

Application of Genetic Algorithms in Data Mining

Pankaj Nagar, Sumit Srivastava

Department of Statistics, University of Rajasthan, Jaipur.

Abstract: Genetic Algorithms are used to create a version of biological evolution on computers. They operate on small computer program that, just like organisms undergoing natural evolution, are subject to mutation, sexual reproduction, and selection of the most fit. It provides several important mathematical analyses as to why simulating evolution on a computer might be a high optimal way of solving the hard problem. The paper try to work on the Genetic algorithm in broad sense of simulated evolutionary system because it is best-known term – although we recognize that the pure usage of genetic algorithm is generally more focused on optimization and search.

I. INTRODUCTION

Genetic algorithms loosely refers to the simulated evolutionary systems, but more precisely they are the algorithms that dictate how, population of organism should be formed, evaluated, and modified. For instance, there are genetic algorithms that determine how to select organisms will be deleted from the population. They can be defining how the genetic material of the simulated chromosomes is converted into a computer program that can solve some real-world problem.

The problem that may be solved by genetic algorithms vary from the optimizing a variety of data mining techniques such as neural networks and nearest neighbor to the optimization of negotiating strategies for oil rights. A simple example of the application of genetic algorithms first proposed by Alex Singer, would be a two gene chromosomes that encoded the solution to a simple direct marketing problem: “what is optimal numbers of the coupons that should be put into a coupons mailers in order to optimize profit?” at first this might seem to be pretty simulated problem to solve – simply mail out as many coupons both receiving and actually using a coupons. The problem is made a little bit more complicated, however, because several other factors affect whether a coupons packet mailer make a profit.

- The more coupons there are, the more the mailer weights and the higher the mailing costs (thus decreasing profit).
- Any coupon that does not appear in the mailer is not used by the consumer resulting in the lost of the revenue.
- If there are too many coupons in the mailer, the consumer will be overloaded and not choose to use any of the coupons

This problem can be encoded into a simple genetic algorithms where each simulated organism has a single gene

that represents that “organisms’ “ best guess as the correct number of coupons. These computers programs are as simple as just one reflects how many coupons to put into the mailer. The genetic algorithms can proceed with this optimization be creating a population of these single- gene organisms at random and through simulated evolutions, modifying the genes, deleting the worst performers and making copies with slight modifications of the best performers. Over time the optimal number of coupons is determined. Figure [1] shows a population of these simple coupons organisms, indicating which ones would be deleted in him next generation because their solution was too far from optimal.

Genetic algorithms can be used for optimal Couponsthe mailers they proposed. The other two simulated organisms reproduced similar copies of themselves into the next generation. In this case the problem was so simple that random guessing of numbers and hence evaluating the guesses would have sufficed. When the problem become much more complicated, random guessing is not sufficient genetic algorithms may be the best way to get to the right solution.

II HOW DO THEY RELATE TO EVOLUTION

In many ways genetic algorithms stay true to the processes available in biological evolution and to the computer evolution – or at least they try to. Some of the analogs in genetic algorithms that appear in natural evolution include

- Organism – which represents the computers program being optimized
- Population – the collection of organisms undergoing simulated evolution.
- Chromosome – in biology the chromosome or chromosomes contain the genetic makeup of the organisms and fully define how the organism will develop from its genotype (genetic definition) with environmental influences to its phenotype (outward appearance and behavior). In genetic algorithms the chromosomes encodes the computer programs
- Fitness – The calculation with which an organisms’ value can be determined for selection and survival of the fittest.
- Gene – the basic building block of the chromosome which defines one particular feature of the simulated organism.
- Locus – the position on the chromosomes that contains a particular gene (e.g. the location that determine eye color).

- Allele – the value of the gene(e.g. blue for the locus of the eye color)
- Mutation: -a random change of the value of a gene
- Mating – the process by which two simulated organisms swap pieces of computer program in a simulated crossover.
- Selection – the process by which the simulated organisms that are best at solving the particular problem are retained and the less successful are weeded out by deleting them from computer memory.

Many other important aspects of natural evolution are not mentioned in this list, some of which can be very important. These represent some of the important research areas for genetic algorithms and include topics such as simulation of old age and death, parasites, diploidy (having two copies of each chromosomes), overcrowding for finite resources, and geographic constraints on mating patterns. Some of the recent research into these topics has resulted in surprisingly good results.

