The Hybrid parameter learning algorithm development for dynamic Bayesian network in the context of the Metropolis — Hastings approach

T V Azarnova and P V Polukhin
Voronezh State University, Voronezh, Russia
E-mail: ivdas92@mail.ru

Abstract. Bayesian networks are an efficient tool for analyzing complex stochastic processes. The generic methods of analysis based on Bayesian networks are probabilistic inference procedures that aimed at solving filtering, prediction and smoothing problems. An essential stage in the probabilistic inference methods implementation is network structure and parameters learning carried out on the special training samples obtained as a result of passive observation of the analyzed process and/or active experiment. Effectiveness of the Bayesian networks parameters learning algorithms directly depends on the preprocessing and organization of the training sample processing, which optimizes the classical learning algorithms capabilities. In the context of research introduced in this article, a hybrid algorithm for the parameters learning of dynamic Bayesian networks with observed and hidden variables has been developed. Algorithm allows increasing the efficiency of probabilities calculation procedure for the initial state and the probabilities of transition between the networks slices involving the variety of special methods for organizing training sample preprocessing. The questions of convergence of the proposed algorithm are also considered.

1. Introduction
Static Bayesian networks are directed acyclic graphs whose vertices are random discrete or continuous values, and arcs reflect conditionally probabilistic relationships between the vertices. Only Bayesian networks with discrete random variables will be considered in this paper. An arc directed from vertex $y$ to vertex $x$ denote that vertex $y$ is the parent of vertex $x$ and ($y \in Parents(x)$). If there are no arcs between $y$ and $x$, the assumption of conditional independence of these vertices is fulfilled. From the conditional independence hypothesis for Bayesian networks, the following statement is true — each node $x$ at known values of its parents $Parents(x)$ does not depend on any set of vertices $X$, such that $X \cap Parents(x) = \emptyset$. The formal representation of the conditional independence hypothesis for the Bayesian network is as follows

$$P(X_i, Y|Parents(X_i)) = Parents(X_i|Parents(X_i))P(Y|Parents(X_i)).$$  \hspace{1cm} (1)

To determine the case of conditional independence of some vertex $x$ from all other nodes of the network, it is necessary to determine its Markov blanket $X^x$. A Markov blanket is a set that includes the vertex itself, its parent, child vertices, and the set of parents of child vertices.
Each variable \( x \) of the Bayesian network is concerned with a conditional probabilities distribution, which for the case of a discrete domain values, is described by a table of conditional probabilities (CPT). CPT determines probabilistic relations between the variable \( x \) and the set of parent nodes \( 	ext{Parents}(x) \). The total joint probability distribution for the Bayesian network has the next form:

\[
P(x_1, x_2, \ldots, x_n) = P(X_1 = x_1, X_2 = x_2, X_n = x_n) = \prod_{i=1}^{n} P(x_i | \text{Parents}(X_i)). \tag{2}
\]

The common way for building Bayesian network is based on a chain rule, which can be represented as follows:

\[
P(x_1, \ldots, x_n) = P(x_n | x_{n-1}, \ldots, x_1) \times P(x_{n-1} | x_{n-2}, \ldots, x_1) \times \cdots \times P(x_2 | x_1) P(x_1) = \prod_{i=1}^{n} P(x_i | x_{i-1}, \ldots, x_1, \ldots, x_1). \tag{3}
\]

Dynamic Bayesian networks are a sequence of static Bayesian networks (slices of a dynamic Bayesian network) taken in chronological order. Dynamic Bayesian network has a set of nodes with probabilistic connections between slices, specified by transition probabilities. The main task of the parameters learning in dynamic Bayesian networks is to obtain the initial probability distribution for static networks describing each time slices, and the probabilities of transition between slices. On practice, hybrid algorithms have a particular interest. Their allow combining different mathematical learning methods in order to optimize the learning quality criteria. The article investigates a hybrid learning algorithm based on the expectation of maximization and Metropolis — Hastings with Markov chains methods. To optimize the training procedure, it supposes using the method of network structure transformation based on the junction tree. Application of the junction tree allows organizing a directed structure for learning the network and improves the network parameters learning process management. Classical algorithms for dynamic Bayesian networks parameters learning are based on the maximum likelihood estimation evaluation. Despite the widespread of these algorithms, adaptive tools to train dynamic Bayesian networks with complex structure, hidden variables, and a large amount of training samples are required. Such tools allow reducing the computational complexity of basic procedures and improve performance of dynamic Bayesian network construction process. Analysis of modern approach in such scientific area shows that the development of new algorithmic solutions make possible to improve the efficiency of learning procedures for the structure and parameters of dynamic Bayesian networks is quite important and requiring a wide variety of researches in common application areas.

