Expert Systems with Applications 38 (2011) 2609–2620

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An incremental genetic algorithm for classification and sensitivity analysis of its parameters

Gözde Bakırlı, Derya Birant [*], Alp Kut

Department of Computer Engineering, Dokuz Eylul University, Izmir, Turkey

ARTICLE INFO

Keywords:
Genetic algorithm
Classification
Data mining
Incremental mining
Sensitivity analysis

ABSTRACT

Traditionally, data mining tasks such as classification and clustering are performed on data warehouses. Usually, updates are collected and applied to the data warehouse frequent time periods. For this reason, all patterns derived from the data warehouse have to be updated frequently as well. Due to the very large volumes of data, it is highly desirable to perform these updates incrementally. This study proposes a new incremental genetic algorithm for classification for efficiently handling new transactions. It presents the comparison results of traditional genetic algorithm and incremental genetic algorithm for classification. Experimental results show that our incremental genetic algorithm considerably decreases the time needed for training to construct a new classifier with the new dataset. This study also includes the sensitivity analysis of the incremental genetic algorithm parameters such as crossover probability, mutation probability, elitism and population size. In this analysis, many specific models were created using the same training dataset but with different parameter values, and then the performances of the models were compared.

1. Introduction

Data mining is the process of extracting hidden patterns from large datasets. Data mining has been widely used in many areas, such as marketing, banking and finance, medicine and manufacturing. Main data mining tasks and methods include classification, clustering, association rule mining, sequential pattern mining and outlier detection.

Classification is a procedure in which individual items are placed into groups based on quantitative information on some characteristics inherent in the items. In the classification process, a collection of labelled classes is provided and a training set is used to learn the descriptions of classes. Classification rules are discovered and then these rules are used to determine the most likely label of a new pattern. The most widely used classification techniques are neural networks, decision trees, *k*-nearest neighbours, support vector machines, and Naive Bayes.

Although, genetic algorithm (GA) is not one of the most widely used classifiers, several studies (Kim & Ryu, 2005; Nath, Rahman, & Salah, 2005; Zhu & Guan, 2004) show that this technique can also be successfully used for classification. These works include traditional genetic algorithm and don't support the incremental usage of the models when new data is added to the existing dataset. In addition, they don't evaluate the performance of the models which are constructed by different values of GA parameters.

A data warehouse is typically not updated immediately when insertions on the operational databases occur. New records are collected over a period of time and then data warehouse is updated at the end of this period. After that, genetic algorithm should be rerun again on the whole dataset in order to discovery classification rules again. Due to the very large size of the datasets, it is highly desirable to run this algorithm incrementally.

This study focuses on the problem of incrementally updating classification rules on changes of the database. It introduces a new incremental genetic algorithm for classification. The purpose of this study is to reduce the execution time of training process when new transactions are inserted. Based on the experimental results, it can be proven that the incremental GA yields significant speed-up factors over traditional GA even for large numbers of new transactions in a data warehouse. The performance evaluation of incremental GA on the "Nursery" dataset is presented, demonstrating the efficiency of the proposed algorithm.

This study also presents the sensitivity analysis of the GA parameters such as crossover probability, mutation probability, with/without elitism and population size. The aim of this analysis is to evaluate the performances of the classification models which are constructed using the same training dataset with different GA parameter values. Each classification model (classifier) consists of input parameters (crossover and mutation probabilities, population size etc.), applied techniques (parent selection type, crossover

[*] Corresponding author. Tel.: +90 232 4127418; fax: +90 232 4127402.
E-mail address: derya@cs.deu.edu.tr (D. Birant).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.08.051

type, different termination criteria etc.), and outputs (average fitness value, classification rules etc.) related to training process. The models are compared by applying *n*-fold cross validation method. In order to implement all these experiments, we developed the tool, named Generic Genetic Classifier Tool.

The rest of this paper is organized as follows: Section 2 includes related works on traditional genetic algorithm and classification using genetic algorithm. Section 3 briefly introduces the principles and basic concepts of incremental GA-based classification. Section 4 presents the new algorithm which can be used to incrementally update the classification rules on insertions of the database Section 5 reports an extensive performance evaluation and experimental results. Section 6 concludes with a summary and future work.

2. Related works

Genetic algorithms are a family of computational models motivated by the process of natural selection in biological system. Evolutionary computing concept is appeared in the 1960's by Rechenberg, 1971. GA was first developed by Holland (1975) and then improved by other many researchers (Booker, Goldberg, & Holland, 1989). Currently, GA is one of the most important techniques of artificial intelligence. GAs are used for soft constraint satisfaction, scheduling problems, finding game strategies, and so forth.

The basis of genetic algorithm is "natural selection". That means, individuals who have sufficient features to live, are transferred in next generation, and other individuals who are not good enough, disappear. The stronger candidates remain in the population, the weaker ones are discarded (Shapiro, 2001). So new generation get closer to the best solution at each step and this operation goes on until termination criteria are met. For the basic concept of genetic algorithms, please refer to Goldberg (1989).

