An Efficient Web Query Optimization Algorithm Based on LDA and MA

Ziqiang Wang
School of Information Science and Engineering
Henan University of Technology
Zhengzhou, China
wzqagent@126.com

Xia Sun
School of Information Science and Engineering
Henan University of Technology
Zhengzhou, China
wzqbox@gmail.com

***Abstract*—With the explosive growth in the Web documents, retrieving document from the large-scale document database has become one of the most active research fields in data mining communities. Thus, developing an efficient document query optimization algorithm to retrieval Web document is of great importance. An efficient document query optimization algorithm based on linear discriminant analysis(LDA) and memetic algorithm(MA) is proposed in this paper. The experimental results demonstrate that the proposed algorithm achieves much better performance than other conventional document query optimization algorithms.**

Keywords- document query; memetic algorithm(MA); linear discriminant analysis(LDA)

I. INTRODUCTION

The amount of online document has grown greatly in recent years because of the increase in popularity of the World Wide Web. In order to make use of this vast amount of data, efficient techniques to retrieve Web document information based on its content need to be developed. As a result, the role of information retrieval (IR) systems is becoming more important[1]. One of the most important and difficult operations in information retrieval is to generate queries that can succinctly identify relevant documents and reject irrelevant documents. In addition, the document data are typically of very high dimensionality, ranging from several thousands to several hundreds of thousand. High-dimensional data often leads to inferior retrieval results due to the curse of dimensionality. To achieve higher efficiency in manipulating the Web document data, it is desirable to first project the documents into a lower-dimensional subspace in which the semantic structure of the document space becomes clear. Once the high-dimensional document space is mapped into a lower dimensional space, the traditional information retrieval algorithms can then be applied.

The document space is generally of high dimensionality and querying in such a high dimensional space is often infeasible due to the curse of dimensionality. Therefore, dimensionality reduction is essential to the design of efficient document retrieval algorithms. Linear Discriminant Analysis (LDA)[2] is one of the most well-known dimensional reduction methods. The goal of LDA is to find a linear transformation that maximizes the between-class scatter and minimizes the within-class scatter so that the class separability can be optimized in the transformed space. LDA has been successfully used as a dimensionality reduction technique to many pattern recognition problems, such as information retrieval, face recognition, and microarray gene expression data analysis. In this paper, we adopt LDA for document dimensional reduction, since it is a popular and widely dimensionality reduction method in pattern recognition.

In addition, one of the most important and difficult operations in information retrieval is to generate queries that can succinctly identify relevant documents and reject irrelevant documents. Relevance feedback (RF) is an important tool to improve the performance of information retrieval (IR)[1]. RF focuses on the interactions between the user and the search engine by letting the user label semantically relevant or irrelevant documents. Recently, Moscato introduced the term "memetic algorithm" (MA)[3] which combines evolutionary algorithms with the intensification power of a local search, and has a pragmatic perspective for better effects than GA. As such MA, a local optimizer is applied to each offspring before it is inserted into the population in order to make it towards optimum and then GA platform as a means to accomplish global exploration within a population. MAs have been successfully applied to several NP optimization problems such as scheduling problem, cell formation problem and TSP problem. Nevertheless, the use of the algorithm for document query optimization is still a research area where few people have tried to explore. In this paper, the objective is to investigate the capability of the memetic algorithm(MA) boosted by LDA-based dimensionality reduction method for Web document query optimization in the context of information retrieval.

The remainder of the paper is organized as follows: Section II introduces the LDA-based dimensionality reduction method. Section III gives a brief review of memetic algorithm(MA). In Section IV, we describe how the memetic algorithm can be applied to Web document query optimization. The experimental results are presented in Section V. Finally, the paper ends with some conclusions.

II. LDA-BASED DOCUMENT DIMENSIONAL REDUCTION

Web document sets are huge, so we need to find a lower dimensional representation of the data. Dimensionality reduction is the representation of high-dimensional patterns in a low-dimensional subspace based on a transformation

978-0-7695-3556-2/08 $25.00 © 2008 IEEE
DOI 10.1109/MMIT.2008.62

which optimizes a specified criterion in the subspace. This is important in information retrieval, since the lower dimensionality approximation is not just a tool for transforming a given problem into another one which is easier to solve, but the reduced representation itself will reduce the cost of the post-processing involved in document retrieval, and will improve retrieval speed and efficiency.

