Pertanika J. Sci. & Technol. 25 (3): 719 - 734 (2017)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Comparison of Scoring Functions on Greedy Search Bayesian Network Learning Algorithms

ChongYong, Chua* and HongChoon, Ong

School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia

ABSTRACT

Score-based structure learning algorithm is commonly used in learning the Bayesian Network. Other than searching strategy, scoring functions play a vital role in these algorithms. Many studies proposed various types of scoring functions with different characteristics. In this study, we compare the performances of five scoring functions: Bayesian Dirichlet equivalent-likelihood (BDe) score (equivalent sample size, ESS of 4 and 10), Akaike Information Criterion (AIC) score, Bayesian Information Criterion (BIC) score and K2 score. Instead of just comparing networks with different scores, we included different learning algorithms to study the relationship between score functions and greedy search learning algorithms. Structural hamming distance is used to measure the difference between networks obtained and the true network. The results are divided into two sections where the first section studies the differences between data with different number of variables and the second section studies the differences between data with different sample sizes. In general, the BIC score performs well and consistently for most data while the BDe score with an equivalent sample size of 4 performs better for data with bigger sample sizes.

Keywords: Bayesian network, greedy search, heuristic search, score-based, scoring function, structure learning

INTRODUCTION

Since its introduction by Pearl (1985), the Bayesian Network (BN) has had a huge impact on data mining as one of the best mining tools. Numerous studies have been done on learning BN in the last few decades. Constraint-based and score-based algorithms are two major classes of BN structural learning algorithms while hybrid-based algorithms that combined both is later introduced. Despite advantages and disadvantages of each algorithm, score-based algorithms, especially the greedy search type algorithm that constantly looks for the most improvement in each iteration, are used frequently due to its simplicity and

Article history:
Received: 11 July 2016
Accepted: 5 December 2016

E-mail addresses:
cychua91@hotmail.com (ChongYong, Chua),
hcong@usm.my (HongChoon, Ong)
*Corresponding Author

ISSN: 0128-7680

effectiveness. With score-based algorithms such as Hill Climbing (Daly & Shen, 2007) and K2 algorithm (Cooper & Herskovits, 1992), a scoring function is chosen to calculate the joint probability distribution using a BN structure from its corresponding database, and then a network searching approach is used to maximise this function. To this end, most score-based algorithm research is done on improving the search phase but scoring functions play a vital role as well. A good search approach can help to sidestep local maxima while a good scoring function can decide which network is closer to the true network. In real world applications, a network with a higher score does not always imply a good output. Instead, a network that can define the true causal relationship of the problem studied is a better output.

Scoring functions are usually derived from basic assumptions on the network parameter distribution. Based on multinomial sampling and dirichlet distribution assumptions, Heckerman et al. (1995) proposed the BD score and further improved it to the BDe score after including the likelihood equivalence assumption. Cooper and Herskovits (1992) introduced the K2 score, which is a particular case of BD score in their K2 algorithm paper. In other cases without the assumption of distribution, log-likelihood score is one of the established scores available. However, it does not represent a good score in BN learning as it leads to overfitting due to its tendency of favouring a complete network. To limit the overfitting problem, AIC score (Akaike, 1974) and BIC score (Schwarz, 1978) are proposed by introducing a penalising term in the log-likelihood score function.

In most cases, improvement of BN structural learning is done independently for searching mechanism and scoring functions. However, there is a relationship between searching mechanism and scoring functions. Each scoring function has its own characteristics and advantages that fit into certain learning algorithms. In this study, we were interested in identifying which scoring function worked better for a greedy search type algorithm. The rest of this paper is organised as follows. Section 2 introduces some preliminary concepts, assumptions and notations of Bayesian networks and scoring functions used. This is followed by Section 3, which shows the results of our experiment and discusses the results, and lastly, Section 4 consists of the conclusion.

MATERIALS AND METHODS

Bayesian Network

BN, also known as Bayesian Belief Network (BBN), is a powerful data mining tool for reasoning under uncertainty (Heckerman, 1996). BN is a directed acyclic graph (DAG) that consists of two parts $G=<S, \Theta>$. For a set of variables $X=\{X_1, X_2,\ldots, X_n\}$ from data D, a network structure, S, is a set of arcs connecting nodes or variable X, which indicates that there exists a dependent relationship between the nodes. Θ is the conditional probability associated with each variable. Suppose that X_i denotes a variable and its corresponding node, π_i, is the parent of each node X_i in S as well as the variable corresponding to the parents. Given the structure S, the joint probability distribution for X is given by:

$$f(X) = \prod_{i=1}^{n} f(X_i | \pi_i) \quad [1]$$

BN is then constructed based on the conditional independence concept, which is defined below:

