Journal of Applied Statistics

ISSN: 0266-4763 (Print) 1360-0532 (Online) Journal homepage: http://www.tandfonline.com/loi/cjas20

An intuitive risk factors search algorithm: usage of the Bayesian network technique in personalized medicine

Evgeny D. Maslennikov, Alexey V. Sulimov, Igor A. Savkin, Marina A. Evdokimova, Dmitry A. Zateyshchikov, Valery V. Nosikov & Vladimir B. Sulimov

To cite this article: Evgeny D. Maslennikov, Alexey V. Sulimov, Igor A. Savkin, Marina A. Evdokimova, Dmitry A. Zateyshchikov, Valery V. Nosikov & Vladimir B. Sulimov (2015) An intuitive risk factors search algorithm: usage of the Bayesian network technique in personalized medicine, Journal of Applied Statistics, 42:1, 71-87, DOI: 10.1080/02664763.2014.934664

To link to this article: http://dx.doi.org/10.1080/02664763.2014.934664

Published online: 08 Jul 2014.

Journal of Applied Statistics, 2015
Vol. 42, No. 1, 71–87, http://dx.doi.org/10.1080/02664763.2014.934664

An intuitive risk factors search algorithm: usage of the Bayesian network technique in personalized medicine

Evgeny D. Maslennikov[a,b*], Alexey V. Sulimov[a,b], Igor A. Savkin[a], Marina A. Evdokimova[c], Dmitry A. Zateyshchikov[d,c], Valery V. Nosikov[d] and Vladimir B. Sulimov[a,b]

[a]Dimonta Ltd., 15 Nagornaya Str., Build. 8, 117186 Moscow, Russian Federation; [b]Research Computer Center of M.V. Lomonosov Moscow State University, 1 Leninskie Gory, bldg. 4, 119991 Moscow, Russian Federation; [c]Educational and Research Medical Center of Russia President General Management, 21 Marshal Timoshenko Str., 121359 Moscow, Russian Federation; [d]Federal Research Clinical Center for specialized types of health care and medical technologies, 28 Orekhovy Boulevard, 115682 Moscow, Russian Federation

(*Received 31 January 2013; accepted 11 June 2014*)

The article focuses on the application of the Bayesian networks (BN) technique to problems of personalized medicine. The simple (intuitive) algorithm of BN optimization with respect to the number of nodes using naive network topology is developed. This algorithm allows to increase the BN prediction quality and to identify the most important variables of the network. The parallel program implementing the algorithm has demonstrated good scalability with an increase in the computational cores number, and it can be applied to the large patients database containing thousands of variables. This program is applied for the prediction for the unfavorable outcome of coronary artery disease (CAD) for patients who survived the acute coronary syndrome (ACS). As a result, the quality of the predictions of the investigated networks was significantly improved and the most important risk factors were detected. The significance of the tumor necrosis factor-alpha gene polymorphism for the prediction of the unfavorable outcome of CAD for patients survived after ACS was revealed for the first time.

Keywords: Bayesian networks; variable correlation; personalized medicine; naive network optimization; acute coronary syndrome; TNF gene polymorphism

1. Introduction

Bayesian networks (BNs) are probabilistic models representing relationships within a set of given variables via directed acyclic graphs with conditional probability tables (CPTs) in their nodes,

*Corresponding author. Email: em@dimonta.com

containing conditional dependencies between neighbor variables. The BN technique has proved to be efficient in many areas of science and applications [32]. Both interesting and extremely important field of BN applications are expert systems of personalized medicine [10,28]. The main idea of personalized medicine is to use the right treatment for every particular patient taking into account his medical status, personal genetics features, environment, lifestyle, medical and family history. The BN technique is well suited to deal with cases containing uncertainties [35]. The BN technique allows combining prior expert knowledge with experimental data, working with different types of patient data and detecting the relationship between variables. BN algorithms perform just as well on multi-core computing systems. There are numerous works including specialized software that apply the BN technique for prophylaxis and treatment of life-threatening diseases [1,2,5,7,8,23,26,37,40].

