Population Based Search Methods in Mining Association Rules

K. Indira[1], S. Kanmani[2], P. Prashanth[2], V. Harish Sivasankar[2], Konda Ramcharan Teja[2], and R. Jeeva Rajasekar[2]

[1] Dept. of CSE
[2] Dept. of IT
Pondicherry Engineering College, Puducherry, India

Abstract. Genetic Algorithm (GA) and Particle swarm optimization (PSO) are both population based search methods and move from set of points (population) to another set of points in a single iteration with likely improvement using set of control operators. GA has become popular because of its many versions, ease of implementation, ability to solve difficult problems and so on. PSO is relatively recent heuristic search mechanism inspired by bird flocking or fish schooling. Association Rule (AR) mining is one of the most studied tasks in data mining. The objective of this paper is to compare the effectiveness and computational capability of GA and PSO in mining association rules. Though both are heuristic based search methods, the control parameters involved in GA and PSO differ. The Genetic algorithm parameters are based on reproduction techniques evolved from biology and the control parameters of PSO are based on particle 'best' values in each generation. From the experimental study PSO is found to be as effective as GA with marginally better computational efficiency over GA.

Keywords: Genetic Algorithm, Particle Swam optimization, Association rules, Effectiveness, Computational efficiency.

1 Introduction

With advancements in information technology the amount of data stored in databases and kinds of databases continue to grow fast. Analyzing and finding the critical hidden information from this data has become very important issue. Association rule mining techniques help in achieving this task. Association rule mining is searching of interesting patterns or information from database [12]. Association rule mining finds interesting associations and/or correlation relationships among large set of data items. Typically the relationship will be in the form of a rule [13], $X \rightarrow Y$ Where X and Y are itemsets and X is called the antecedent and Y the consequent.

Genetic algorithm and particle swarm optimization are both evolutionary heuristics and population based search methods proven to be successful in solving difficult problems. Genetic Algorithm (GA) is a procedure used to find approximate solutions to search problems through the application of the principles of evolutionary biology. Particle swarm optimization (PSO) is a heuristic search method whose mechanics are inspired by the swarming or collaborative behavior of biological populations. The

V.V. Das and J. Stephen (Eds.): CNC 2012, LNICST 108, pp. 255–261, 2012.

major objective of this paper is to verify whether the hypothesis that PSO has same effectiveness as that of GA but better computational efficiency is valid or not. This paper is organized as follows. Section 2 discusses the related works carried out so far on GA and PSO in association rule mining. Section 3 describes the methodology adopted for mining ARs. In section 4 the experimental results are presented followed by conclusions in section 5.

2 Related Works

During last few decades many researches were carried out using evolutionary algorithm in data mining concepts. Association rule mining shares major part of research in data mining. Many classical approaches for mining association rules have been developed and analyzed. GA discovers high level prediction rules [1] with better attribute interaction than other classical mining rules available. The mechanism to select individuals for a new generation based on the technique of elitist recombination [2] simplifies the implementation of GA.

In [3], cross probability and mutation probability are set up in dynamic process of evolution. When new population evolves, if every individual is comparatively consistent, then cross probability Pc and mutation probability P_m are increased. Noda et al. [4] has proposed two relatively simple objective measures of the rule Surprisingness (or interestingness). By contrast, genetic algorithms (GAs) [5] maintain a population and thus can search for many non-dominated solutions in parallel. GA's ability to find a diverse set of solutions in a single run and its exemption from demand for objective preference information renders it immediate advantage over other classical techniques.

Particle Swarm Optimization is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995 [6], inspired by social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as GA. However unlike GA, PSO has no evolution operators such as crossover and mutation. A binary version of PSO based algorithm for fuzzy classification rule generation, also called fuzzy PSO, is presented in [7]. PSO has proved to be competitive with GA in several tasks, mainly in optimization areas. The PSO variants implemented were Discrete Particle Swarm Optimizer [8] (DPSO), Linear Decreasing Weight Particle Swarm Optimizer [9] (LDWPSO) and Constricted Particle Swarm Optimizer [10] (CPSO).

