International Journal of Database Theory and Application
Vol. 3, No. 2, June, 2010

A Particle Swarm Optimization and Immune Theory-Based Algorithm for Structure Learning of Bayesian Networks

Xiao-Lin Li
School of Business, Nanjing University,Nanjing 210093, China
lixl@nju.edu.cn

Abstract

Bayesian network is a directed acyclic graph. Existing Bayesian network learning approaches based on search & scoring usually work with a heuristic search for finding the highest scoring structure. This paper describes a new data mining algorithm to learn Bayesian networks structures based on an immune binary particle swarm optimization (IB-PSO) method and the Minimum Description Length (MDL) principle. IBPSO is proposed by combining the immune theory in biology with particle swarm optimization (PSO). It constructs an immune operator accomplished by two steps, vaccination and immune selection. The purpose of adding immune operator is to prevent and overcome premature convergence. Experiments show that IBPSO not only improves the quality of the solutions, but also reduces the time cost.

Keywords: *Bayesian network; structure learning; minimum description length; immune theory; particle swarm optimization*

1. Introduction

During the last two decades there has been a steadily expanding interest in the use of rigorous probabilistic networks as a technique for the research in pattern recognition. A probabilistic network is a graphical model that describes dependencies or independencies between variables or attributes of interest. Such a model records qualitative influences between variables in addition to the numerical parameters of the probability distribution. As such it provides an ideal form for combining prior knowledge, which might be limited solely to experience of the influences between some of the variables of interest, and data.

Bayesian networks are graphical representations of dependency relationships between variables. They are intuitive representations of knowledge and are akin to human reasoning paradigms. They are powerful tools to deal with uncertainties, and have been extensively used to uncertainty knowledge representation, inference and reasoning. In the past decades, they have been successfully applied in medical diagnose, software intelligence, finance risk analysis, DNA functional analysis, web mining and so on, and have become a rapidly growing field of research and have seen a great deal of activity.

A Bayesian network consists of two components: structure and parameters. They can respectively be used in qualitative and quantitative causal analysis. Bayesian network learning includes two parts, one for structure learning and one for parameters learning. It is a trivial process to learn parameters given structure and complete data, so the key attention for learning Bayesian networks has been focused on structure learning. A lot of researches have been given in this line. To this end, researchers have developed many algorithms to induct a Bayesian network from a given database [1][2][3][4][5][6]. However, it has been noted [7] that the search landscape is large and multimodal, and deterministic search algorithms find local optima. An obvious choice to combat the problem is to use a stochastic search method.

Particle swarm optimization (PSO)[9], rooting from simulation of swarm of bird, is a new branch of Evolution Algorithms based on swarm intelligence. The concept of PSO, which can be described with only several lines of codes, is more easily understood and realized than some other optimization algorithms. PSO has been successfully applied in many engineering projects [10][11][12][13][14].

This paper proposes a new data mining algorithm to learn Bayesian networks structures based on an improved swarm intelligence method and the Minimum Description Length (MDL) principle. An important characteristic of the algorithm is that, in order to prevent and overcome premature convergence, some concepts in immune systems are introduced into binary particle swarm optimization. Furthermore, the algorithm, like some previous work, does not need to impose restriction of having a complete variable ordering as input.

This paper will begin with a brief introduction to Bayesian network and MDL principle. In section 3, the immune binary particle swarm optimization algorithm will be proposed. Then, in section 4 and 5, the performance of our algorithm will be demonstrated by conducting a series of experiments as well as a summary of the whole paper be made.

2. Bayesian networks and MDL metric

2.1. Bayesian networks

A Bayesian network is a direct acyclic graph (DAG). It can be used for both inferential exploration of previously undetermined relationships among attributes as well as descriptions of these relationships upon discovery. The structural part of a Bayesian graphical model is a directed acyclic graph consisting of arcs and nodes, which correspond to domain variables $X_1, \cdots, X_n$. And the arcs between nodes represent direct dependencies between the variables. Likewise, the absence of an arc between two nodes X_1 and X_2 represents that X_2 is independent of X_1 given its parents. The joint probability distribution (JPD) is then expressed in the following formula:

$$P(x_1, \cdots\cdots, x_n) = \prod_{k=1\cdots n} P(x_k \mid \pi(x_k)) \tag{1}$$

where $\pi(x_k)$ is the configuration of X_k's parent node set $\Pi(X_k)$.

