopefully capture the true semantic relationship between documents. SSM is an automatic approach which can solve the problems by using statistically derived conceptual indices instead of individual words. It utilizes singular value decomposition (SVD) [27,43] to decompose the large term-by-document matrix into a set of $k$ orthogonal factors.

3.1. The proof of SSM to simulate VSM

We use document-by-term matrix $D(n \times m)$ to represent the original corpus matrix, assuming there are $m$ terms in an $n$ documents data set. The transpose of matrix $D$ is then represented by the term-by-document matrix $A(m \times n)$

$$D = A^T$$  \hspace{1cm} (6)$$

The singular value decomposition of $A$ is defined as

$$A = USV^T$$  \hspace{1cm} (7)$$
where $U$ is an $m \times m$ orthogonal matrix, $\Sigma$ is an $m \times n$ diagonal matrix containing the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)}$. All off-diagonal elements of $\Sigma$ are zeros. $V^T$ is an $n \times n$ orthogonal matrix. In common data set, the number of terms is much more than that of documents, namely, $m$ is much greater than $n$. The singular value decomposition of matrix $A$ when $m > n$ is illustrated in Fig. 1.

Hence, the new corpus matrix $C$ is defined by

$$C = DU_1$$

(8)

We only use $U_1$ to construct the new corpus matrix $C$. The size of the $U_1$ is the economy size of $U$, depending on the number of the nonzero singular values in matrix $\Sigma$ and $\Sigma_1(n \times n)$ is such nonzero part. Fig. 2 illustrates the construction of matrix $C$ when $m > n$. We can see that the newly constructed corpus $C$ is a $n \times n$ matrix. That is, the number of dimensions is reduced from $m$ in the original corpus matrix $D$ to $n$ in the new corpus matrix $C$.

Here we will demonstrate the similarity, i.e. cosine similarity and Euclidean distance, between two randomly selected rows in matrix $C$ are same to those of the corresponding rows in the original corpus $D$.

Proof. Suppose $D_1$ and $D_2$ are two randomly selected rows in the document-by-term matrix $D(n \times m)$, $C_1$ and $C_2$ are the two corresponding rows in matrix $C$, therefore, from (8) we have:

$$C_1 = D_1 U_1$$

(9)

$$C_2 = D_2 U_1$$

(10)

From (6) and (7) we obtain that

$$D = A^T = (U \Sigma V^T)^T = (U_1.U_2)((\Sigma_1,0)V^T)^T = V(\Sigma_1,0)(0,U_1^TU_2) = V \Sigma_1 U_1^T$$

(11)

Then, from (11) we have

$$DU = D(U_1.U_2) = (DU_1,DU_2) = (DU_1,V \Sigma_1 U_1^TU_2)$$

(12)

According to the property of unitary matrix, it’s easy to obtain

$$U_1^TU_2 = 0$$

(13)

Subsequently, from (12) and (13) we can obtain

$$DU = (DU_1,0)$$

(14)
That is,
\[
\begin{pmatrix}
D_1 \\
D_2 \\
\vdots \\
D_n
\end{pmatrix} = \begin{pmatrix}
\begin{pmatrix}
D_1 \\
D_2 \\
\vdots \\
D_n
\end{pmatrix} U_1, 0
\end{pmatrix}
\]
where we represent the document-by-term matrix \(D(n \times m)\) by its rows \(D_1, D_2, \ldots, D_n\). Hence, from (15) we have
\[
D_1 U = (D_1 U_1, 0)
\]
\[
D_2 U = (D_2 U_1, 0)
\]
Suppose \(D_1\) and \(D_2\) are represented by \(\{w_{1,1}, w_{1,2}, \ldots, w_{1,m}\}\) and \(\{w_{2,1}, w_{2,2}, \ldots, w_{2,m}\}\) respectively, therefore, the inner product between \(D_1\) and \(D_2\) is defined by
\[
\langle D_1, D_2 \rangle = \sum_{k=1}^{m} w_{1,k} \cdot w_{2,k}
\]
where \(\langle D_1, D_2 \rangle\) is the standard inner product. \(w_{1,k}\) and \(w_{2,k}\) stand for the \(k\)th elements for \(D_1\) and \(D_2\). Since matrix \(U\) is an unitary matrix, in terms of the property of unitary matrix, we have:
\[
\langle D_1, D_2 \rangle = \langle D_1 U, D_2 U \rangle
\]
\[
\langle D_1, D_1 \rangle = \langle D_1 U, D_1 U \rangle
\]
\[
\langle D_2, D_2 \rangle = \langle D_2 U, D_2 U \rangle
\]
Hence, from (9), (10), (16), (17) and (19) and the property of inner product we have
\[
\langle D_1, D_2 \rangle = \langle D_1 U, D_2 U \rangle = \langle (D_1 U_1, 0), (D_2 U_1, 0) \rangle = \langle D_1 U_1, D_2 U_1 \rangle = \langle C_1, C_2 \rangle
\]
where \(C_1\) and \(C_2\) are the two corresponding rows in matrix \(C\), respectively. Similarly, we can obtain
\[
\langle D_1, D_1 \rangle = \langle D_1 U, D_1 U \rangle = \langle (D_1 U_1, 0), (D_1 U_1, 0) \rangle = \langle D_1 U_1, D_1 U_1 \rangle = \langle C_1, C_1 \rangle
\]
\[
\langle D_2, D_2 \rangle = \langle D_2 U, D_2 U \rangle = \langle (D_2 U_1, 0), (D_2 U_1, 0) \rangle = \langle D_2 U_1, D_2 U_1 \rangle = \langle C_2, C_2 \rangle
\]
In VSM cosine measure is commonly used to compute the similarity between two documents. The cosine similarity between \(D_1\) and \(D_2\) are defined by
\[
\cos(D_1, D_2) = \left( \frac{\sum_{k=1}^{m} w_{1,k} \cdot w_{2,k}}{\sqrt{\sum_{k=1}^{m} w_{1,k}^2 \cdot \sum_{k=1}^{m} w_{2,k}^2}} \right) = \frac{\langle D_1, D_2 \rangle}{\sqrt{\langle D_1, D_1 \rangle \cdot \langle D_2, D_2 \rangle}}
\]
Suppose \(C_1\) and \(C_2\) in our new constructed corpus matrix \(C\) are represented by \(\{v_{1,1}, v_{1,2}, \ldots, v_{1,n}\}\) and \(\{v_{2,1}, v_{2,2}, \ldots, v_{2,n}\}\) respectively, therefore, the cosine similarity between \(C_1\) and \(C_2\) are defined by
\[
\cos(C_1, C_2) = \left( \frac{\sum_{k=1}^{n} v_{1,k} \cdot v_{2,k}}{\sqrt{\sum_{k=1}^{n} v_{1,k}^2 \cdot \sum_{k=1}^{n} v_{2,k}^2}} \right) = \frac{\langle C_1, C_2 \rangle}{\sqrt{\langle C_1, C_1 \rangle \cdot \langle C_2, C_2 \rangle}}
\]
Hence, form (22)–(26) we can obtain
\[
\cos(D_1, D_2) = \cos(C_1, C_2)
\]
What is more, if Euclidean distance is used to calculate the similarity between the two documents in VSM, the Euclidean distance between \(D_1\) and \(D_2\) is defined by
\[
\text{dis}(D_1, D_2) = \sqrt{\sum_{k=1}^{m} (w_{1,k} - w_{2,k})^2} = \sqrt{\sum_{k=1}^{m} (w_{1,k}^2 + w_{2,k}^2 - 2w_{1,k} \cdot w_{2,k})} = \sqrt{\langle D_1, D_1 \rangle + \langle D_2, D_2 \rangle - 2\langle D_1, D_2 \rangle}
\]
and the Euclidean distance between \(C_1\) and \(C_2\) is defined by
\[
\text{dis}(C_1, C_2) = \sqrt{\sum_{k=1}^{n} (v_{1,k} - v_{2,k})^2} = \sqrt{\sum_{k=1}^{n} (v_{1,k}^2 + v_{2,k}^2 - 2v_{1,k} \cdot v_{2,k})} = \sqrt{\langle C_1, C_1 \rangle + \langle C_2, C_2 \rangle - 2\langle C_1, C_2 \rangle}
\]
Subsequently, from (22), (23), (24), (28) and (29) we obtain that
\[
\text{dis}(D_1, D_2) = \text{dis}(C_1, C_2)
\]
Here we have demonstrated that in SSM matrix \( C(n \times n) \) precisely simulates the original document-by-term matrix \( D(n \times m) \) in VSM via cosine and Euclidean distance similarities calculation. Note that, in comparison with matrix \( D \), the number of dimensions in \( C \) has decreased from \( n \) to \( m \), where \( n \) is much smaller than \( m \).

