Algebraic Bayesian Networks: a Frequentist Approach to Knowledge Pattern Parameters Machine Learning

Alexandr Tulupyev[a,b], Nikita Kharitonov[b]

[a] *Saint Petersburg State University, University Embankment, 7/9, St Petersburg, 199034, Russia*
[b] *SPIIRAS, 14-Ya Liniya V.O., 39, St Petersburg, 199178, Russia*

Abstract
Algebraic Bayesian networks are related to the class of probabilistic graphical models. As a machine learning model they are required to be trained on some data set. This work is dedicated to the frequentist approach to machine learning of a knowledge pattern as a local learning of the Algebraic Bayesian network. The theoretical explanation of approach is provided and the algorithm is described. The algorithm's pseudocode is presented, its theoretical complexity is calculated. Then an experiment is conducted and real estimates of the algorithm's implementation time of work are received.

Keywords 1
Machine learning, probabilistic graphical model, Algebraic Bayesian networks, frequentist approach

1. Introduction

One of the main tasks of machine learning is the training of a model. This task is also applicable for Algebraic Bayesian networks (ABN), as one of the machine learning models.

The goal of this work is to produce a frequentist approach to machine learning of Algebraic Bayesian network parameters. In this work, ABN is represented as a knowledge pattern.

To reach the goal of the work, the next steps are provided. First of all, the short overview of researches in machine learning and Algebraic Bayesian networks is presented in part 2. After that, an alternative ABN learning approach is considered in part 3 in order to give a fuller picture of this task in Algebraic Bayesian network theory. The essence of a frequentist approach to ABN parameters machine learning is described in part 4, meanwhile the algorithm's pseudocode for training is presented in part 5. The theoretical complexity of the provided algorithm is calculated in part 6 and estimates of algorithm's implementation time of work are shown in part 7.

2. Relevant works

Neural networks [1, 2, 3, 4, 5] and probabilistic graphical models are one of the machine learning models. The last can be separated into Algebraic Bayesian networks [6, 7], which are the subject of research, Belief Bayesian networks [8, 9, 10] and Markov chains[11, 12, 13].

Decomposition is one of the main ideas of probabilistic graphical models construction [7]. Each model is a set of smaller parts, which stores some information. In Algebraic Bayesian networks theory these objects are knowledge patterns [7]. The example of two types of graphical representation of ABN is on Figure 1.

Russian Advances in Fuzzy Systems and Soft Computing: selected contributions to the 8-th International Conference on Fuzzy Systems, Soft Computing and Intelligent Technologies (FSSCIT-2020), June 29 – July 1, 2020, Smolensk, Russia

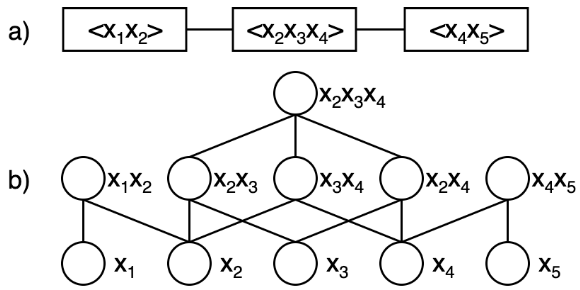

Figure 1: Algebraic Bayesian network:
a) as a set of knowledge patterns
b) as a set of conjuncts

As mentioned above, each knowledge pattern is a part of information stored in ABN. It can be presented as an ideal of conjuncts, what is shown on Figure 1. Each conjunct has scalar or interval estimates of its probabilities [6]. There are other representations of knowledge patterns, such as ideals of conjuncts or sets of quants, and matrices, which allows to convert the probability estimates of one representation to another [14].

The main operations in Algebraic Bayesian networks are maintaining consistency, a priori and a posteriori inference [15]. Maintaining consistency is the checking and correction of probability estimates in ABN [16]. A priori inference is a receipt of probabilistic formula probability, based on information in a network [6]. A posteriori inference is an update of probabilities in an Algebraic Bayesian network based on some new received information (evidence) and the calculation of probability of this evidence receipt. The paper [17] is dedicated to the sensitivity of this process.

During the creation of an Algebraic Bayesian network it is necessary to receive the probabilities of its elements. Most sets for network training have empty values. The work [18] describes the approach for receiving the ABN with interval probability estimates based on a set of data with missing values. This work is dedicated to receiving of the Algebraic Bayesian network with scalar probability estimates based on a similar data set.

