Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1999 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1999

Automatic Generation of Matching Function by Genetic Programming for Effective Information Retrieval

Weiguo Fan
University of Michigan Business School

Michael Gordon
University of Michigan Business School

Praveen Pathak
University of Michigan Business School

Automatic Generation of a Matching Function by Genetic Programming for Effective Information Retrieval

Weiguo Fan   Michael D. Gordon   Praveen Pathak
Department of Computer and Information Systems
University of Michigan Business School, 701 Tappan, 48109, USA
{wfan, mdgordon, praveen}@umich.edu

Abstract

With the advent of the Internet, online resources are increasingly available. Many users choose popular search engines to perform an online search to satisfy their information need. However, these search engines tend to turn up many non-relevant documents, which make their retrieval precision very low. How to find appropriate ranking metrics to retrieve more relevant documents and fewer non-relevant documents for users remains a big challenge to the information retrieval community. In this paper, we propose a new framework that combines the merits of genetic programming and relevance feedback techniques to automatically generate and refine the matching functions used for document ranking. This approach overcomes the shortcoming of traditional ranking algorithms using a fixed ranking strategy. It also gives some new ideas and hints for information retrieval professionals.

Introduction

There is a virtual explosion in the availability of electronic information. The advent of the Internet or World Wide Web (WWW) has brought far more information than any human being can absorb. The aim of information retrieval (IR) systems is to organize and store such information, and retrieve useful information when a user submits a query to the IR systems. Vector space models are normally used to represent both the contents of the query and the document collections. When a query representing the user's information need is submitted to the IR system, the system will first extract the keywords (terms or phrases) from the query and represent them with a keyword vector, then utilize its search engine to match this vector with vectors representing documents. A list of documents which are ranked in descending order of their relevance are returned to the user. The effectiveness of an IR system predicted is normally measured by two ratios: *recall* and *precision*. *Recall* is the ratio of the number of relevant documents retrieved to the total number of relevant documents in the whole collection. *Precision* is defined as the number of the relevant documents retrieved to the total number of documents retrieved.

A recent study, by (Gordon et al. 1999) indicated the precision and recall of the commonly used search engines like AltaVista, Infoseek, etc. are normally very low. So users often have to sift through many web pages to find a small set of documents that satisfy their information needs. One important factor that determines the final ranking of documents to users is the matching function (also called similarity function). There are many matching functions reported in the IR literature, like Cosine, Dice, and others (Salton 1989). The most commonly used and extensively studied matching function is the generalized Cosine matching function, which can be summarized using the following formula:

$$Sim(Q,D) = \frac{\sum_{i=1}^{T} w_{qi} \times w_{di}}{Normalizing\_factor} \qquad (1)$$

where Q and D represent the query vector and document vector respectively. T is the number of common terms between the query vector and the document vector. w_{qi} and w_{di} are the weighting factors for the term t_i in query vector and document vector respectively. Different term weighting strategies may produce quite different retrieval results. We will focus our discussion on this set of matching functions because of its wide usage in most commercial IR systems and search engines. However, our framework can be easily generalized to other similarity functions.

Extensive research studies have been done on the effectiveness of various term weighting strategies on final document rankings (Salton et al. 1988; Singhal 1997). However, these studies indicate that there is no single term weighting strategy that can work consistently well among collections of different size as well as queries of different length. Moreover, all the experiments done in these studies were based on a previously determined set of term weighting strategies.

One exception is the work done by Pathak (Pathak 1998). He proposed a new weighted matching function, which is the linear combination of different similarity functions. The weighting parameters were estimated by an adaptive genetic algorithms based on relevance feedback from users. Simulation experiments indicated that this new weighted matching function performs better than individual ones. We propose in this paper a new framework that automatically generates a matching function for different queries or collections. A set of commonly used feature weighting strategies, such as term frequency *tf*, document frequency *df* , inverse document frequency *idf*, etc., are composed adaptively by genetic programming. Novel matching functions for different queries are generated automatically based on feedback from users.

Our paper is organized as follows: First we briefly review the idea underlying genetic programming and relevance feedback in section 2 and section 3 respectively, then we propose our framework in section 4. Section 5 gives the conclusions and the directions for future research.