Bayesian network structure learning is a challenging problem. The main difficulty here is how to find a good dependency structure among the many possible ones, which can be infinitely many. As in most learning algorithms, there are three important issues that need to be addressed here: hypothesis space, scoring function, and search algorithm. For Bayesian networks, we know that the task of finding the highest scoring network is NP-hard [8]. Thus, we must resort to techniques such as heuristic search in order to find the highest scoring structure. The success of this approach depends on the choice of the potential parents. Clearly, a wrong initial choice will result in poor structures.

2.2. The MDL metric

One of the key components of learning a dependency structure from data automatically is the ability to evaluate different structures in order to choose one that fits the data well. We can choose structures by a scoring function, such as the minimum description length (MDL) principle [15]. The MDL metric is derived from information theory and incorporates the MDL principle. With the composition of the description length for network structure and the description length for data, the MDL metric tries to balance between model accuracy and complexity. Using the metric, a better network would have a smaller score. Similar to other metrics, the MDL score for a Bayesian network, S , is decomposable and could be written as in equation 2. The MDL score of the network is simply the summation of the MDL score of $\Pi(X_k)$ of every node X_k in the network.

$$MDL(S) = \sum_{k} MDL(X_k, \Pi(X_k)) \tag{2}$$

According to the resolvability of the MDL metric, equation 2 can be written when we learn Bayesian networks form complete data as follows:

$$MDL(S) = N\sum_{k=1}^{N} \sum_{X_k, \Pi(X_k)} P(X_k, \Pi(X_k)) \log P(X_k, \Pi(X_k)) \tag{3}$$
$$-\sum_{k=1}^{N} \frac{\log N}{2} \| \Pi(X_k) \| (\| X_k \| - 1)$$

Where N is database size, $\| X_k \|$ is the number of different values of X_k , and $\| \Pi(X_k) \|$ is the number of different parent value combinations of $\Pi(X_k)$.

3. Particle Swarm Optimization

PSO, originally developed by Kennedy and Elberhart [9], is a method for optimizing hard numerical functions on metaphor of social behavior of flocks of birds and schools of fish. It is an evolutionary computation technique based on swarm intelligence. A swarm consists of individuals, called particles, which change their positions over time. Each particle represents a potential solution to the problem. In a PSO system, particles fly around in a multi-dimensional search space. During its flight each particle adjusts its position according to its own experience and the experience of its neighbors, making use of the best position encountered by itself and its neighbors. The effect is that particles move towards the better solution areas, while still having the ability to search a wide area around the better solution areas. The performance of each particle is measured according to a predefined fitness function, which is related to the problem being solved and indicates how good a candidate solution is. The PSO has been found to be robust and fast in solving non-linear, non-differentiable, multi-modal problems. The mathematical abstract and executive steps of PSO are as follows.

Let the i th particle in a D -dimensional space be represented as $X_i = (x_{i1}, \ldots, x_{id}, \ldots, x_{iD})$. The best previous position (which possesses the best fitness value) of the i th particle is recorded and represented as $P_i = (p_{i1}, \ldots, p_{id}, \ldots, p_{iD})$, which is also called *pbest*. The index of the best *pbest* among all the particles is represented by the symbol g. The location P_g is also called *gbest*. The velocity for the i th particle is represented as $V_i = (v_{i1}, \ldots, v_{id}, \ldots, v_{iD})$. The concept of the particle swarm optimization consists of, at each time step, changing the velocity and location of each particle towards its *pbest* and *gbest* locations according to Equations (4) and (5), respectively:

$$V_i(k+1) = \omega V_i(k) + c_1 r_1 (P_i - X_i(k)) / \Delta t + c_2 r_2 (P_g - X_i(k)) / \Delta t \tag{4}$$

$$X_i(k+1) = X_i(k) + V_i(k+1)\Delta t \tag{5}$$

where ω is the inertia coefficient which is a constant in interval [0, 1] and can be adjusted in the direction of linear decrease [16]; c_1 and c_2 are learning rates which are nonnegative constants; r_1 and r_2 are generated randomly in the interval [0, 1]; Δt is the time interval, and commonly be set as unit; $v_{id} \in [-v_{\max}, v_{\max}]$, and $v_{\max}$ is a designated maximum velocity. The termination criterion for iterations is determined according to whether the maximum generation or a designated value of the fitness is reached.