Definition 1 (Conditional Independence)
Two variables X_1 and X_2 are conditionally independent given if the variable X_3

$$f(X_1 \mid X_3) = f(X_1 \mid X_2, X_3) \quad [2]$$

Bayesian Network Structural Learning

Structural learning in BN refers to selecting the structure that most accurately defines the causal relationships between variables from a set of structure candidates. Typically, a structural learning algorithm can be separated into three major categories, which are constraint-based, score-based and hybrid-based. The constraint-based structural learning algorithm learns the network structure by analysing the probabilistic relations entailed by the Markov property of BN with a conditional independence test and then constructs a graph that satisfies the corresponding d-separation statement (Scutari, 2010). A few common conditional independence tests used for constraint-based algorithms are mutual information (Kullback, 1959) and Pearson's χ^2 (Spirtes et al., 1993). These tests are able to determine the existence of an edge between two nodes based on the conditional independence property as defined in Definition 1. A constraint-based algorithm begins with learning the skeleton of the network that is a completely undirected network. Most of the learning algorithms restrict the search to the Markov blanket of each node (including the parents, the children and the parents of the children of that particular node). The next step is to set the direction of the arcs that are a part of the v-structure (a triplet of nodes incident on a converging connection X_j->X_i<-X_k). Lastly, direction of other arcs will be set to satisfy the acyclicity constraint (Neapolitan, 2004). The Incremental Association Markov Blanket by Tsamardinos et al. (2003) and PC algorithm by Spirtes and Glymour (1991) are among the well-established and most applied constraint-based algorithms.

On the other hand, the score-based algorithm works differently from the constraint-based algorithm. This algorithm is a type of maximisation problem that assigns a score to each network structure candidate and tries to maximise it with a certain heuristic search algorithm. Hill-climbing and Tabu search (Glover & Laguna, 1993) are known as greedy search algorithms while the Genetic algorithm (Goldberg, 1989) and Particle Swarm Optimisation (Kennedy & Eberhart, 1995) are examples of metaheuristic algorithms. Greedy search algorithms always opt for the best improvement of each iteration, which is referred to as the greedy property while metaheuristic algorithms explore the search space with simple or complex procedures inspired by natural phenomena. The score of a network structure is a guideline or criterion used to measure the fitness of the structure to prior knowledge and data. A score-based algorithm begins with an empty or a random structure and modifies the structure afterwards. Some transition functions such as adding arcs, deleting arcs or reversing arcs will be applied to the network structure to improve the structure's score. The iteration stops when the score converges at one point. However, one of the disadvantages of score-based algorithms is the iteration might be stuck at the local maxima and not return an optimum solution.

The hybrid-based algorithm is a combination of both score-based and constraint-based algorithms. It begins with constructing a skeleton or partially directed DAG (PDAG) using a conditional independence test. It then continues with performing a constrained score-based algorithm on the network obtained in the previous stage. MMHC (Tsamardinos et al., 2006) and H2PC (Gasse et al., 2014) are examples of hybrid-based algorithms. Despite the existence of various learning algorithms, score-based algorithms are frequently used in real-life applications compared to other methods. Constraint-based algorithms are efficient and faster especially when the data consist of a large number of variables, but it is strongly dependent on sample size of the data, where the results of conditional independence tests used are not entirely reliable with finite data (Fast, 2010). This weakness is crucial as most real-life data are limited and even incomplete. Although score-based algorithms suffer from limited data as well, the impact is not as significant as constraint-based algorithms. Hybrid-based algorithms on the other hand are not well established yet and have limited a choice of algorithms compared to the previous two types of algorithm. Hence, we are interested in improving score-based algorithms by reviewing existing scoring functions and their relationship with learning algorithms, specifically, greedy search algorithms.

Data Assumptions

The following are a few assumptions about the data considered in this study.

I. All variables X_i, $i = 1, 2, \ldots, n$ are discrete and observable and X_i has r_i possible values.
II. All data are complete i.e. all instances have values for all variables that have no missing data. There are no latent variables in the database.
III. All cases occur independently given a Bayesian network model.

Scoring Functions

Bayesian Dirichlet likelihood-equivalence (BDe). As an extension of the the Bayesian Dirichlet score, the BDe score included two more assumptions:

Assumption 1: Likelihood equivalence
Given two directed acyclic graphs, G and G', such that $P(G)>0$ and $P(G')>0$, if G and G' are equivalent then $P(\Theta|G)=P(\Theta|G')$.

