Computers and Mathematics with Applications 57 (2009) 1901–1907

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Genetic algorithm for text clustering based on latent semantic indexing☆

Wei Song*, Soon Cheol Park

Division of Electronics and Information Engineering, Chonbuk National University, Jeonju, 561756, Republic of Korea

ARTICLE INFO

Keywords:
Document representation model
Genetic algorithm
Latent semantic indexing
Text clustering

ABSTRACT

In this paper, we develop a genetic algorithm method based on a latent semantic model (GAL) for text clustering. The main difficulty in the application of genetic algorithms (GAs) for document clustering is thousands or even tens of thousands of dimensions in feature space which is typical for textual data. Because the most straightforward and popular approach represents texts with the vector space model (VSM), that is, each unique term in the vocabulary represents one dimension. Latent semantic indexing (LSI) is a successful technology in information retrieval which attempts to explore the latent semantics implied by a query or a document through representing them in a dimension-reduced space. Meanwhile, LSI takes into account the effects of synonymy and polysemy, which constructs a semantic structure in textual data. GA belongs to search techniques that can efficiently evolve the optimal solution in the reduced space. We propose a variable string length genetic algorithm which has been exploited for automatically evolving the proper number of clusters as well as providing near optimal data set clustering. GA can be used in conjunction with the reduced latent semantic structure and improve clustering efficiency and accuracy. The superiority of GAL approach over conventional GA applied in VSM model is demonstrated by providing good Reuter document clustering results.

1. Introduction

Clustering is an unsupervised pattern classification technique which is defined as group n objects into m clusters without any prior knowledge. The number of partitions/clusters may or may not be known a priori. The task of document clustering is both difficult and intensively studied [1,2]. Several algorithms for clustering data when the number of clusters is known a priori are available in the literature. K-means algorithm [3], one of the most widely used, attempts to solve the clustering problem into a fixed number of clusters K known in advance. It is an iterative hill-climbing algorithm and solution suffering from the limitation of the sub optimal which is known to depend on the choice of initial clustering distribution [4]. In [5], a branch and bound algorithm uses a tree search technique to search the entire solution space. It employs a criterion of eliminating sub trees which do not contain the optimal result. In this scheme, the number of nodes to be searched becomes huge as the size of the data set becomes large. Several types of biologically inspired algorithms have been proposed in the literature. Ant clustering algorithm [6] is to project the original data into bidimensional output grid and position that are similar to each other in their original space of attributes. By doing this, the algorithm is capable of grouping together items that are similar to each other. Genetic algorithm (GA) [7,8] belongs to search techniques that mimic the principle of natural selection. GA performs a search in complex, large and multimode landscapes, and provides near-optimal solutions for objective or fitness function of an optimization problem. However, the cost of computational time is high because its long string representation evolves in high dimensional space typical for textual data [9].

☆ This work was partially supported by the Korea Research Foundation (Grant Nos. KRF-2006-321-A00012) and partially supported by Brain Korea 21.
* Corresponding author.
E-mail addresses: songwei9988@yahoo.com.cn (W. Song), scpark@chonbuk.ac.kr (S.C. Park).

0898-1221/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2008.10.010

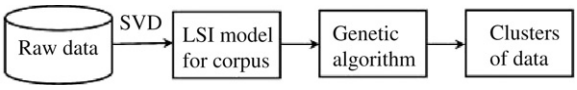

Fig. 1. The process of GA for text clustering based on LSI.

The most general and straightforward approach to represent text is the vector space model (VSM), it means each unique term in the vocabulary represents one dimension in feature space. Unfortunately, it needs a large number of features to represent high dimensions, and it is not suitable for GA since the scalability will be poor and the cost of computational time will be high. Meanwhile, if we represent all texts in this way many documents that are related to each other semantically might not share any words and thus appear very distant, and occasionally documents that are not related to each other might share common words and thus appear to be closer. This is due to the nature of text, where the same concept can be represented by many different words, and words can have ambiguous meanings. Latent semantic indexing (LSI) [10] is an automatic method that reduces this large space to one that hopefully captures the true relationships between documents [11]. LSI uses the singular value decomposition (SVD) technique to decompose the large term-by-document matrix into a set of k orthogonal factors. In this semantic structure, even two documents do not have any common words, we also can find the associative relationships, because similar contexts in the documents will have similar vectors in semantic space. The process of GA for text clustering based on LSI is shown in Fig. 1.

In this paper, we propose a variable string length GA using a gene index to encode chromosomes in the semantic space. The gene index indicates the location of each gene in the chromosome, which has a greater chance of obtaining the appropriate center combination and to find the proper number of clusters.

In the next section, we give a brief review of LSI, and describe how we use it for text clustering. Details of genetic algorithms for text clustering based on the LSI model are described in Section 3. Experiment results are given in Section 4. Conclusions and future works are given in Section 5.

2. Latent semantic indexing model for documents representation

The purpose of LSI is to extract a smaller number of dimensions that are more robust indicators of meaning than individual terms. Once a term-by-document matrix is constructed, LSI requires the singular value decomposition of this matrix to construct a semantic vector space. Due to the word-choice variability, the less important dimensions corresponding to "noise" are ignored. A reduced rank approximation to the original matrix is constructed by dropping these noisy dimensions.

2.1. Singular value decomposition

Our corpus can be firstly represented as a term-by-document matrix $X(m \times n)$, assuming there are m distinct terms in an n documents collection. The singular value decomposition of X is given by

$$X = U\Sigma V^{\mathrm{T}} \tag{2.1}$$

where U and V are the matrices of the left and right singular vectors. Σ is the diagonal matrix of singular values. LSI approximates X with a rank k matrix.

$$X_k = U_k \Sigma_k V_k^{\mathrm{T}} \tag{2.2}$$

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 matrix V^{T}. $\Sigma_k = \mathrm{diag}(\sigma_1, \ldots, \sigma_k)$ is the first k factors. That is, the documents are represented in the k dimensional LSI space spanned by the basis vectors.

2.2. General LSI model for information retrieval

When LSI is used for the purpose of information retrieval [12,13] query q is a $m \times 1$ matrix, where m is the number of terms in the documents collection. The query vector $\hat{q}$ is constructed by

$$\hat{q} = q^{\mathrm{T}} U_k \Sigma_k^{-1}. \tag{2.3}$$

2.3. Our approach of LSI model for document representation

When LSI is used for the purposes of document representation, a document d is firstly initialized as a $m \times 1$ matrix, where m is the number of terms. Because matrix U in (2.1) represents the matrix of terms vectors in all documents and the proper number of rank U_k spans the basis vectors of U. In our approach we remove Σ_k^{-1} matrix and use the multiplying of matrices d^{T} and U_k to represent the document vector. The results of our experiment also show that the representation of multiplying of two matrix U_k and Σ_k^{-1} is not as good as that of multiplying the single U_k matrix. So each document vector $\hat{d}$ is represented by $1 \times k$ matrix.