Structure Learning and Implementation in Bayesian Network

Dingxin Yu DINGXIN.YU@POSTGRAD.MANCHESTER.AC.UK
School of Computer Science, The University of Manchester

Abstract

This work is aimed to evaluate performance of Bayesian network for different directed acyclic graphs (DAGs) and explore Bayesian network property. Implementation and all experiments are accomplished in Java and Matlab environment using different test data sets in three categories. Experiments including how the likelihood and log-likelihood measure of Bayesian Network differs as the number of training data set changes, scoring function and fill distribution strategy in Bayesian network. Besides, the performances of different Bayesian network DAGs will be evaluated against various data set as well.

1. Introduction

In order to reach conclusions based on current information. A general approach for this purpose is given by probabilistic graphical models is allowing interpretable models to be constructed and then processed by reasoning algorithms by corresponding different models. Due to uncertainty of most real-world application, it is almost impossible to manually creating a model from various data. In this case, these probabilistic models can be learned automatically from data by using different algorithms. A variety of models discussed by probabilistic graphical models including spanning Bayesian networks, discrete and continuous models, undirected Markov networks, and extensions to deal with dynamical systems and relational data(Koller & Friedman, 2009). Bayesian network will be mainly discussed in later chapters with corresponding experiments.

Bayesian networks is a compact representation of conditional dependency relation between random variables. It consists of a graph and conditional probability tables. The graph is set of nodes that represents a set of random variables. Each variable has a finite set of values. Set of directed edges that represent dependency relationship between the corresponding random variables. Bayesian Network is a graph augmented with Conditional Probability Tables(CPTs) whereas vertices represent random variables and edges represent dependency between random variables.

This article consists mainly of introduction, background, experiments, analyses and conclusion. Bayesian network structure and its measuring properties are described in background section. In addition, techniques used during investigation and outlines are listed. All experimental work including performance tests are described and presented in graphs. Results will be analysed in details. Besides, conclusions and future concern are included as well.

2. Background

Numbers of methodologies are included in this project.

2.1. Markov Blanket and Value Elimination

Markov blanket and value elimination algorithm are two foundations in Bayesian network. A variable in a Bayesian network is conditionally independent from all other, given its Markov blanket. Markov blanket of a node X consists of parents of X, children of X and parents of children of X(Dalmao, 2003). Value elimination is an algorithm for Bayesian network inference(Bui & Jun, 2012). The idea of value elimination algorithm is to use conditional independence to reduce the number of calculation combinations. That is also considered as one of the most important advantages of Bayesian network.

For example, by using those terminologies, if there is a n binary variable $X_1, X_2.....X_n$ and we want to computer probability of X_n equals 1, then in general we have to marginalize the probability over $n-1$ variables which is sum of 2^{n-1} elements. However, if conditional independence relationship can be represented in Bayesian network in Figure 1.

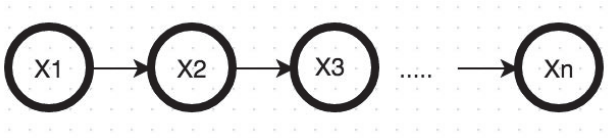

Figure 1. Random Chain Structure

Algorithm 1 Value Elimination

Input: A Bayesian network with variables $X_1, ..., X_m$ and an elimination order.
Initialize $noChange = true$.
for $i = m$ **to** 1 **do**
  Remove all the factors that mention X_i
  Multiply those factors, getting a value for each combination of mentioned variables.
  Sum over X_i
  Put this new factor into the factor set
end for

Each summation can be computed first by value elimination algorithm. Probability of X_n equals 1 can be computed with $n-1$ rather than 2^{n-1} summations.

2.2. Goodness of Fit Measure

Likelihood is used to test which Bayesian network has better goodness of fit. Generally, a higher likelihood or log-likelihood represents better Bayesian network. It means this data set is more likely to have occurred under this model.

Likelihood:$P(Data|Model)$

$$= \prod_i P(Data_i|Model) \quad (1)$$

$$= \prod_i \prod_j P(N_i^j|Parent(N^j), Model) \quad (2)$$

Log-Likelihood:$P(Data|Model)$

$$= log(\prod_i \prod_j P(N_i^j|Parent(N^j), Model)) \quad (3)$$

$$= \sum_i \sum_j log(P(N_i^j|Parent(N^j), Model))) \quad (4)$$

The two likelihoods are linked very simply. Basically, the log-likelihood is just the log of the likelihood calculation. However, which one gives a better way to measure Bayesian network will be experimented in later chapter.