Assumption 2: Complete structural possibility
For any complete directed acyclic graph G, we have $P(G)>0$.
The BDe score with equivalent sample size, , can be expressed (Heckerman et al., 1995) as:

$$P(G, D) = P(G) \times \prod_{i=1}^{n} \prod_{j=1}^{q} \left(\frac{\Gamma(N'_{ij})}{\Gamma(N_{ij}+N'_{ij})} \times \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk}+N'_{ijk})}{\Gamma(N'_{ijk})} \right) \quad [3]$$

where N_{ijk} denotes the number of instances in the database D where the variable x_i assigned its kth value ($k = 1, 2, \ldots, r_i$), and its parent π_i assigned its jth value ($j = 1, 2, \ldots, q_i$), $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$, $N'_{ijk} = N' \times P(X_i = x_{ik}, \prod_{X_i} = w_{ij}|G)$ and $N'_{ij} = \sum_{k=1}^{r_i} N'_{ijk}$.

Akaike Information Criterion (AIC) & Bayesian Information Criterion (BIC). Log-likelihood (LL) score in BN is a measure of likelihood in log form between the network and data parameter. The LL score is written as:

$$LL(G|D) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk}\log\left(\frac{N_{ijk}}{N_{ij}}\right) \quad [4]$$

The LL score has decomposable property, where the score is the summation of conditional probability of each node given their parent sets. It also assumes likelihood equivalence assumption as explained in Assumption 1. Due to its likelihood equivalence and decomposable property, the LL score is computational efficient when used in structure learning. However, the LL score is not commonly used in BN structure learning as it tends to favour a complete network. When used in a structural learning algorithm, it tends to generate a fully connected DAG, causing overfitting problem. In order to eliminate overfitting problem, a penalised term is introduced to limit the number of arcs in the final network. A general penalised LL score, PLL, is shown as follows:

$$PLL(G|D) = LL(G|D) - f(N)|G| \quad [5]$$

where, $|G| = \sum_{i=1}^{n}(r_i - 1)q_i$ denotes network complexity, which is the number of parameters in Θ for the network G and $f(N)$ is a non-negative penalisation function. When $f(N)$=1, the penalised score function is the Akaike Information Criterion score (AIC) as shown below (Akaike, 1974):

$$AIC(G|D) = LL(G|D) - |G| \quad [6]$$

Later on, Schwarz (1978) proposed a stricter penalised scoring function, Bayesian Scoring Criterion score (BIC), which is given as:

$$BIC(G|D) = LL(G|D) - \frac{1}{2}log(N)\cdot|G| \quad [7]$$

K2 Score. Cooper and Herskovits (1992) proposed the K2 score, which is a particular case of BD score with the uninformative assignment $N'_{ijk} = 1$ (corresponding to zero pseudo-counts). The K2 score can be expressed as:

$$K2(G,D) = \log\big(P(B)\big) + \sum_{i=1}^{n}\sum_{j=1}^{q_i}\left(\log\left(\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\right) + \sum_{k=1}^{r_i}\log\big(N_{ijk}!\big)\right) \quad [8]$$

Decomposability of the K2 score makes it a computational efficient scoring function but it does not have a score equivalence property as the previous three scoring functions. However, score equivalence property might or might not help in network learning. When we want to

learn the causal relationship between variables, we would need to eliminate the equivalence network from the true network as they infer a different causal relationship; score equivalence property skips this process. On the other hand, score equivalence property can reduce the time consumed if we only need the inference of one variable given another as either equivalence network can do this task.

RESULTS AND DISCUSSION

Experimental Design

In this experiment, five scoring functions were compared on seven sets of data ranging from 8 to 417 variables using three benchmark structure learning algorithms. The five scoring functions compared were the BDe score (ESS of 4), BDe (ESS of 10), AIC score, BIC score and K2 score. Hill Climbing (HC), Tabu Search (TS) and the K2 algorithm were used to test the accuracy of each scoring function. Since the optimal equivalent sample size (ESS) of the BDe score was unknown, we identified the ESS value by including two variations of the BDe score in this study with an ESS of 4 and 10, respectively. The length of the Tabu list used in the TS was set as 10 as a simple test was conducted to show that a Tabu list of more than 10 does not improve the quality of the network learnt. Node ordering required for K2 algorithm was obtained from the true network such that, if x_i preceded x_j in the ordering, x_j must not be in the parent sets of the x_i. True network in this study referred to the original network constructed by the authors of the paper of each dataset. True networks for each dataset are shown in Figures 1 to 7. Since all variables in the datasets studied did not have a number of parents exceeding 10 in their respective true network, the maximum number of parents for each algorithm was capped at 10.