3. Interval probabilities estimates and local training

The goal of machine learning of an Algebraic Bayesian network is to receive the probability distribution which represents the information in the data set as close as possible.

But one distribution wouldn't represent all of the information in case of missing values in the data set. They only allow you to get a particular, averaged information. One of the solutions in that case is to use the set of distributions. From the technical point of view that means the receipt of interval estimates of probabilities instead of saclar.

As it was mentioned above, the approach to obtaining an Algebraic Bayesian network with interval estimates was described in [18]. Here we will show how it works.

The data set is represented by the set of atoms, which has one of 0 (false), 1 (true) or * (missing value) values. On the first step all the values of conjuncts are received based on this set with the next logic:

0 & 0 = 0;
0 & 1 = 1;
1 & 1 = 1;
* & 0 = 0;
* & 1 = *;
* & * = *.

On the next step all the missing values are replaced by interval [0;1], all the 0 are replaced by [0;0] and 1 are replaced by [1;1].

Then the mean value for left and right bounds of intervals are calculated for each conjunct. And this is the result of training.

The example is on Figure 2.

x_1	x_2
1	0
*	1
0	*
1	1

x_1	x_2	x_1x_2
1	0	0
*	1	*
0	*	0
1	1	1

x_1	x_2	x_1x_2
[1;1]	[0;0]	[0;0]
[0;1]	[1;1]	[0;1]
[0;0]	[0;1]	[0;0]
[1;1]	[1;1]	[1;1]

x_1	[0.5;0.75]
x_2	[0.5;0.75]
x_1x_2	[0.25;0.5]

Figure 2: The example of receiving interval probability estimates

4. The receiving of a scalar probabilities as a result of a training

However it can be useful to receive the one probability distribution as a result of machine learning, which means the creation of an Algebraic Bayseian network with scalar probability estimates. The reason is that algorithms for analysing and processing ABN with scalar probability estimates have noticeably less computational complexity.

This work is dedicated to the process of training of an Algebraic Bayesian network with scalar probability estimates based on a data set with missing values. The Algebraic Bayesian network in this work is represented as a knowledge pattern.

The goal of the training is the same: the receiving of ABN which represents information from a data set as close as possible.

The main idea of a process is changing the estimates in the network step by step. For that the data set is presented as a set of quants. On each step the quant is taken from it and estimates are corrected based on this quant value.

If a quant has missing information, it is replaced by the subset of inconsistent quants. Each element in that subset receives the weight based on probabilities from the network. After that, the probabilities in networks are changed depending on the element's weights.

The algorithm is presented in pseudocode below.

5. The pseudocode of the knowledge pattern training algorithm

The input data for a training is a set of quants, which can be presented as an array. This set is divided in two parts: one part with quants which has missing values, and the second part is others.

The first part is used to receive the first knowledge pattern iteration. The probability of quant in knowledge pattern is a frequency of its occurrence in this sub data set.

For each quant in the second part all the quants which are not consistent to it are taken. Then weight is calculated and probability of quant in the knowledge pattern is changed.

```
//reading of input vector of quants
set = ReadInput()

// dividing input in two parts:
// with and without missing value
setWithMissVal, setWithoutMissVal = divide(input)

// Creating of Knowledge pattern with zeroes prob
//n - number of atoms
KP = arrayOfZeroes(n^2)
for q in setWithoutMissVal do
   // index(q) - index of q,
   // equal to binary representation of q
   KP[index(q)] += 1
KP = KP/len(setWithoutMissVal)


count = 0
for q in setWithMissVal do
   //taking all non-consistent quants to given
```

```
    quants = takeNonConsistentQuants(q)
    //array of 0 with len(KP)
    weigths = [len(KP)]{0}
    for i in len(KP)
        weigths[i] = KP[i]/sumProb(KP, quants)
    for i := len(KP)
        KP[i] = (count*KP[i]+weigths[i])/(count+1)
return KP
```

sumProb(knowledge pattern, quants) - sum of probabilities of given quants from given knowledge pattern

6. The algorithm's complexity

Let us consider the algorithm's complexity.

The first part of the algorithm (creating an initial knowledge pattern) has a linear time of work. This time is proportional to the number of quants in the data set without missing values.

In the second part of the algorithm each quant with missing values aligns 2^k quants without missing values, where k is a number of missing values.

Let us assume that all the types of quants with missing values in the data set have equal probabilities of occurrence (except quants without missing values — they are used in the first part, or quants with only missing values — they are non-informative for training).