However, many optimization problems are set in space featuring discrete, qualitative distinctions between variables and between levels of variables. As any problem, discrete or continuous, can be expressed in a binary notation, it is seen that an optimizer which operates on two-valued functions might be advantageous. Kennedy and Eberhart also developed the discrete binary version of the PSO. Then the particle changes its value by [17].

$$V_i(k+1) = \omega V_i(k) + c_1 r_1 (P_i - X_i(k)) / \Delta t + c_2 r_2 (P_g - X_i(k)) / \Delta t \tag{6}$$

$$\text{if } \rho_i(k+1) < sig\left(v_i(k+1)\right) \text{ then } x_i(k+1) = 1 \text{ ; else } x_i(k+1) = 0 \tag{7}$$

4. Immune Binary Particle Swarm Optimization Method

PSO described above can be considered as the conventional particle swarm optimization, in which as time goes on, some particles become inactive quickly because they are similar to the *gbest* and lost their velocities. In the following generations, they will have less contribution for their very low global and local search capability and this problem will induce the emergence of the prematurity.

In this paper, an immune binary particle swarm optimization (IBPSO) is proposed. The aim of combining immune concepts and methods with PSO is theoretically to utilize the locally characteristic information for seeking the ways and means of finding the optimal solution when dealing with difficult problems. In immune systems, for immature B cells the receptor editing is stimulated by B cell receptor and provides an important means of maintaining self-tolerance. The process of the receptor editing may diversify antibodies not only to jump local affinity optima, but also across the entire affinity landscape. In order to improve the performance of binary particle swarm optimization, we introduce the immune operator, which is accomplished by two steps, a vaccination and an immune selection. They are explained as follows.

Given an individual, a vaccination means modifying the genes on some bits in accordance with priori knowledge so as to gain higher fitness with greater probability [18]. A vaccine is abstracted from the prior knowledge of the pending problem, whose information amount and validity play an important role in the performance of the algorithm. But in most cases of dealing with some problems, it is difficult to abstract the characteristic information of them because we know little about the priori knowledge. On the other hand, the work of searching the local scheme used for the global solution makes the workload increase greatly and the efficiency decrease, so that the value of this work is lost. Similarly to the techniques of the vaccine inoculation, we take full advantage of some characteristic information and knowledge in the process of solving problem to distill vaccines, and the vaccinate particles. After each generation, the proposed algorithm abstracts information from genes of the present optimal particle to make vaccines during the evolutionary process. Therefore, we can obtain vaccines easily and withhold the evolutionary information simultaneously.

The immune selection accomplish by the following two steps. The first one is the immune test. If the fitness is smaller than that of the parent, which means serious degeneration must have happened in the evolution process, the parent will instead of the current individual and participate in the next generation to compete. The second one is the probability selection strategy based on antibody concentration. Define

$$D(x_i) = \frac{1}{\sum_{j=1}^{N} \left| f_i - f_j \right|}, i = 1, 2, \ldots, N \tag{8}$$

Where f_i is the fitness value of the i th particle, N is the number of the particles in the population, and $D(x_i)$ is the i th antibody concentration [19]. Define

$$P(x_i) = \frac{\sum_{j=1}^{N} \left| f_i - f_j \right|}{\sum_{i=1}^{N} \sum_{j=1}^{N} \left| f_i - f_j \right|}, i = 1, 2, \ldots, N \tag{9}$$

$P(x_i)$ is the probability selection formula based on antibody concentration.

From formula (8) and (9), we can find the selection probability of i th antibody becomes less and less if more and more antibodies are similar with i th antibody. Otherwise, the selection probability of i th antibody becomes more if less antibodies are similar with i th antibody. So the probability selection strategy can comeback the population variety and overcome premature convergence.

The IBPSO method is summarized in table 1.

Table 1: The IBPSO method

1. Set to 0.
2. Create an initial population, *Pop(t)*. The initial population size is *N*.
3. If the current population contains the optimal individual, then the evolutionary stop; otherwise, continues.
4. Abstract vaccines according to the prior knowledge.
5. Create *N* offspring according to formula (6) and (7), then randomly create *M* new particles, and all new particles are stored in the intermediate population *Pop'(t)*.
6. Perform vaccination on *Pop'(t)* and obtain *Pop''(t)*.
7. Perform immune selection on *Pop''(t)* and obtain the next generation *Pop(t+1)*.
8. Go to 3.

Then the Bayesian networks structures can be learned by using IBPSO. The algorithm we propose is shown in Table 2.

5. Experiments

We conduct a number of experiments to evaluate the performance of the immune binary particle swarm optimization algorithm. The learning algorithms take the data set only as input. The data set is derived from ALARM network (http://www.norsys.com/netlib/alarm.htm).