2.3. Scoring Function

Due to overfitting issue, simple structure is preferred. In other word, a simpler network with higher log-likelihood is better. And in this case, another terminology named scoring function(Tang & Xu, 2014)is a way to measure with following arithmetic expression.

$Score = log(P(Data|Model)) - C * E$

Where E represents total number of elements in the CPT and constant C is a parameter to trade-off goodness of fit and simplicity of the network. Corresponding experiments designed for scoring function with constant C.

2.4. Bayesian Network With Different DAGs

There are various of different DAGs in Bayesian network. Three typical DAGs are chosen for this project which are No Edge, Random Chain and Best Tree.

No Edge is the simplest with smallest complexity among DAGs. In other word it is an empty graph.

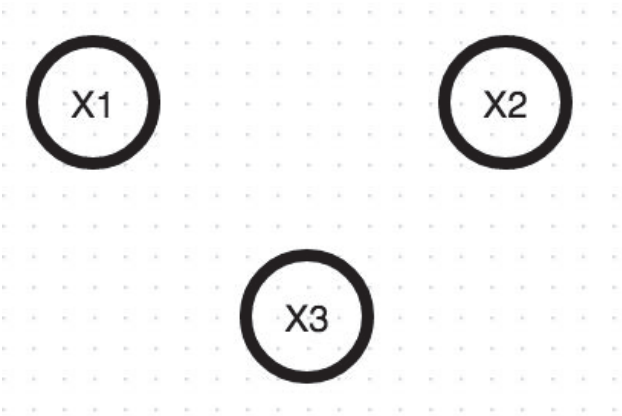

Figure 2. No Edge Structure

Random chain can be generated by choosing random ordering of the vertices, e.g., $V_1, ..., V_n$. Create a chain, i.e., Vi is the only child of V_{i-1}.

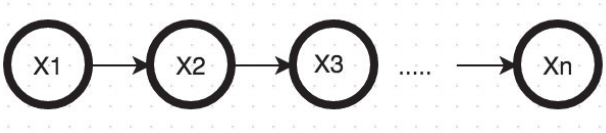

Figure 3. Random Chain Structure

Best tree is a fully connected graph G, i.e., there is al-

ways en edge between each pair of vertices. Weigh of the edges computed by mutual information between each pair of random variables by following formula. It also known as minimum weight spanning tree. $MI(A,B) = \sum P(A=a, B=b) log(\frac{P(A=a,B=b)}{P(A=a)*P(B=b)})$

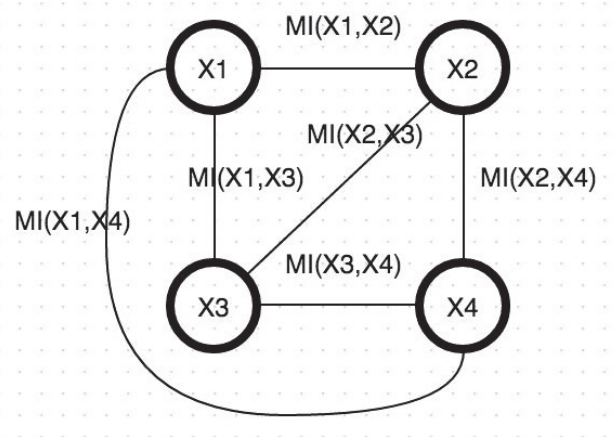

Figure 4. Best Tree Structure

A numbers of experiments are designed for structure evaluation against different kind of data sets by the same search strategy with different initial DAGs.

Search possible structures

- S - All possible DAGs with vertices as random variables
- A - add or remove edge, reverse direction
- Deterministic
- Goal - Stop when score reaches limit or after certain time.

In this case, suppose there is no missing data in data set.

2.5. Missing Data Handling

For a given data from Table 1, assume each experimental data is independent.

$$P(Data, H|Model)$$
$$= P(Data_3, H|Model) \quad (5)$$
$$= P(A|\ B, Model) \quad (6)$$
$$= P(\ B|A, Model)P(A|Model) \quad (7)$$

CPTs can be computed by counting with expected counts for H. A probability of H, will be generated by thoes method.

Table 1. Missing Data Example

Exp	A	B
1	1	1
2	0	1
3	H	0

Table 2. Likelihood and Log-likelihood

Data set	Log-likelihood	Likelihood
d1	-10.54193498725965	0.0
d2	-57.064187498416835	0.0
d3	-10320.88431900996	0.0

3. Experiments

It is assumed that all variables are binary variables. The value are either 0 or 1 for all experiments.

