Evolutionary Information Retrieval

Pawan Lingras
Department of Mathematics and Computing Science
Saint Mary's University, Halifax, Nova Scotia
Canada B3H 3C3

Abstract

The evolutionary information retrieval differs from traditional information retrieval in that it evolves over a length of time with the user's changing information needs based on the user's preference for a wide variety of documents. The World Wide Web and Internet have increased the amount and variety of information available from a user's desktop. An evolutionary information retrieval system searches the web and randomly chooses documents to present to the user. Based on user's interest in the document, a genetic algorithm determines the fitness of various chromosomes and constructs next generation queries for further retrieval.

1. Introduction

Most traditional information retrieval methods are designed for requests initiated by the information consumer. The information retrieval system identifies documents that may be relevant to the user's information needs. The World Wide Web and other related Internet technologies have changed the traditional role of information retrieval. Newer tools, such as information agents, periodically retrieve information deemed relevant to the user's information needs. Often, an information agent categorizes the available information. The user then selects categories of interest, for which the information agent regularly collects all new information from its database. Some recent information agents attempt to adapt themselves to the particular information needs of an individual. However, the quantity of information on the Internet is growing at an exponential rate. Therefore, it is increasingly difficult for the information agents to bring all the relevant information to the attention of the user and limit the search appropriately.

This paper outlines a genetic algorithm based strategy that creates an evolving representation of a user's information needs. Each chromosome consists of various attributes (genes) such as geographic location, information category and a list of keywords. The genetic algorithm randomly generates chromosomes and retrieves titles of documents from a large collection. The titles are then made available to the user. If a title attracts the user's attention, it is considered relevant. The record of these choices is used to determine the fitness of chromosomes in a genetic pool. The genetic algorithm uses a crossover operator to develop queries that are most suited to a user. The mutation operator ensures that the evolutionary information retrieval system continues to explore the world wide document collection for other information of possible relevance. The retrieval performance or fitness of a chromosome depends upon the retrieval model. The proposed approach is demonstrated using the linear vector space model. However, the evolutionary information retrieval strategy outlined in this paper can also incorporate other popular retrieval models.

2. Vector Space Model

An Information Retrieval (IR) system identifies the relevant documents in response to a user's information request. Many retrieval models have been proposed for this purpose [2]. A list of keywords is one of the most primitive representations of information needs. More sophisticated techniques use the Boolean operators such as AND, OR, NOT. Some of the other information retrieval techniques allow the users to attach weights to the keywords based on their level of importance [2].

In a vector space model a document d_j is represented by a vector using a list of keywords $k_i, i = 1 \cdots n$, as $d_j = (w_{1j}, \cdots, w_{nj})$, where w_{ij} is the weight for keyword k_i in d_j. Similarly, a query q is represented as a vector $q = (w_{1q}, \cdots, w_{nq})$. The similarity between document d_j and query q is given as follows:

$$sim(d_j, q) = \frac{\sum_{i=1}^{n} w_{ij} \times w_{iq}}{\sum_{i=1}^{n} w_{ij} \times \sum_{i=1}^{n} w_{iq}}.$$

```
generate initial population G(0)
      evaluate G(0)
      for(t = 1; solution is not found; t++)
            generate G(t) from G(t – 1)
            evaluate G(t)
```

Fig. 1. Abstract view of a genetic algorithm

Web Search	Information Retrieval	Genetic Algorithms	q
0.6	0.4	0.9	

Web Search	Information Retrieval	Machine Learning	r
0.3	0.5	0.8	

Fig. 2. Initial Queries Represented as Chromosomes

Documents in the collection are ranked or classified as relevant or non-relevant using $sim(d_j, q)$.

Constructing queries based on keywords can be difficult for a novice user. Some of the research in information retrieval describes various ways of learning the weighted query from a user [3].

3 Genetic Algorithms

The origin of Genetic Algorithms (GAs) is attributed to Holland's [1] work on cellular automata. There has been significant interest in GAs over the last two decades. The range of applications of GAs includes such diverse areas as: job shop scheduling, training neural nets, image feature extraction, and image feature identification. This section gives a brief overview of GAs.

GAs follow the principles of evolution through natural selection. The domain knowledge is represented using a candidate solution called an *organism*. In most cases an organism is a single chromosome represented by a vector of length *n*: $c = (c_i \mid 1 \le i \le n)$, where c_i is called a gene.

A group of organisms is called a *population*. Successive populations are called *generations*. A GA starts from initial generation $G(0)$, and for each generation $G(t)$ generates a new generation $G(t + 1)$. An abstract view of a GA is given in Fig. 1.

The genetic algorithm is based on two fundamental evolutionary concepts:

1) a Darwinian notion of *fitness*, which describes an organism's ability to survive; and
2) *genetic* operators, which determine the next generation's genetic makeup based upon the current generation.

Conventionally, (2) is achieved through *crossover* and *mutation* operators. The crossover operator creates new organisms called *offspring*, by recombining the genetic material of two existing organisms, the *parents*. Parents are chosen to "pass on" their genes to the next generation based on their fitness. Such crossovers ensure the exploitation of successful subspaces of the solution space. The mutation operator randomly alters one or more genes in an organism. Mutations add genetic diversity to the population. Through mutation, GAs search previously unexplored sections of the solution space. Mutations consequently ensure that the entire search space is connected. Through the combination of crossovers and mutations, GAs are able to simultaneously explore new subspaces and exploit successful ones.