2. Description of classical and modified learning parameters algorithms of the dynamic Bayesian networks

First of all, we will focus on the application of the junction tree construction algorithm for the directed Bayesian networks structure transformation. Inherently, the junction tree method allow to reorganize the generic structure of the dynamic Bayesian network in the form of the tree graph, which permit to simplify the evidence distribution in the network. The algorithm for constructing a junction tree for a dynamic Bayesian network includes the stages of domain, moral, and triangular graph formation. The entire process of the junction tree formation represents as the next sequence of procedures [5]:

- domain graph \( G_m \) moralization procedure. The domain graph is obtained from the original Bayesian network graph by eliminating information about the probability distribution of
vertices. In the process of moralization, a click is formed for each vertex of the $x_i$ graph. The click $C_{x_i} = \{\text{Parents}(x_i)\}$ assume a set of related parent nodes associated with the vertex $x_i$.

- triangulation procedure. A graph $G_t$ becomes triangular if it contains cycles from at most four vertices. The triangulation procedure consists from the elimination of graph nodes in a specific order. Each step of this algorithm selects any vertex $V$ with the minimum number of child vertices. Vertices related to node $V$ are united, and then resulting structure is added to the triangular graph $G_t$. The current vertex $V$ is removed from the graph $G_m$. The algorithm is repeated until the graph $G_m$ becomes empty.

- the procedure for constructing a junction tree. The node $V$ of the triangular graph $G_t$ with the minimum number of neighbor vertices is selected. The set of vertices including the selected vertex $V$ and all neighbor vertices is a clique $C_t$ of the triangular graph $G_t$, and the set of vertices of graph $G_t$ not included in the clique is the set of separators $S_t$. All vertices neighbor to $V$ that do not belong to the set of separators $S_t$ are removed. Then each node of the $C_t$ clique link to the separator $S_t$, and then $S_t$ to $C_j$, provided that $S_t$ is a subset of the clique $S_t \subseteq C_j$. As a result, the triangular graph $G_t$ is transformed into a click tree $T_c$. After that, introduce the potential term, which is a discrete function describing probability distribution $P(X_1|Y_1, Y_2, \ldots, Y_n) = \phi(X_1, Y_1, Y_2, \ldots, Y_n)$. The marginalization procedure can reduce the set of potential variables [6]:

$$\phi(X_1, X_2, \ldots, X_k, X_{k+1}, \ldots, X_n) = \sum_{X_k} \phi(X_1, X_2, \ldots, X_{k-1}, X_{k+1}, \ldots, X_n). \quad (4)$$

Junction tree allows conducting a network query efficiency. For the junction tree query procedure it is appropriate to use algorithm of the propagation distribution — Shefer — Shenoy algorithm [3, 4]. This algorithm performs a two-way approach of evidence distribution for all edges of the tree graph. Calculated value in the transmission between two nearby cliques $C_i$ and $C_j$, linked by the potential $S_{i,j} = C_i \cap C_j$, defined as the multiplication of all potentials associated with the clique $C_j$ during the process of obtaining all data for the clique $C_i$ except $C_j$. The results are summarized for all variables not included in the set of separators $S_{i,j}$:

$$e_{ij}(S_{ij}) = \sum_{C_i \setminus S_{ij}} \phi_{C_i} \prod_{k \neq j} e_{ki}(S_{ki}). \quad (5)$$

