Robust Inference of Bayesian Networks Using Speciated Evolution and Ensemble

Kyung-Joong Kim, Ji-Oh Yoo, and Sung-Bae Cho

Dept. of Computer Science, Yonsei University,
134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, Korea
{uribyul, taiji391}@sclab.yonsei.ac.kr
sbcho@cs.yonsei.ac.kr

Abstract. Recently, there are many researchers to design Bayesian network structures using evolutionary algorithms but most of them use the only one fittest solution in the last generation. Because it is difficult to integrate the important factors into a single evaluation function, the best solution is often biased and less adaptive. In this paper, we present a method of generating diverse Bayesian network structures through fitness sharing and combining them by Bayesian method for adaptive inference. In the experiments with Asia network, the proposed method provides with better robustness for handling uncertainty owing to the complicated redundancy with speciated evolution.

1 Introduction

One commonly used approach to deal with uncertainty is a Bayesian network (BN) which represents joint probability distributions of domain. It has already been recognized that the BN is quite easy to incorporate expert knowledge. BN and associated schemes constitute a probabilistic framework for reasoning under uncertainty that in recent years has gained popularity in the community of artificial intelligence. From an informal perspective, BN is directed acyclic graph (DAG), where the nodes are random variables and the arcs specify the dependency between these variables. It is difficult to search for the BN that best reflects the dependency in a database of cases because of the large number of possible DAG structures, given even a small number of nodes to connect. Recently, there are many researchers to design BN structures using evolutionary algorithms but most of them use the only one fittest solution in the last generation [1].

The main problem with standard evolutionary algorithms appears that they eventually converge to an optimum and thereby loose their diversity necessary for efficiently exploring the search space and its ability to adapt to a change in the environment when a change occurs. In this paper, we present a method of generating diverse evolutionary Bayesian networks through fitness sharing and combining them by Bayes rule. It is promising to use multiple solutions which have different characteristics because they can deal with uncertainty by complementing each other for better robustness. Several works are concerned with machine learning based on evolutionary computation (EC) and combining the solutions found in the last population [2, 3, 4]. Fig. 1 shows the basic idea of the proposed method. To show the usefulness of the method, we conduct experiments with a benchmark problem of ASIA network.

M.-S. Hacid et al. (Eds.): ISMIS 2005, LNAI 3488, pp. 92 – 101, 2005.

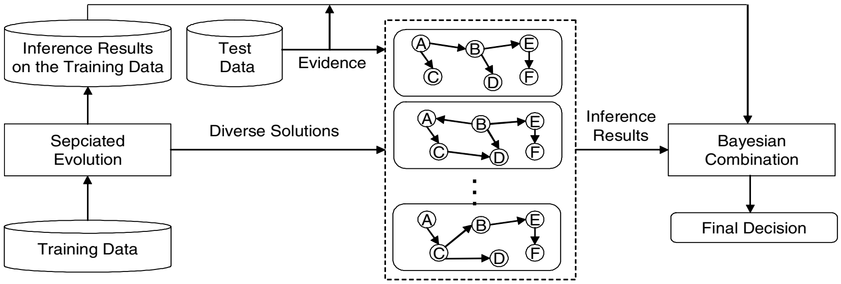


Fig. 1. The proposed ensemble model of speciated Bayesian networks

2 Evolutionary Bayesian Networks

As indicated by [5], evolutionary algorithm is suitable for dynamic and stochastic optimization problems but there are a few works to deal with the issue using evolutionary Bayesian network. One important approach to learning Bayesian networks from data uses a scoring metric to evaluate the fitness of any given candidate network for the database, and applies a search procedure to explore the set of candidate networks [6]. Because learning Bayesian networks is, in general, an NP-hard problem and exact methods become infeasible, recently some researchers present a method for solving this problem of the structure learning of Bayesian network from a database of cases based on evolutionary algorithms [7].