4. Genetic Encoding and Operations

Fig. 2 shows examples of query chromosomes, while Fig. 3 shows examples of document chromosomes. Each gene in the chromosome consists of a pair (k_i, w_{ic}), where w_{ic} is weight attached to the keyword k_i in chromosome *c*. Such a pair makes it possible to eliminate keywords with zero weights from the memory as well as computations. In addition to the contents

Web Search	Information Retrieval	Intelligent agents	Genetic Algorithms	d_1
0.2	0.9	0.1	0.1	

Web Search	Information Retrieval	Intelligent agents	Machine Learning	d_2
0.5	0.6	0.3	0.5	

Web Search	Information Retrieval	Genetic Algorithms	Machine Learning	d_3
0.1	0.1	0.3	0.9	

Web Search	Information Retrieval	Intelligent agents	Genetic Algorithms	d_4
0.2	0.4	0.1	0.6	

Fig. 3. Sample Document Collection

```
Fitness = 0.0;
For each document d in the sample collection
      If d is relevant then
            Fitness = Fitness + sim(c,d);
      Else
            Fitness = Fitness - sim(c,d);
Return Fitness;
```

Fig. 4. Fitness Function for Chromosome *c*

of the document, the keywords may consist of such information as categories of the document (e.g. sports or entertainment) and regional information (e.g. North America or Asia). The system will continuously update a list of keywords as it explores the document space.

A genetic algorithm may start the process by generating random queries. The current generation of query chromosomes will be used to retrieve documents. The documents will be retrieved based on the values of $sim(d_j, q)$. Let us assume that documents d_1, d_2, d_3 and d_4 were retrieved based on the generation consisting of queries *q* and *r* shown in Fig. 2.

Given a sample set, user may be able to identify relevant and non-relevant documents. Let us assume that the user deemed d_1 and d_2 as relevant. The documents d_3 and d_4 were considered non-relevant to the user. Based on this information the genetic algorithm will evaluate the fitness of current generation of query chromosomes and construct next generation of query chromosomes.

Fig. 4 shows an example of a fitness function that can be used to evaluate a query chromosome. The fitness is calculated as a difference between cumulative similarity of relevant and non-relevant documents. In our example, the fitness of *q* and *r* will be given as:

$$
\begin{aligned}
Fitness(q) &= sim(d_1,q) + sim(d_2,q) \\
&\quad - sim(d_3,q) - sim(d_4,q) \\
&= 0.23+0.15-0.14-0.33 \\
&= -0.09 \\
Fitness(r) &= sim(d_1,r) + sim(d_2,r) \\
&\quad - sim(d_3,r) - sim(d_4,r) \\
&= 0.25+0.28-0.36-0.13 \\
&= +0.04
\end{aligned}
$$

Therefore, query *r* will be considered more fit than query *q*. The queries with high fitness

AND Offspring

Web Search	Information Retrieval
0.3	0.4

OR Offspring

Web Search	Information Retrieval	Genetic Algorithms	Machine Learning
0.6	0.5	0.9	0.8

Fig. 5. Boolean Crossover Operator

Query chromosome before mutation:

Web Search	Information Retrieval	Genetic Algorithms
0.6	0.4	0.9

Query chromosome after mutation:

Web Search	Information Retrieval	Intelligent Agents
0.6	0.4	0.6

Fig. 6. Extended Mutation Operator

values from the current generation are used to create queries for the next generation through crossover and mutation operators. In addition to the conventional crossover operator, a Boolean crossover operator (Fig. 5) may be useful in information retrieval. The Boolean crossover produces *AND* and *OR* offspring. In the *AND* offspring, the weight of a given keyword k_i is given by the minimum of the weights of k_i in the parents. As mentioned before, the keywords with zero weights are dropped from the chromosomes. On the other hand, for the OR offspring, weight of k_i is given by the maximum of the weights of k_i in the parents. Fig. 6 shows an example of an extended mutation operation. The extended mutation operator may reduce weight of one of the keywords to zero and optionally add an additional keyword. For example, in Fig. 6, the weight for the keyword "*Genetic Algorithms*" was reduced to zero and a new keyword "*Intelligent Agents*", which was encountered in documents d_1, d_2 and d_4, was added with a random weight of 0.6.

With the use of a fitness function, conventional as well as Boolean crossovers and mutation operations, the information retrieval system may continue to evolve and adapt to the changing nature of a user's information needs.

5. Summary and Concluding Remarks

Conventional information retrieval responds to user's information requests. Internet technology has made a large amount of information accessible from a user's desktop. The size and distribution of information makes it impossible for a human user to explore for documents that may be relevant to his interests. Evolutionary information retrieval uses a genetic algorithm to continuously explore the document space. The crossover operator attempts to maximize the relevance of retrieved documents for the user. The mutation operator, on the other hand, ensures continued exploration of new document collections in an attempt to increase the variety of documents. This paper described a design of an evolutionary information retrieval system based on vector space model. The description included examples of a fitness function, extended mutation operation, and a new Boolean crossover operator. The strategy described in this paper is general enough to be used in conjunction with many other information retrieval models.

References

[1] Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.

[2] Salton, G. and McGill, M.H. (1983), *Introduction to Modern Information Retrieval*, McGraw-Hill, New York.

[3] Wong, S.K.M. and Yao, Y.Y. (1990), Query Formulation in Linear Retrieval Models, *Journal of the American Society for Information Science* 41(5), 334-341.