A disadvantage of the Shefer — Shenoy sampling algorithm is that the complexity of the algorithm is proportional to the total number of neighbor vertices $n$ for the current node $X_i$. Therefor algorithm needs to store two tables of evidence values for each separator $S_{i,j}$ and $S_{ji}$ in forward and back propagation. To obviate shortcomings, it is advisable to use the Hugin algorithm in the network polling process. As counter to Shefer — Shenoy algorithm, this algorithm implies a single marginal calculation. Structurally, Hugin algorithm is divided into two main stages: data collection and data distribution. Throughout the data collection process, the variable T is chosen, it distributes data to all nearby vertices and recursively repeat this operation for their neighbor vertices, except the current vertex itself. As a result, there is validity check only for the variable T, and allows to discard some parts of the data and start propagation. During the data propagation process, vertex T request neighbor vertices, if the propagation operation is feasible, then for each neighbor vertices data is propagated to all their neighbor vertices except vertex T.

The application of algorithms for construction and the junction tree sampling, allows increasing the efficiency of the dynamic Bayesian network parameters learning by optimizing evidence propagation procedures across the network. To refine the procedure for calculating the
initial probability distributions of network nodes, it is proposed to use various hybrid approaches. The adaptation of this approach considerably depends on the presence or absence case of hidden (unobservable) variables. The classical network parameters learning based on the maximum likelihood method. The maximum likelihood calculation construction upon the solution of the following optimization problem:

$$\theta_{MAP}^G = \operatorname{argmax}_\theta \log(P|\theta) + \log(P(\theta)),$$

where \( \log(P(X|\theta)) = L(G, \theta^G, D) = \sum i = 1^n \sum j = 1^q \sum k = 1^r N_{i,j,k} \ln \theta_{i,j,k}^G \) — is the likelihood logarithm, \( G \) — Bayesian network structure, \( \theta^G \) — the set of network parameters.

To determine the maximum likelihood values for the parameters \( \theta^G \), we differentiate \( L(G, \theta^G, D) \) by \( \theta^G \) and get the following equations as a result:

$$\frac{\partial L(G, \theta^G, D)}{\partial \theta^G} = \frac{N_{i,j,k}}{\theta_{i,j,k}^G} - N_{i,j} = 0 \Rightarrow \theta_{i,j,k}^G = \frac{N_{i,j,k}}{N_{i,j}}, \quad (7)$$

where \( N_{i,j} = \sum_{k=1}^{r_i} N_{i,j,k} = \{ l \in \{1, \ldots, N\}|\text{Parents}(X_i) = j \} \).

The maximum likelihood estimate has a frequency nature. Employment of the method in some cases when there are hidden variables becomes more complicated. This is stipulating that it is not possible to assign concrete value to the node with hidden variables and so training data for such node is missing. Procedure of learning with hidden variables uses the assumption that the hidden variables are independent from the observation variables. To overcome the presence of hidden variables limitations, the full data model \( Z = (X, H) \) (\( H \) — hidden variables, \( X \) — observed variables) is initially used. This model described by the density distribution \( f(Z|\theta) \) with the undefined parameter \( \theta \). Then the maximum likelihood logarithm for the Bayesian network may be written in integral form:

$$L(\theta|X) = \int f(X, H|\theta) dH. \quad (8)$$

As long as the function \( L(\theta|X) \) is differentiable and single-modal, the maximum likelihood estimate of the parameter \( \theta \) has the next form:

$$S(\theta|X) = \frac{\partial \log(L(\theta|X))}{\partial \theta} = 0. \quad (9)$$

If it is difficult to find an explicit solution to the equation, the parameter \( \theta \) estimated using Newton — Raphson algorithm

$$\theta^{t+1} = \theta^t + I^{-1}(\theta^t|X) S(\theta^t|X),$$

$$I(\theta^t|X) = - \frac{\partial^2 L(\theta|X)}{\partial \theta \partial \theta}. \quad (10)$$

