Bayesian Network Parameter Learning Algorithm for Target Damage Assessment

Xiuli Du, Guichuan Fan*, Yana Lv and Shaoming Qiu
School of Information Engineering, Dalian University, Dalian, Liaoning, 116622, China
*Corresponding author’s e-mail: 13654918696@163.com

Abstract. Aiming at the problem that the existing methods of target damage assessment based on Bayesian network mainly determine the structural parameters of Bayesian network by giving conditional probability tables based on expert experience, which results in too subjective and having large errors in the evaluation results, an improved learning algorithm of conditional probability tables of Bayesian network is proposed in this paper. We divided the E step of the EM algorithm (Expectation Maximization Algorithm) into three steps. Firstly, the range of the missing variable is determined by expert experience, then the Gibbs sampling algorithm is used to complete the sample set, and finally the sample is weighted. The proposed algorithm is compared with EM algorithm, Gibbs algorithm, EM and Gibbs algorithm. The experimental results show that the proposed algorithm has good stability and high precision.

1. Introduction
Target damage assessment is an important and key step to achieve a precise strike in modern combat systems. In traditional decision-making, there are neural network method and fuzzy comprehensive evaluation method to assess target damage. Bayesian network method is the most widely used method for target damage assessment [1-2]. However, the existing conditional probability table of Bayesian network is mainly based on expert experience. Because the expert experience has strong subjective, it is easy to introduce the judgment error [3-5], and the damage assessment result is different according to the conditional probability table established by different experts.

At present, the most classical Bayesian network parameter learning algorithms mainly include maximum likelihood estimation method MLE (Maximal Likelihood Estimate) [6], EM algorithm [7-8] and Gibbs sampling algorithm [9-10]. MLE can achieve better parameter learning only when the sample set is completely. The disadvantage of EM algorithm is that the calculation of expectation in step E is more complex, and it can work normally when the Bayesian network is small in scale. But when the network structure is increasing, the learning efficiency will be greatly reduced because of the complexity of calculation, which will eventually lead to the paralysis of the algorithm. The Gibbs sampling algorithm is lacking in convergence, and it takes a lot of time in the application.

In this paper, we mainly aim at the problem that the results of damage assessment based on the conditional probability table given by experts’ experience have strong subjective and large errors, and at present, Bayesian network parameter learning algorithm has problems such as high computational complexity or slow convergence rate. At the same time, considering at the practical application of target damage assessment, the learning sample set often has different degrees of data loss, A Bayesian network parameter learning algorithm combining expert experience, classical EM algorithm and Gibbs sampling algorithm is proposed.
2. Bayesian network parameter learning

Bayesian network parameter learning is divided into missing value learning and non-missing value learning. The difference between the two is mainly the data set used for learning is whether complete. In practical applications, the data set of Bayesian network is often incomplete. Therefore, this paper mainly studies the Bayesian network parameter learning algorithm when the missing of sample value.

2.1. Classic EM Algorithm

The EM Algorithm is used in statistics to find probabilistic models that depend on unobservable implicit variables, which solves the parameter learning problem with hidden variables based on the maximum likelihood estimation.

First, let the Bayesian network structure be \( N = (G, \Theta) \), One of the data sets is \( D = (d_1, d_2, ..., d_m) \). These \( n \) samples are independent of each other. Any data \( d_i \) may be composed of a missing value \( x \) and an observable variable \( y \). At this point, the log likelihood function contains \( n \) probability distribution model parameters as below:

\[
L(\Theta) = \sum_{i=1}^{n} \log \sum_{x} P(X = x, Y = D)
\]  

(1)

Because there is a missing value variable, the likelihood function cannot be solved directly, introducing the probability distribution function \( Q(X = x_i | Y) \) about the missing value variable \( X \) here. It represents the probability under the condition \( X = x_i \) that the variable \( Y \) can be observed. For discrete random missing variables, there is \( \sum Q(X = x_i | Y) = 1 \).

The EM algorithm first determines the expected value of the log likelihood function from \( Q(X = x_i | Y) \). Then modify the value of \( Q(X = x_i | Y) \) so that the expected \( L \) of the log likelihood function reaches the maximum value (ie, M step), and repeated iterative E step and M step to get the final estimated value.