The fixing up of the best position [16) for particles after velocity updation by using Euclidean distance helps in generating the best particles. The chaotic operator based on Zaslavskii maps when used in velocity update equation [17] proved to enhance the efficiency of the method. The soft adaptive particle swarm optimization algorithm [18] exploits the self adaptation in improving the ability of PSO to overcome optimization problems with high dimensionality. The particle swarm optimization with self adaptive learning [19] aims in providing the user a tool for various optimization problems. The problem of getting struck at local optimum and hence premature convergence is overcome by adopting self adaptive PSO [20] where the diversity of population is maintained. This copes up with the deception of multiple local optima and reduces computational complexity.

3 Methodology

Genetic algorithm and particle swarm optimization both population based search methods are applied for mining association rule from databases. The self adaptive GA [15] is found to perform marginally better than traditional GA. This section describes the methodology adopted for mining AR based on both SAGA and PSO.

Fitness value decides the importance of each itemset being evaluated. Fitness value is evaluated using the fitness function. Equation 3 describes the fitness function.

$$Fitness(x) = con(x) \times \log(\sup(x) \times length\ (x) + 1) \tag{1}$$

Where sup(x) and conf(x) are as described in equation 2 and 3, *length*(x) is length of the association rule type x.

$$\sup(x) = \frac{No.of\ transactions\ containing\ X}{Total\ No.of transactions} \tag{2}$$

$$conf\ (X \rightarrow Y) = \frac{\sup(X \cup Y)}{\sup(X)} \tag{3}$$

The effectiveness of the rules mined is measured in terms of predictive accuracy.

$$Predictive\ accuracy = \frac{|X\ \&\ Y|}{|X|} \tag{4}$$

where |X&Y| is the number of records that satisfy both the antecedent X and consequent Y, |X| is the number of rules satisfying the antecedent X.

3.1 Mining AR Based on SAGA

Genetic Algorithm (GA) is an adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. The evolutionary process of a GA [11] is a highly simplified and stylized simulation of the biological version. The algorithm is as given below.

Step 1. [Start] Generate random population of *n* chromosomes
Step 2. [Fitness] Evaluate the fitness *f(x)* of each chromosome *x* in the population
Step 3. [New population] Create a new population by repeating the following steps
- [Selection] Select two parent chromosomes from a population according to their fitness
- [Crossover] With a crossover probability cross over the parents to form a new offspring (children)
- [Mutation] With a mutation probability mutate new offspring at each locus
- [Accepting] Place new offspring in a new population

Step 4. [Replace] Use new generated population for a further run of algorithm
Step 5. [Test] If the end condition is satisfied, stop, and return the best solution in current population
Step 6. [Loop] Go to step 2

The mutation rate is made self adaptive in SAGA as follows:

$$p_m^{(n+1)} = \lambda\, p_m^0 \sqrt{\frac{\sum_{i=1}^{m}(f_{max}^{(n+1)} - f_i^m)^2}{\sum_{i=1}^{m}(f_{max}^{(n)} - f_i^n)^2}} \tag{5}$$

p_m^n is the nth generation mutation rate, $p_m^{(n+1)}$ is the $(n+1)^{th}$ generation mutation rate. The first generation mutation rate is p_m^0, $f_i^{(m)}$ is the fitness of the nth individual itemset i. $f_{max}^{(n+1)}$ is the highest fitness of the $(n+1)^{th}$ individual stocks. $f_i^{(n)}$ is the fitness of the nth individual i. m is the number of itemsets. λ is the adjustment factor. The fitness criterion is as described in equation 5.

3.2 Mining AR Based on PSO

PSO is initialized with a group of random particles (solutions) and then searches for optimum value by updating particles in successive generations. In each iteration, all the particles are updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. This value is called ***pbest***. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called ***gbest***. The outline of basic particle swarm optimizer is as follows

Step1. Initialize the population : locations and velocities
Step 2. Evaluate the fitness of the individual particle (pbest)
Step 3. Keep track of the individuals highest fitness (gbest)
Step 4. Modify velocities based on pBest and gBest position
Step 5. Update the particles position
Step 6. Terminate if the condition is met
Step 7. Go to Step 2