III GENETIC ALGORITHMS, ARTIFICIAL LIFE, AND SIMULATED EVOLUTION

Here we describe some of the fields which simulated there working with the genetic algorithms and how they differ:

- Artificial life – A field of computer science that simulate evolution and natural processes on computer generally for the purpose of creating complex life-like Behavior. Examples include the automated growth of realistic-looking plants and the realistic flocking, schooling and herding behaviors of large simulated organisms
- Simulated evolution – A field of computer science and biology that simulates evolution and biological systems on the computer for the purpose of better understanding how evolution works. An example would be the simulation of an extremely simple single-gene organism to understand the effects of recessive genes
- Simulated systems, emergent systems, and complex systems – Systems that are constructed simple performed organisms that don't evolve but do interact with each other for the purpose of understanding large-scale effects over time (i.e. the evolution of the system is of more interest than any evolution of an organism) – for example, a system that models telephone customers and how they react to rate changes and competitors' offers over time in order to understand how to optimize calling rates.
- Cellular automata – systems that creates quite complex macro behavior through the interaction of very simple predefined rule. Cellular

automata have been used fro everything from creating

- Optimization systems – a systems used for the express purpose of optimizing the solution to some well-defined problem. Genetic algorithms are typically used for these kinds of systems as well as hill-climbing and simulated annealing algorithms.

IV BUSINESS USAGE OF THE GENETIC ALGORITHM

Although they would have to be classified generally as an emerging science, genetic algorithm has a wide variety of uses in business. There are three main areas to which they can be applied:

- Optimization – Give a business problem with certain variables and a well defined definition of profit, a genetic algorithm can be used to automatically determine the optimal value for the variables that optimize the profit.
- Prediction – Genetic algorithms have been used as meta level operators that are used to help optimize other data mining algorithms. For instance, they have been used to optimize the weights in a neural network or to find the optimal association rules in market-analysis
- Simulation – Sometimes a specific business problem is not well defined in terms what the profit is or whether one solution is better than the other. The business person instead just has large number of entities (usually customer or competitors) that they would like to simulate via simple interaction rules overtime.

V SCORE CARD FOR THE BUSINESS APPLICATION:

Although several companies now provide products that utilize genetic algorithms, these products are still in their infancy for use in the business community. In some successful implementations they have been used with a great deal of consulting resources or on well-defined problems that map easily into easily where they have been used before. Perhaps their greatest strength is their clarity and simple use of generating a proposed solution and then testing it via survival of the fittest (which can be viewed as a form of statistical hypothesis testing and cross-validation).

Table 1 show the business score card for genetic algorithm. Some of the disadvantage that these systems offer to the business community is that they scale extremely well to parallel software and hardware systems and in general are fun to use. They can be slow when wielded by a native user and can get stuck in suboptimal solutions even when wielded by experienced users.

Table[1] Business score card for genetic algorithms

Data mining Measures	Description
Automation	Genetic algorithms are relatively automated once the fittest function and how the problem will be encoded into simulated genetic material on computer are defined.
Clarity	When genetic algorithms are used to optimize existing data mining techniques, they cannot afford any further clarity than is inherent in the underlying technique
ROI	One of the great advantages of genetic algorithms is that because the problem solution being optimized can be fairly general, many different factors relating to overall ROI of entire business process can be taken into account at the same time.

1. The values from various proposed solutions can be well defined
2. The problem is complex and cannot be solved directly
3. The problem is relatively new and not well understood, and no one has yet been able to determine other optimization techniques to be used for its solution
4. The problem involves a large numbers of variables working together to produce a large scale effect.

If these four key attributes occur in the problem that you are trying to solve, then genetic algorithms may be good technique to try. If any of the attributes are not present, then some other techniques probably exists and is preferable. For instance, if the proposed solution cannot be defined as good or bad, it will not be possible for the system to evolve to a better solution. If the problem is simple, then it is likely that there will be better and faster direct method; and if the problem – even if complex – has been actively researched, then there are also probably better solutions available. The last attribute is not necessarily a requirement, but it does seem to be common theme among the successful applications of the genetic algorithms. For instances, it is often the case that you can model the behavior of an individual customer but not be able to understand the overall behavior(and profit) of a dynamic systems that includes all your customers.

VI THE AREA OF GENETIC ALGORITHM USAGE:

At heart, Genetic algorithms are optimizing technique. They are systems that take difficult-to-solve and very complex problems and come up with a pretty good solution without a lot of detailed understanding about the problem except how to evaluate a good solution. They can be applied to an incredibility diverse set of problems and will come up with solutions superior to random guessing and often better solution than could be achieved via hill climbing techniques. Beyond that, though, it is difficult to characterize where the application of genetic algorithm is appropriate as much depends on how the problem is encoded and how effective the fitness function is at evaluating whether a solution is good or bad.

At the other end of the scale, a genetic algorithm can almost never replace or outperform a well-thought-out algorithm designed specifically to solve one particular problem. For example, back propagation will work better and faster to train neural networks, and simple rules-of-thumb optimization techniques can solve the traveling-salesman problem much better and more quickly than can genetic algorithm. Their use then comes often as a replacement for the time involved in detailed analysis of given problem – which is business may mean quite a few problems that are quite complex but so new that there is not precedent for a previous well-thought-out solution. In cases like these, genetic algorithms provide optimized solutions where no optimization was performed before.