Larranaga *et al.* carry out performance analysis on the control parameters of the genetic algorithms (population size, local optimizer, reproduction mechanism, probability of crossover, and mutation rate) using simulations of the ASIA and ALARM networks [1]. Also, they propose searching for the best ordering of the domain variables using genetic algorithms [8]. Wong *et al.* have developed a new approach (MDLEP) to learn Bayesian network structures based on the Minimum Description Length (MDL) principle and evolutionary programming (EP) [9]. Wong *et al.* propose a novel data mining algorithm that employs cooperative co-evolution and a hybrid approach to discover Bayesian networks from data [10]. They divide the network learning problem of n variables into n sub-problems and use genetic algorithms for solving the sub-problems.

3 Speciated Evolutionary Bayesian Network Ensemble

The system sets each BN with random initial structures. The fitness of Bayesian network for training data is calculated using general Dirichlet prior score metric (DPSM). Yang reports that DPSM performs better than other scoring metrics [11]. Fitness sharing using MDL difference measure which needs only small computational cost rescales the original fitness for diversity. Once the fitness is calculated, genetic algorithm selects the best 80% individuals to apply genetic operators. The genetic operators, crossover and mutation, are applied to those selected individuals. After applying genetic operations, the new set of individuals forms a new population. It finishes when stop criterion is satisfied. Using clustering, representative individuals are selected and combined with Bayesian scheme.

3.1 Representation

In evolutionary algorithm, it is very important to determine the representation of an individual. There are several methods to encode a Bayesian network such as connection matrix and variable ordering list. Although connection matrix representation is simple and genetic operators can be easily implemented, additional "repair" operation is needed. In the structure learning of Bayesian network an ordering between nodes of the structure is often assumed, in order to reduce the search space. In variable ordering list, a Bayesian network structure can be represented by a list L with length n, where its elements l_j verify

$$\text{if } l_j \in Pa(l_i) \text{ then } j < i,\ (l_j, l_i \in V)$$

Although variable ordering list can reduce search space, it has a shortcoming that only one Bayesian network structure can be constructed from one ordering.

In this paper, we devise a new chromosome representation which combines both of them. It does not need a "repair operator" but also provides enough representation power for searching diverse Bayesian network structures. In this representation, a Bayesian network structure can be represented by a variable ordering list L with length n, and $n \times n$ connectivity matrix C. The definition of L is the same as the above formula but an element of matrix C is used for representing an existence of arcs between variables.

$$c_{ij} = \begin{cases} 1 & \text{if there is an arc between } \mathrm{x}_i \text{ and } \mathrm{x}_j\ (i > j), \\ 0 & \text{otherwise.} \end{cases}$$

Lower left triangle describes the connection link information.

3.2 Genetic Operators

The crossover operator exchanges the structures of two Bayesian networks in the population. Since finding an optimal ordering of variables resembles the traveling salesman problem (TSP), Larranaga *et al.* use genetic operators that were developed for the TSP problem [8]. In their work, cycle crossover (CX) operator gives the best results and needs a small population size to give good results while other crossover operators require larger population sizes. CX operator attempts to create an offspring from the parents where every position is occupied by a corresponding element from one of the parents. The $n \times n$ connection matrix can be represented as a string with length ${}_nC_2$, because only lower left triangle area in the matrix describes useful information. Two connection matrix represented as a string can be exchanged by using 1-point crossover.

The displacement mutation operator (DM) is used for mutation of variable order. The combination of cycle crossover and the mutation operator shows better results in extensive tests [8]. The operator selects a substring at random. This substring is removed from the string and inserted in a random place. Mutation operator for the connection matrix performs one of the two operations: addition of a new connection and deletion of an existing connection. If the connection link does not exist and the connection entry of the BN matrix is ‘0’, a new connection link is created. If the connection link already exists, it removes the connection link.