The linear discriminant analysis (LDA)[2] algorithm is a popular dimensional reduction algorithm which aims to find an optimal transformation that maps the data into a lower-dimensional space that minimizes the within-class distance and simultaneously maximizes the between-class distance, thus achieving maximum discrimination. LDA takes the relationship between different documents into consideration, and ensures the minimum redundancy among the features in the reduced space, which makes it very attractive for document query optimization applications.

Suppose that we have a set of *n d*-dimensional samples $x_1,\ldots,x_n$ belonging to *c* different classes with samples in the subset D_i labeled l_i, $i=1,\ldots,c$ .Then the objective of LDA is to seek the transform W not only maximizing the between-class scatter of the projected samples, but also minimizing the within-class scatter, such that the following criterion function is maximized.

$$J(W)=\arg\max_{W}\frac{\left|W^T S_b W\right|}{\left|W^T S_w W\right|} \tag{1}$$

where S_b and S_w denotes the between-class scatter matrix and within-class scatter matrix, respectively. Their definitions are as follows:

$$S_b=\sum_{i=1}^{c} n_i\left(\mu_i-\mu\right)\left(\mu_i-\mu\right)^T \tag{2}$$

$$S_w=\sum_{i=1}^{c}\sum_{j=1}^{n_i}\left(x_j-\mu_i\right)\left(x_j-\mu_i\right)^T \tag{3}$$

where n_i is the number of samples in the *i*-th class, $\mu_i=\left(1/n_i\right)\sum_{j=1}^{n_i}x_j$ is the sample mean of the *i*th class, and $\mu=\left(1/n\right)\sum_{i=1}^{n}x_i$ is the total samples mean.

It has been proved that if S_w is nonsingular, the solution vector W is the eigenvector of the matrix $S_w^{-1}S_b$ corresponding to the largest eigenvalues. For document classification, a problem arises that the matrix S_w cannot be guaranteed to be nonsingular, since the number of terms in the document collection is larger than the total number of documents. To overcome this problem, we can first apply PCA to project the documents into a subspace by throwing away the components corresponding to zero eigenvalue, then the matrix S_w becomes nonsingular.

After the dimensional reduction process, the query search algorithm is ready to be used for categorizing a new document in the reduced low-dimensional document feature space into relevant or irrelevant document. For the query search algorithm in the document query optimization algorithm, we adopt memetic algorithm[3] that show significant improvement over other evolutionary algorithms when applied to function optimization problems.

III. MEMETIC ALGORITHM(MA)

Memetic algorithms(MA) is inspired by Dawkins' notion of a meme [3,4]. MA is similar to GA but the elements that form a chromosome are called memes, not genes. The unique aspect of the MAs algorithm is that all chromosomes and offsprings are allowed to gain some experience, through a local search, before being involved in the evolutionary process. A pseudocode for a MA procedure is given in as follows[3,5]:

Begin;
Generate random population of *P* solutions (chromosomes);
For each individual $i\in P$: calculate fitness (i);
For each individual $i\in P$: do local-search(i);
For *i*=1 to number of generations;
Randomly select an operation (crossover or mutation);
If crossover;
Select two parents at random i_a and i_b;
Generate on offspring $i_c=crossover(i_a,i_b)$;
$i_c=local-search(i_c)$;
Else If mutation;
Select one chromosome *i* at random;
Generate an offspring $i_c=mutate(i)$;
$i_c=local-search(i_c)$;
End if;
Calculate the fitness of the offspring;
If i_c is better than the worst chromosome then replace the worst chromosome with i_c;
Next *i*;
Check if termination=true;
End.

From the above pseudocode of memetic algorithm (MA), we can observe that the parameters involved in MAs are the same four parameters used in GA: population size, number of generations, crossover rate, and mutation rate in addition to a local-search mechanism.