If the logarithm \( L(\theta|X) \) is convex, the sequence of parameter values \( \theta^t \) converges to the maximum likelihood estimate \( \theta^* \) for the parameter \( \theta \). One possible method allowing to achieve convergence is the method of the score function, which makes possible to replace the observed information \( I(\theta^t|X) \) by the expected \( \Lambda(\theta) \)

$$\theta^{t+1} = \theta^t + \Lambda^{-1}(\theta^t) S(\theta^t|X), \Lambda(\theta) = E(I(\theta|X)|\theta) = - \int \frac{d^2 L(\theta|X)}{d \theta d \theta} f(X|\theta) dX. \quad (11)$$
Analyzing the Newton—Raphson algorithm and the score function method [9], it can be concluded that an augmentation number of parameters $\theta$ characterized by increasing time complexity. To negotiate such limitations of considered algorithms, the most adaptive and efficient learning algorithms are algorithms based on the expectation maximization method. These algorithms have quite good convergence. Convergence is achieved by increasing the value of the log-likelihood $L(\theta|X)$ at each iteration cycle of the algorithm. As long as the logarithm $L(\theta|X)$ is restricted, the sequence of the logarithms of $L(\theta_t|X)$ converges to a stationary value of $L(\theta|X)$. The convergence rate of the algorithm inversely proportional to the number of hidden variables in the Bayesian network. In the classical version, the algorithm replaces the hidden parameters with their estimates, and then performs a cyclic evaluation of the parameters to archive the essential process convergence.

With a convex likelihood function, Jensen’s inequality performs for any convex combination of variables:

$$f(\sum_{i=1}^{n} \alpha_i \xi_i) \leq \sum_{i=1}^{n} \alpha_i f(\xi_i).$$  \hspace{1cm} (12)

For the likelihood logarithm, Jensen’s inequality takes the form:

$$L(\theta) = \log\left(\sum_H q(H) \frac{P(X,H|\theta)}{q(H)}\right) \geq \sum_H q(\theta;H) \log\left(\frac{P(X,H|\theta)}{q(H)}\right),$$  \hspace{1cm} (13)

where $q(\theta;H)$ — is the distribution over all hidden variables,

$$q(\theta;H) : 0 \leq q(\theta;H) \leq 1, \sum_H q(\theta;H) = 1.$$  \hspace{1cm} (14)

Jensen’s inequality describes the lower bound of the likelihood logarithm $L(\theta)$. For the case where Jensen’s inequality turns to equality, we can obtain the following expression for the distribution $q(\theta;H)$:

$$q(\theta;H) = \frac{P(X,H|\theta')}{\sum_H P(X,H|\theta')} = P(X,H|\theta') = P(H|X,\theta).$$  \hspace{1cm} (15)

From this expression obtaining that $q(H)$ is a posteriori distribution over all hidden variables $H$ as long as observed variables $X$ and a fixed set of parameters $\theta'$ has given. At the maximization stage of the algorithm the following optimization problem has been solved [10]

$$\theta' = \arg\max_{\theta'} \left[ \sum_H q(\theta;H) \log\left(\frac{P(X,H|\theta)}{q(\theta;H)}\right) \right].$$  \hspace{1cm} (16)

In the course of solving the optimization problem, a situation may be arise when the function has a sufficiently large number of local extremes that leads to arise dependency output values from initial approximation and convergence process deceleration. To analyze convergence, the Kullbak—Leibler metric or the Akaike criterion is used.

To overcome the common expectation maximization algorithm challenges, using stochastic version of the maximization algorithm is required. The generic approach introduces in this algorithm permit to ‘shake’ observation variables values at every iteration step of the expectation maximization algorithm. Along with the stochastic version of expectation maximization algorithm, there are a number of another approaches that allow to improve the estimation of parameters without necessity to divide the parameters into several classes, but help obtain a sufficiently high accuracy of the approximation. One of the basic algorithms using this idea is a modified algorithm of expectation maximization using regularization [1]. The most suitable
The main point of this strategy is to maximize the sum of a certain number of regularizes and the likelihood logarithm, while the expectation stage remains unchanged:

\[ L(\theta) = \log(P(X|L(\theta))) + \beta R(\theta), \]

\[ \theta' = \arg\max_{\theta} \left( \sum_H q(H) \log \left( \frac{P(X,H|\theta)}{q(H)} \right) \right) + \beta R(\theta), \quad (17) \]

where \( \beta \) is a non-negative regularization coefficient. From current maximization expression with regularization the coefficient for maximum likelihood estimation represents as \( \beta R(\theta) = \log(P(\theta)) \).