2.2. Gibbs sampling algorithm

The Gibbs sampling algorithm is a kind of random simulation sampling algorithm. It is a special sampling algorithm based on the MCMC method that proposed by German in 1984 [10]. First, you need to know the conditional probability of an attribute in the sample under all other attributes, and then use this conditional probability to get the sample value of each attribute. The process is as follows:

Given initial state \( X^{(0)} = (x_1^{(0)}, x_2^{(0)}, ..., x_d^{(0)}) \)

(1) get \( X_1^{(i+1)} \sim \pi(X_1 | x_2^{(i)}, ..., x_d^{(i)}) \)

(2) get \( X_2^{(i+1)} \sim \pi(X_2 | x_1^{(i)}, x_3^{(i)}, ..., x_d^{(i)}) \)

……

(n) get \( X_d^{(i+1)} \sim \pi(X_d | x_1^{(i)}, x_2^{(i)}, ..., x_{d-1}^{(i)}) \)

A sample of \( X \) can be obtained by 1~n steps, and then 1~n can be repeated to output a sample set that is sampled. Finally, the maximum likelihood estimation of Bayesian network parameters is obtained from the complete sample set.

3. Bayesian network parameter learning algorithm based on EM algorithm

Applying the expert experience to the EM algorithm in this paper can make the initial process of solving the missing value variable into the convergence curve faster. Applying the Gibbs sampling algorithm to the EM algorithm can make the missing value variable solve more quickly.

Set the Bayesian network structure to \( N = (G, \Theta) \), One of the data sets is \( D = (d_1, d_2, ..., d_m) \), \( X = (x_1, x_2, ..., x_n) \) is a set of missing values. When the EM algorithm performs the t-th iteration, given the observable variable set \( Y \) and the known parameter and the conditional probability that the missing value a is \( X_i \) certain value is \( P(X_i = x | \Theta') \). The improved Bayesian network parameter learning
algorithm takes the EM algorithm as the skeleton and divides the E step of the EM algorithm into three steps:

E1: Approximate determination of the likely range of a missing value variable through expert experience;
E2: Randomly extract \( x_1, x_2, \ldots, x_i \) by conditional probability \( P(X_i = x_i | \theta') \) to complete the sample data;
E3: Assign weight to each sample after completing the sample.

The following three steps of E1, E2 and E3 are introduced in detail:

In the E1 step, the conditional probability parameter of each missing node of the Bayesian network is recorded as \( \theta \). Here we assume that the prior probability \( P(\theta) \) obeys the normal distribution, according to the expert's prior knowledge distribution characteristics, the normal distribution function obeyed by \( p \) is calculated, and the value range of the missing node is determined.

In the E2 step, the Gibbs sampling algorithm is used to randomly sample the normal distribution interval obtained in the E1 step, and the probability value of the sampled sample is recorded as \( 12, \ldots, i p_p \). Normalize the sample probability to \( 12, \ldots, i p_p \), and \( 1 = \sum_i p_i \). Take the random variable \( \lambda \), which obeys the interval distribution of \([0, 1]\), as the auxiliary judgment tool to select the sampling value, and the sampling result of \( X_i \) is:

\[
X_i = \begin{cases} 
  x_i, & 0 < \lambda \leq p_i \\
  x_i, & \sum_{k<i} p_k < \lambda \leq \sum_{k<i} p_k \\
  x_i, & \lambda > \sum_{k<i} p_k 
\end{cases}
\] (2)

In the E3 step, the corresponding weight is added to the missing node, and the weight of the observable variable is taken as 1. Update the corresponding weights while updating the missing nodes each iteration, and patches each missing node sequentially to get the matrix \( \sum_i p_i = 1 \). Take the random variable \( \lambda \), which obeys the interval distribution of \([0, 1]\), as the auxiliary judgment tool to select the sampling value, and the sampling result of \( X_i \) is:

\[
\theta_{\theta'}^{i+1} = \begin{cases} 
  \theta_{\theta'}^i, & \text{if } \sum_{k=1}^i \theta_{\theta'}^{jk} > 0 \\
  1/r_i, & \text{if } \sum_{k=1}^i \theta_{\theta'}^{jk} \leq 0
\end{cases}
\] (3)

Where \( \theta_{\theta'}^{jk} \) represents the weight when \( x'_j = k \) and the parent node of \( x'_i \) is \( \pi(x'_i) = j \).