IV. THE MA-BASED QUERY ALGORITHM

The proposed query optimization method is based on a vector space model[1]in which both documents and queries are represented as vectors. After dimension reduction by the above described LDA algorithm, the memetic algorithm

(MA) was used in the reduced dimensional space. The goal of the MA is to find an optimal set of documents which best match the user's requirement by exploring different regions of the document space simultaneously. The detail steps of the MA-based document query optimization algorithm are outlined as follows.

A. The Encoding of Query Individual

The first step toward implementation of the MA is the definition the chromosomes to be evolved (i.e., solution space).In this paper, the chromosome is represented by query vector space. Each query individual representing a query is of the following form.

$$Q_u(q_{u1}, q_{u2}, \cdots, q_{uT}) \tag{4}$$

where T is total number of stemmed terms automatically extracted from the documents, q_{ui} is the weight of the ith term in Q_u , Initially, a term weight is computed as the following formula[6].

$$q_{ui} = \frac{(1+\log(tf_{ui}))\cdot\log(\frac{N}{n_i})}{\sqrt{\sum_{k=1}^{T}((1+\log(tf_{uk}))\cdot\log(\frac{N}{n_k}))^2}} \tag{5}$$

where tf_{ui} is the frequency of term t_i in document d_u , N is the total number of documents, n_i is the number of documents containing the term i.

B. Fitness Function Calculation

The fitness represents the effectiveness of a query during the retrieving stage. It is computed according to the relevance of the retrieved documents. The definition of fitness function is as follows:

$$Fitness(Q_u^{(s)}) = \frac{\sum_{d_j \in D_r^{(s)}} Sim(d_j, Q_u^{(s)})}{\sum_{d_j \in D_{nr}^{(s)}} Sim(d_j, Q_u^{(s)})} \tag{6}$$

where N is the total number of documents, $D_r^{(s)}$ is the set of relevant documents retrieved at the generation(s) of the MA, d_j is the *jth* document , $D_{nr}^{(s)}$ is the set of non-relevant documents retrieved at the generation(s) of the MA, $Sim(d_j, Q_u^{(s)})$ is a similar measure function defined as follows.

$$Sim(d_j, Q_u^{(s)}) = \frac{\sum_{i=1}^{T}(q_{ui}^{(s)} \cdot d_{ji})}{\sqrt{\sum_{i=1}^{T} q_{ui}^2} \cdot \sqrt{\sum_{i=1}^{T} d_{ui}^2}} \tag{7}$$

C. Local Search

For each individual $Q_u \in P$: do the following local-search(Q_u) procedure.

Begin;

Select an incremental value $d = a * Rand()$, where a is a constant that suits the variable values, and $Rand()$ is random generator in [0,1];

For a given chromosome $Q_u \in P$: calculate fitness $Fitness(Q_u)$;

For j=1 to number of memes in chromosome Q_u;

$$q_{uj} = q_{uj} + d \tag{8}$$

If chromosome fitness not improved then

$$q_{uj} = q_{uj} - d \tag{9}$$

If chromosome fitness not improved then retain the original value q_{uj};

Next j;
End.

D. Loop

For i=1 to number of generations;

Select two parents at random Q_a and Q_b;

Generate on offspring $Q_c = crossover(Q_a, Q_b)$;

$Q_c = local - search(Q_c)$;

Generate an offspring $Q_c = mutate(Q_c)$;

Calculate the fitness of the offspring in terms of Equation (6);

If Q_c is better than the worst chromosome;

Then replace the worst chromosome with Q_c;

Next i;

E. Relevant Documents Merging

At each generation of MA, these retrieved relevant documents by all the individual queries of the query population are merged to a single document lists, and presented to user. Our adopted merging methods according to the following formula:

$$\mathrm{Re}l^{(s)}(d_j) = \sum_{Q_u^{(s)} \in Pop^{(s)}} Fitness(Q_u^{(s)}) \cdot RSV(Q_u^{(s)}, d_j) \tag{10}$$

where $Pop^{(s)}$ is the population at the generation(s) of the MA; $RSV(Q_u^{(s)}, d_j)$ is the retrieval status value(RSV)[7] of the document d_j for the query $Q_u^{(s)}$ at the generation(s) of the MA.