### 3.2. Latent semantic indexing technology to further decrease dimensions

Latent semantic indexing (LSI) [9] is a technology that projects the original high dimensional document vectors into a space with “semantic” indexed dimensions. We use the concept of LSI to further decrease the dimensions, as well as construct a semantic structure for corpus. In LSI, in order to find the best rank \( k \) approximation of \( A \), we can simply choose the \( k \) largest singular values first. The approximation matrix \( A_k \) is given by

\[
A_k = U_k \Sigma_k V_k^T
\]

where \( U_k \) is comprised of the first \( k \) columns of the matrix \( U \) and \( V_k^T \) is comprised of the first \( k \) rows of the matrix \( V^T \). \( \Sigma_k = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_k) \) is the diagonal matrix. The best rank of approximation \( A_k \) captures most of the important underlying structure while ignoring noise via word choice [46].

LSI is firstly used in query-based information retrieval (IR) system [7,24], i.e. a query \( q \) is represented as a vector in a reduced \( k \)-dimensional space. Hence, the traditional LSI method represents the query vector \( q \) as

\[
\tilde{q} = q^T U_1 \Sigma_1^{-1}
\]

Nowadays, it is also widely utilized in clustering [6] and categorization [41] system, i.e. a document is regarded as a query in the query-based system, and each document vector is given by

\[
\tilde{d} = d^T U_1 \Sigma_1^{-1}
\]

However, the main difference between our approach and the traditional LSI is that we only use \( U_1 k \) matrix, rather than the multiplying of \( U_k \) and \( \Sigma_k^{-1} \) in the traditional LSI, because we have demonstrated that the similarities, i.e. cosine similarity and Euclidean distance, between two randomly selected rows in matrix \( C \) (8) are equal to those of the corresponding rows in original corpus \( D \), respectively. That is to say, the whole dimension of \( C \) can precisely simulate the matrix \( D \). Meanwhile, the appropriately reduced dimensions of matrix \( C \) can hopefully reveal the true semantic relationships between documents with the help of LSI. Furthermore, our experimental results show that the traditional approach is not as good as our method. Thus, in our SSM the semantic corpus \( C_k \) is ultimately defined as

\[
C_k = \frac{D}{n \times k} U_{1k}
\]

where \( U_{1k} \) is the first \( k \) columns of the economy part of the matrix \( U \), and each document vector is newly formed by

\[
\hat{d} = \tilde{d}^T U_{1k}
\]

Once the new corpus matrix is formed, we use cosine similarity measure to evaluate similarity between pairs of documents. Suppose two document vectors \( d_1 \) and \( d_2 \) are represented by \((v_{1,1}, v_{1,2}, \ldots, v_{1,k})\) and \((v_{2,1}, v_{2,2}, \ldots, v_{2,k})\), respectively, the cosine similarity between \( d_1 \) and \( d_2 \) in our SSM is defined as

\[
\text{sim}_{SSM}(d_1, d_2) = \left( \sum_{p=1}^{k} v_{1,p} \cdot v_{2,p} \right) \sqrt{\sum_{p=1}^{k} v_{1,p}^2 \cdot \sum_{p=1}^{k} v_{2,p}^2}
\]

Therefore, a hybrid strategy which combines the thesaurus-based similarity (WordNet) and corpus-based similarity (SSM) is proposed in this study. The new document similarity measure is given by:

\[
\text{sim}(d_1, d_2) = \bar{\lambda} \text{sim}_{SSM}(d_1, d_2) + (1 - \bar{\lambda}) \text{sim}_{ONTO}(d_1, d_2)
\]

where \( \lambda \) is a real constant within the range from 0 to 1. \( \text{sim}_{ONTO} \) is the thesaurus-based similarity in (5). In the next section a fuzzy control genetic algorithm based on the proposed similarity strategy is proposed for text clustering.