In some cases the process of calculating \( q(\theta; H) \) at the expectation stage of the algorithm is quite complex, and the effectiveness of learning depends on the training sample completeness. To bypass these limitations, it is appropriate to amplify the expectation maximization algorithm with the Monte Carlo randomized method. The main idea of the algorithm is based on the comparison of the function \( q(\theta; H) \) with the function \( q'(\theta; H) \), whose value can be calculated according to a simulation model [7]:

\[ q'(\theta; H) = \frac{1}{N_m} \sum_{i=1}^{N_m} \log(f_\theta(H,X_i)), \quad (18) \]

where \( X = X_1, X_2, \ldots, X_{N_m} \) are independent random implementations of the observed variable \( X \) generated from the random distribution \( f_\theta(X|H) \). For large values of \( N_m \), the following approximation is valid

\[ q'(\theta; H) \approx q(\theta; H). \quad (19) \]

Further, considering the adaptation of the algorithm to the expectation and maximization stages of the algorithm, perform the search of the parameter \( \theta \) value on each iteration \( m+1 \) for the current iteration \( \theta^m \). This process can be specified by the following inequality [11]

\[ q'(\theta^{m+1}; H) \geq q(\theta^m; H), \quad \forall \theta \in \Theta. \quad (20) \]

At the maximization stage, the analyzed approximation leads to the appearance of an additional error \( \xi \)

\[ \xi^m(N_m) = \left| \int \log(f_\theta(X,H))f_\theta(H|X)\mu_H dH - \frac{1}{N_m} \sum_{i=1}^{N_m} \log(f_\theta(H,X_i)) \right|. \quad (21) \]

where \( \mu_H \) — median of random variable \( H \). Meanwhile the algorithm is still sensitive to the value of the parameter \( N_m \), even if the initial iterations do not require a large number of samples, each subsequent iteration requires an increase the number of samples in order to achieve the required level of algorithm accuracy and, as a result, leads to a decrease the error value \( \xi^m \).

Each of the considered expectation maximization algorithms have advantages and disadvantages, some algorithms allow to increase the estimation of parameters, but not take into account their initial distribution, others generate insufficient number of samples to achieve the required level of accuracy. To overcome these limitations, we introduce hybrid algorithms for learning the parameters of the Bayesian network based on the expectation maximization algorithm. One of these types of algorithms family is a stochastic maximization expectation algorithm with Monte Carlo sampling.
3. Hybrid algorithm for learning parameters of dynamic Bayesian networks based on the Markov Chain Monte Carlo method

The complete joint distribution for a dynamic Bayesian network, represented as a set of static Bayesian networks, taken in chronological order on the interval \([1, t]\), has the following form:

\[
P(X_{1:t}) = \prod_{i=t-1}^{t} \prod_{j=1}^{N} P(X_i^j | \text{Parents}(X_i^j)). \tag{22}
\]

The transition probability for time slices \(t - 1\) and \(t\) is calculated by the formula:

\[
P(X_{t-1}|X_t) = \prod_{j=1}^{N} P(X_t^j | \text{Parents}(X_t^j)). \tag{23}
\]

To obtain a complete joint distribution, as well as a transition model, a procedure for learning the parameters of a dynamic Bayesian network rest on an expectation maximization algorithm. If there are no hidden variables in the network, the learning algorithm is reduced to determining the probability distribution for each variable owing to the likelihood logarithm calculation.

To improve the expectation maximization algorithm statistical index and increase the accuracy of the generated probability distributions for the time slices, it is advisable to obtain several initial approximations, and confine to a single approximation, as in the generic form of this method build upon generation samples [2]. The application of samples from the generic training set, and then appreciate the likelihood of these samples.

Expectation maximization algorithm. To achieve this, it is necessary to generate a finite number of samples according to this distribution. The most common type of algorithm is the Metropolis — Hastings algorithm. This algorithm builds a Markov chain for each hidden parameter \(H\), generated during the execution of the Metropolis — Hastings algorithm. This algorithm builds a Markov chain for each hidden parameter \(H\) with stationary probability distribution \(P(H^t+1 | H^t, \theta)\), \(H^t \sim P(H^t+1 | H^t, \theta)\). This distribution satisfies the following equality

\[
\int P(H^t+1 | H^t, \theta) dH^t = 1. \tag{24}
\]

Metropolis found that the transition from the state of \(H^t\) to \(H^t\) occurs much more often, as a result the probability distribution \((H^t+1 | H^t, \theta) < 1\) was introduced, which appoint the excluding criteria for the sample \(H^t\), generated during the execution of the Metropolis — Hastings algorithm. Next consider the Metropolis — Hastings with the relation to the estimation of the parameters for Bayesian network with the expectation maximization algorithm.