Apart from BN there are also Markov random field (MRF) probabilistic graphical models [4] which are based on the undirected graphs. They are very similar to BN, but MRF models need considerable computational resources for the calculations. MRF are used mainly in computer vision tasks [41]. Common tasks of extracting information from data sets (data mining) are classification, clustering, and regression [15]. Classification identifies to which a set of categories a new observation belongs on the basis of a training set of data. Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). Regression is the task of searching the function which models training data with least error estimating the relationships among variables. For application in personalized medicine, the classification method is the most important: there are two types of patients: diseased and healthy, our goal is to predict class of a new patient, who has no diagnosis. Most popular classification techniques are: neural networks, decision trees models, and support vector machines (SVM) [22]. The neural network is the mathematical model represented by a set of interconnected elements (neurons) which can compute output values from inputs. Their main disadvantages: it is impossible to give reasons for the results (black box) and they need large computational resources. The former is disagreeable for most medical scientists. The decision tree method is the graphical model represented by a treelike graph with conditions of graph path choice in the nodes. They have limitations for handling missing data, they are very sensitive to training data, and therefore this technique is not optimal for working with patient databases. SVM [42] is based on the search of the hyperplane dividing examples of different classes in the examples space. Complexity of the optimal hyperplane construction for the tasks with a large number of variables is the main problem of the SVM method. This feature restricts usage of SVM for learning and prediction for patients with a large number of clinical and genetic parameters contained in contemporary patient databases. So, BN probabilistic models are most convenient to be used for purposes of personalized medicine dealing as well with missing and erroneous data inherent in patient databases and be able to deal with large numbers of variables.

For purposes of personalized medicine, both for a better understanding of the origins of the disease and use of the most effective treatment, it is interesting to identify the subset of risk factors that occasionally cannot be captured by traditional statistical testing (for example, by the popular Pearson product–moment correlation coefficient [36] method, which cannot work properly with missing data or sensitivity analysis (SA) [6,24], which requires preliminary network topology optimization, and the result depends strongly on the network topology and threshold parameters). One technique solving this problem has been proposed in [3] and applied to the epidemiology problem.

In the given work we present an intuitive and simple algorithm of revealing most significant BN variables, which, in terms of personalized medicine, act as the most crucial factors determining the course of the target disease. This algorithm results in the BN optimization and improvement of the prediction quality within the naive network topology. We also consider

parallel implementation of the algorithm on multi-core computer systems, which makes it possible to perform the optimization of the very large BN including many thousand parameters per patient. We apply the developed method for solving one of the most critical problems of personalized medicine – prediction of an outcome of coronary artery disease (CAD) which is the most common cardiovascular disease and one of the main causes of death in the world [33]. Contemporary treatment of patients with ACS is based on the risk estimation strategy. There are numerous stratification scales including GRACE, PURSUIT, TIMI, etc. [19]. The main problem is that all of them estimate a patient's individual prognosis using only several risk factors and ignoring a lot of other important features. As distinct from traditional statistical methods, the application of the BN algorithm can improve the situation. In the present work the proposed algorithm was applied for the prediction for the outcome of CAD among patients who had already suffered acute coronary syndrome (ACS). Considerable improvement of the prediction quality was demonstrated and the significance of the tumor necrosis factor (TNF)-alpha gene polymorphism was revealed for the first time. Results have been compared with ones obtained using SA.

2. Material and methods

2.1 *Network's performance measure*

We compare the predictive quality of different BNs to identify the most significant variables. For this purpose we use the receiver operating characteristic (ROC) curves method [14,21,34]. The area under such curves (AUC) is used as the numerical measure of the predictive quality of a BN for a given training database. Here, we must specify that such a method measures predictive quality for the root variable which is responsible for the identification of the core of the problem under consideration; in the case of personalized medicine, for example, it is the variable responsible for the course of the considered disease. Also we assume that there are no missing data about the root variable in the given training database.