### 4. Document clustering using genetic algorithm

The searching capability of GA has been utilized in this study to evolve the fixed number of clustering for textual data. GA is known to offer significant advantages over conventional methods by using several biologically inspired principles. The basic steps of GA are now described in detail.

#### 4.1. Chromosome representation

Each chromosome is a string of real numbers representing \( K \) cluster centers. For an \( N \) dimensional space, the length of a chromosome is \( N \times K \) bits (genes), where the first \( N \) bits represent the \( N \) dimensions of the first cluster center, and the next \( N \)
and the mutation probability \( C_0 \) are set as 0.8. Goldberg [11] made extensive experiments and proposed that \( C_0 \) and \( C_1 \) are two real constants between 0 and 1. The values of \( C_0 \) and \( C_1 \) are critical for searching success of GA. In order to avoid premature convergence of GA to a suboptimum, we need to increase the diversity of population by increasing the value of \( C_1 \) and \( C_0 \). That is, when \( \bar{fit} \) is close to \( fit_{\text{max}} \) while far from \( fit_{\text{min}} \), we need to scale up \( p_c \) and \( p_m \), and when \( \bar{fit} \) is close to \( fit_{\text{min}} \) while far from \( fit_{\text{max}} \), we need to scale down \( p_c \) and \( p_m \). Thus we have

\[
\begin{align*}
    p_c &= \frac{k_1}{\Var} \\
    p_m &= \frac{k_2}{\Var}
\end{align*}
\]

where \( k_1 \) and \( k_2 \) are two real constants. Here we have defined \( \Var \) which is determined by the distribution of all individuals in the current population. However, when the population is likely to converge to a solution of the global optimal, the mutual high values of \( p_c \) and \( p_m \) for each solution may destroy the potential optimal individuals. Thus, in order to preserve the excellent individuals, we need to decrease the values of \( p_c \) and \( p_m \) for high fitness individuals. We use \( \text{Dif} \) to depict the relative difference between the individual fitness value and the maximum fitness value, where higher the value of \( \text{Dif} \) is, the bigger \( p_c \) and \( p_m \) should be.

\[
\text{Dif} = \frac{(\text{fit}_{\text{max}} - \text{fit})}{(\text{fit}_{\text{max}} - \text{fit}_{\text{min}})}
\]

and the values of \( p_c \) and \( p_m \) are proportional to values of \( \text{Dif} \)

\[
\begin{align*}
    p_c &= k_1 \cdot \frac{\text{Dif}}{\Var} = k_1 \frac{(\text{fit}_{\text{max}} - \text{fit})}{(\text{fit}_{\text{max}} - \bar{fit})} \\
    p_m &= k_2 \cdot \frac{\text{Dif}}{\Var} = k_2 \frac{(\text{fit}_{\text{max}} - \text{fit})}{(\text{fit}_{\text{max}} - \bar{fit})}
\end{align*}
\]

for crossover, \( p_c \) is determined by the fitness values of the parents. We choose the larger one as \( \text{fit} \) in (42). We can see from the definition of (42) and (43) that when \( \text{fit} < \bar{fit} \), the values of \( p_c \) and \( p_m \) would become bigger than \( k_1 \) and \( k_2 \), respectively. Such situations may cause an over scattered problem. Thus we have

\[
\begin{align*}
    p_c &= k_1 \cdot \frac{\text{Dif}}{\Var}, \quad \text{fit} \geq \bar{fit} \\
    p_c &= k_3, \quad \text{fit} < \bar{fit} \\
    p_m &= k_2 \cdot \frac{\text{Dif}}{\Var}, \quad \text{fit} \geq \bar{fit} \\
    p_m &= k_4, \quad \text{fit} < \bar{fit}
\end{align*}
\]

where \( k_1, k_2, k_3 \) and \( k_4 \) are four real constants between 0 and 1. Goldberg [11] made extensive experiments and proposed that \( p_c \) with value less or equivalent to 0.8 and \( p_m \) with value less or equivalent to 0.1 are critical for searching success of GA. In order to evolve for global optimum and prevent convergence to a suboptimal solution, the poor solutions \( (\text{fit} < \bar{fit}) \) will be destroyed for more extensive search. Therefore, for poor individuals \( (\text{fit} < \bar{fit}) \) we empirically set a large value for \( p_c \) and a moderately large value for \( p_m \) \((k_3 = 0.8, k_4 = 0.1)\) in (45) and (47) respectively. As for good individuals \( (\text{fit} \geq \bar{fit}) \), only when \( \text{fit} = \bar{fit} \), can \( p_c \) get the maximum value \( (p_c = k_1) \). Otherwise, \( p_c \) will be smaller than \( k_1 \). In this case, we empirically set \( k_1 \) as 0.8 to be the highest crossover probability \( p_m \) since the individuals with fitness of \( \bar{fit} \) should also be destroyed. Accordingly, we set \( k_2 \) as 0.1 empirically.