3.3 Fitness Evaluation

In order to induce a Bayesian network from data, researchers proposed a variety of score metrics based on different assumptions. Yang *et al.* compared the performance of five score metrics: uniform prior score metric (UPSM), conditional uniform prior score metric (CUPSM), Dirichlet prior score metric (DPSM), likelihood-equivalence Bayesian Dirichlet score metric (BDe), and minimum description length (MDL) [11]. They concluded that the tenth-order DPSM is the best score metric. If the Dirichlet distribution is assumed, then the score metric can be written as

$$P(B,D) = P(B)\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij}+N_{ij})}\times\prod_{k=1}^{r_i}\frac{\Gamma(N'_{ijk}+N_{ijk})}{\Gamma(N'_{ijk})}.$$

Here, N_{ijk} denotes the number of cases in the given database D in which the variable x_i takes the *k*th value ($k=1,2,\ \ldots,r_i$), and its parent $Pa(x_i)$ is instantiated as the *j*th value ($k=1,2,\ \ldots,q_i$), and $N_{ij}=\sum_{k=1}^{r_i}N_{ijk}$. N'_{ijk} is the corresponding Dirichlet distribution orders and $N'_{ij}=\sum_{k=1}^{r_i}N'_{ijk}$. As [11], we investigate a special case where the Dirichlet orders are set to a constant, say $N'_{ijk}=10$.

3.4 Fitness Sharing with MDL Measure

There are several ways to simulate speciation. In this paper, fitness sharing technique is used. Fitness sharing decreases the fitness of individuals in densely populated area and shares the fitness with other BN’s. Therefore, it helps genetic algorithm search broader solution space and generate more diverse BN’s. With fitness sharing, the genetic algorithm finds diverse solution. The population of BN’s is defined as $\{B_1,B_2,\ldots,B_{pop\_size}\}$.

Given that f_i is the fitness of an individual B_i and $sh(d_{ij})$ is a sharing function, the shared fitness fs_i is computed as follows :

$$fs_i = \frac{f_i}{\sum_{j=1}^{pop\_size} sh(d_{ij})}$$

The sharing function $sh(d_{ij})$ is computed using the distance value d_{ij} which means the difference of individuals B_i and B_j as follows:

$$sh(d_{ij}) = \begin{cases} 1-\dfrac{d_{ij}}{\sigma_s}, & 0 \le d_{ij} < \sigma_s \\ 0, & d_{ij} \ge \sigma_s \end{cases}$$

Here, σ_s means the sharing radius. σ_s is set with a half of the mean of distances between each BN in initial population. If the difference of the individuals is larger than σ_s, they do not share the fitness. Only the individuals that have smaller distance value than σ_s can share the fitness. Fitness of individuals on the highest peak decreases when the individuals in the area are dense.

Although there is no consensus on the distance measure for the difference of individuals B_i and B_j, an easily acceptable measure is structural difference between them. Lam defines the structural difference measure in the minimum description length principle which is well established in machine learning [12]. To compute the description length of the differences we need develop an encoding scheme for representing these differences. It is clear that the structure of B_i can be recovered from the structure of B_j and the following information.

- A list of the reverse arcs
- The missing arcs of B_i
- The additional arcs of B_i

Let *r*, *m*, and *a* be, respectively, the number of reverse, missing, and additional arcs in B_i, with respect to a network B_j. Since there are $n(n-1)$ possible directed arcs, we need $\log_2[n(n-1)]$ bits. The description length B_i given B_j is as follows.

$$DL(B_i \mid B_j) = DL(B_j \mid B_i) = (r+m+a)\log_2[n(n-1)]$$

Because it is formally defined and has low computational cost, we have adopted the measure to calculate the difference between Bayesian networks.

3.5 Combination of Multiple Bayesian Networks

Single linkage clustering is used to select representative Bayesian networks from a population of the last generation. The number of individuals for the combination is automatically determined by the predefined threshold value. Bayesian method takes each BN's significance into accounts by allowing the error possibility of each BN to affect the ensemble's results [13].