The application of AUC takes more computational resources in comparison with other scoring methods, such as the belief scoring function [9] or minimal description length method [30]. However, it is the only method we can use to compare networks with different number of variables adequately in as much as the other methods were primarily designed for network learning [9,30] or topology optimizing [17]. The results of scoring two networks with different number of variables using those methods depend strictly on the accepted coefficients.

In the present work we used the 'excluding-by-one' method (Appendix 1). In the table for building the ROC-curve (ROC-table) each line corresponds to a single line in a given database, and the first column of the table contains real values of the root variable in respective lines of the database, whereas other columns contain probabilities that the root variable could have one or another value. Every line in the ROC-table is obtained by the exclusion of a single line from the entire training database, learning the network on the database without the excluded line and calculating the probability for the root variable of the excluded line. That probability value is added to the ROC-table together with the real evidence that the root variable has. The 'excluding-by-one' method is slow (perhaps, the slowest AUC calculating method), but we use it for the sake of high accuracy and to minimize influence of randomness on the results of our calculations.

2.2 *Network optimization algorithm*

We propose the algorithm of the BN optimization with respect to its variables. For this purpose we include all given variables to the so-called naive network which has the simplest topology – the root variable is the parent one with respect to others, which are not interlinked, see Figure 1

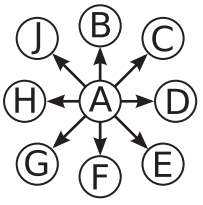

Figure 1. Example of the naive BN. *A* is the root variable.

(in some papers such a network is called 'native'). In spite of its simplicity, the naive BN topology is widely used for a broad spectrum of problems [27,32,38,39]. The use of naive topology appears to be one of the best options compared to more complex topologies [13,16]. Also, most of basic BN' algorithms (such as network learning or evidence collection [25,29] which are necessary for AUC calculation) perform faster with the naive topology than with any other one (brief description of those algorithms are presented in Appendix 1). Moreover, learning the naive BN, besides its simplicity, is precise, non-iterative and has no randomness (Appendix 1).

The naive topology suits well our goal – the network optimization with respect to its variables: elimination of any variable from the network (except the root one) does not change the network topology. For any other topology except the naive one, elimination of a variable from the network can change its topology, and the BN optimization with respect to its variables must involve the topology optimization, so the results will depend on the topology optimization algorithm used.

The presented algorithm is iterative (Algorithm 1). A description of the used functions can be found in Appendix 1, Table A1. At every iteration we calculate AUC of the entire network (line 3), then calculate AUC of every network that can be obtained by excluding (line 6) one variable from the entire network (except for the root variable, line 7), storing them in the previously allocated array (line 4). From the obtained networks we choose the one having the maximal value of AUC (line 8). If this value is larger than one of the entire network, we begin the new iteration of the algorithm with this newly obtained network (line 10). We stop the algorithm execution as soon there is no possibility to improve the AUC value by excluding a variable from the network (line 12).

Because our main purpose is not only to optimize the network but also to find out the most significant variables influencing the problem under consideration, we can continue to exclude variables from the optimized network as follows. Any variable can be removed from the network if the respective AUC decreases less than a specified small quantity. The least crucial variables are excluded first – their removal results in the smallest AUC decrease. In the present work, the specified small value has been taken equal to 0.003 (Section 3).

2.3 *Performing optimization on parallel computers*

The presented network optimization algorithm may be easily launched on several parallel cores to increase its performance. The most expedient way is to calculate AUC of a single network at every iteration of the optimization algorithm (Algorithm 1) at a single core. The most obvious advantage of such a method is the absence of numerous time-consuming data transfers between cores. Applying this method, we can obtain linear acceleration with an increase in the number of computational cores.