They are probably four key attributes for a business problem that could benefit from the application of genetic algorithm

SUMMARIZATION

Genetic algorithms will continue to borrow more and more from biology and natural evolution as it is better understood. In this paper, we have discussed the tremendous benefits of using sharing to prevent premature convergence and in doing so have learned something about natural and simulated evolution. Computer code itself allowing tremendous latitude in the genetic organism to find powerful and creative solutions to difficult problems in large development. In this case we can work upon the LISP computer language can be used in the genetic material, and modified mutation and crossover have led to programs able to do everything from a park a simulated track to create desirable network.

REFERENCES

1. Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm, *Proceeding ICGA 2*, pp. 14-21, Lawrence Erlbaum Associates, Publishers, 1987.
2. Bala J., De Jong K., Huang J., Vafaie H., and Wechsler H. Using learning to facilitate the evolution of features for recognizing visual concepts. *Evolutionary Computation 4(3) - Special Issue on Evolution, Learning, and Instinct: 100 years of the Baldwin Effect*. 1997.

3. Bandyopadhyay, S., and Muthy, C.A. "Pattern Classification Using Genetic Algorithms", *Pattern Recognition Letters*, (1995). Vol. 16, pp. 801-808.
4. De Jong K.A., Spears W.M. and Gordon D.F. (1993). Using genetic algorithms for concept learning. *Machine Learning* 13, 161-188, 1993.
5. Duda, R.O., Hart, P.E., and Stork, D.G. *Pattern Classification*. 2nd Edition, John Wiley & Sons, Inc., New York NY. (2001).
6. Falkenauer E. *Genetic Algorithms and Grouping Problems*. John Wiley & Sons, (1998).
7. Freitas, A.A. A survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery, See: www.pgia.pucpr.br/~alex/papers. A chapter of: A. Ghosh and S. Tsutsui. (Eds.) "Advances in Evolutionary Computation". Springer-Verlag, (2002).
8. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*, MA, Addison-Wesley. (1989).
9. Guerra-Salcedo C. and Whitley D. "Feature Selection mechanisms for ensemble creation: a genetic search perspective". In: Freitas AA (Ed.) *Data Mining with Evolutionary Algorithms: Research Directions – Papers from the AAAI Workshop*, 13-17. Technical Report WS-99-06. AAAI Press, (1999).
10. Jain, A. K.; Zongker, D. "Feature Selection: Evaluation, Application, and Small Sample Performance", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 2, February (1997).
11. Kuncheva, L.I., and Jain, L.C., "Designing Classifier Fusion Systems by Genetic Algorithms", *IEEE Transaction on Evolutionary Computation*, Vol. 33 (2000), pp 351-373.
12. Martin-Bautista MJ and Vila MA. A survey of genetic feature selection in mining issues. *Proceeding Congress on Evolutionary Computation (CEC-99)*, 1314-1321. Washington D.C., July (1999).
13. Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs*. 3rd Ed. Springer-Verlag, (1996).
14. Muhlenbein and Schlierkamp-Voosen D., Predictive Models for the Breeder Genetic Algorithm: I. Continuous Parameter Optimization, *Evolutionary Computation*, (1993) Vol. 1, No. 1, pp. 25-49.
15. Park Y and Song M. A genetic algorithm for clustering problems. *Genetic Programming 1998: Proceeding of 3rd Annual Conference*, 568-575. Morgan Kaufmann, (1998).
16. Pei, M., Goodman, E.D., and Punch, W.F. "Pattern Discovery from Data Using Genetic Algorithms", *Proceeding of 1st Pacific-Asia Conference Knowledge Discovery & Data Mining (PAKDD-97)*. Feb. (1997).
17. Pei, M., Punch, W.F., and Goodman, E.D. "Feature Extraction Using Genetic Algorithms", *Proceeding of International Symposium on Intelligent Data Engineering and Learning '98 (IDEAL '98)*, Hong Kong, Oct. (1998).
18. Punch, W.F., Pei, M., Chia-Shun, L., Goodman, E.D., Hovland, P., and Enbody R. "Further research on Feature Selection and Classification Using Genetic Algorithms", In *5th International Conference on Genetic Algorithm*, Champaign IL, pp 557-564, (1993).
19. Siedlecki, W., Sklansky J., A note on genetic algorithms for large-scale feature selection, *Pattern Recognition Letters*, Vol. 10, Page 335-347, (1989).
20. Skalak D. B. (1994). Using a Genetic Algorithm to Learn Prototypes for Case Retrieval and Classification. *Proceeding of the AAAI-93 Case-Based Reasoning Workshop*, pp. 64-69. Washington, D.C., American Association for Artificial Intelligence, Menlo Park, CA, 1994.
21. Vafaie H and De Jong K. "Robust feature Selection algorithms". *Proceeding 1993 IEEE Int. Conf on Tools with AI*, 356-363. Boston, Mass., USA. Nov. (1993).