The number of selected BN's is *K*. $B=\{B_1, B_2, \ldots, B_K\}$. *B* denotes the set of the BN's. *Q* is the target node. *M* is the number of states of *Q* that has $g_1, g_2, \ldots, g_M$ states. The *k*th BN produces the probability that *Q* is in state *i* using Bayesian inference, and it is defined as $P_i(k)$. Using training data, $P(P_i(k) > 0.5 \mid Q = q_i)$ is calculated. Finally, $P(Q = q_i \mid B_1, B_2, \ldots, B_k)$ is represented as follows.

$$P(Q = q_i \mid B_1, B_2, \ldots, B_k) = \prod_{k=1}^{K} \frac{P(P_i(k) > 0.5 \mid Q = q_i)}{P_i(k)}$$

4 Experimental Results

The ASIA network, introduced by Lauritzen and Spiegelhalter [14] to illustrate their method of propagation of evidence considers a small piece of fictitious qualitative me dical knowledge. Fig. 2 presents the structure of the ASIA network. The ASIA network is a small Bayesian network that calculates the probability of a patient having tuberculosis, lung cancer or bronchitis respectively based on different factors, for exa mple whether or not the patient has been to Asia recently.

There are several techniques for simulating BN's, and we have used probabilistic logic sampling, with which we develop a database of 1000 cases. Population size is 50 and the maximum generation number is 1000. Crossover rate is 0.5, selection rate is 0.8 and mutation rate is 0.01.

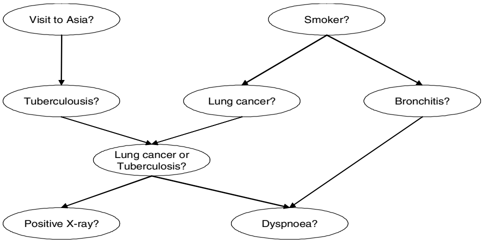


Fig. 2. The network structure for a benchmark problem

Randomly selected 100 cases from the 1000 cases are used as test data while the remaining data are used for training. Because the size of the network is relatively small, both of genetic algorithm without speciation and that with speciation can find near-optimal solutions. Unlike other classifiers such as neural networks and support vector machine, Bayesian network can infer the probability of unobserved target states given the observed evidences. It is assumed that "Tuberculousis," "Lung cancer," "Bronchitis," and "Lung cancer or Tuberculosis" nodes are unobserved target variables and the remaining nodes are observed variables. The four nodes are regarded as target nodes and represent whether a person gets a disease. Although the data are generated from the original ASIA network, inference results for the data using the original network may be incorrect because there are only four observed variables.

Using single linkage clustering method, the last generation of speciated evolution is clustered and there are five clusters when threshold value is 90. If the number of individuals in the cluster is more than two, the one with the highest fitness is chosen. An individual labeled as 49 in the largest cluster shows the best fitness (-2123.92) while the others' fitness are ranged from -2445.52 to -2259.57. Although the fitness values of four individuals in other clusters are relatively smaller than the best one, the combination can be stronger one by complementing each other. Inference accuracies of original network, the best individual from simple genetic algorithm without speciation (Fig. 3), and the combination of the speciated Bayesian networks (Fig. 4) for the "Lung cancer or Tuberculosis" node are 98%, 98%, and 99%, respectively. For "Tuberculousis," "Lung cancer," and "Bronchitis" nodes, they show the same accuracy. Both of the best individual in simple genetic algorithm and the individual labeled as 49 in the speciation have a similar structure with the original network. However, the other four networks in the speciation have more complex structures than the original network. Table 1 summarizes the inference results of the case where only the combination of BN's produces correct result.