However, the above-mentioned parallelization method has several disadvantages. First, it is pointless to use more cores than the number of variables in the network under consideration. Second, if we perform a lot of iterations of the algorithm and the number of cores is comparable

Algorithm 1 Optimization algorithm for naive network

Function:	OPTIMIZENAIVENETWORK	
Input:	$\mathcal{B}$	naive Bayesian network with variables $\mathbf{X} = \{X_0, .., X_N\}$, X_0 is root variable
	$\mathcal{D}$	database with M data lines, first line has index 0
Output:	$\mathcal{B}$	optimized naive Bayesian network

1: *loop* ← **true**
2: **while** *loop* **do**
3: $a \leftarrow \text{AUC}(\mathcal{B}, \mathcal{D})$
4: ALLOCATE(**b**) {{Array of real numbers}}
5: **for** $i \leftarrow 1$ **to** N **do**
6: $\mathcal{B}' \leftarrow$ EXCLUDEVARIABLE($\mathcal{B}, i$)
7: $b_i \leftarrow \text{AUC}(\mathcal{B}', \mathcal{D})$
8: $\{b, j\} \leftarrow$ MAX(**b**) {{b - maximum value, j - index of this value in array **b**}}
9: **if** $b > a$ **then**
10: $\mathcal{B} \leftarrow$ EXCLUDEVARIABLE($\mathcal{B}, j$)
11: **else**
12: *loop* ← **false**

with the number of variables, many cores may become idle when variables are excluded. This leads to a significant efficiency decrease rather than acceleration.

In the case of the large training database and a network with a small number of variables, it is reasonable to parallelize the AUC calculating procedure (Appendix 1). This suggests learning network and calculating target probabilities after temporarily excluding every single data line at a single core. Thus, efficiency would not drop with the increase in the number of iterations of the optimization algorithm, but it needs a lot of data transfers between cores which may adversely affect the performance.

3. Example: prediction of the CAD outcome after the ACS

In this section we apply the developed method of the network optimization, revealing the most significant variables, for the problem of prediction of the outcome of CAD. We used the database related to patients who had already suffered ACS in the past, so they have an increased risk of the unfavorable outcome.

3.1 *Database, networks, and program*

In our research we used the database from the multicenter prospective observational Russian study performed from December 2004 until June 2009 in 16 medical centers of seven Russian cities: Moscow, Kazan, Perm, Chelyabinsk, Stavropol, Rostov-on-Don, and St. Petersburg. It involved 1193 patients hospitalized due to acute CAD (no more than 10 days before acute myocardial infarction (MI) with ST-segment elevation and no more 3 days before for Non ST MI and unstable angina) and survived for 10 days after the index event. The database containing more than 400 variables described the patient's status, demographic parameters, routine biomarkers' level, electrocardiographic and echocardiographic data, medical and family history, genetic data, and the follow-up results. In this work we did not use variables that were presented for less

than 40% patients in order to decrease statistical noise, so 204 variables only were included into naive BNs (Appendix 2, Table A2).

The study end points were: fatal and non-fatal MI, fatal and non-fatal stroke, unstable angina with hospitalization, sudden cardiac death, and non-vascular death.

We compare the results obtained from two different networks: the first one was designed for the prediction of the unfavorable outcome after ACS within the next half-year period (hereinafter called *net-06*) and the second one – for prediction of the unfavorable outcome after ACS within the next one and a half-year period (*net-18*). Initially, they have identical sets of variables, apart from the root variables. In our research we used only data for patients including men under the age of 74 and women under the age of 82, as there is a strictly different mechanism of disease for aged patients. The patients lost for follow-up between baseline and a six-month visit were excluded from the six-month analysis. So this analysis was performed for 980 patients. Similarly, by the 18th month the analysis of the data for 722 patients was available.