The process of samples generation executes sequentially. Each next step generates a sample of \(H^t+1\). The process of generating \(H^t+1\) consists of two steps. At the first step, a random value \(H^t\) is generated from the distribution \(P(H^t+1 | H^t, \theta)\). On the next step we calculate the probability distribution \(\varphi(H^t+1 | H^t, \theta)\) taking into account the Hastings transformation [8]

\[
\varphi(H^t+1 | H^t, \theta) = \min \left( 1, \frac{P(H^t+1 | \theta) P(H^t | H^t, \theta)}{P(H^t) P(H^t+1 | H^t, \theta)} \right). \tag{25}
\]

From the last expression flow out that sample \(H^t\) will be accepted \(H^t+1 = H^t\) with probability \(\varphi(H^t+1 | H^t, \theta)\) and rejected \(H^t+1 = H^t+1\) with probability \(1 - \varphi(H^t+1 | H^t, \theta)\).
\( \varphi(H^{t+1}|H^t, \theta) \), it is necessary to fulfill the condition of the detailed balance. This condition results from the symmetry property

\[
P(H^t)\varphi(H^{t+1}|H^t, \theta) = P(H^{t+1})\varphi(H^t|H^{t+1}, \theta). \tag{26}
\]

Integrating the expression \( \varphi(H^{t+1}|H^t, \theta) \) by the parameter \( H^{t+1} \), we obtain the following equation

\[
\int P(H^t)\varphi(H^{t+1}|H^t, \theta) = \varphi(H^{t+1}|H^t, \theta). \tag{27}
\]

According to the latter equality, we can formulate a detailed balance condition for the Metropolis — Hastings algorithm

\[
P(H^{t+1})\varphi(H^t|H^{t+1}, \theta) = \min(P(H^{t+1})\varphi(H^t|H^{t+1}, \theta), P(H^t)) \varphi(H^{t+1}|H^t, \theta) \tag{28}
\]

\[
= \min(P(H^t)\varphi(H^{t+1}|H^t, \theta), P(H^{t+1})\varphi(H^t|H^{t+1}, \theta)) = P(H^t)P(H^{t+1}|H^t, \theta).
\]

The Markov chain used in the Metropolis — Hastings algorithm consider to the expression for \( \varphi(H^{t+1}|H^t, \theta) \) has the following form

\[
M = P(H^{t+1}|H^t, \theta) \min\left(1, \frac{P(H^t|\theta)P(H^{t+1}|H^t, \theta)}{P(H^t)P(H^{t+1}|H^t, \theta)}\right), H^{t+1} \neq H^t. \tag{29}
\]

The hybrid expectation maximization and Metropolis — Hastings algorithms creation procedure allow to include Metropolis — Hastings algorithm into expectation step with the aim of discarding mismatched samples and generating probabilistic distributions for all hidden variables \( Q(H) \). Employment of this approach makes it possible to learn a network with hidden variables just as a network with full observability. In this regard, the expectation maximization and Metropolis — Hastings algorithms can be considered as a unified algorithm for learning parameters of dynamic Bayesian networks. The completion criterion of the algorithm is approximation of the distribution \( \varphi(H^{t+1}|H^t, \theta) \) to the stationary distribution of \( H^t \) and \( H^{t+1} \) transitions.

At the maximization stage, the samples that have successfully passed the verification on the stage maximization and Metropolis — Hastings expectation algorithms are evaluated. The formation of conditional probability tables for all vertices of the dynamic Bayesian network implemented by searching maximum value of likelihood logarithm for every variable within dynamic Bayesian network.

