Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1997 Proceedings | Americas Conference on Information Systems (AMCIS)

8-15-1997

Using Genetic Programming as a Learning Tool in Discovering Financial Trading Rules

Kai Lung HUI
Hong Kong University of Science and Technology

Kar Yan TAM
Hong Kong University of Science and Technology

Using Genetic Programming as a Learning Tool in Discovering Financial Trading Rules

Kai Lung HUI
Kar Yan TAM*
Department of Information and Systems Management
School of Business and Management
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong

Abstract

The growth of database systems has called for more advanced information retrieval and knowledge discovery tools. Genetic programming is proposed as one such tool and its characteristics and strengths are discussed in this paper. The application of genetic programming in learning security trading rules is also discussed.

Introduction

In the past decades the development of relational database systems have been outrageous and more and more organizations retain their own computerized database systems. The growth in data volume has called for advanced information retrieval or knowledge discovery algorithms and tools. These algorithms or tools learn through the content and relationships between the data and attempt to extract useful trends or exceptions which are of interest to the users. The rules or models thus extracted may help in supporting a wide array of decision activities. Neural networks and decision trees are typical examples of knowledge discovery tools that are commonly used.

Since knowledge discovery in databases (or data mining, as it is usually called) involves searching the best fitting rules or knowledge through analyzing the data, genetic programming can be applied in this regime. The merit of genetic programming lies on its implicit parallelism and extensive search capability throughout the entire population space. In other words, it would less likely be trapped in local optima, which is a major problem of many searching algorithms.

The concept and implementation of genetic programming will be discussed in the next section. This is followed by its application in discovering financial trading rules. Concluding remark will be given as to the kind of analysis that can be performed in evaluating the efficiency of genetic programming in doing this kind of data mining tasks.

Genetic algorithms and genetic programming

Developed by Holland (1975), genetic algorithms mimic the biological evolution of organisms by manipulating the inherent "genes" of solutions. The idea is that natural selection will favor the better and superior organisms and the poor alternatives should be disposed. This way of propagation biases towards the better alternatives, and therefore sooner or later the population will contain only the best organisms, which is the desirable result that we opt for. The heuristic proceeds as follows: A population of solutions is maintained. The relative fitness of each solution is evaluated. With the provision that better solutions have higher chances to be selected, solutions are selected at random for combination or modification to produce new generations of solutions. The child are then compared with the parents and better solutions are maintained while the worse are discarded. The process repeats for typically several hundreds or thousands of generations until some terminating conditions are met.

Notice that under such specification of evolution, genetic algorithm can proceed in learning the inherent features of the data without an explicit domain knowledge. What is needed is simply the way to assess the fitness of the solutions, which can be easily identified in most situations. In other words, it is suitable for depicting unobserved trends or exceptions in data where no or limited domain knowledge is available.

In genetic algorithms, the genetic structures of organisms are represented by lists of fixed length character or binary strings called *chromosomes*. While this representation is efficient, it is difficult to map different problems into such schema. Koza (1992) extended the framework by relaxing the fixed length character string restriction. This results in genetic programming which allows flexible presentation of solutions as hierarchies of different functions in tree-like structures.

The key operations in genetic algorithms or programming are *crossovers* and *mutations*. Crossover involves cutting a piece of genetic structure from each of the two parent organisms at particular positions and then join them together to produce an offspring. This child organism maintains parts of the salient characteristics of the parent organisms while at the same time improves through inheriting different characteristics of different organisms. On the other hand, mutation involves changing a particular attribute of the organism so that different genetic structures may emerge. These two operations together ensure the exhaustive search in the global solution population.

Applications of genetic programming in discovering trading rules

Economists have for long time denied the possibility of existence of financial trading rules (rules that are generated based on interpretation of data and that can explain and predict the trend of market fluctuations). However, given a database on financial markets, it would be beneficial to decision makers and investors if systematic trends or patterns can be discovered simply out of the data. This is an area where data mining techniques could be applied, and as such the current study applies genetic programming in trying to explore possible trading rules out of the daily data from the security market.

Well established trading rules like moving averages, channel rules or filter rules will be used in this study. Their specifications and characteristics will be closely examined. Rather than using them exclusively as complete decision rules, they act as building blocks in the current study and may combine with each other, with different function parameters, to form complex decision rules. Essentially, they (and the function parameters) are the "genes" and may be propagated and modified throughout the different generations of solutions.

To allow the genetic programming to learn through the data, a sample of actual data on the security market will be used as a training set to "train" the algorithm. The fitness of the solutions are assessed by the returns that they can generate if they are applied as the guiding investment strategies during the training period. The best decision rules will then be used for later analysis.

Anticipated analysis

The best decision rules discovered by the genetic programming will be applied to another series of security trading data and the returns that can be generated would be calculated. The performance of the decision rules will then be compared with other models like random walk or the simple buy-and-hold strategy. It is expected that the returns generated by the rules will be higher than other classical models.

The decision rules would be examined in detail and see if any systematic pattern exists. This helps in understanding the essential ingredients that lead to good (by means of generating high return) investment strategy in security markets.

In addition to assessing the fitness of the solutions, analysis can also be performed on the implicit parallelism of the algorithm. The implementation of genetic algorithms or programming on parallel machines have started to receive attentions from researchers (for a few examples, see Pettey and Leuze, 1989 and Tanese, 1989). A parallel version of the genetic programming that works on MIMD parallel computers can be developed. We would then be able to compare the performance of the parallel architecture with the serial one and see if there is any difference in terms of elapsed time in learning or solution quality. This certainly helps in advancing knowledge of applying genetic algorithms to knowledge discovery tasks.

References

Holland, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press: Ann Arbor, 1975.

Koza, J. R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, 1992.

Pettey, C. C. and Leuze, M. R., *A Theoretical Investigation of a Parallel Genetic Algorithm*, Proceedings of the Third International Conference on Genetic Algorithms (1989), p. 398-405.

Tanese, R., *Distributed Genetic Algorithms*, Proceedings of the Third International Conference on Genetic Algorithms (1989), p. 434-439.