The end result of networks generated by each scoring function and learning algorithm was compared with the true network using the Structural Hamming Distance (SHD) from "bnlearn" package in R. A brief explanation of SHD is quoted from Tsamardinos et al. (2006):

> *SHD directly compares the structure of the learned and the true networks and its use is fully oriented toward discovery rather than inference. SHD between two PDAGs is defined as the number of the following operators required to make the PDAGs match: add or delete an undirected edge, and add, remove, or reverse the orientation of an edge. Thus, an algorithm will be penalized by an increase of the score by 1 for learning a PDAG with an extra un-oriented edge and by 1 for not orienting an edge that should have been oriented. Algorithms that return a DAG are converted to the corresponding PDAG before calculating this measure. The reason for defining the SHD on PDAGs instead of DAGs is so that we do not penalize for structural differences that cannot be statistically distinguished.*

The first section of this experiment studied the effect of scoring functions on different types of datasets with varying numbers of variables as summarised in Table 1. Each dataset was run with three learning algorithms using five scoring functions. The final network of each simulation was compared to the true network using SHD as explained in the previous

paragraph. The smaller the SHD between the network obtained and the true network, the better the score. The second section of this experiment investigated the effect of scoring functions on same datasets with different sample sizes. We selected three datasets (Alarm, Win95pts and Andes) for this study. For each dataset, we generated three sets of data with 5,000, 10,000 and 20,000 instances, respectively, based on the conditional probability table of the true network. The method for comparing is the same as the first section of the experiment with different sets of data.