Let n be the number of atoms in quant. The probability to gave k missing values in this quant is Figure 3:

$$\frac{C_n^k}{\sum_{i=1}^{n-1} C_n^i}.$$

Figure 3: Probability to gave k missing values in this quant

The denominator is a count of quants with k missing values. The numerator is a sum of these counts for all quants with at least one missing and one non-missing value.

Then the mean count of quants which are matched to quant with missing value is Figure 4:

$$\frac{\sum_{i=1}^{n-1} C_n^i 2^i}{\sum_{i=1}^{n-1} C_n^i}$$

Figure 4

The value of this formula for n = 2..10 is in the table 1.

Table 1

The mean count of quants matched to the quant with missing value

n	2	3	4	5	6	7	8	9	10
Value	2	4	7	12.6	22	38.4	66.4	114.4	196

The table 1 shows that the mean count of quants which are matched to the quant with missing values increases with the increasing number of atoms in quant.

In the theory of Algebraic Bayesian networks knowledge patterns with 2-4 atoms are considered. The complexity of analysing and processing of knowledge patterns increases exponentially with the increasing number of atoms. So, the best approach to handle a large knowledge pattern is to divide it into smaller ones and to work with the received ABN.

7. The algorithm’s implementation time of work

The goal of the experiment was to receive the dependency of the algorithm's implementation time of work from the number of atoms in quants..

The input data for the algorithm was a set of 1000 quants with the random count of missing values. The training was run for 1000 times. The new training data set was created on each turn to minimize the influence of data set peculiarity.

The result of an experiment is the dependency of training time on the number of atoms in data set quant. The result is shown in the form of a graph on Figure 5.

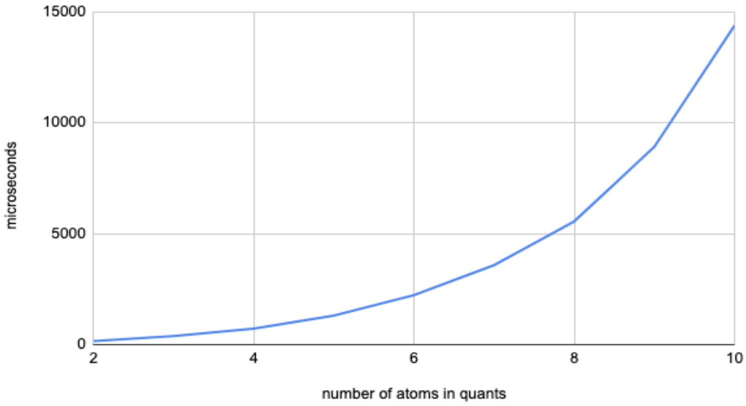

Figure 5: The example of receiving interval probability estimates

As shown on Figure 5, the time of work increases exponentially.

8. Acknowledgements

This work was carried out with the financial support of the Russian Foundation for Basic Research (project No. 18-01-00626) and within the framework of the project under the state order SPIIRAS No. 0073-2019-0003.

9. Conclusion

The frequentative approach to local machine learning of the Algebraic Bayesian network was provided in the work. The algorithm was described and pseudocode was provided. Then the algorithm’s theoretical complexity was calculated. The experiment was conducted and estimates of the algorithm's implementation time of work were received.

The received result is related to the research in the area of machine learning of the Algebraic Bayesian networks structure. The next steps are the research of conjugate priors as an application to the machine learning of the Algebraic Bayesian networks and the use of the results in the imitation of trees of social-engineering attacks and synthesis over the space of such trees of summary indicators of the security/vulnerability of users [19, 20] with taking into account the usage of incomplete, inaccurate information and expert knowledge with uncertainty.

10. References

[1] C. De la Parra, A. Guntoro, A. Kumar, Improving approximate neural networks for perception tasks through specialized optimization, Future Generation Computer Systems, 113 (2020) 597-606. doi:10.1016/j.future.2020.07.031.

[2] J. T. Hancock, T. M. Khoshgoftaar, Survey on categorical data for neural networks, Journal of Big Data, 10, 28 (1) (2020). doi: 10.1186/s40537-020-00305-w.

[3] M. Petrolo, E. Carrera, On the use of neural networks to evaluate performances of shell models for composites, Advanced Modeling and Simulation in Engineering Sciences, 7, 31 (1) (2020). doi: 10.1186/s40323-020-00169-y.