The program *SiMBA* ('Simple Multi-core Bayesian Analyzer') implementing the presented network optimization algorithm (Algorithm 1) applicable on several computing cores was developed with the use of *C++* programming language with *MPI* library. We used *GNU Compiler* version 4.6.3 and *OpenMPI* version 1.5. *SiMBA* can optimize network by maximal AUC or by the minimal size of network with restricted AUC decrease value (that is useful to reveal most important variables as it has been shown in Section 2.2). Because we deal with networks with a relatively large number of variables (comparable to the database size) we perform parallelization as calculation of AUC of a single network at a single core (as described above).

3.2 *Results and discussion*

After applying the optimization procedure to investigated networks we significantly increased the prediction quality of both of them. Thus, AUC increased from 0.6080 (for the initial network) to 0.8031 (for the optimized one) for *net-06* and from 0.6479 to 0.7614 for *net-18*. Numbers of variables have decreased significantly after optimization: from 204 (for both initial networks) to 48 and 54 for *net-06* and *net-18*, respectively. The optimized networks have only 19 common variables (Appendix 3, Table A3). This can be interpreted as follows: essentially different sets of factors affect the progress of CAD within the next half-year period after ACS and within the next 1.5-year period.

To reveal the most important factors determining the course of CAD after ACS, we proceeded to remove variables from the optimized networks further, excluding every variable that decreased AUC by less than 0.003. We used SA to choose the best threshold. From Figure 2 it is possible to calculate derivatives. The second derivative has the minimum value at the point with the threshold equal to 0.003. The process was stopped after an attempt to exclude any variable leading to a significant (larger than 0.003) decrease of AUC. As a result, we obtained 17 and 16 variables for minimal networks *net-06* and *net-18*, respectively. Comparing variables in the resulting minimal networks we concluded that the most important factors are the TNF-alpha gene polymorphism, history of MI and usage of spironolactone within 10 days of hospitalization. These were the only variables common for both networks after the post-optimization variables removal. The exclusion of these variables causes a significant AUC decrease (for both networks). The most intriguing of these factors is the TNF gene polymorphism. Importance of the TNF gene polymorphism for determining the course of CAD after ACS is revealed for the first time. This is a very interesting result as far as this protein plays a significant role as a promoter of inflammation [31]. Meanwhile, it is obvious that the history of MI also influences the risk of the unfavorable outcome of CAD. The need for diuretics (spironolactone) is one of rough characteristics of severity of the disease. So, the obtained results are not only useful in terms of personalized medicine, but also they comply with common sense considerations.

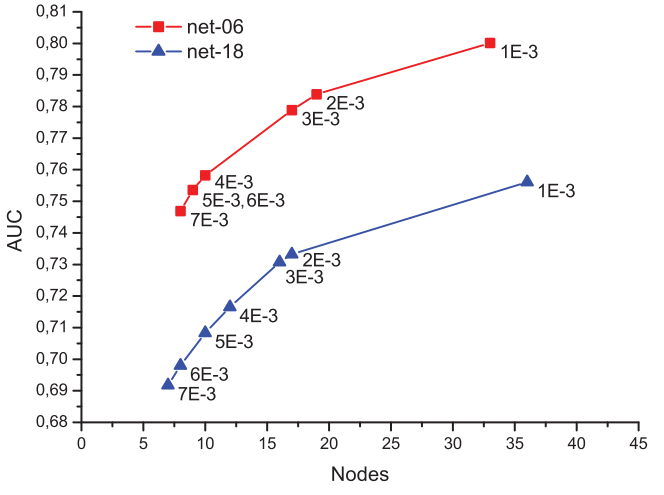

Figure 2. The dependence of AUC on the number of variables.