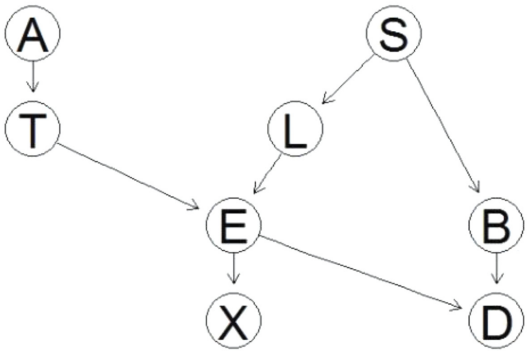

Figure 1. Asia

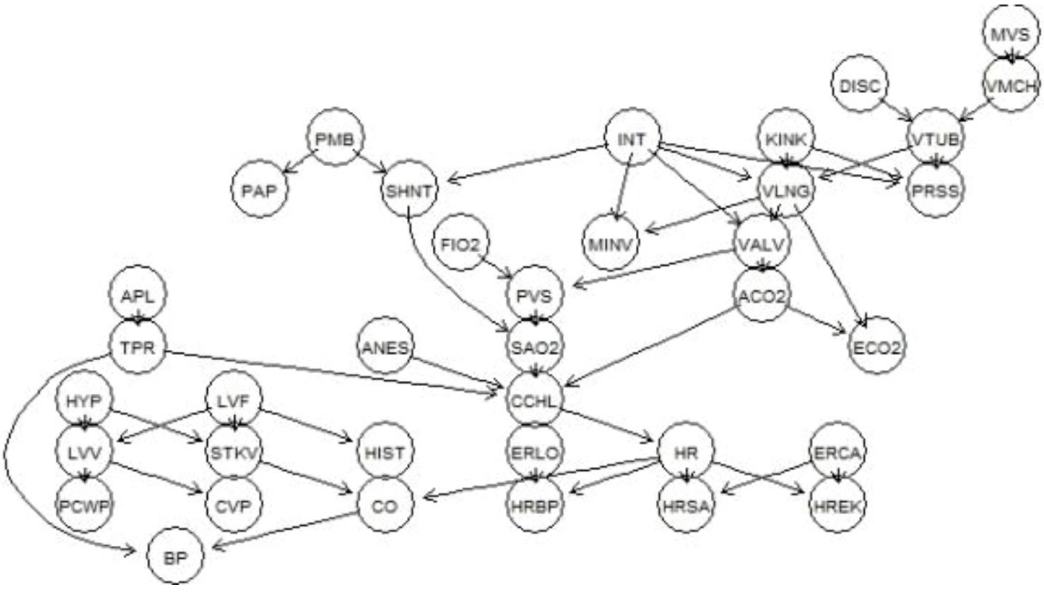

Figure 2. Alarm

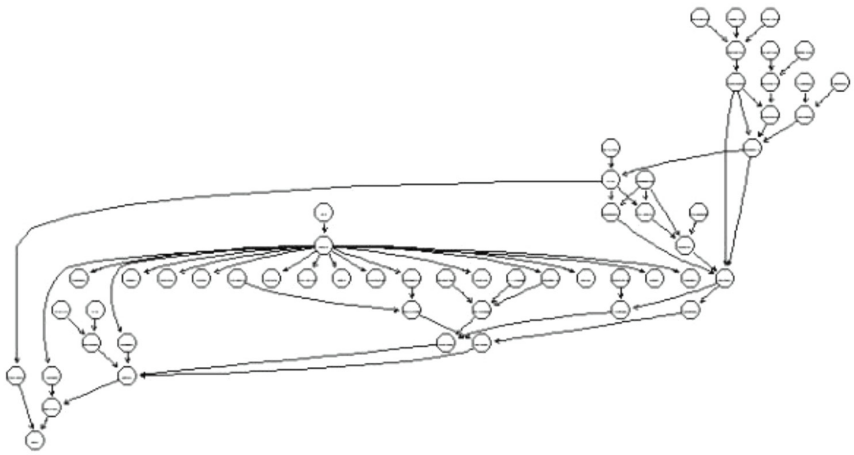

Figure 3. Hailfinder

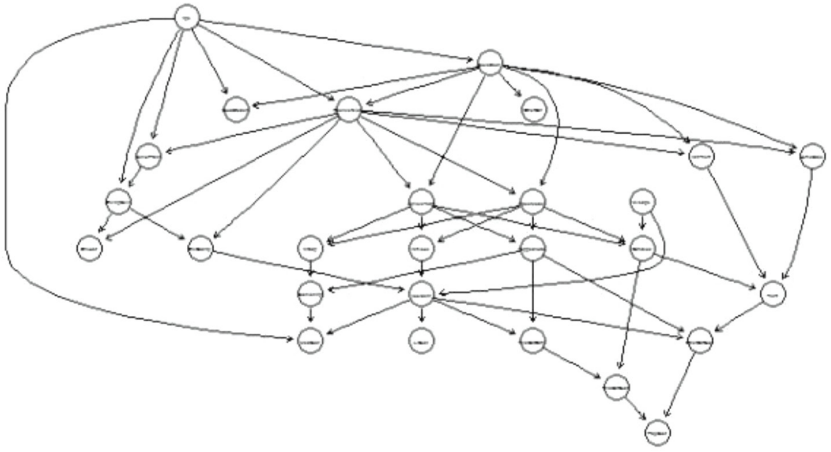

Figure 4. Insurance

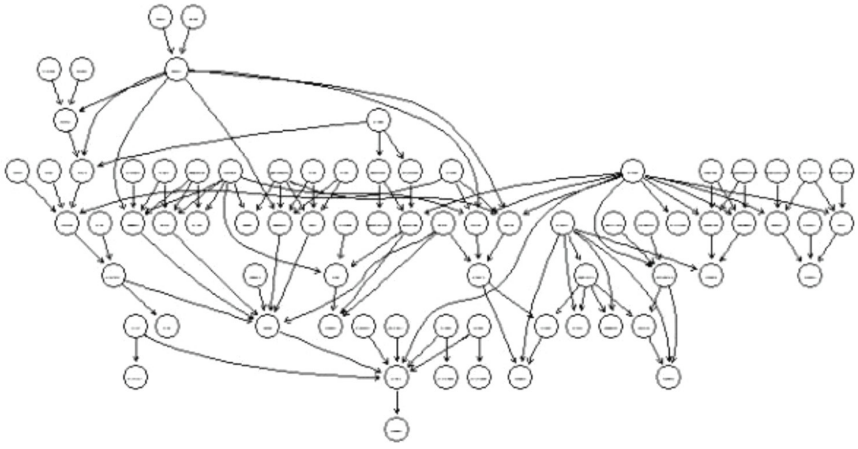

Figure 5. Win95pts

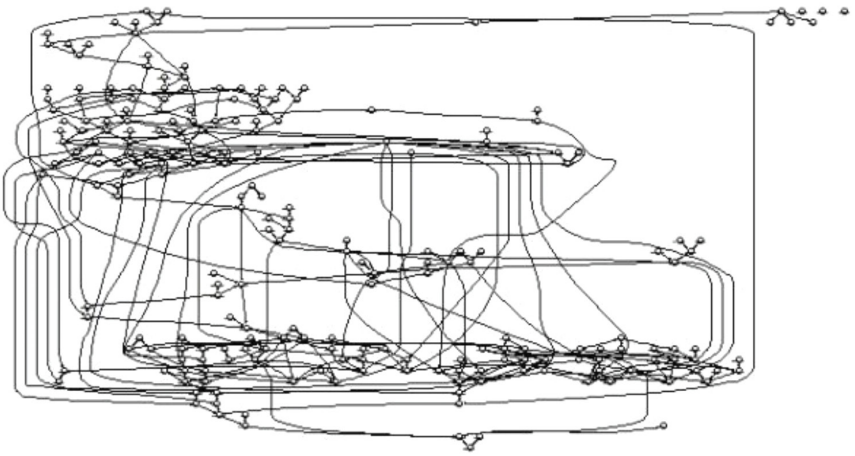

Figure 6. Andes

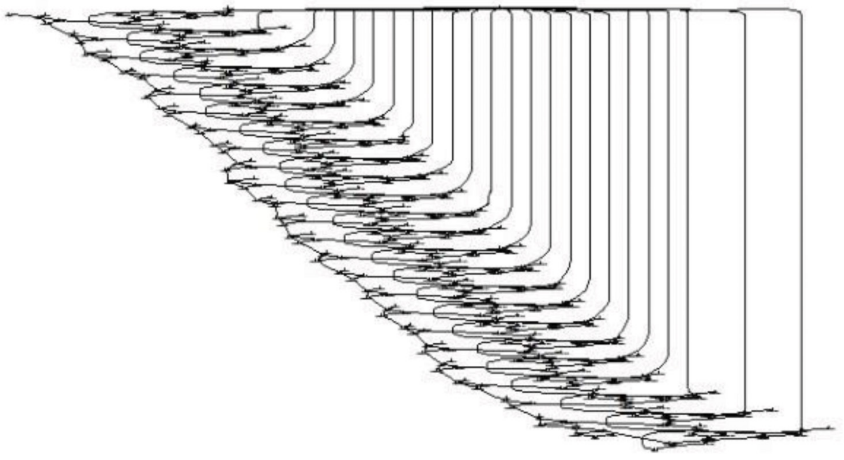

Figure 7. Diabetes

Table 1
Summary of Datasets

Data	No. of Variables	No. of Instances	Reference
Asia	8	5 000	Lauritzen and Spiegelhalter (1988)
Insurance	27	20 000	Binder et al. (1997)
Alarm	37	20 000	Beinlich et al. (1989)
Hailfinder	56	20 000	Abramson et al. (1996)
Win95pts	76	20 000	Developed at Microsoft Research and contributed to the community by Jack Breese.
Andes	223	20 000	Conati et al. (1997)
Diabetes	417	20 000	Andreassen et al. (1991)

Experiment Results

As explained in the previous section, the experiment was divided into two sections. In order to simplify the evaluation of the results, we categorised the data into small data (less than 50 variables), medium data (51-100 variables) and large data (more than 100 variables). The results of the first section are tabulated in Table 2 with bold font indicating the lowest SHD for each row.

For small data, the BDe score (ESS of 4) gives the best performance by generating 7 out of 9 networks with the lowest SHD. This is followed by the BIC score and K2 score with four and two networks of the lowest SHD, respectively. However, the BDe score with an ESS of 4 does not outperform other scoring functions as the differences were not convincing. For medium and large data, the BIC score performed best by generating seven networks with the smallest SHD from a total of 12 networks. Albeit only seven networks generated by BIC score had the smallest SHD, the difference between the BIC score and other scoring functions was significant with minimal exception. Surprisingly, the AIC score generated networks with the lowest SHD for data on diabetes but it did not differ much from the BIC score and both BDe scores.

