Available online at www.sciencedirect.com

ScienceDirect

Decision Support Systems 42 (2006) 1338–1349

Decision Support Systems

www.elsevier.com/locate/dss

Nonlinear ranking function representations in genetic programming-based ranking discovery for personalized search

Weiguo Fan [a,*], Praveen Pathak [b], Linda Wallace [c]

[a] Virginia Polytechnic Institute and State University, 3007 Pamplin Hall, Blacksburg, VA 24061, United States
[b] University of Florida, United States
[c] Virginia Polytechnic Institute and State University, United States

Received 14 October 2004; received in revised form 24 October 2005; accepted 3 November 2005
Available online 7 December 2005

Abstract

Ranking function is instrumental in affecting the performance of a search engine. Designing and optimizing a search engine's ranking function remains a daunting task for computer and information scientists. Recently, genetic programming (GP), a machine learning technique based on evolutionary theory, has shown promise in tackling this very difficult problem. Ranking functions discovered by GP have been found to be significantly better than many of the other existing ranking functions. However, current GP implementations for ranking function discovery are all designed utilizing the Vector Space model in which the same term weighting strategy is applied to all terms in a document. This may not be an ideal representation scheme at the individual query level considering the fact that many query terms should play different roles in the final ranking. In this paper, we propose a novel nonlinear ranking function representation scheme and compare this new design to the well-known Vector Space model. We theoretically show that the new representation scheme subsumes the traditional Vector Space model representation scheme as a special case and hence allows for additional flexibility in term weighting. We test the new representation scheme with the GP-based discovery framework in a personalized search (information routing) context using a TREC web corpus. The experimental results show that the new ranking function representation design outperforms the traditional Vector Space model for GP-based ranking function discovery.

Keywords: Information routing; Information retrieval; Genetic programming; Ranking function

1. Introduction

Nowadays, more and more people are using web search engines to find information online to help them make better informed decisions. Major search engines such as Google, Yahoo!, etc., receive millions of search requests per day according to searchenginewatch.com. Although many of the search engines have fast search response times and reasonably good search quality, most of them lack the personalized search capabilities which would allow different people to receive different search results for the same query, depending on their own personal preferences and profiles. This kind of personalized search is highly desirable in the business intelligence arena where people/organizations constantly look for individual-tailored information [12,32].

Prior research on personalized search has focused on the user profiling perspective in which historic feedback information (relevance feedback information or user clickthrough data) is used to infer a user's real informa-

* Corresponding author. Tel.: +1 540 231 6588; fax: +1 540 231 2511.
E-mail address: wfan@vt.edu (W. Fan).

0167-9236/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2005.11.002

tion need [12,32]. The inferred profile is then used for later information routing or personalized search when new unseen documents become available. In this paper, we approach the personalized search task from the user preference modeling perspective. In particular, we look at how to use common information heuristics or cues to model a user's ranking preference towards information. We will use an advanced machine learning technique called genetic programming (GP) to combine all of the available information heuristics in order to obtain best permissible personalized search results. The remainder of this section provides a brief introduction to the GP technique, explains its application to information retrieval, and outlines the primary goals of this paper.

Genetic algorithms (GAs) [23] and genetic programming [26] are search algorithms based on evolutionary theory. They represent the solution to a problem as an individual (also called a chromosome) in a population pool. They evolve the population of individuals (chromosomes), generation by generation, following the genetic transformation operations – such as reproduction, crossover, and mutation – with the aim of discovering chromosomes with better fitness values. A fitness function is used to assign the fitness value for each individual.

The difference between GAs and GP is the internal representation – or data structure – of the individual chromosome. In GAs, each individual is commonly (though not always) represented by a fixed-length bit string, like (1101110...) or a fixed-length sequence of real numbers (1.2, 2.4, 4, ...). In GP, more complex data structures (e.g., tree, linked list, or stack) are used [27]. Moreover, the length or size of the data structure is not fixed, although it may be constrained within a certain size limit by implementation. GAs are often used to solve difficult optimization problems, while GP is typically used to approximate complex, nonlinear functional relationships [26]. Because of the intrinsic parallel search mechanism and powerful global exploration capability in a high-dimensional space, both GAs and GP have been used to solve a wide range of difficult optimization problems that often have no best known solutions.

Because of these merits, there has been increasing interest in applying GAs and GP to intelligent information retrieval (IR) in recent years [3–12,17,18,24,28,29,31,41]. The application areas cover a wide range of IR topics such as document indexing; query induction, representation, and optimization; document clustering; and. document matching and ranking.

GP has previously been applied to discover ranking functions for both individual queries (personalized search or information routing tasks) [7,8,10,15,16] and multiple queries (consensus search or ad hoc retrieval tasks) [9,11–13,15]. Given a query (or a set of queries), a ranking function is used by a search engine to rank documents according to their match with the query. Since there is no known best ranking function for a query (or a set of queries), we model this problem as a GP search problem. Candidate ranking functions are represented as individuals in a GP population using a tree structure, and then evolved by GP to discover ranking functions with better fitness values.

So far, previous ranking function discovery efforts have centered on the Vector Space model (VSM), in which all documents and queries are represented as vectors and the same term weighting strategy used in a ranking function is applied to all terms in a document. For example, given a term weighting strategy *tf*, a ranking function will count all the term frequencies (*tf*) of all the terms matching a user query and use the sum of scores (also called retrieval status value—RSV) for final ranking. Ranking functions based on the Vector Space model have performed very well in various IR experimental evaluations and TREC (Text Retrieval and Evaluation Conference) competitions [19–22,35,37].

The advantage of the VSM is that it is simple, effective and easy to implement. This, however, does not necessarily mean that it is optimal for all application contexts. The VSM is very good for generic information retrieval (IR) engines such as web search engines or library catalog search systems, as these systems can effectively leverage the simplicity and powerfulness of the VSM to strive for speed and accuracy. However, the VSM may not be ideal for a personalized search (information routing) context where a user is consistently searching for the same information over a long period of time. In the new world of constant information, such a personalized search has become a critical component in the arsenal of the knowledge worker. For example, consider a stock analyst who has to constantly monitor the information about a company (for example Intel) or an industry and come up with recommendations with this new information. The information requirements of this analyst are fairly static – the analyst is interested in any news item that may affect the stock she is covering. In generic information retrieval (also known as ad hoc), a query given to the retrieval system to find documents about "intel" might come up with documents about CIA, FBI, Intelligence about Iraq, etc. But the stock analyst is not interested in this type of "intel". The stock analyst is interested in the documents that talk about Intel as a company. Thus, a generic retrieval system is not enough to satisfy the requirements of this analyst. What is needed is a targeted personalized search (or information routing) mechanism which will deliver appropriate documents to this analyst.