The specific pseudo-code of Bayesian network parameter learning algorithm based on EM algorithm is as follows:
Bayesian network parameter learning algorithm input: $G$ -- Bayesian network structure;  
$D$ -- Sample set;  
$\delta$ -- Convergence threshold;  

Output: Parameter estimation of the network  
(1) Conditional probability table value of randomly initializing Bayesian network  
(2) Give the scope of the value of the missing variable through expert experience  
(3) Set the convergence threshold $\sigma$ of Gibbs sampling and start iteration  

(4) for $i=0 : \sigma + n - 1$  
\[ x_{i+1}^{i} \] is sampled according to the conditional probability distribution $P(x_i | x_1, x_2, ..., x_i)$, and the transition probability is assigned to the weight;  
\[ x_{i+1}^{i} \] is sampled according to the conditional probability distribution $P(x_i | x_1, x_2, ..., x_i)$, and the transition probability is assigned to the weight;  
......  
\[ x_{i+1}^{i} \] is sampled according to the conditional probability distribution $P(x_i | x_1, x_2, ..., x_i)$, and the transition probability is assigned to the weight;  
......  
\[ x_{i+1}^{i} \] is sampled according to the conditional probability distribution $P(x_i | x_1, x_2, ..., x_i)$, and the transition probability is assigned to the weight;  

(5) Sample set for output sampling completion:  
\[
\begin{bmatrix}
(x_1, x_2, ..., x_n) \\
(x_1^{i+1}, x_2^{i+1}, ..., x_n^{i+1})
\end{bmatrix}
\]

(6) Calculate the parameters by the following formula by taking the complete sample set and weights:  
\[
\theta_{ij}^{i+1} = \begin{cases} 
\frac{\omega_{ij}^{i}}{\sum_{k=1}^{n} \omega_{ik}^{i}}, & \text{if } \sum_{k=1}^{n} \omega_{ik}^{i} > 0 \\
\frac{1}{\epsilon}, & \text{if } \sum_{k=1}^{n} \omega_{ik}^{i} \leq 0 
\end{cases}
\]

Iteration termination condition: When $|\theta_{ij}^{i+1} - \theta_{ij}^{i}| < \delta$, the iteration is stopped.  

4. Algorithm Simulation and Performance Analysis  
The learning network used in this experiment is a battalion level combat overall target damage Bayesian network structure, as shown in the figure. Because the combat data is difficult to obtain, the experimental sample set is approximated by two experts, the sample size is 10, 20, 30, 40 and all the samples are small samples, and the degree of variable loss is 10% and 30% respectively.
the extent of damage to the command post
degree of damage to command vehicle
damage degree of traffic hub
damage degree of Main Battle Tank
degree of damage to chariots
damage degree of Antitank Missile

Command function
Transportation function
Fire function

Overall combat Functions of the Enemy

Fig. 1 Target Damage Bayesian Network Architecture

In order to describe the accuracy of parameter learning more objectively, this paper compares the accuracy and convergence speed of classical EM algorithm, Gibbs sampling algorithm and EM combined Gibbs algorithm by calculating KL divergence and Euclidean distance between learning value and real value, the Euclidean distance formula between two points in the n-dimensional space is given below:

\[ d = \sqrt{\sum_{i=1}^{n}(x_i - \bar{x}_i)^2} \]  \( (6) \)

Where \( x_i \) and \( \bar{x}_i \) are calculated and true values, respectively.

The KL distance is used to describe the difference between two probability distributions of the same variable. The corresponding solution formula is as follows:

\[ KL(p \mid q) = \sum_{x} p(x) \log \frac{p(x)}{q(x)} \]  \( (7) \)

Where \( q(x) \) and \( p(x) \) are calculated and true values, respectively.

The expert experience value of the overall Bayesian network damage assessment modeling and generating samples of the battalion combat unit is taken as the true value, and the results of Bayesian network parameter learning algorithm learning are taken as the calculated values and solved by substituting them into the formula respectively.

Firstly, the convergence speed of the algorithm in small samples is investigated, and the experiment conducted parameter learning on samples with a sample size of 40 and a deletion degree of 10%. The number of iterations was 30, and the KL divergence and Euclidean distance between the parameter estimation value and the real value were calculated. The variation curves of KL divergence and Euclidean distance with the number of iterations are given below when the sample size of this algorithm is 40 and the loss degree is 10%.
It can be clearly seen from Fig. 2 and Fig. 3 that with the increase of the number of iterations, the parameter learning result of the sample of the algorithm in this paper converges rapidly and finally converges to a fixed value. The conditional probability table corresponding to the convergence value is the parameter learning result of the sample, so the convergence performance of the algorithm in this paper is good under the small sample.