The BDe score was asymptotically equivalent to the BIC score but one of the reasons the BDe with an ESS of 4 performed better for smaller data was due to the relatively larger sample size. BIC scores tend to penalise complex networks more heavily compared to BDe scores especially for smaller sample sizes. Since all datasets except Asia had the same sample size, which was 20,000 instances, the performance of the BDe score dropped as the number of variables increased. On the other hand, BIC scores favour a simpler network for larger data with a smaller sample size. Hence, the BIC score performed better than the BDe score as the number of variables increased.

In terms of an ESS parameter for the BDe score, an ESS of 4 outperformed an ESS of 10 for all networks generated. According to two studies done on finding the optimal ESS for BDe scores (Steck & Jaakkola, 2002; Silander et al., 2007), a lower ESS tended to favour deletion of arcs while a higher ESS favoured addition of arcs. With a large sample, the learnt network becomes an empty graph as the ESS approaches zero and tends to become a fully connected graph as the ESS increases. This explains why 4 is better than 10 as the ESS for the BDe score in this study as nearly all the networks generated had overfitting problem. Hence, a smaller number of ESS reduced the complexity of networks generated and was closer to the true network.

In the comparison with structural learning algorithms, the BIC score performed well for both Hill Climbing and Tabu search. Meanwhile, the K2 score occasionally performed better when the K2 algorithm was used. Although the K2 score performed better for certain data, the difference was small and the time consumed for the K2 score was significantly higher than for the others. For the second section of this experiment, the results are tabulated in Table 3. The results of different sample sizes on the Alarm data once again strengthened the belief that the BDe score performed better for larger sample sizes. For Alarm data with 20,000 instances, the BDe score with an ESS of 4 performed better with HC and TS. However, when the sample size was reduced to 10,000, the BIC score started to perform better and this was proven when the sample size was reduced to 5,000.