4.2. Fuzzy control operators

Three biologically inspired operators, such as selection, crossover and mutation, are exerted to yield new offspring in GA [12,17]. To dynamically regulate the GA operators, we use several control parameters to adaptively manipulate the crossover probability \( p_c \) and the mutation probability \( p_m \). In order to avoid premature convergence of GA to a suboptimum, we need to know whether GA is likely to converge to an optimum or not. We use \( \Var \) to depict the distribution of the population in the current generation.

\[
\Var = \frac{\text{fit}_{\text{max}} - \bar{fit}}{\text{fit}_{\text{max}} - \text{fit}_{\text{min}}}
\]

where \( \bar{fit}, \text{fit}_{\text{max}} \) and \( \text{fit}_{\text{min}} \) are the average the maximum and the minimum fitness values respectively in the current generation. \( \Var \) depicts the closeness of the distribution which may cause a potential convergence. It is intuitive that when the average fitness is close to the best one while far from the worst one, the current distribution will be dense which may cause a premature convergence, so we need to increase the diversity of population by increasing the value of \( p_c \) and \( p_m \). That is, when \( \bar{fit} \) is close to \( \text{fit}_{\text{max}} \) while far from \( \text{fit}_{\text{min}} \), we need to scale up \( p_c \) and \( p_m \), and when \( \bar{fit} \) is close to \( \text{fit}_{\text{min}} \) while far from \( \text{fit}_{\text{max}} \), we need to scale down \( p_c \) and \( p_m \). Thus we have

\[
\begin{align*}
    \text{Dif} &= \frac{(\text{fit}_{\text{max}} - \text{fit})}{(\text{fit}_{\text{max}} - \text{fit}_{\text{min}})}
\end{align*}
\]

and the values of \( p_c \) and \( p_m \) are proportional to values of \( \text{Dif} \)

\[
\begin{align*}
    p_c &= k_1 \cdot \frac{\text{Dif}}{\Var} = k_1 \frac{(\text{fit}_{\text{max}} - \text{fit})}{(\text{fit}_{\text{max}} - \bar{fit})} \\
    p_m &= k_2 \cdot \frac{\text{Dif}}{\Var} = k_2 \frac{(\text{fit}_{\text{max}} - \text{fit})}{(\text{fit}_{\text{max}} - \bar{fit})}
\end{align*}
\]

for crossover, \( p_c \) is determined by the fitness values of the parents. We choose the larger one as \( \text{fit} \) in (42). We can see from the definition of (42) and (43) that when \( \text{fit} < \bar{fit} \), the values of \( p_c \) and \( p_m \) would become bigger than \( k_1 \) and \( k_2 \), respectively. Such situations may cause an over scattered problem. Thus we have

\[
\begin{align*}
    p_c &= k_1 \cdot \frac{\text{Dif}}{\Var}, \quad \text{fit} \geq \bar{fit} \\
    p_c &= k_3, \quad \text{fit} < \bar{fit} \\
    p_m &= k_2 \cdot \frac{\text{Dif}}{\Var}, \quad \text{fit} \geq \bar{fit} \\
    p_m &= k_4, \quad \text{fit} < \bar{fit}
\end{align*}
\]

where \( k_1, k_2, k_3 \) and \( k_4 \) are four real constants between 0 and 1. Goldberg [11] made extensive experiments and proposed that \( p_c \) with value less or equivalent to 0.8 and \( p_m \) with value less or equivalent to 0.1 are critical for searching success of GA. In order to evolve for global optimum and prevent convergence to a suboptimal solution, the poor solutions \( (\text{fit} < \bar{fit}) \) will be destroyed for more extensive search. Therefore, for poor individuals \( (\text{fit} < \bar{fit}) \) we empirically set a large value for \( p_c \) and a moderately large value for \( p_m \) \((k_3 = 0.8, k_4 = 0.1)\) in (45) and (47) respectively. As for good individuals \( (\text{fit} \geq \bar{fit}) \), only when \( \text{fit} = \bar{fit} \), can \( p_c \) get the maximum value \( (p_c = k_1) \). Otherwise, \( p_c \) will be smaller than \( k_1 \). In this case, we empirically set \( k_1 \) as 0.8 to be the highest crossover probability \( p_m \) since the individuals with fitness of \( \bar{fit} \) should also be destroyed. Accordingly, we set \( k_2 \) as 0.1 empirically.
4.3. Fitness function and termination criterion

In our algorithm we use Davies–Bouldin index \([3,8]\) to evaluate the fitness of each chromosome. Davies–Bouldin is defined as the ratio of the summation of within-cluster scatter to inter-cluster separation. The hybrid similarity measure in \((37)\) is proposed in this study to evaluate the similarity between pairs of documents. The definition of the within-cluster scatter for cluster \(C_i\) is given by

\[
S_i = \frac{1}{n_i} \sum_{x \in C_i} \text{sim}(x, z_i)
\]

where \(z_i\) is the center of \(C_i\) and \(n_i\) is the number of documents in this cluster. That is, \(S_i\) represents the average similarity in cluster \(C_i\). Subsequently, we compute

\[
R_i = \max_{j \neq i} \left\{ \frac{S_i + S_j}{s_{ij}} \right\}
\]

where \(s_{ij}\) is the inter-cluster separation which is valued by the similarity between the centers of \(C_i\) and \(C_j\). Note that, we may change the value of gene (term weight) during mutation process, but the value of \(\text{sim}_{onto}\) in \((5)\) does not change. That is because the value of \(\text{sim}_{onto}\) in \((5)\) only depends on whether it appears or not, which will not be affected by the variance of the term weight. Therefore, we only count the effect of \(\text{sim}_{SSM}\) in \((36)\) during mutation process. The Davies–Bouldin DB is defined as

\[
DB = \frac{1}{K} \sum_{i=1}^{K} R_i
\]

Since the separation between documents varies inversely with the value of the hybrid similarity in \((37)\), and the goal of clustering is to minimize the within-cluster spread and to maximize the inter-cluster separation. In our algorithm the fitness function is defined as \(DB\) in \((50)\). The higher the value of fitness is, the better clustering should be. The process of the three evolution operators and the fitness calculation continues for several generations till the termination criterion is satisfied. The algorithm is terminated when the iterations without improvement reach the consecutive \(n_{\text{max}}\) generations. Note that in case the process does not terminate normally, it is executed for the maximum fixed number of iterations \(N_{\text{max}}\).