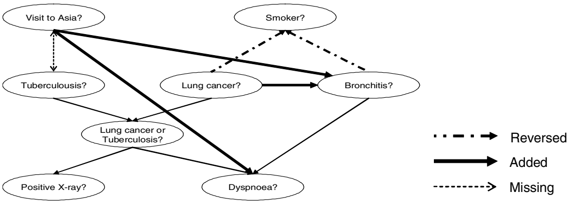
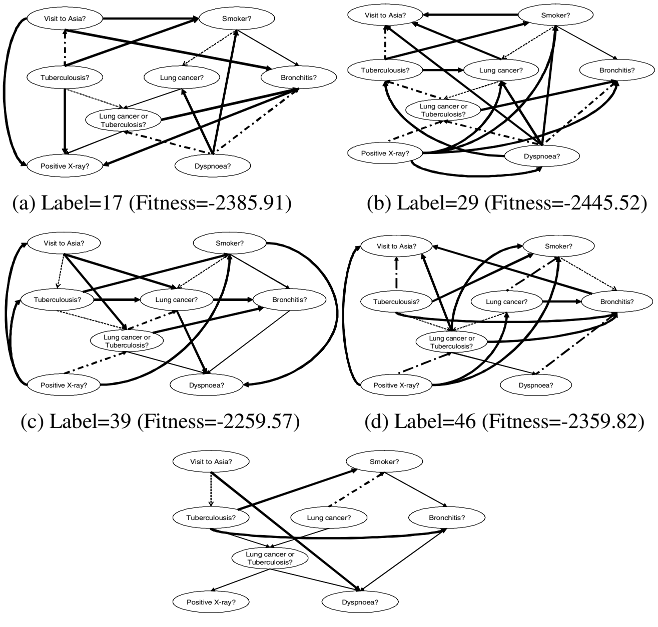


Fig. 3. The best individual evolved using simple genetic algorithm without speciation

Predictive accuracy is the most popular technique for evaluating models, whether they are Bayesian networks, classification trees, or regression models [15]. However, a fundamental problem lies in that predictive accuracy entirely disregards the confidence of the prediction. For example, a prediction of the target variable with a probability of 0.51 counts exactly the same as a prediction with a probability of 1.0. In recognition of this kind of problem, there is a growing movement to employ cost-sensitive classification methods. Good invented a cost neutral assessment measure [16]. Good's definition is

$$IR = \sum_i [1 + \log_2 P(x_i = v)]$$

where i indexes the test cases, v is the actual class of the ith test case and $P(x_i)$ is th e probability of that event asserted by the learner.



(a) Label=17 (Fitness=-2385.91)

(b) Label=29 (Fitness=-2445.52)

(c) Label=39 (Fitness=-2259.57)

(d) Label=46 (Fitness=-2359.82)

(e) Label=49 (Fitness=-2123.92)

Fig. 4. The five individuals evolved using genetic algorithm with speciation

Table 1. Inference results of "Lung cancer or Tuberculosis" when "Visit to Asia" is "No visit," "Smoker" is "Non-Smoker," "Positive X-ray" is "Abnormal" and "Dyspnoea" is "Present." The original value of "Lung cancer or Tuberculosis" is "True." (T=True, F=False)

	The original network	The best individual in simple GA	Label 17	Label 29	Label 39	Label 46	Label 49	The combination of five BN's
	False	False	False	True	True	True	False	True
P(T)	0.494	0.471	0.118	0.709	0.676	0.576	0.464	0.503
P(F)	0.506	0.529	0.882	0.291	0.324	0.424	0.536	0.497

Table 2. Information rewards for inferring "Bronchitis" when "Dyspnoea" is unobservable. Experimental results are the average of ten runs

The original network	The best individual in simple GA	The best individual in speciation	The combination of BN's
-2.08	-3.14 ± 0.18	-3.02 ± 0.22	0.86 ± 0.24

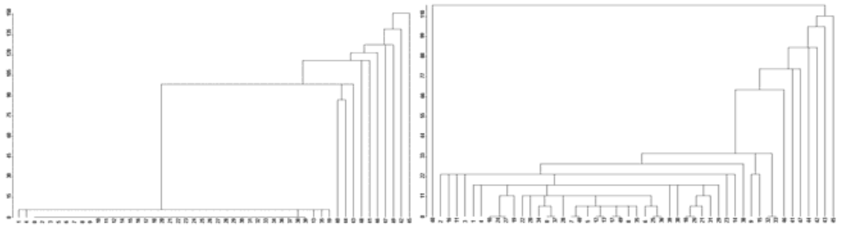

(a) Genetic algorithm (b) Speciated evolution

Fig. 5. The comparison of dendrograms

The combination of the speciated BN's can improve the predictability even when some variables are unobserved. Table 2 summarizes information reward values for the situation and the combination of BN's shows the best performance. If the reward value is high, it means that the classifier performs well. Although the best individual in the speciated evolution returns similar reward value with that from the best individual in genetic algorithm, the combination returns improved information reward value that is larger than that of original network. Fig. 5 shows the comparison of dendrogram. In Fig. 5(a), 40 individuals in the left side form one cluster and they are almost the same. Meanwhile, 40 individuals in the left side in Fig. 5(b) form a number of clusters.