Table 2
Comparison Between Datasets

Data	Structural learning algorithm	Structural hamming distance (SHD)				
		BIC	AIC	BDe(4)	BDe(10)	K2
Asia	HC	**1**	4	**1**	8	8
	TS	**1**	4	**1**	8	8
	K2	**1**	4	**1**	8	4
Insurance	HC	45	43	42	45	**39**
	TS	44	42	**36**	39	39
	K2	**9**	11	10	29	10
Alarm	HC	35	53	**25**	38	33
	TS	31	49	**14**	37	33
	K2	**6**	19	10	14	7
Hailfinder	HC	**12**	25	19	19	39
	TS	**12**	44	35	42	39
	K2	**12**	15	18	18	25
Win95pts	HC	**38**	106	125	187	66
	TS	**38**	106	125	188	66
	K2	**19**	90	109	154	**19**
Andes	HC	**30**	512	112	192	216
	TS	**30**	515	113	194	216
	K2	**21**	473	94	168	53
Diabetes	HC	497	**445**	431	464	695
	TS	497	**445**	431	464	695
	K2	176	**77**	111	149	118

Table 3
SHD Comparison Between Sample Sizes

Data	Structural learning algorithm	Structural hamming distance (SHD)				
		BIC	AIC	BDe(4)	BDe(10)	K2
Alarm20000	HC	35	53	**25**	38	33
	TS	31	49	**14**	37	33
	K2	**5**	19	10	14	7
Alarm10000	HC	**12**	36	18	18	23
	TS	14	36	**13**	15	23
	K2	**2**	23	12	13	**2**
Alarm5000	HC	**12**	32	15	32	28
	TS	**12**	32	15	33	21
	K2	**4**	24	6	24	5
Win95pts20000	HC	**38**	106	125	187	66
	TS	**38**	106	125	188	66
	K2	21	90	109	154	**19**
Win95pts10000	HC	**46**	114	146	212	86
	TS	**46**	114	147	212	88
	K2	**23**	89	121	181	40
Win95pts5000	HC	**48**	100	148	217	80
	TS	**48**	100	152	217	80
	K2	**27**	86	152	197	36
Andes20000	HC	**30**	512	112	192	216
	TS	**30**	515	113	194	216
	K2	**21**	473	94	168	53
Andes10000	HC	**45**	493	127	240	267
	TS	**45**	493	134	242	267
	K2	**31**	443	110	211	65
Andes5000	HC	**72**	521	173	297	262
	TS	**72**	525	174	298	262
	K2	**57**	455	152	259	105

CONCLUSION

In this paper, five scoring functions were compared: BIC, AIC, BDe (ESS of 4), BDe (ESS of 10) and the K2 score. Two factors were manipulated to study the effects of scoring functions on data with a different number of variables and sample sizes. The first part of this study compared the performance of scoring functions between seven sets of data with different numbers of variables, while the second part of the study compared the scoring functions between three sets of data with different sample sizes.

The performance of the scoring functions in this study were measured using structural hamming distance (SHD) between the network generated for each scoring function and the true network. A smaller number of SHD indicated better similarities between the two networks, which implied that a better network was learnt. Albeit the higher scoring was advocated in network learning, defining a true causal relationship was more important as an overfitting network does not always imply true causal relationship.

When the sample size was relatively large, the BDe score with an ESS of 4 performed well. But as the number of variables increased, the data sample size was relatively smaller and the performance of the BDe score descended. On the other hand, the BIC score performed well and was consistent for all data regardless of the number of variables and sample size. Most of the networks generated using the AIC score was too complex and faced the problem of overfitting. Although the K2 score performed better with the K2 algorithm in certain cases, it is not recommended for use as the difference was insignificant while the time consumed was significantly greater. Different greedy search learning algorithms used have minimal impact on the performance of scoring functions. Based on the results, all networks generated had more arcs than the true network. In this case, it implies that stricter penalided terms tend to produce a network more similar to the true network. However, the penalised term is hard to determine as strict terms will produce barely connected network for data with few variables while loose terms will produce a fully connected graph for a large network in extreme cases. In summary, the BIC score is definitely the best benchmarked score for a greedy search type network learning algorithm and the BDe score can perform equally well provided the sample size is large.