5. Experiments results and analysis

In order to evaluate the performance of our algorithm, two most-widely adopted standard data sets, Reuter (21578 version) corpus and 20-newsgroup (18828 version) corpus, are utilized for test in this paper. In Reuter corpus, we select 500 documents from 5 topics (Coffee, Crude, Grain, Money-fx and ship) and in the 20-newsgroup, we select 300 documents from 5 topics (Comp.windows.x, Rec.motorcycles, Sci.space, Sci.med, and Talk.politics.mideast). After preprocessing, i.e. word extraction, stop word removing, and Porter’s stemming \([30]\), there are 5253 indexing terms from Reuter corpus and 8595 indexing terms from 20-newsgroup corpus in the vocabulary. Thus, in the initial VSM, the feature vectors are represented by

\[
d_i = \{w_{i,1}, w_{i,2}, \ldots, w_{i,5253}\}
\]

\[
d'_i = \{w'_{i,1}, w'_{i,2}, \ldots, w'_{i,8595}\}
\]

for each document in Reuter corpus and 20-newsgroup corpus, respectively, where \(w_{ij}\) is the term weight of the \(jth\) indexing term in document \(i\).

5.1. Feature selection and reduction

It has been known that the main difficulty in the application of VSM to genetic algorithm is that there are thousands or even tens of thousands of indexing terms for representing feature vectors, typical for textual data. Savio and Lee \([37]\) recommend many approaches for feature space reduction. In our method, we select an amount of terms with the highest term weights, 1000 from Reuter data set and 1500 from 20-newsgroup data set, for representing document vectors. That is

\[
d_i = \{w_{i,1}, w_{i,2}, \ldots, w_{i,1000}\}
\]

\[
d'_i = \{w'_{i,1}, w'_{i,2}, \ldots, w'_{i,1500}\}
\]

and, instead of using simple \(tf \cdot idf\) term weights scheme, symmetric Okapi BM25 \([14,28]\), the most popular retrieval model in the probabilistic framework, is utilized for indexing term weights

\[
W_{ij} = \frac{t_{f_{ij}}}{t_{f_{ij}} + 0.5 \times \frac{a_i}{\text{argmax} \times idf_j}}
\]
and the classical inverse document frequency $idf$ derived from the binary independence model \cite{33} is usually computed as
\begin{equation}
$idf_i = \log(N/n)$
\end{equation}

where $N$ is the total number of test documents, and $n$ is the number of documents in which the $j$th indexing term occurred; $tf_i$ is the term frequency of $j$th indexing term occurred in document $i$; $dl_i$ is the length of document $i$, and $avgdl$ is the average document length. This formula normalizes the length of documents rather than the simple $tf \times idf$ method.

When the proposed SSM in \cite{8} is implemented for document representation, we can reduce the number of dimensions from 1000 and 1500 in VSM to 500 and 300 in SSM for representing Reuter documents and 20-newsgroups documents, respectively, where 500 and 300 are the number of test documents in our experiments. That is
\begin{equation}
\hat{d}_i = \{v_{i1}, v_{i2}, \ldots, v_{i500}\}
\end{equation}
\begin{equation}
\hat{d}_j = \{v'_{j1}, v'_{j2}, \ldots, v'_{j300}\}
\end{equation}

where $v_{ij}$ is the $j$th element in document vector $i$. Namely, in SSM $\hat{d}_i$ is the $ith$ row of elements from corpus matrix $C$ in \cite{8}. For the final dimensions reduction, as well as for the construction of the latent semantic space with the help of LSI technology in \cite{35}, we can further decrease the number of dimensions from 500 to $k_1$ and from 300 to $k_2$, respectively, that is
\begin{equation}
\hat{d} = \{v_{11}, v_{12}, \ldots, v_{k_1}\}
\end{equation}
\begin{equation}
\hat{d} = \{v'_{11}, v'_{12}, \ldots, v'_{k_2}\}
\end{equation}

where $k_1(k_1 < 500)$ and $k_2(k_2 < 300)$ are two integers. For termination criterion, we use a value of 20 for $n_{max}$ and the maximum fixed number of iterations $N_{max}$ is set as 2000.

In our experiments, we take advantage of OpenCV (Open Source Computer Vision) for single value decomposition of term-by-document matrix $A$ in \cite{7}. OpenCV is a library of programming functions designed mainly for real time computer vision computing. We use JWNL (Java WordNet Library) as API to access WordNet in our program. The fuzzy control GA program is written in C++ language and compiled in Visual C++ 6.0. Then, we access JWNL using C++ program in VS, and run it on a Pentium 3.2 GHz PC. All the figures in the experiments part are drawn by Matlab 7.0. In the subsequent part we will show how our SSM works well to simulate the original VSM and to provide a latent semantic space by choosing an appropriate dimension $k$ in \cite{34}.

5.2. The evaluation of our SSM for text representation

It has been abundantly proved in multidimensional scaling (MDS) literature \cite{2,42} that the detected ‘distance’ between objects can depict the similarity between them, since objects can be regarded as points in $N$ dimensional space. We apply MDS to perceive how our proposed corpus matrix $C_k$ in SSM simulates the original document-by-term matrix $D$ in VSM and provides a semantic space with the help of LSI technology. SStress criterion is utilized in this paper to evaluate the similarity between the two matrices. SStress is given by
\begin{equation}
SStress = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{s_{ij}^2 - s_{ij}^2}{2}\right)
\end{equation}

where $n$ is the total number of the documents, $s_{ij}$ represents the cosine similarity between documents $d_i$ and $d_j$, where $d_i$ and $d_j$ are the $ith$ row and the $j$th row of elements from the original document-by-term matrix $D$ in VSM and $s'_{ij}$ represents the cosine similarity between documents vector $\hat{d}_i$ and $\hat{d}_j$ chosen from the $ith$ row and the $j$th row of elements from the objective matrix $C_k$ in SSM. Suppose that $d_i$ and $d_j$ are represented by $(w_{i1}, w_{i2}, \ldots, w_{im})$ and $(w_{j1}, w_{j2}, \ldots, w_{jm})$ respectively, $s_{ij}$ is then defined as
\begin{equation}
\frac{\sum_{p=1}^{m}w_{i,p} \cdot w_{j,p}}{\sqrt{\sum_{p=1}^{m}w_{i,p}^2} \cdot \sqrt{\sum_{p=1}^{m}w_{j,p}^2}}
\end{equation}