We applied the SA to our networks in order to verify these results. For this purpose we introduced a special type score for each variable of the network. This score reflects sensitivity of the root variable probability to the evidence of the given variable. This score is calculated as follows. First, we learn the given naive network using our entire database. Second, the root variable probability Z_j^i is calculated with evidence of value j of X_i variable with evidences of other variables stay missing. Then the score C_i of X_i variable is calculated as follows:

$$C_i = \sum_j |Z_j^i - P_0| \times E_j^i,$$

where the reference root probability P_0 (first value of two for definiteness) is calculated for the data line with all missing values, E_j^i is the number of evidences of value j of variable X_i. Actually, P_0 is a priori probability of the root variable to possess the first value (index '0'). Finally, all network variables are ranked with respect to their scores. The variables in the top have the largest score, and the root variable probability is the most sensitive to the top variables. This analysis was applied to our initial networks *net-06* and *net-18* with 204 variables.

As a result the significance of MI history was confirmed (the top score for both networks) and the TNF gene polymorphism (in top 10 for both networks). Significance of spironolactone usage was revealed only in *net-06*. Tables with the best SA results for both networks *net-06* and *net-18* are presented in Appendix 3, Tables A4 and A5.

We calculated also AUC values for every naive network that could be constructed using the ranking sequence of variables obtained by the SA presented above. The results are shown in Figure 3, where the *x*-axis represents the naive networks with the respective number of variables (except the root variable): from the top variables with the highest SA score to our initial networks *net-06* and *net-18* with 204 variables. One can see that maximal AUC values 0.6389 (*net-06*) and 0.6442 (*net-18*) obtained by this method are much lower than ones, 0.8031 and 0.7614, obtained by our network optimization method, respectively.

To illustrate how the outcome is distributed across different variables we presented lists of variables of the minimal optimized networks, *net-06* and *net-18*, together with their SA scores in Tables A6 and A7, respectively (Appendix 3).

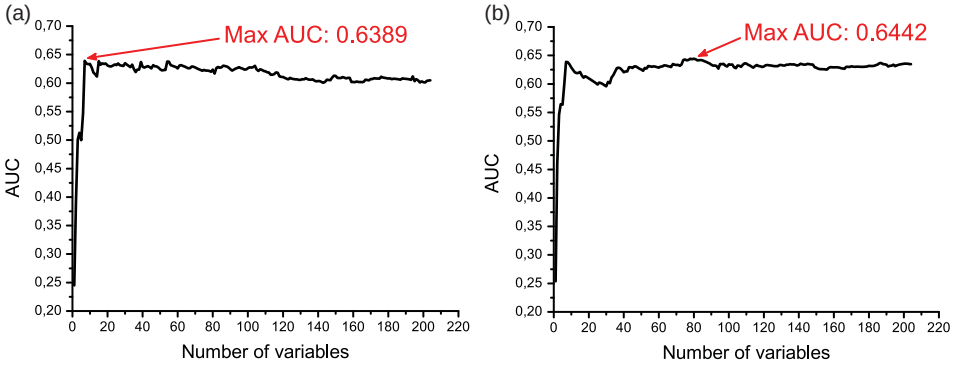

Figure 3. AUC values for naive networks constructed using the variable sequence obtained by SA for *net-06* (a) and *net-18* (b), where the *x*-axisrepresents the naive networks with the respective number of variables (except the root variable): from the top variables with the highest SA score to our initial networks *net-06* and *net-18* with 204 variables.