Firstly, we generate 5,000 cases from this structure and learn a Bayesian network from the data set ten times. Then we select the best network structure as the final structure. We also compare our algorithm with binary particle swarm optimization and classical GA algorithm [6]. The MDL metric of the original network structures for the ALARM data sets of 5,000 cases is 81,219.74.

The population size *N* is 30 and the maximum number of generations is 5,000. We employ our learning algorithm to solve the ALARM problem. Some parameters in the experiment are taken as: c_1=1.9, c_2=0.8, ω=0.5, ρ=0.5, $v_{\max}$=8. We also implemented a classical GA to learning the ALARM network. The one-point crossover and mutation operations of classical GA are used. The crossover probability p_c is 0.9 and the mutation probability p_m is 0.01. The MDL metric for immune binary particle swarm optimization algorithm, binary particle swarm optimization algorithm and the classical GA are delineated in Figure 1.

From Figure 1, we see that the value of the average of the MDL metric for IBPSO is 81223.3, the value of the average of the MDL metric for binary particle swarm optimization algorithm is 81268.3 and the value of the average of the MDL metric for the GA is 8,1789.4. We find immune binary particle swarm optimization algorithm evolves good Bayesian

Table 2: The Algorithm for learning Bayesian networks by IBPSO

1. Set to 0.
2. Create an initial population, *Pop(t)*, of *N* random DAGs. The initial population size is *N*.
3. Each DAG in the population *Pop(t)* is evaluated using the MDL metric.
4. While t is smaller than the maximum number of generations G
   a) Each DAG in *Pop(t)* produces one offspring by formula (6) and (7). If the offspring has cycles, delete the set of edges that violate the DAG condition. If choices of set of edges exist, we randomly pick one choice.
   b) Create *M* new particles randomly, and all new individual are stored in the intermediate population *Pop'(t)*. The size of Pop'(t) is *N*+*M*.
   c) Perform vaccination on *Pop'(t)* and obtain *Pop''(t)*.
   d) Perform immune test on *Pop''(t)*. If the fitness of the offspring individual in *Pop''(t)* is not as good as the parent in *Pop(t)*, the parent will instead of the offspring to compete.
   e) Perform immune selection operations according to formula (8) and (9). The individual will be selected if $P(x_i) \geq threshold$.
   f) Select *N* DAGs from *Pop''(t)* and store them in the new population *Pop(t+1)*.
   g) Increase t by 1
5. Return the DAG with lowest MDL metric found in any generation of a run as the result of the algorithm.

network structures at an average generation of 3991.6. Binary particle swarm optimization algorithm and GA obtain the solutions at average generation of 4083.7 and 4495.4, respectively. From Figure 1, we can also find that the proposed algorithm performs more poorly than Binary particle swarm optimization algorithm does at the early generations. But the performance of the proposed algorithm is better at the end of the generations. The reason of the phenomenon is that the proposed algorithm randomly creates *M* new particles in order to prevent and overcome the prematurity and use the best particle of the generation as the vaccination. Thus, we can conclude that immune binary particle swarm optimization algorithm finds better network structures at earlier generations than binary particle swarm optimization algorithm and the GA does.

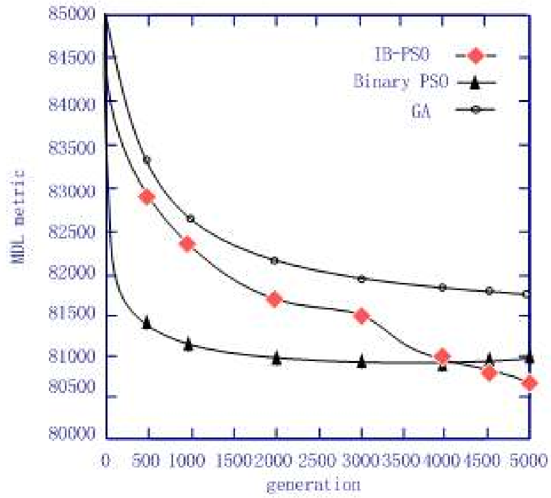

Fig.1. The MDL metric for the ALARM network

5. Conclusions

Bayesian network is a directed acyclic graph. Existing Bayesian network learning approaches based on search & scoring usually work with a heuristic search for finding the highest scoring structure. This paper proposes an immune binary particle swarm optimization method by combining the immune system with PSO. In the IBPSO, the immune operator is designed for preventing and overcoming premature convergence. IBPSO constructs an immune operator accomplished by two steps, vaccination and immune selection. Then, the proposed method is used as a search algorithm to learn Bayesian networks from data. Experiments show that the proposed method is effective for learning Bayesian networks from data. Furthermore, studying the influence of the parameters for the evolutionary process on the performance of IBPSO is left for future work.