In recent years, only in a few studies, GAs has been applied for classification problem to discover classification rules. Ishibuchi, Nakashima, and Murata (2001) constructed a fuzzy classifier system in which a population for fuzzy if-then rules is evolved from genetic algorithms. Avcı (2009) implemented classification method by combining genetic algorithm and support vector machine techniques. Fan, Chen, Ma, and Zhu (2007) created an approach for proposal grouping, in which knowledge rules are designed to interact with proposal classification, and the genetic algorithm is developed to search for the expected groupings. Yuen, Wong, Qian, Chan, and Fung (2009) proposed a hybrid model which combines genetic algorithm and neural network for classifying garment defects. Kwong, Chang, and Tsim (2008) used genetic algorithm to discover knowledge about the fluid dispensing. Dehuri, Patnaik, Ghosh, and Mall (2007) used an elitist multi-objective genetic algorithm for mining classification rules from large databases. Yılmaz, Yıldırım, and Yazıcı (2007) used genetic algorithm to make classification segments of video to objects.

According to the review of GA-based classification methods, previous studies use either traditional genetic algorithm or combination of genetic algorithm with another AI technique such as fuzzy, neural network. They don't propose the incremental usage of the genetic algorithm for classification when new data is added to the existing dataset.

The problem of incrementally updating mined patterns on changes of the database, however, has been proposed for other data mining tasks such as clustering, association rule mining. Lin, Hong, and Lu (2009) propose an efficient method for incrementally modifying a set of association rules when new transactions have been inserted to the database. Lühr and Lazarescu (2009) introduce an incremental graph-based clustering algorithm to both incrementally cluster new data and to selectively retain important cluster information within a knowledge repository. Fan, Tseng, Chern, and Huang (2009) propose an incremental technique to solve the issue of added-in data without re-implementing the original rough set based rule induction algorithm for a dynamic database.

Sensitivity analysis is the study to determine how a given model output depends upon the input parameters (Saltelli, 2008). In other words, it is the process of varying input parameters over a reasonable range and observing the relative change in model response. It is an important process for checking the quality of a given model, as well as a powerful tool for checking the robustness and reliability of the model. A sensitivity analysis can be conducted by changing each parameter value by +/−10% and +/−50% (Cacuci, 2003). This study compares the performance of the classification models constructed by different GA parameter settings.

3. Basic concepts for incremental GA-Based classification

Each phase in GA (Fig. 1) produces a new generation of potential solutions for a given problem. In the first stage, an initial *population*, which is a set of encoded bit-strings (*chromosomes*), is created to initiate the search process. The performance of the strings is then evaluated with respect to the *fitness function* which represents the constraints of the problem. After the sorting operation, the individuals with better performance (fitness value) are selected for a subsequent genetic manipulation process. The selection policy is responsible for assuring survival of the best-fit individuals. In the next stages, a new population is generated using two genetic operations: *crossover operation* (recombination of the bits/genes of each two selected strings/chromosomes) and *mutation operation* (alteration of the bits/genes at one or more randomly selected positions of the strings/chromosomes). This process is repeated until certain criteria are met.

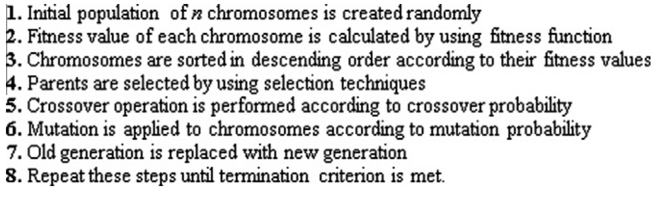
3.1. Initial population

In GA, *search space* is a set which includes all possible solutions of the problem. *Population* is a subset of *n* randomly chosen solutions from the search space. In GA-based classification, a different population is created randomly for each class. For example; if there are five classes in the dataset then population is created five times and training process is also repeated for each class over related population of this class.

Population consists of *chromosomes* (individuals) which are encoded as strings to represent candidate solutions to the problem. Various encoding methods such as binary encoding, real number encoding, integer or literal permutation encoding, general data structure encoding have been proposed for particular problems to provide effective implementation of GAs (Gen & Cheng, 2000). For classification problem, binary encoding should be used to construct chromosomes.

3.2. Fitness function

After the initial population is created, a fitness function is used to evaluate chromosomes (individuals) and to determine whether a chromosome is a good answer to the problem or not. In other words, it is used to determine which chromosomes are the "best"



1. Initial population of *n* chromosomes is created randomly
2. Fitness value of each chromosome is calculated by using fitness function
3. Chromosomes are sorted in descending order according to their fitness values
4. Parents are selected by using selection techniques
5. Crossover operation is performed according to crossover probability
6. Mutation is applied to chromosomes according to mutation probability
7. Old generation is replaced with new generation
8. Repeat these steps until termination criterion is met.

Fig. 1. Basic genetic algorithm.