where $m$ is the total number of terms. Suppose that $\hat{d}_i$ and $\hat{d}_j$ in matrix $C_k$ are represented by $(v_{i1}, v_{i2}, \ldots, v_{ik})$ and $(v_{j1}, v_{j2}, \ldots, v_{jk})$ respectively, $s'_{ij}$ is subsequently defined as
\begin{equation}
\frac{\sum_{p=1}^{k}v_{i,p} \cdot v_{j,p}}{\sqrt{\sum_{p=1}^{k}v_{i,p}^2} \cdot \sqrt{\sum_{p=1}^{k}v_{j,p}^2}}
\end{equation}

where $k(k \leq n)$ is the number of the dimensions selected, according to the definition of $C_k$ in \cite{34}, to construct a semantic space in SSM. The values of SStress against the number of dimensions $k$ for Reuter data set and 20-newsgroup data set are shown in Fig. 3.
From Fig. 3 we can see that for Reuter data set and 20-newsgroup data set, both SStress(s) decrease as the raising of dimension $k$. That is, the objective matrices $C_k(s)$ are more and more matching to their related original matrices $D(s)$, respectively, via cosine similarity calculation. At the beginning of several dimensions, SStress(s) decrease very rapidly, but later they decrease slowly as the dimension increases. When the dimension increases to around the amount of documents, i.e. 500 for Reuter data set and 300 for 20-newsgroup data set, the SStress(s) level off. For Reuter data set, the value of SStress is less than 150 with the dimensions ranging from 230 to 500. For 20-newsgroup data set, the value of SStress is less than 100 with the dimensions ranging from 170 to 300 dimensions. We empirically choose the dimensions within these two ranges, which can hopefully construct a semantic structure with the help of LSI. We will prove the effectiveness of these two ranges for a better clustering in subsequent experiments. Note that, it is these slight differences that render our SSM a robust space.

5.3. Fuzzy control genetic algorithm for text clustering

In this part we use precision, recall and F-measure to evaluate the performance of the proposed clustering algorithm. The precision $P_{ij}$ of cluster $j$ in type $i$ is defined as

$$P_{ij} = \frac{n_{ij}}{N_i}$$

and the recall $R_{ij}$ of cluster $j$ in type $i$ is defined as

$$R_{ij} = \frac{n_{ij}}{N_j}$$

where $n_{ij}$ is the number of documents in cluster $j$ which belongs to type $i$, $N_j$ is the number of documents in the cluster $j$, and $N_i$ is number of documents in the type $i$, which is predefined for the purpose of testing. The average precision $P$ is defined as

$$P = \frac{1}{K} \sum_{i=1}^{K} P_{ij}$$

and the average recall $R$ is defined as

$$R = \frac{1}{K} \sum_{i=1}^{K} R_{ij}$$

where $K$ is the number of clusters. Thus, F-measure $F$ is defined as

$$F = \frac{2 \times P \times R}{P + R}$$

To evaluate the performance of clustering, we divide our experiments into two steps. Firstly, fuzzy control GA is tested with SSM to detect exactly which dimension in SSM is most appropriate to represent the features of document, and makes fuzzy control GA get the best performance. And then, the dimension obtained from the first step is used to initialize $sim_{SSM}$ in hybrid similarity strategy in (37). We count the best F-measures of fuzzy control GA with SSM and VSM respectively. In our experiment we repeat the program 30 times and choose the average value of the best five F-measures of each method as the best F-measure to represent the performance of each method. We compare the performance of fuzzy control GA in SSM by
varying the number of dimensions $k$ from 50, 60, 70, 80, 90, 100, 120, 140, 160, 180, 200, 250, 300, 350, 400, 450 to 500 for Reuter data set and from 50, 60, 70, 80, 90, 100, 120, 140, 160, 180, 200, 220, 240, 260, 280 to 300 for 20-newsgroup respectively. Figs. 4 and 5 show the performance of the proposed fuzzy control GA. Moreover, we compare it with GA (conventional GA without fuzzy control) [23], commonly used K-means [38] and FCM [16] all with VSM (1000 terms for Reuter data set, the document representation in (53); 1500 terms for 20-newsgroup data set, the document representation in (54)).

From Fig. 4 we can see that the value of the F-measure of fuzzy control GA with VSM (1000 terms) is 0.720 which is higher than that of GA with VSM (1000 terms), with value of 0.691. Since K-means usually suffers from the limitation of the sub-optimum which depends on the choice of initial centers distribution, the F-measure of K-means with VSM (1000 terms) is very low, with value of 0.643. At the very beginning, the F-measure of fuzzy control GA with SSM increases very rapidly as the dimension increases, but when the dimension increases to about 90, the F-measure begins to increase very slowly. Fuzzy control GA achieves its best performance when dimension increases to 350, the value of F-measure being 0.765, that is obviously better than the performances of other three methods. When the dimension is close to about 450, the F-measure of fuzzy control GA slightly decreases, in comparison with its best performance.

From Fig. 5 we can see that in VSM the respective F-measures of fuzzy control GA, GA, and K-means, with 1500 terms, are 0.732, 0.701, and 0.652. The F-measure of fuzzy control GA with SSM increases with the raising of dimension. When dimension increases to 180, fuzzy control GA achieves its best performance, with value of 0.768. In comparison with Fig. 3, within dimensions from 170 to 300, the value of SStress is less than 100, where our method of SSM hopefully constructs a robust semantic space which will accordingly enhance our ability of distinguishing documents. When the dimension reaches to about 240, the F-measure of fuzzy control GA slightly decreases.