3.1. Compare Log-likelihood and Likelihood

The training data sets for this part are the txt files whose names start with CPTNoMissingData. The format of each data file is: The first line contains two numbers separated by a white space. The first number is the number of nodes in the Bayesian Network. The second number is the number of data in the file. Lets denote the number of nodes as K and the number of data as N. Each line at line 2 to K+1 represents a node of the Bayesian Network and its parents. Each line contains one or more words, separated by a white space. The first word represents the name of the node, while the rest represents the names of the nodes parents. For example, A B C means Node B and C are the parents of node A.

Each line at line $K+2$ to $K+N+1$ represents the data, in the same order the nodes are written. For instance, in CPTNoMissingData-d1.txt, each line of data represents the value of A B C. Table 2 are concluded from result file cpt-CPTNoMissingData-dx-log and cpt-CPTNoMissingData-dx. Last Line is the log-likelihood of the data given the Bayesian Network model.

```
Log base is 10 for this experiment.
```

3.2. Change Constant C in Scoring Function

The training data sets for this experiment are the txt files whose names start with noMissingData. The format of each data file is the same as the input format for experiment 3.1, but without the parents information,

i.e.: The first line contains two numbers separated by a white space. The first number is the number of nodes in the Bayesian Network. The second number is the number of 2 data in the file. The number of nodes are donated as K and the number of data as N.

The second line represents the names of the nodes, separated by a white space.

Each line at line 3 to $N + 2$ represents the data.

Results are generated by using different constant C. Result files for this experiment are result-ConstantCompare-bn-noMissingData Last Line consists of two number separated by a white space. The first number is the log-likelihood of the data given the Bayesian Network model. The second number is the score of the Bayesian Network. Bayesian network model structure and corresponding CPT for each node are presented above log-likelihood and score.

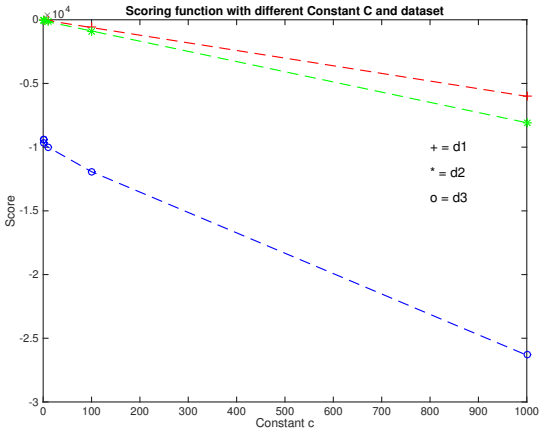

Figure 5. Score with different constant C.

3.3. Compare Different Initial DAGs

The training data sets for this experiment are the same files as experiment 3.2. And some proper constant C and threshold are set for different data set. Three minutes are time limitation for each data set and corresponding DAGs in this experiment. Table 3, 4 and 5 are generated from result file starting with noEdge, rChain and btree. Result format is the same as experiment 3.2. Same constant C is set for corresponding data set.

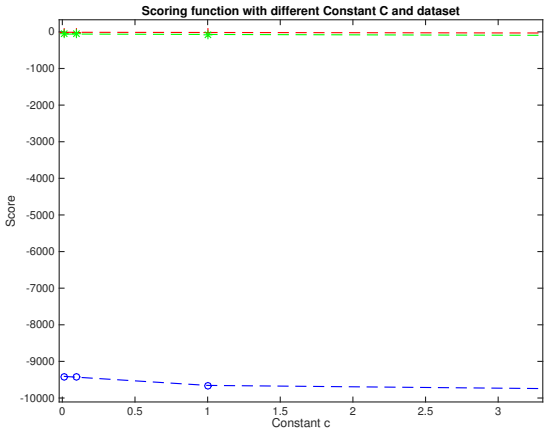

Figure 6. C with value of 0.01, 0.1, 1.

3.4. Calculate Missing Data Using Fill In Distribution

The training data sets for this part are files that start with someMissingData. The format of each data file is the same as the input format for experiment 3.2 and experiment 3.3, but each data input may have a value of 0, 1, or H1/H2/.../HM, where Hi means missing data and M is the number of missing data.

In this experiment, missing data separated by a white space, where M is the number of missing data. The first word is the name of the missing data (e.g., H1, H2, etc.). The second word is the probability that the missing data has value 1. For example, if $P(H1|Data, Model) = 0.6$, then the output will be: H1 0.6.

Table 6 shows log-likelihood and score for best tree structure with some missing data set.

By using of fill in distribution strategy, missing data are calculated in file bn-someMissingData for corresponding data set.