Acknowledgments

This work was supported by National Science Foundation of China (60803055), the China MOE Research Project of Humanities and Social Science (08JC630041), the China Postdoctoral Science Foundation (20080441031) and the Jiangsu Planned Projects for Post-doctoral Research Funds (0801038C).

References

[1] J Suzuki. A construction of Bayesian networks from databases based on a MDL scheme, *Proc of the 9th Conference of Uncertainty in Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann, (1993) 266~273,
[2] Y Xiang, S K M Wong. Learning conditional independence relations from a probabilistic model, Department of Computer Science, University of Regina, CA, (1994) Tech Rep: CS-94-03,.

[3] D Heckerman. Learning Bayesian network: The combination of knowledge and statistic data, *Machine Learning*, Vol. 20, No. 2, (1995), 197 ~ 243,.
[4] Cheng J, Greiner R, Kelly J. Learning Bayesian networks from data: An efficient algorithm based on information theory, *Artificial Intelligence*. Vol.137, No.1-2, (2002) 43-90.
[5] Lam, W. and Bacchus, F., Learning Bayesian belief networks: An algorithm based on the MDL principle, *Computational Intelligence*, Vol 10, No.4, (1994).
[6] P. Larrañaga, M. Poza, Y. Yurramendi, R. Murga, and C. Kuijpers, Structure Learning of Bayesian Network by Genetic Algorithms: A Performance Analysis of Control Parameters, *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 9, Sept. (1996) 912-926.
[7] Friedman, N. (1998a). The Bayesian Structural EM Algorithm. Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, Morgan Kaufmann Publishers.
[8] D.M. Chickering.: Learning Bayesian networks is NP-complete. In: D. Fisher, H. J.Lenz, (eds.), Learning from Data: Artificial Intelligence and Statistics V. Springer, Berlin (1996) 121-130.
[9] J. Kennedy, R.C. Eberhart, Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks. December. (1995) 1942–1948.
[10] J. Kennedy, R.C. Eberhart, Y. Shi, Swarm Intelligence, Morgan Kaufmann, San Mateo, CA, 2001.
[11] Y. Shi, R. Eberhart, Empirical Study of Particle Swarm Optimization, Proceedings of the Congress on Evolutionary Computation (CEC99). July, 1999, pp. 1945–1950.
[12] P.N. Sugantha, Particle Swarm Optimization with Neighborhood Operator, Proceedings of the Congress on Evolutionary Computation (CEC99). July, 1999, pp. 1958–1962.
[13] E. Ozcan, C.K. Mohan, Particle Swarm Optimization: Surfing the Waves, Proceedings of the Congress on Evolutionary Computation (CEC99) (July), 1999, pp. 1939–1944.
[14] M. Clerc, The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization, Proceedings of the Congress on Evolutionary Computation (CEC99) (July), 1999, pp. 1951–1957.
[15] W. Lam and F. Bacchus. Learning Bayesian belief networks: an algorithm based on the MDL principle, *Computational Intelligence*, Vol.10, No.4, pp: 269–293, 1994.
[16] Shi Y. and Eberhart R.C., A modified particle swarm optimizer. *Proceedings of IEEE International Conference of Evolutionary Computation*, Anchorage, Alaska, May,69-73, 1998.
[17] Kennedy J. and Eberhart R., A discrete binary version of the particle swarm optimization algorithm. *Proceedings of the Conference on Systems, Man, and Cybernetics.* 4104-4109, 1997.
[18] L. C. Jiao, L. Wang. A novel genetic algorithm based on Immunity. *IEEE Trans. on Systems, Man, and Cybernetics-Part A Systems and Humans.* 2000, 30(5): 552-561.
[19] Gang Lu and De-jian Tan. Improvement on regulating definition of antibody density of immune algorithm. *Proceeding of the 9th international conference on neural information processing.* 5: 2669-2672. 2002.

Authors

Xiao-Lin Li received her PhD in Computer Software and Theory from Jilin University, China, in 2005. From December 2005 to December 2007, she was a postdoctor in National Laboratory for Novel Software Technology, Nanjing University, China. Since December 2007, she had been an assistant professor in School of Business, Nanjing Univeristy, China. Her research interests include Business Intelligence, Machine Learning, Data Mining and Decision making.