The graph of the relationship between the number of iterations of the algorithm in parameter learning and the sample size and the degree of data loss is given below.

It can be seen from the figure that when learning with small samples, the number of iterations of the algorithm starts to increase with the increase of the total sample size, which is related to the increase of the complexity of data processing when the algorithm is actually executed. As the total sample size increases, the number of iterations of the algorithm decreases, because the number of iterations decreases as the number of observable variables increases.

In order to illustrate the learning accuracy and time performance of this algorithm, this experiment compares the learning results of classical EM algorithm, Gibbs sampling algorithm, EM+Gibbs algorithm and algorithm of this paper with the Euclidean distance and KL divergence, as shown in the figure.
As can be seen from Fig. 5 and Fig. 6, when the training sample set is the same, the Euclidean distance and the KL divergence decrease as the sample size increases, and the larger the sample size, the closer the final learning result of the four algorithms is to the true value. Among the four algorithms, the learning result of the algorithm in this paper is the most accurate.

5. Conclusion
This paper takes target damage assessment as the background, and aims at the problem that the existing target damage assessment methods based on Bayesian network mainly determine the structural parameters of Bayesian network through the conditional probability table given by expert experience, which leads to the large error of assessment results, an improved Bayesian network conditional probability table learning algorithm is proposed. Firstly, the constraint condition of the missing node is given by expert experience, then the E step of the classical EM algorithm is improved, and then combined with the Gibbs sampling algorithm to solve the Bayesian network structural parameters. The simulation results show that the parameter learning results of this method can be improved on the existing basis and the accuracy is improved.

References
[1] Li, Q.R., Shi, Y.B., Chen, Z.H., et al. (2018) Damage Assessment Model of Civil Airfield in an Air Raid Based on Dynamic Bayesian Network. Journal of Ballistics, 30: 93-96.
[2] Zhong, C.Y., Shu, J.S., Wu, J., et al. (2017) Status and development of target damage assessment technology. Aerodynamic Missile Journal, 34: 68-72+77.
[3] Qu, W.J., Xu, Z.L., Yuan, Y.W. (2016) Air Intelligence Radar Battle Damage Assessment Based on Dynamic Bayesian Networks. Tactical Missile Technology, 23: 93-100.
[4] Ma, X.M., Ding, P., Yan, W.D. (2016) Warship-damage Assessment Based on Bayesian Networks. Ordnance Industry Automation, 35: 72-75.
[5] Sui, H.G, Hua, F., Fan, Y.D., et al. (2016) Road Damage Extraction from High-Resolution SAR Image Based on GIS Data and Bayes Network. Geomatics and Information Science of Wuhan University, 41: 578-583.
[6] Grossman, D., Domingos, P. (2004) Learning Bayesian Network Classifiers by Maximizing Conditional Likelihood. In: Proceedings of the 21st International Conference on Machine Learning, New York. pp. 46-53.
[7] Steffen, L., Lauritzen. (1995) The EM algorithm for graphical association models with missing data. Computational Statistics and Data Analysis, 19: 90-95.
[8] Reed, E., Mengshoel, O.J. (2014) In: Bayesian network parameter learning using EM with parameter sharing. Proceedings of the Eleventh UAI Conference on Bayesian Modeling Applications Workshop, Sun SITE Central Europe. pp. 48-59.

[9] Wang, S.C., Leng, C.P., Du, R.J., (2009) Noise Smoothing in Learning Parameters of Bayesian Network. Journal of System Simulation, 21: 5053-5056+5060.

[10] Bouchardcote, A., Doucet, A., Roth, A., (2017) Particle Gibbs split-merge sampling for Bayesian inference in mixture models. Statistics, 18: 46-50.

[11] Zhang, Y. (2017) Parameter learning of Bayesian network based on the financial data of stock. Journal of Science of Teachers’ College and University, 37: 23-27+31.

[12] Cao, R.S., Ni, S.H., Zhang, P., et al. (2016) EM Parameter Learning Algorithm of Bayesian Network Based on Cloud Model. Computer Science, 43: 194-198.