5 Conclusion

In this paper, we have proposed an ensemble of multiple speciated Bayesian networks using Bayesian combination method. Although some Bayesian networks in ensemble provides incorrect inference, the ensemble network can perform correctly by reflecting each Bayesian network's behavior in training data and reducing the effect of incorrect results. Experimental results on ASIA network show that the proposed method can improve simple genetic algorithm which converges to only one solutions (sometimes the best solution performs poorly in changed environments) and the fusion of results from speciated networks can improve the inference performance by compensating each other. Future work of this research is to develop real-world applications based on the proposed method. Highly dynamic problems such as robot navigation, user context recognition, and user modeling can be considered as candidates.

Acknowledgements

This research is supported by the 21st century frontier R&D program of ministry of science and technology in Korea.

References

1 Larranaga, P., *et al.*: Structure learning of Bayesian networks by genetic algorithm. IEEE Trans. on Pattern Analysis and Machine Intelligence, 18(9) (1996) 912-926

2 Immamura, K., Heckendorn, R. B., Soule, T., and Foster, J. A.: Abstention reduces errors-Decision abstaining N-version genetic programming. GECCO (2002) 796-803

3 Iba, H.: Bagging, boosting, and bloating in genetic programming. GECCO (1999) 1053-1060

4 Anglano, C., Giordana, A., Bello, G. L., and Saitta, L.: Coevolutionary, distributed search for inducing concept description. ECML (1998) 322-333

5 Branke, J.: Evolutionary Optimization in Dynamic Environments. Kluwer (2001)

6 Neapolitan, R. E.: Learning Bayesian Networks. Prentice Hall (2003)

7 Neil, J. R., and K. B. Korb.: The evolution of causal models: A comparison of Bayesian metrics and structure priors. Pacific-Asia Conference on Knowledge Discovery and Data Mining (1999)

8 Larranaga, P., *et al.*: Learning Bayesian network structures by searching for the best ordering with genetic algorithm. IEEE Trans. on Systems, Man and Cybernetics – Part (A), 26(4) (1996) 487-493

9 Wong, M. L., Lam, W., and Leung, K.S.: Using evolutionary programming and minimum description length principle for data mining of Bayesian networks. IEEE Trans. on Pattern Analysis and Machine Intelligence, 21(2) (1999) 174-178

10 Wong, M. L., Lee, S.Y., and Leung, K.S.: Data mining of Bayesian networks using cooperative coevolution. Decision Support Systems (2004) (in press)

11 Yang, S. and Chang, K.-C.: Comparison of score metrics for Bayesian network learning. IEEE Trans. on Systems, Man and Cybernetics-Part A, 32(3) (2002) 419-428

12 Lam, W.: Bayesian network refinement via machine learning approach. IEEE Trans. on Pattern Analysis and Machine Intelligence, 20(3) (1998) 240-251

13 Xu, L., Krzyzak, A. and Suen, C.Y.: Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. on Systems, Man and Cybernetics, SMC-22(3) (1992) 418-435

14 Lauritzen, S. L. and Spiegelhalter, D. J.: Local computations with probabilities on graphical structures and their applications on expert systems. Journal Royal Statistical Society B, 50(2) (1988) 157-224

15 Korb, K. B. and Nicholson, A. E.: Bayesian Artificial Intelligence. CHAPMAN & HALL, (2003)

16 Good, I.: Relational decisions. Journal of the Royal Statistical Society B, 14 (1952) 107-114