Note that, in SSM when dimension $k$ is equal to the number of documents, $C_k$ in SSM precisely simulates the original document-by-term matrix $D$ in VSM, and the value of SStress is zero. That is to say, in this case, the cosine similarity between any pair of documents in SSM is same to that in VSM. Therefore, in principle, the method will just have the same performance in both VSM and SSM as long as the method is good enough. But the difference lies that, in VSM, the respective

![Fig. 4. Clustering performance a for Reuter data set.](image1)

![Fig. 5. Clustering performances for 20-newsgroup data set.](image2)
features for document representation are 1000 terms for Reuter data set, and 1500 terms for 20-newsgroup data set, but in SSM, when dimension \( k \) is equal to the number of documents, the respective features for document representation are 500 for Reuter data set, and 300 for 20-newsgroup data set. It’s this difference that makes the complexity for the chromosome of searching for the best solution, in VSM, increase exponentially. Therefore, in practice, it’s very hard to produce the best global optimal solution. In spite of the fact that, only when dimension \( k \) equals to the number of documents, is the similarity between documents equal to each other in VSM and SSM, fuzzy control GA in SSM has a better performance than that in VSM.

It has been well recognized in semantic similarity literature \([21,45]\) that the properly defined parameters can provide dramatic results for similarity calculation. That is, how to define the value of \( \lambda \) in (37) is critical for the successful working of our clustering method. The choice of the various \( \lambda \) has a significant impact on which two semantic measures proposed will be selected to evaluate the similarity between documents. And consequently, the effectiveness of the clustering method will also be affected. It is intuitive that \( \text{sim}_{\text{SSM}} \) plays more important role than \( \text{sim}_{\text{ONTO}} \) for calculating the hybrid semantic similarity in spite of the fact that it could reveal semantic relations between two orthogonal indexing words from concept-based tree like hierarchy structure. Since \( \text{sim}_{\text{ONTO}} \) is only a secondary factor to install semantic relations from the third party software, and it is usually restricted by lexical resource in Wordnet, e.g., a document, in some specialized domains, does not necessarily include WordNet lexicon words or after stemming some formal words are broken up into incomplete forms which will not be included in WordNet lexicon. Hence, some important concepts will be lost. In view of these facts, \( \lambda \) is well initialized in the interval of \((0.50 < \lambda < 1.00)\), and a gradient descent approach is utilized to vary the value of \( \lambda \) from 1.00 to 0.50 with interval of 0.05. Note that, we have calculated \( \text{sim}_{\text{SSM}} \) by selecting the fixed dimensions \( k \), that is, 350 dimensions for Reuter data set gained from Fig. 4 and 180 for 20-newsgroup data set gained from Fig. 5. In Figs. 6 and 7, we count \( \lambda \) where fuzzy control GA(s) achieve their best results, and \( \lambda \) in (37) is finally assigned with the value of 0.65. In our experiments, various value of \( \lambda \) may cause the same F-measure results, e.g., \( \lambda \) varied from 0.90 to 1.00 generates the same F-measure of 0.765 in Fig. 6, and \( \lambda \) varied from 0.85 to 1.00 generates the same F-measure of 0.768 in Fig. 7, due to the same final clustering distribution.
In order to achieve our ultimate goal, that is, evaluating the performance of our clustering algorithm with the hybrid similarity strategy in [37], we calculate the hybrid similarity by selecting the dimensions, where fuzzy control GA(s) achieve their best performances with SSM, to initialize \( \text{sim}_{\text{SSM}} \) in (37), namely, 350 dimensions for Reuter data set and 180 for 20-newsgroup data set. We compare fuzzy control GA with the GA (conventional GA without fuzzy control), FCM and K-means via the same similarity measures. The average precision \( P \), average recall \( R \), average F-measure \( F \), and the approximate convergent generations \( G \) for each method are illustrated in Table 1.

From Table 1 we can see that for Reuter data set, fuzzy control GA with the hybrid similarity, gets the best clustering performance, according to the results of precision, recall and F-measure. The precision, recall and F-measure of fuzzy control GA with SSM are same to those of the conventional GA with the hybrid similarity respectively, due to the same clustering results distribution. However, the clustering results of these two methods are both better than those of the conventional GA with sole SSM. We can notice that fuzzy control GA(s), with the hybrid similarity and SSM, perform better than the conventional GA(s) with the same similarity measures, because with the help of the appropriate fuzzy control the adaptive \( p_c \) and \( p_m \) makes GA effectively prevent premature convergence and explore better solutions. For 20-newsgroup data set, the precision provided by fuzzy control GA with the hybrid similarity method, is only inferior to that of fuzzy control GA with SSM, but the recall of the former method is the best. We also count the F-measure for each method. For Reuter data set, the F-measures for fuzzy control GA with the hybrid similarity is 0.782, for that with SSM is 0.765, for that with VSM is 0.720, for GA with the hybrid similarity is 0.765, for GA with SSM is 0.745, for GA with VSM is 0.691, for FCM with VSM is 0.652, and for K-means with VSM is 0.643. For 20-newsgroup data set, the F-measures for these methods are 0.775, 0.768, 0.732, 0.760, 0.730, 0.701, 0.692 and 0.652, respectively. It has known that in the hybrid strategy and SSM, the number of dimensions, i.e. 350 for Reuter data set and 180 for 20-newsgroup set, for document representation is much less than that in VSM, i.e. 1000 terms for Reuter data set and 1500 terms for 20-newsgroup. The reduction of dimension greatly improves the convergence speed of GA indirectly. Here we only compare the speed of fuzzy control GA and the conventional GA both with hybrid strategy, SSM and VSM. For each method, the initial population is created with a random distribution and the convergent generation is different in spite of the same method. Therefore, it is difficult to count the convergence speed of each method exactly. Because we select the best five repeats to represent the performance of each method, we can only use an approximate number to represent the speed. We have assumed that in case the process does not terminate normally, the process is executed for the maximum fixed number of 2000 iterations. From Table 1 we can see that for Reuter data set, the conventional GA(s) with the hybrid similarity and SSM usually terminate at 2000 generations, so does the conventional GA with SSM on 20-newsgroup. Fuzzy control GA is much faster than the conventional GA, when they use the same hybrid similarity strategy. That is because the well defined adaptive parameters accelerate the convergence speed. Moreover, fuzzy control GA with the hybrid similarity is faster than that with SSM, because the hybrid similarity strategy enhances its ability of distinguishing documents.