Genetic programming (GP)

The paradigm of genetic programming is based on Darwin's principle of survival of the fittest. Starting from a randomly generated initial population of computer programs, it evolves new populations following this principle. "Fitter" programs are those that come closest to solving a given problem or performing a particular task. These programs are generally hierarchically organized and of dynamically varying size and shape. The new individuals (programs) are a product of genetic operations, like crossover and mutation, on the current population's better individuals. Like genetic algorithms, genetic programming is heuristic and stochastic.

The following steps are generally performed by genetic programming systems (Koza 1992):

1) Generate an initial population of computer programs each formed from a random composition of a set of operators;

2) Perform the following sub-steps iteratively until a final termination criterion has been satisfied:

   a) Execute every individual's program and assign it a fitness value, according to its ability to solve the problem;

   b) Create a new population by applying genetic operators on the selected individuals:

      - Reproduction;
      - Crossover;

3) Take the best solution of the final population.

Relevance feedback (RF)

Relevance feedback is one of the processes in an information retrieval system that seeks to improve the system's performance based on a user's feedback. It modifies queries using judgments of the relevance of a few, highly-ranked documents and has historically been an important method for increasing the performance of information retrieval systems. Specifically, the user's judgments of the relevance or non-relevance of some of the documents retrieved are used to add new terms to the query and to reweight query terms. For example, if all the documents, that the user judges as relevant contain a particular term, then that term may be a good one to add to the original query. It is shown (Salton et al. 1990) that relevance feedback has improved the system's overall performance by 60% to 170% for different document collections. Given the apparent effectiveness of relevance feedback techniques, it is important that any proposed model of information retrieval include these techniques.

In our system, rather than modifying the query vector, we modify the matching function by genetic programming to reflect a user's feedback about relevance.

A new framework using genetic programming and relevance feedback

Following the notion of Koza (Koza 1992), the following settings are defined for our genetic programming system:

Table 1 Genetic programming system settings

Terminal	$tf\ ,df\ ,idf\ ,tf_{max}\ ,tf_{avg}\ ,\Re$
Functions	+, -, *, / , log, sqrt
Fitness measure	$1-\frac{1}{\frac{\alpha}{P}+\frac{(1-\alpha)}{R}}$

where *tf* is the term frequency, *df* is the document frequency, *idf* is the inverted document frequency, α is a parameter defined by the user, *P* is the precision of the retrieval results, *R* is the recall of the retrieval results. $\Re$ is the random constant terminal which allows various random floating point constants to be inserted at random.

We propose the following framework that adaptively generates matching functions:

1) Generate an initial population of random compositions of the terminals and functions;

2) Perform the following substeps iteratively until the final termination criterion has been satisfied:

   a) Execute every individual's program by calculating the matching score using Equation1 for a given query and all documents; assign the program a fitness value based on the feedback from the user;

   b) Create a new population applying genetic operators on the selected individuals:

      - Reproduction;
      - Crossover;

3) Report the final matching function and return the relevant documents.

The genetic programming terminates when the termination criterion is satisfied. Usually, we terminate a run either when a pre-specified number of generations has been reached, or when the precision or recall are above a predefined threshold.

Conclusions and future directions

Computer simulations have been conducted to demonstrate the effectiveness of applying genetic programming to automatically generate a matching function that produces more relevant documents for users. We have implemented a prototype of our genetic programming approach in C. In the future we will compare the quality of the solutions proposed by the GP approach with other existing approaches.

References:

Gordon, M. D. & Pathak, P. “Finding information on the WWW: Retrieval effectiveness of search engines”, Information Processing and Management, (To Appear).

Koza, J. R. “Genetic Programming: On the Programming of Computers by Means of Natural Selection”, MIT Press, Cambridge, MA, USA, 1992.

Pathak, P. “A Simulation model of document information retrieval system with relevance feedback”, Proceedings of the Fourth Americas Conference on Information Systems, Baltimore, 1998.

Salton, G. “Automatic Text Processing”, Addison-Wesley Publishing Co., Reading, MA, 1989.

Salton, G. & Buckley, C. “Term-weighting approaches in automatic text retrieval”, Information Processing and Management, 24(5), 1988, pp:513–523.

Salton G. & Buckley, C. “Improving Retrieval performance by relevance feedback”, Journal of American Society for Information Science, 41(4), 1990, pp:288-297.

Singhal, A. K. “Term Weighting Revisited”, Ph.D. thesis, Cornell University, 1997.