Genetic Algorithms in Information Retrieval

Vrajitoru Dana

Université de Neuchâtel
Institut Interfacultaire d'Informatique
Pierre-à-Mazel 7, 2000 Neuchâtel, Switzerland
e-mail: dana.vrajitoru@seco.unine.ch

Abstract
Information retrieval proposes solutions for searching, in a given set of objects, for those replying to a given description. Often, these objects are documents, however other forms like image, video, sounds can be considered. Besides statistical approaches, artificial intelligence models present an attractive paradigm to improve performance in IR systems, and the genetic algorithm (GA) represents one of them. How can the GA be used in IR and under which conditions? What are the reasons of its failures? What can be done to improve its performance ? In this paper we try to answer these questions.
Keywords: *information retrieval, genetic algorithm, feedback.*

Introduction

One of the developments of computer science in the last years tends to bring computer scientists closer to researchers in neuroscience or natural systems. In this vein, we can cite the GA which represents a special interest for us.
The GA is a probabilistic algorithm simulating the mechanism of natural selection of living organisms and is often used to solve problems having expensive solutions. Its traditional application is the optimization of a fitness function on a given set. We have chosen it for its strongness and because it can be applied without any special knowledge of the domain.

1. GA and information retrieval

This section presents the GA and our application in information retrieval.

The GA is generally used to solve optimization problems. If P is the set of all possible individuals, we search for an individual i_o presenting the best value of a fitness function $f : P \rightarrow R$. The GA starts with a limited number of individuals from P (initial population). The iterative search process is based on the competition of these individuals and their descendants during a number of generations.

The individuals are coded according to the chromosome model as a string of length l, where each position (locus) contains a gene: $i = (i_1, i_2, , ..., i_l)$. The gene values i_j (allele) are frequently binary. The simplest GA constructs a new generation from an old one following three steps: reproduction, crossover, and mutation.

In information retrieval, the individuals are document descriptions in the collection. For a given document D_j, where $j = 1..m$, and a set of terms T_k where k=1..n, a description of D_j takes the form: $dj = (t_{1j}, t_{2j}, ..., t_{nj})$ in which the value t_{ij} shows the presence of the term T_i in the document D_j. We have used a weighted indexing, where t_{ij} has a value between 0 and 10, according to the importance of the term. We have chosen these values between 0 and 10 by scaling of ntf·nidf indexing (Salton, Fox and Wu 1983).

An individual is build by putting together the description of all the documents in the collection: $i = (d_1, d_2, ..., d_m) = (t_{11}, ..., t_{n1}, t_{12}, ..., t_{n2}, ..., t_{1m}, ..., t_{nm})$. We have evaluated our models on the CACM (3204 documents) and CISI (1460 documents) collections.

The similarity between queries and documents is computed with the cosine (Salton, Fox and Wu 1983). The retrieved documents are sorted by their similarity value with the query. The fitness function is computed as the average precision at 11 fixed recall points (Vrajitoru 1995) in a transient way, which means that the GA only learns on the current query.

We have changed the GA in order to ensure its monotony. If the best fitness value from the new generation is lower that the best fitness value from the old

generation, the best individual from the old generation replaces the worse individual from the new generation.

2. Evaluation

We are interested in our research in using the relevance judgments of the past queries in order to improve the performance of the system on the current query. Our evaluation of the GA starts with an initial population containing an individual build by automatic indexing, and a variable number of individuals build from the relevance judgments of the other queries in the following manner:

```
individual = 1;
for each (Q : query, T : term Q, D : relevant document for Q)
  { add (T, D) to individual;
    individual = (individual mod population size) + 1;
```

In order to find the optimum size of the initial population, we have taken care that the number of generations build for each evaluation multiplied by the number of individuals in the initial population remains constant.

Table 1. Results of the GA by variation of the initial population size

Collection	**CACM**	**CISI**
Individuals/generations	**Precision (%change)**	
baseline	32.70	19.83
4/20	37.37 (+14.28)	21.71 (+9.48)
8/10	38.16 (+16.7)	**24.90 (+25.57)**
10/8	39.43 (+20.58)	24.14 (+21.73)
14/6	**40.62 (+24.22)**	24.77 (+24.91)
20/4	**41.61 (+27.25)**	**24.96 (+25.87)**

Our evaluations try to estimate the error rate in a reliable way. To do this, one must separate the information used for the evaluation from the information used in the training phase (Efron 1986). To follow this principle, the construction of the initial population ignores the relevance judgments of the current query.

The results of our evaluation, presented in Table 1, show a significant improvement from the baseline.

3. Improvement of the GA for information retrieval

An analysis of the evolution of the individual performance has shown that its values get more uniform as we advance through generations. As uniformity refrains performance improvement, we have searched for some of its causes. Our conclusion is that the classical crossover operator leads to the fact that the individuals resulting from a crossover operation are seldom more performant than their parents, phenomenon also occurring in the linear combinations.

To avoid this, we have constructed a new crossover operator, called α - β, using two cross points instead of one and treating differently the two input individuals (more details are given in Vrajitoru 96) (results in Table 2 presents the results of the new operator (8 individuals and 10 generations) compared to the classical one.

Table 2: Results of the new operator

Collection	**Baseline**	**Classical**	α - β
CACM		38.16 (+16.70)	43.95 (+34.41)
CISI	19.83	24.90 (+25.59)	25.74 (+29.82)

Conclusion

The conclusions of this paper are that the GA can be successfully applied to information retrieval, and that based on the field specificities, the GA can improve its performance.

References

Efron, B. (1986). How Biased Is the Apparent Error Rate of a Prediction Rule. *Journal of the American Statistical Association,* 81 (394), 461-470.

Salton, G., Fox, E., Wu, U. (1983). Extended Boolean Information Retrieval. *Communications of the ACM*, 26(12), 1022-1036.

Vrajitoru, D. (1995) : *Modification de l'indéxation par apprentissage dans le modèle vectoriel.* Cahier de recherche en informatique, CR-I-95-02.