4. Analysis

4.1. Compare Log-likelihood and Likelihood

According to Table 2, likelihood is 0.0 for all three training sets. The reason why it is 0.0 is because double limitation in Java. However, likelihood should trend to be 0 for the increasing number of data or node in Bayesian network. Log-likelihood is therefore better in measuring Bayesian network. With a higher value of log-likelihood means the corresponding data set is more likely to have occurred under this model .

Table 3. No Edge Structure

No Edge	Log-likelihood	Score
d1	-10.6916372	-40.6916372
d2	-60.5681333	-110.568133
d3	-9457.04647	-9518.04647

Table 4. Random Chain Structure

Random Chain	Log-likelihood	Score
d1	-10.6916372	-40.6916372
d2	-60.5681330	-110.568133
d3	-9448.46799	-9517.46799

Table 5. Best Tree Structure

Best Tree	Log-likelihood	Score
d1	-10.6916372	-40.6916372
d2	-60.5681330	-110.568133
d3	-9454.58890	-9499.58890

Table 6. Score for Some Missing Data in Best Tree Structure

Data set	Log-likelihood	Score
d1	-10.775436	-25.775436
d2	-62.621634	-97.621634
d3	-9435.40280	-9594.40280

4.2. Change constant C in Scoring Function

Several positive constant C (0.01, 0.1, 1, 10, 100, 1000) are chosen for scoring function. The reason why C cannot be negative is, it will actually end up increasing the score as the Bayesian network get more complex. According to Figure 5 and Figure 6, it is hard to distinguish score between data set 1 and data set 2 when C is a small value by using same Bayesian network structure. Thus, larger C is suitable in simpler Bayesian network structure. And smaller C is better for complex structure, due to the consideration of accuracy.

4.3. Compare different Initial DAGs

According to Table 3, Table 4 and Table 5, all three different structure generated the same score for data set1 and data set2 within three minutes. Nevertheless, No Edge and Random Chain have less ram cost than Best Tree in processing with data set1 and data set2.However, with the increasing number of samples and node in Bayesian network, Best Tree got the highest score among three DAGs.

In conclusion, under proper constant C for scoring function, Best Tree is proper structure for complicated node relationship with Bayesian network, No Edge is better for simple node relationship and small data set. Random Chain is an alternative choice if node relationship is unknown.

4.4. Missing Data Handling

Fill in distribution is a good solution to missing data. The result file simply shows the probability that the missing data has value 1. After replacing 1 or 0 by using of probabilities from result file, result of Bayesian network would not change significantly. However, if the percentage of missing data is not very high, Bayesian network can still generate reasonable result by ignoring those missing data. In this case, cost of time and RAM will be reduced.

5. Conclusions

This work is aimed to evaluate performance of Bayesian network for different DAGs and explore Bayesian network property.

Experiments 3.1 and 3.2 are focusing on model evaluation properties including how the likelihood and log-likelihood measure of Bayesian Network differs as the number of training dataset changes and scoring function with different constant C. The value of log-likelihood will increase, as the number of node relationship increase. But simple Bayesian network structure is preferred, due to consideration of over-fitting. Score function is therefore involved in this case. Constant C could be treated as balance constant between accuracy and model complexity.

Experiment 3.3 shows the performances among different DAGs. A fully connected graph is better for complex CPT relationship whereas an empty graph suits simple data set very well.

In addition, fill in distribution strategy is well explained with support data set in Experiment 3.4.

Furthermore, there are some limitations and future considerations for this project. The details are listed below.

- Should use big decimal instead of double in Java. In this case, value of likelihood would not be 0.0.
- Should use natural log instead of log base 10. It might be more reasonable.
- Time limitation should be smaller for simple data

set. More straightforward sight in comparing different structure.

- Should have implementation on ignoring strategy.

In conclusion, all above discussed properties and various structures make Bayesian networks become increasingly being used to model environmental systems, particularly in concluding information from different data sources and handling missing data and uncertainty.

References

Bui, A. T. and Jun, C. Learning bayesian network structure using markov blanket decomposition. *Pattern Recognition Letters*, 16(33):2134, 2012.

Dalmao, S. E. *Value elimination: A new algorithm for bayesian inference*. PhD thesis, Department of Computer Science, University of Toronto, 2003.

Koller, D. and Friedman, N. (eds.). *Probabilistic graphical models: Principles and techniques*. MIT Press, Cambridge, 2009.

Tang, Y. and Xu, Z. A score based approach towards improving bayesian network structure learning. pp. 39–44, 2014.