The experimental part of the research supposes to use the water dynamic Bayesian network. We suppose that structure of the static network associated with dynamic Bayesian network has been known yet. This network has 32 nodes, 66 arcs, 10082 parameters and includes three time slices \( t-1, t \) and \( t+1 \). Training samples for the water network include about 1 GB of data for each time slice with high level of entropy, that allow to simulate different dynamic Bayesian network parameters learning behavior. The process of Metropolis — Hastings sampling originate step-by-step reduction of samples required to acquire optimal dynamic Bayesian network parameters estimation under transition from one neighbor slice to another. It is assumed that the procedure of learning network parameters depend on presence of conditional links between related time slices that collaborate in the probability distribution forming process for each slice of the dynamic Bayesian network.

To evaluate the proposed algorithm, we give a comparison with the standard algorithm of expectation maximization. The comparison of the algorithms allow to estimate quality and time complexity of each algorithm.
Figure 1. Estimation of the resource efficiency of the expectation maximization algorithm with Metropolis — Hastings algorithm.

From figure 1 taking that for dynamic Bayesian networks with a complex structure and a sufficiently large number of parameters the developed hybrid algorithm can significantly reduce the time spent on the network parameters learning, while the number of iterations of the algorithm required to achieve the desired value of the logarithmic function stay optimal. Application of the expectation maximization and Metropolis — Hastings algorithm allows to optimize expectation step of the expectation maximization algorithm and to form a sample that allows obtaining a good approximation of the probability distribution for hidden variables as a network component, regardless of the time slices total number. In this case, the Markov chain is generated for each variable that is a part of the network, including variables with transitive links between neighbor time slices. The Markov chain creates according to the distribution of all samples generated by the Monte Carlo method. Markov chain in this case is stationary.

4. Conclusion
Modern research in the learning dynamic Bayesian networks direction concert important aspects related to the modeling time probabilistic processes. The learning algorithms assign to obtain the initial probability distribution, transition and sensor models require determining the conditional probabilities tables of network nodes, building Markov chains to simulate transitive links between time slices, as well as the essential conditions definition for implementation of the logical inference procedure. The development of hybrid stochastic algorithms for learning parameters of dynamic Bayesian networks allows optimizing the training procedure by reducing training sample size and permits to adopt such algorithms to dynamic Bayesian networks with complex structure and large amount of parameters. The usage of the Monte Carlo method with Markov chains generation, especially the Metropolis — Hastings algorithm allows improving samples processing organization procedure by removing samples that do not adjust with evidence and reduces the probability of an error at the maximization stage of the expectation maximization algorithm. Adding regularization to the maximization stage of the developed hybrid algorithm allows increasing the accuracy of determining the maximum value of the likelihood logarithm by reducing the probability of falling into local optima. In this regard, the accuracy of generating desired full probability distribution for a dynamic Bayesian network with a fixed set of time
slices has been increased. The practical results of the research corroborate the effectiveness of the proposed algorithm in the process of learning the parameters of complex dynamic Bayesian networks, both in the mode of full and partial network observability.

5. References

[1] Dempster A, Laird N and Robin D 1997 *J. Of Royal Statistical Society* B 7 pp 1–38
[2] Perl J 1982 *Proc. American Association of Artificial Int. Nat. Conf.* B 1 pp 133–136
[3] Shenoy P P 1997 *J. of Approximate Reasoning* B 17 pp 239–263
[4] Shenoy P P 1986 *IEEE Expert* B 1 pp 43–52
[5] Maslennikov E D and Sulimov V B 2010 *Computational methods and programming* B 4 pp 94–107
[6] Sirotkin A V 2006 *SPIRAS Proceedings* B 3 pp 229–239
[7] Boudreau J F and Swanson E.S. 2018 *Applied Computational Physic* (Oxford: Oxford University Press)
[8] Brooks S, Gelman A, Jones G and Meng X 2011 *Handbook of Markov Chain Monte Carlo* (Boston: CRC Press)
[9] Winkler G 1995 *Image Analysis, Random fields and Dynamic Monte Carlo Methods* (Berlin: Springer)
[10] Kolmogorov A N 2005 *Selected works. Probability theory and mathematical statistics* (Moscow: Science)
[11] Tulupyev A L 2009 *Bayesian networks. A probabilistic logic approach* (Saint-Petersburg: Saint-Petersburg State University press)