The chromosome encoding approach adopted in this scheme is binary encoding. Particles which have larger fitness are selected for the initial population. The particles in this population are called initial particles. Initially the velocity and position of all particles randomly set within predefined range. In each iteration, the velocities of all particles are updated based on velocity updating equation

$$V[t+1] = V[t] + c1 \times rand(\ ) \times (pbest[t] - present[t]) + c2 \times rand(\ ) \times (gbest\ [t] - present[t]) \tag{6}$$

$$present[t+1] = present[t] + v[t] \tag{7}$$

v[] is the particle velocity, present[] is the current particle. pbest[] and gbest[] are local best and global best position of particles. rand () is a random number between (0,1). c1, c2 are learning factors. Usually c1 = c2 = 2. 3. The position of particles is then updated based on equation 4. During position updation if the acceleration exceeds the user defined V_{max} then position is set to V_{max}. The above process is repeated until fixed number of generations or the termination condition is met.

4 Experimental Results

To confirm the effectiveness of GA and PSO, both the algorithms were coded in Java. Lenses, Haberman and Car evaluation datasets from UCI Irvine repository [14] were taken up or the experiment. Self adaptive GA and PSO based mining of ARs on the above dataset when performed resulted in predictive accuracy as potted in figure 1. The predictive accuracy when achieved maximum during successive iterations was recorded. PSO is found to be equally effective as SAGA in mining association rules. The predictive accuracy for both the methods is close to one another.

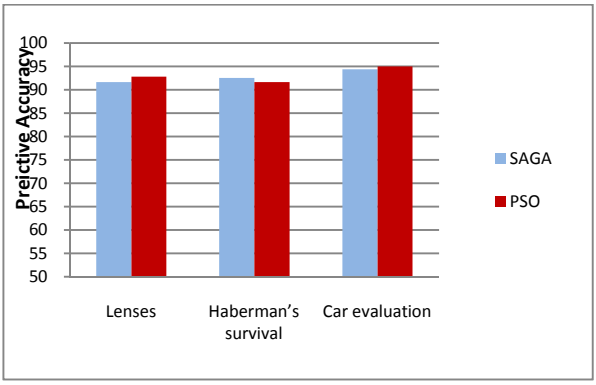

Fig. 1. Predictive Accuracy Comparison

In terms of computational effectiveness PSO is found to be marginally fast when compared to SAGA. This can be seen from the figures 2.

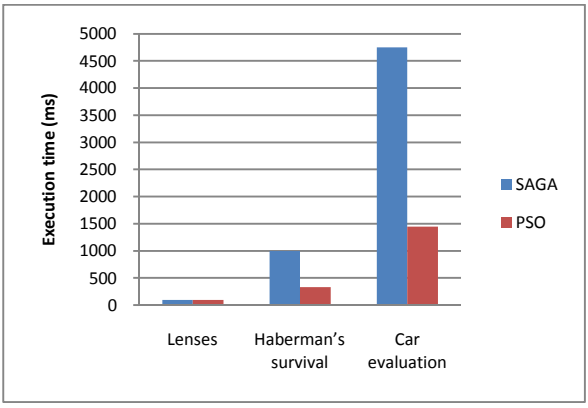

Fig. 2. Predictive Accuracy Comparison

Particle Swarm optimization shares many similarities with Genetic Algorithms. Both methods begin with a group of randomly initialized population, evaluate their population based on fitness function. Genetic operators namely crossover and mutation preserves the aspects of the rules and in avoiding premature convergence. The

main difference between PSO and GA is that PSO does not have the genetic operators as crossover and mutation. In PSO only the best particle passes information to others and hence the computational capability of PSO is marginally better than SAGA.

5 Conclusions

Particle swarm optimization is a recent heuristic search method based on the idea of collaborative behavior and swarming in populations. Both PSO and GA depend on sharing information between populations. In GA information is passed from one generation to other through the reproduction method namely crossover and mutation operator. GA is well established method with many versions and many applications. The objective of this study is to analyze PSO and GA in terms of effectiveness and computational efficiency.

From the study carried out on the three datasets PSO proves to be as effective as GA in mining association rules. In term of computational efficiency PSO is marginally faster than GA. The pbest and gbest values tends to pass the information between populations more effectively than the reproduction operators in GA. PSO and GA are both inspired by nature and more effective for optimization problems. Setting of appropriate values for the control parameters involved in these heuristics methods is the key point to success in these methods.