REFERENCES

Abramson, B., Brown, J., Edwards, W., Murphy, A., & Winkler, R. L. (1996). Hailfinder: A Bayesian system for forecasting severe weather. *International Journal of Forecasting*, *12*(1), 57–71. doi: 10.1016/0169-2070(95)00664-8. doi:10.1016/0169-2070(95)00664-8 doi:10.1016/0169-2070(95)00664-8 doi:10.1016/0169-2070(95)00664-8

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723. doi: 10.1109/TAC.1974.1100705

Andreassen, S., Hovorka, R., Benn, J., Olesen, K. G., & Carson, E. R. (1991). A model-based approach to insulin adjustment. In *Proceedings of the 3rd Conference on Artificial Intelligence in Medicine, Springer-Verlag* (pp. 239–248). doi: 10.1007/978-3-642-48650-0_19

Beinlich, I. A., Suermondt, H. J., Chavez, R. M., & Cooper, G. F. (1989). The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine*, *Springer-Verlag* (pp. 247–256). doi: 10.1007/978-3-642-93437-7_28

Binder, J., Koller, D., Russell, S., & Kanazawa, K. (1997). Adaptive probabilistic networks with hidden variables. *Machine Learning*, *29*(2-3), 213–244. doi: 10.1023/A:1007421730016

Conati, C., Gertner, A. S., Van Lehn, K., & Druzdzel, M. J. (1997). On-line student modeling for coached problem solving using Bayesian networks. In *Proceedings of the 6th International Conference on User Modeling*, *Springer-Verlag* (pp. 231–242). doi: 10.1007/978-3-7091-2670-7_24

Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, *9*(4), 309–347. doi: 10.1007/BF00994110

Daly, R., & Shen, Q. (2007). Methods to accelerate the learning of Bayesian network structures. In *Proceedings of the 2007 UK Workshop on Computational Intelligence, Imperial College, London* (pp. 1-9). Retrieved from http://hdl.handle.net/2160/421

Fast, A. (2010). *Learning the structure of Bayesian networks with constraint satisfaction.* (Ph.D. thesis). University of Massachusetts Amherst, Department of Computer Science, U.S.A.

Gasse, M., Aussem, A., & Elghazel, H. (2014). A hybrid algorithm for Bayesian network structure learning with application to multi-label learning. *Expert Systems with Applications*, *41*(15), 6756–6772. doi: 10.1016/j.eswa.2014.04.032

Glover, F., & Laguna, M. (1993). Tabu search. In C. R. Reeves (Ed.), *Modern heuristic techniques for combinatorial problems* (pp. 70-150). Oxford: Blackwell Scientific Publishing.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimisation and machine learning*. Boston: Addison-Wesley Longman.

Heckerman, D. (1996). A tutorial on learning Bayesian networks. *Microsoft Research, Technical Report: MSRTR-95-06*. Retrieved from https://www.microsoft.com/en-us/research/publication/a-tutorial-on-learning-with-bayesian-networks/

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, *20*(3), 197–243. doi: 10.1023/A:1022623210503

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks* (pp. 1942–1948).

Kullback, S. (1959). *Information theory and statistics.* New York: Wiley.

Lauritzen, S., & Spiegelhalter, D. (1988). Local computation with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *50*(2), 157–224. Retrieved from http://www.jstor.org/stable/2345762

Neapolitan, R. E. (2004). *Learning Bayesian networks*. New Jersey: Pearson Prentice Hall.

Pearl, J. (1985). Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine* (pp. 329–334).

Schwarz, G. E. (1978). Estimation of the dimension of a model. *Annals of Statistics*, *6*(2), 461–464. Retrieved from http://www.jstor.org/stable/2958889

Scutari, M. (2010). Learning Bayesian network with the BNlearn r package. *Journal of Statistical Software*, *35*(3), 1–22. doi: 10.18637/jss.v035.i03

Silander, T., Kontkanen, P., & Myllymaki, P. (2007). On sensitivity of the MAP Bayesian network structure to the equipment sample size parameter. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence* (pp. 360–367). Retrieved from arXiv:1206.5293

Spirtes, P., & Glymour, C. (1991). An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, *9*(1), 62–72. doi: 10.1177/089443939100900106

Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causation, prediction and search*. New York, NY: Springer.

Steck, H., & Jaakkola, T. S. (2002). On the Dirichlet prior and Bayesian regularization. In *Advances in Neural Information Processing Systems 15, MIT Press* (pp. 697–704). Retrieved from http://hdl.handle.net/1721.1/6702

Tsamardinos, I., Aliferis, C. F., & Statnikov, A. (2003). Algorithms for large scale Markov blanket discovery. In *The 16th International FLAIRS Conference, St. Augustine, Florida* (pp. 376–381).

Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, *65*(1), 31–78. doi: 10.1007/s10994-006-6889-7