6. Conclusions

In this paper, a fuzzy control genetic algorithm (GA) is proposed, in conjunction with a hybrid semantic similarity measure, for document clustering. The common problem in the field of text clustering is that the document is represented as a bag of words, while the conceptual similarity is ignored. We take advantage of thesaurus-based and corpus-based semantic methods to solve this problem. The hierarchical structure and the broad-coverage taxonomy of WordNet are utilized as the thesaurus-based ontology. However, the sole thesaurus-based method is restricted by its vocabulary. e.g., a document, in some specialized domains, does not necessarily include WordNet lexicon words or after stemming some formal words are broken up into incomplete forms which will not be included in WordNet lexicon. Hence, some important concepts will be lost. A semantic space model, named SSM, which can reveal the true semantic relationships between documents, is proposed as the corpus-based similarity measure in this study. We have demonstrated that our SSM with whole dimensions precisely simulates the original document-by-term matrix and the properly reduced dimensions provide a dramatic feature space for document representation. Thus, our method of SSM is combined with the thesaurus-based measure as a hybrid strategy to evaluate the similarity. According to the nature of GA evolution, we use fuzzy control to adaptively determine

<table>
<thead>
<tr>
<th>Data set</th>
<th>Reuter data set (k = 350)</th>
<th></th>
<th></th>
<th></th>
<th>20-newsgroup data set (k = 180)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
<td>G</td>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
</tr>
<tr>
<td>Fuzzy control GA</td>
<td>0.774</td>
<td>0.790</td>
<td>0.782</td>
<td>1400</td>
<td>0.763</td>
<td>0.787</td>
<td>0.775</td>
<td>900</td>
</tr>
<tr>
<td>with the hybrid similarity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzzy control GA</td>
<td>0.746</td>
<td>0.785</td>
<td>0.765</td>
<td>1600</td>
<td>0.774</td>
<td>0.762</td>
<td>0.768</td>
<td>1200</td>
</tr>
<tr>
<td>with SSM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzzy control GA</td>
<td>0.716</td>
<td>0.724</td>
<td>0.720</td>
<td>2000</td>
<td>0.726</td>
<td>0.738</td>
<td>0.732</td>
<td>2000</td>
</tr>
<tr>
<td>with VSM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GA with the hybrid similarity</td>
<td>0.746</td>
<td>0.785</td>
<td>0.765</td>
<td>2000</td>
<td>0.746</td>
<td>0.774</td>
<td>0.760</td>
<td>1700</td>
</tr>
<tr>
<td>GA with SSM</td>
<td>0.731</td>
<td>0.768</td>
<td>0.745</td>
<td>2000</td>
<td>0.706</td>
<td>0.762</td>
<td>0.730</td>
<td>2000</td>
</tr>
<tr>
<td>GA with VSM</td>
<td>0.667</td>
<td>0.717</td>
<td>0.691</td>
<td>2000</td>
<td>0.688</td>
<td>0.714</td>
<td>0.701</td>
<td>2000</td>
</tr>
<tr>
<td>FCM with VSM</td>
<td>0.642</td>
<td>0.662</td>
<td>0.652</td>
<td>–</td>
<td>0.688</td>
<td>0.696</td>
<td>0.692</td>
<td>–</td>
</tr>
<tr>
<td>K-means with VSM</td>
<td>0.635</td>
<td>0.653</td>
<td>0.643</td>
<td>–</td>
<td>0.649</td>
<td>0.655</td>
<td>0.652</td>
<td>–</td>
</tr>
</tbody>
</table>
the probabilities of crossover and mutation (\(p_c\) and \(p_m\)). For all individuals in population, if the average fitness is close to the best one while far from the worst one, the current distribution will be dense which may cause a premature convergence, so we need to increase the value of \(p_c\) and \(p_m\). Otherwise, we need to decrease them. For each individual, in order to preserve good solutions, we decrease the values of \(p_c\) and \(p_m\) for high fitness individuals. In contrast, we increase the values of \(p_c\) and \(p_m\) for low fitness individuals for an extensive exploration. It is these dynamically regulated parameters that accelerate the convergence speed as well as prevent premature convergence of GA to a suboptimum. The combination of the hybrid similarity strategy with fuzzy control GA, not only leads a dramatic dimensions reduction for document representation which improves the search capacity of GA indirectly, but also enhances its ability of distinguishing documents in terms of concepts, rather than individual terms. 500 documents from Reuter document collection and 300 documents from 20-newsroup corpus are selected for testing. We divide our experiments into two steps. Fuzzy control GA is implemented with SSM first to detect exactly which dimension in SSM is most appropriate to represent the features of documents, and makes fuzzy control GA get the best performance. The experiment results show that when the dimensions are respectively reduced to 350 for Reuter data set and 180 for 20-newsroup data set, fuzzy control GA(s) achieve their best performances, which are much better than those of the conventional GA, FCM and K-means in VSM. Then in order to evaluate the performance of our clustering algorithm with the hybrid similarity strategy, we use the dimensions obtained from the previous step to initialize \(sim_{SSM}\) in the hybrid similarity strategy. The results show that fuzzy control GA with the hybrid strategy performs best, in comparison with that with sole SSM and the conventional GA(s) with the hybrid strategy and SSM. We also notice that fuzzy control GA(s) perform better than conventional GA(s), when they use the same similarity measures. Moreover, with the same clustering algorithms, i.e. fuzzy control GA and the conventional GA, the hybrid semantic similarity measure is better than SSM.

Acknowledgements

The authors thank the Editor in Chief and the reviewers for providing very helpful comments and suggestions. Their insight and comments led to a better presentation of the ideas in this paper. This work was sponsored by National Natural Science Foundation of China (61103129), Natural Science Foundation of Jiangsu Province (SBKZ201122266), SRF for ROCS, SEM, Brain Korea 21, and the Specialized Research Fund for the Doctoral Program of Higher Education (20100093120004).

References