On the Suitability of Genetic-Based Algorithms for Data Mining*

Sunil Choenni

Nat. Aerospace Lab., NLR, P.O. Box 90502, 1006 BM Amsterdam, The Netherlands, and
Univ. of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
email: choenni@nlr.nl

Abstract. Data mining has as goal to extract knowledge from large databases. A database may be considered as a search space consisting of an enormous number of elements, and a mining algorithm as a search strategy. In general, an exhaustive search of the space is infeasible. Therefore, efficient search strategies are of vital importance. Search strategies on genetic-based algorithms have been applied successfully in a wide range of applications. We focus on the suitability of genetic-based algorithms for data mining. We discuss the design and implementation of a genetic-based algorithm for data mining and illustrate its potentials.

1 Introduction

Research and development in data mining evolves in several directions, such as association rules, time series, and classification. The latter field has our attention. We have developed an algorithm to classify tuples in groups and to derive rules from these groups. In our view, a user formulates a mining question and the algorithm selects the group(s) that satisfy this question. For example, in an insurance environment, a question may be to identify persons with (more than average) chances of causing an accident. Then, the algorithm searches for the (group) profiles of these persons.

In general, the search spaces that should be inspected in order to find answers on mining questions are very large, making exhaustive search infeasible. So, heuristic search strategies are of vital importance to data mining. Genetic algorithms, which are heuristic search strategies, have been successfully used in a wide range of applications. A genetic algorithm is capable of exploring different parts of a search space [10].

In this paper, we discuss the applicability of a genetic-based algorithm to the search process in data mining. We show how a genetic algorithm can be suited for data mining problems. In our approach, a search space consists of expressions. An expression is a conjunction of predicates and each predicate is defined on a database attribute. Initially, a random number of expressions, called initial population, is selected. Then, the initial population is manipulated

* This research has been sponsored by the Dutch Ministry of Defense.
by applying a number of operations. The best individuals are selected to form the next generation and the manipulation process is repeated until no significant improvement of the population can be observed.

In general, data mining algorithms require a technique that partitions the domain values of an attribute in a limited set of ranges, simply because considering all possible ranges of domain values is infeasible. Suppose that we have an attribute \textit{age} which has a domain between 18 to 65, and an expression of the form \textit{age} \textbf{in} \([v_i, v_k]\), in which \(v_i\) and \(v_k\) are values from the domain of \textit{age}, defining a range of values. The problem is how to choose the values for \(v_i\) and \(v_k\). As illustrated in [11], this is in general an NP-complete problem. Our solution to this problem is based on a suitable choice of the mutation operator (see Section 3.3). Furthermore, we have chosen a representation for individuals that seamlessly fits in the field of databases. The same holds for the manipulation operators and the function to rank individuals (fitness function). The fitness function discussed in this paper is close to our intuition and gives rise to a speed up of the optimization process. Based on our approach, we have implemented a (prototype) tool for data mining, and have performed a preliminary evaluation. The results will be presented in this paper.

A genetic approach has been proposed in [2] to learn first order logic rules and in [7] a framework is proposed for data mining based on genetic programming. However, the authors neither come up with a implementation nor with experiments. The effort in [2] is focussed towards machine learning, and the important data mining issue of integration with databases is superficially discussed. The effort in [7] describes a framework for data mining based on genetic programming, and stresses on the integration of genetic programming and databases. However, an elaborated approach to implement and evaluate the framework is not presented. Other related research has been reported in [1, 8, 9]. While in [8, 9] variants of a hill climber are used to identify the group(s) of tuples satisfying a mining question, the approach in [1] is based on decision trees. However, the problem of partitioning attribute values has not been discussed in these efforts. We note that a genetic-based algorithm has, by nature, a better chance to escape from a local optimum than a hill climber.

The remainder of this paper is organized as follows. In Section 2, we outline some preliminaries and problem limitations. In Section 3, we identify the issues that play a role in genetic-based algorithms and adapt them in a data mining context. In Section 4, we point out a number of rules that may speed up the search process of a genetic-based algorithm. Section 5 is devoted to an overall algorithm for data mining. In Section 6, we discuss the implementation of the algorithm and some preliminary results. Finally, Section 7 contains conclusions and further work.

\section{Preliminaries & problem limitations}

In the following, a database consists of a universal relation [6]. The relation is defined over some independent single valued attributes, such as \(att_1, att_2, ..., att_n\),