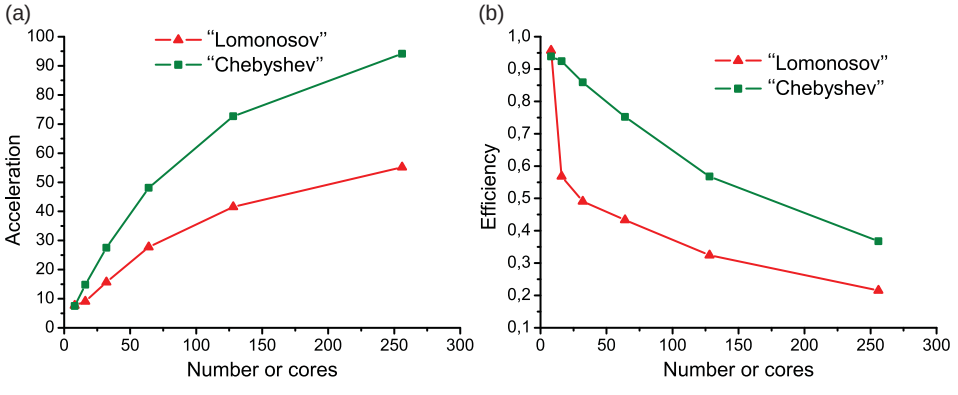

Figure 4. Acceleration (a) and efficiency (b) dependencies on number of computational cores on supercomputers 'Lomonosov' [12] and 'Chebyshev' [11].

We also examined efficiency of multi-core parallel performance of the developed program. We use two values for parallel performance characterization: acceleration

$$A(N) = \frac{t(N=1)}{t(N)}, \tag{1}$$

where t is working time, N is the number of computational cores, and efficiency

$$E(N) = \frac{A(N)}{N}. \tag{2}$$

Both functions' dependencies on the number of computational cores were calculated for two supercomputers of Research Computing Center of Moscow State University: 'Lomonosov' [12] and 'Chebyshev' [11]. Results are presented in Figure 4. As we see, *SiMBA* shows good efficiency and very well acceleration with the number of cores comparable with the number of variables in the investigated networks.

4. Conclusion and further work

In the present paper we proposed the algorithm of naive BNs optimization with respect to the number of their variables. This optimization increases prediction quality of these networks and reveals the most important variables. The algorithm has been implemented in the parallel program able to operate in multiple computational cores mode at supercomputers. The program has been successfully applied to the problem of predicting risk of unfavorable outcome after an ACS.

The networks optimization results in a considerable decrease in the number of network variables from more than two hundred to several dozens in the optimal networks and to less than 20 in the minimal ones. The prediction quality demonstrates a sizable increase in AUC from 0.6 up to 0.8 after the optimization. This effect possibly is connected with a decrease in statistical noise provided by the large number of variables and inherent fluctuations in the patients databases. The networks optimization reveals the most important variables in the problem under consideration, and among them the important role of the TNF-alpha gene polymorphism is discovered for the first time. The results were verified with the SA method. The presented method allows to obtain networks with much larger AUC values than with SA.

The proposed optimization algorithm has been implemented in the parallel program *SiMBA* that has demonstrated good scalability with the increase in the number of computational cores, and it can be applied to the large patients database containing thousands of variables.

The presented results confirmed the effectiveness of BNs technique application for personalized medicine expert systems.

Further work will be mainly aimed at network topology change in the course of the optimization process. This idea was inspired by results of tree augmented naive networks researches [18]. The application of topology optimization may be done in the following way: as we try to exclude any single variable we calculate AUC not for the resulting naive network but for network with the most optimal topology. This can be obtained by learning network with topology optimization, for example, via Structural expectation-maximization algorithm [17]. Other possible way is to combine our method with the recently presented approach [20]. Perhaps it will help not only to detect most significant variables, but also to reveal links between them. We are planning to validate and implement the proposed method for other important problems of personalized medicine.

Acknowledgements

This work was funded partly by the grant from Close Join-Stock Company Taatta Bank (Yakutsk, Russian Federation) and partly by Dimonta, Ltd (Moscow, Russian Federation).

References

[1] S. Andreassen, M. Woldbye, B. Falck and S.K. Andersen, *MUNIN – A Causal Probabilistic Network for Interpretation of Electromyographic Findings*, Proceedings 10th International Joint Conference on Artificial Intelligence (IJCAI), Milan, Italy, 1987, pp. 366–372.

[2] M. Athanasiou and J. Clark, *A Bayesian network model for the diagnosis of the caring procedure for wheelchair users with spinal injury*, Comput. Methods Programs Biomed. 95 (2009), pp. S44–S54.