References

1. Alex, A. F.: A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery, Postgraduate Program in Computer Science. Pontificia Universidade catolica do Parana Rua Imaculada Conceicao, Brazil
2. Shi, X.-J., Lei, H.: Genetic Algorithm-Based Approach for Classification Rule Discovery. In: International Conference on Information Management, Innovation Management and Industrial Engineering, ICIII 2008, vol. 1, pp. 175–178 (2008)
3. Zhu, X., Yu, Y., Guo, X.: Genetic Algorithm Based on Evolution Strategy and the Application in Data Mining. In: First International Workshop on Education Technology and Computer Science, ETCS 2009, vol. 1, pp. 848–852 (2009)
4. Noda, E., Freitas, A.A., Lopes, H.S.: Discovering Interesting Prediction Rules with Genetic Algorithm. In: Proceedings of Conference on Evolutionary Computation (CEC 1999), Washington, DC, USA, pp. 1322–1329 (1999)
5. Michalewicz, Z.: Genetic Algorithms + Data Structure = Evolution Programs. Springer, Berlin (1994)
6. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks, pp. 1492–1948. IEEE Press (1995)
7. He, Z., et al.: Extracting Rules from Fuzzy NeuralNetwork by Particle Swarm Optimization. In: IEEE Conference on Evolutionary Computation, USA, pp. 74–77 (1995)
8. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann (2001)
9. Shi, Y., Eberhart, R.C.: Empirical Study of Particle Swarm Optimization. In: Proceedings of the 1999 Congress of Evolutionary Computation, Piscatay (1999)
10. Clerc, M., Kennedy, J.: The particle Swarm-explosion, Stability and Convergence in a Multidimensional Complex Space. IEEE Transactions on Evolutionary Computation 6, 58–73 (2002)

11. Dehuri, S., Mall, R.: Predictive and Comprehensible Rule Discovery Using a Multiobjective Genetic Algorithm: Knowledge Based Systems, vol. 19, pp. 413–421. Elsevier (2006)
12. Wang, M., Zou, Q., Liu, C.: Multi-dimension Association Rule Mining Based on Adaptive Genetic Algorithm. In: IEEE International Conference on Uncertainty Reasoning and Knowledge Engineering, pp. 150–153 (2011)
13. Dehuri, S., Patnaik, S., Ghosh, A., Mall, R.: Application of Elitist Multi-objective Genetic Algorithm for Classification Rule Generation: Applied Soft Computing, pp. 477–487 (2008)
14. Merz, C.J., Murphy, P.M.: UCI Repository of Machine Learning Databases. University of California Irvine. Department of Information and Computer Science (1996), http://kdd.ics.uci.edu
15. Indira, K., Kanmani, S., Gaurav Sethia, D., Kumaran, S., Prabhakar, J.: Rule Acquisition in Data Mining Using a Self Adaptive Genetic Algorithm. In: Nagamalai, D., Renault, E., Dhanuskodi, M. (eds.) CCSEIT 2011. CCIS, vol. 204, pp. 171–178. Springer, Heidelberg (2011)
16. Kuo, R.J., Chao, C.M., Chiu, Y.T.: Application of Particle Swarm Optimization in Association Rulemining. Applied Soft Computing, 323–336 (2011)
17. Atlas, B., Akin, E.: Multi-objective Rule Mining Using a Chaotic Particle Swarm Optimization Algorithms. Knowledge Based Systems 23, 455–460 (2009)
18. Mohammed, Y., Ali, B.: Soft Adaptive Particle Swarm Algorithm for Large Scale Optimization. In: Fifth International Conference on Bio Inspired Computing, pp. 1658–1662. IEEE Press (2010)
19. Wang, Y., Li, B., Weise, T., Wang, J., Yun, B., Tian, Q.: Self-adaptive Learning Based on Particle Swarm Optimization. Information Science 181, 4515–4538 (2011)
20. Lu, F., Ge, Y., Gao, L.: Self Adaptive Particle Swarm Optimization Algorithm for Global Optimization. In: Sixth International Conference on Natural Computation, pp. 2692–2696. IEEE Press (2010)