[4] Y. Wang, A. Wu, B. Li, Synchronization of bidirection multiple neural networks with impulsive coupling control, Advances in Difference Equations, 2020, 332 (1) (2020). doi: 10.1186/s13662-020-02796-6.

[5] J. Zhang, C. Huang, Dynamics analysis on a class of delayed neural networks involving inertial terms, Advances in Difference Equations, 2020, 120 (1) (2020). doi:10.1186/s13662-020-02566-4.

[6] A. L. Tulupyev, S. I. Nikolenko, A. V. Sirotkin, Bayesian networks: a logical-probabilistic approach, Saint Petersburg, Nauka, 2006.

[7] A. L. Tulupyev, A. V. Sirotkin, S. I. Nikolenko. Bayesian trust networks: probabilistic inference in acyclic directed graphs, SPb.: Publishing house of SPb University, 2009.

[8] J. G. Dai, J. Ren, W. C. Du, V. Shikhin, J. X. Ma, An improved evolutionary approach-based hybrid algorithm for Bayesian network structure learning in dynamic constrained search space, Neural computing & applications, 32(5) (2020). doi: 10.1007/s00521-018-3650-7.

[9] D. Delen, K. Topuz, E. Eryarsoy, Development of a Bayesian Belief Network-based DSS for predicting and understanding freshmen student attrition, European journal of operational research, 281(3) (2020). doi:10.1016/j.ejor.2019.03.037.

[10] Y. F. Zhao, J. J. Tong, L. G. Zhang, G. H. Wu, Diagnosis of operational failures and on-demand failures in nuclear power plants: An approach based on dynamic Bayesian networks, Annals of nuclear energy, 138 (2020). doi: 10.1016/j.anucene.2019.107181.

[11] J. A. R. Cruz, Sensitivity of the stationary distributions of denumerable Markov chains, Statistics and Probability Letters, 166, 108866 (2020). doi:10.1016/j.spl.2020.108866.

[12] J. L. Kirkby, D. H. Nguyen, D. Nguyen, A general continuous time Markov chain approximation for multi-asset option pricing with systems of correlated diffusions, Applied Mathematics and Computation, 386, 125472 (2020). doi:10.1016/j.amc.2020.125472.

[13] Y. Liu, F. Lyu, Kemeny's constant for countable Markov chains, Linear Algebra and Its Applications, 604 (2020) 425-440. doi: 10.1016/j.laa.2020.07.001.

[14] E. A. Malchevskaya, A. A. Zolotin, A. L. Tulupyev, Algorithms of a posteriori inference in algebraic Bayesian networks: refining the matrix-vector representation, Fuzzy systems and soft computing. industrial applications, Ulyanovsk: UlSTU, 2017.

[15] A. A. Zolotin, E. A. Malchevskaya, N. A. Kharitonov, A. L. Tulupyev, Local and global logical-probabilistic inference in algebraic Bayesian networks: matrix-vector description and sensitivity issues, Fuzzy systems and soft computing, Tver: TvGTU, 2017.

[16] A. G. Maksimov, A. D. Zavalishin, On Analytical Solutions to the Problems of Maintaining Local Consistency, Conference on Artificial Intelligence, Springer, Cham, 2020.

[17] A. A. Zolotin, A. L. Tulupyev, Sensitivity Statistical Estimates for Local A Posteriori Inference Matrix-Vector Equations in Algebraic Bayesian Networks over Quantum Propositions, Vestnik St. Petersburg university-mathematics. 51(1) (2018) 42-48. doi: 10.3103/S1063454118010168.

[18] N. A. Kharitonov, A. G. Maximov, A. L. Tulupyev, Algebraic Bayesian Networks: Naïve Frequentist Approach to Local Machine Learning Based on Imperfect Information from Social Media and Expert Estimates, Communications in Computer and Information Science, 1093 (2019). doi: 10.1007/978-3-030-30763-9_20.

[19] M. V. Abramov, A. A. Azarov, Identifying user's of social networks psychological features on the basis of their musical preferences, Soft Computing and Measurements (SCM), XX IEEE International Conference on. IEEE, 2017, 90-92. doi: 10.1109/SCM.2017.7970504.

[20] A. Khlobystova, M. Abramov, A. Tulupyev, An approach to estimating of criticality of social engineering attacks traces, Studies in Systems, Decision and Control, 199 (2019). doi: 10.1007/978-3-030-12072-6_36.