F. Termination Condition

Check if the iteration number approaches to the predefined maximum iteration.

V. EXPERIMENTAL RESULTS

In order to demonstrate the effectiveness of the proposed document query optimization algorithm based on MA, we conducted a series of experiments to compare the MA-based algorithm with other document query optimization algorithm

based on relevant feedback(RF) approach[8] and genetic algorithm(GA)[9]. The parameter values of the proposed memtic algorithm (MA) are empirically set as follows: population size=50, maximum number of generation=4, selection rate=0.75, crossover rate=0.75, mutation rate=0.03, and constant a=0.5.

The standard document collections Reuters-21578[10] was used in our experiments.The actual computational time of different algorithms is given in Table I. All of these three algorithms can respond to the user's query very fast, that is, within 0.1s. Our memtic algorithm (MA) is slightly faster than other two algorithms. The reason is that the LDA-based dimensionality reduction method reduces the cost of the post-processing involved in document retrieval, and improves retrieval speed and efficiency.

TABLE I. AVERAGE RUNNING TIME COMPARISON

Method	Ite-1(s)	Ite-2(s)	Ite-3(s)	Ite-4(s)
MA	0.046	0.058	0.063	0.067
GA	0.067	0.072	0.076	0.085
RF	0.074	0.082	0.086	0.091

TABLE II. RELEVANT DOCUMENT COMPARISON

Method	Ite-1	Ite-2	Ite-3	Ite-4
MA	112	70	54	57
GA	89	62	58	51
RF	83	60	56	54

In addition, we also compare the number of relevant document retrieved using MA, GA and RF. Table II gives the number of relevant document retrieved at each iteration of the three optimization algorithm. We can clearly see that our MA more effective than GA and RF in retrieving relevant documents. Indeed the cumulative total number of relevant documents using MA through all the iterations is higher than using GA and RF. Therefore, our proposed document query optimization algorithm efficiently improves the performance of the query search.

VI. CONCLUSIONS

In this paper, an efficient document query algorithm based on LDA and memtic algorithm (MA) is proposed. The experimental results show that the proposed algorithm achieves much better performance than other conventional document query optimization algorithms.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant No.70701013, the Natural Science Foundation of Henan Province under Grant No. 0611030100 and 072300410020.

REFERENCES

[1] G.Salton and M.J.McGill, Introduction to Modern Information Retrieval.New York:McGraw-Hill,1983.

[2] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, Second edition. Hoboken:Wiley-Interscience,2000.

[3] M.Pablo, “On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms,” Technical Report Caltech Concurrent Computation Program Report 826. Pasadena:California Institute of Technology,1989,pp.1-68.

[4] D.Richard,The Selfish Gene. Cary:Oxford University Press,1990.

[5] E.Elbeltagi, T.Hegazy, and D.Grierson, “Comparison among five evolutionary-based optimization algorithms,” Advanced Engineering Informatics, vol.19,pp.43-53,January 2005.

[6] A.Singhal, C.Buckley, and M.Mitra,“Pivoted document length normalisation,” Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval(SIGIR 96), ACM Press,August 1996,pp.21-29.

[7] A.I.Aguilera and L.J.Tineo,“Flexible query processor for information,” Proceedings of the Ninth IEEE International Conference on Fuzzy Systems(FUZZY 00), IEEE Press,May 2000, pp.1009-1012.

[8] B.T.Bartell,G.W.Cortell,and R.K.Belew, “Optimising similarity using multiquery relevance feedback,” Journal of the American Society for Information Science,vol.49,pp.742-761,December 1998.

[9] J.-T.Horng and C.-C.Yeh, “Applying genetic algorithms to query optimization in document retrieval,” Information Processing and Management, vol.36,pp.737-759,September 2000.

[10] D.D.Lewis,Reuters-21578 text categorization test collection distribution 1.0.http://www.research.att.com/lewis,1999.