[3] A. Aussema, S.R. de Moraisa, and M. Corbex, *Analysis of nasopharyngeal carcinoma risk factors with Bayesian networks*, Artif. Intell. Med. 54 (2012), pp. 53–62.

[4] D. Barber, *Bayesian Reasoning and Machine Learning*, Cambridge University Press, Cambridge, 2011, pp. 57–73.

[5] E.S. Burnside, D.L. Rubin, J.P. Fine, R.D. Shachter, G.A. Sisney, and W.K. Leung, *Bayesian network to predict breast cancer risk of mammographic microcalcifications and reduce number of benign biopsy results: Initial experience*, Radiology 240 (2006), pp. 666–673.

[6] E. Castillo, J.M. Gutiérrez, and A.S. Hadi, *Sensitivity analysis in discrete Bayesian networks*, IEEE Trans. Syst. 27 (1997), pp. 412–423.

[7] J.P. Choi, T.H. Han, and R.W. Park, *A hybrid Bayesian network model for predicting breast cancer prognosis*, J. Korean Soc. Med. Inform. 15 (2009), pp. 49–57.

[8] G. Cooper, *Nestor: A computer-based medical diagnosis that integrates causal and probabilistic knowledge*, Technical Report HPP-84-48, Stanford University, CA, 1984.
[9] G.F. Cooper and E. Herskovits, *A Bayesian method for the induction of probabilistic networks from data*, Mach. Learn. 9 (1992), pp. 309–347.
[10] J. Davis, E. Lantz, D. Page, J. Struyf, P. Peissig, H. Vidaillet, and M. Caldwell, *Machine Learning for Personalized Medicine: Will This Drug Give Me a Heart Attack?* Proceedings of Machine Learning in Health Care Applications Workshop. In conjunction with ICML 2008, July 9, Helsinki, Finland, 2008.
[11] Description of 'Chebyshev' supercomputer, Research Computing Center of Moscow State University, Moscow. Available at http://www.parallel.ru/cluster/skif_msu.html.
[12] Description of 'Lomonosov' supercomputer, Research Computing Center of Moscow State University, Moscow. Available at http://www.parallel.ru/cluster/lomonosov.html.
[13] P. Domingos and M. Pazzani, *Beyond independence: Conditions for the optimality of the simple Bayesian classifier*, Proceedings of the 13th International Conference on Machine Learning, Bari, Italy, 1996, pp. 105–112.
[14] T. Fawcett, *ROC Graphs: Notes and Practical Considerations for Researchers*, Kluwer Academic Publishers, Netherlands, 2004.
[15] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, *From data mining to knowledge discovery in databases*, AI Mag. 17 (1996), pp. 37–54.
[16] J. Friedman, *On bias, variance, 0/1 - loss, and the curse-of-dimensionality*, Data Min. Knowl. Discov. 1 (1997), pp. 55–77.
[17] N. Friedman, *The Bayesian Structural EM Algorithm*, Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI), 1998, pp. 129–138.
[18] N. Friedman, D. Geiger, and M. Goldszmidt, *Bayesian networks classifiers*, Mach. Learn. 29 (1997), pp. 131–163.
[19] A.P. Goncalves, J. Ferreira, C. Aguiar, and R. Seabra-Gomes, *TIMI, PURSUIT, and GRACE risk scores: Sustained prognostic value and interaction with revascularization in NSTE-ACS*, Eur. Heart J. 26 (2005), pp. 865–872.
[20] A. Gruber and I. Ben-Gal, *Efficient Bayesian network learning for system optimization in reliability engineering*, Qual. Technol. Quant. Manag. 9 (2012), pp. 97–114.
[21] J.A. Hanley and B.J. McNeil, *The meaning and use of the area under a receiver operating characteristic (ROC) curve*, Radiology 143 (1982), pp. 29–36.
[22] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Science+Business Media, LLC, New York, 2009.
[23] D. Heckerman, E. Horwitz, and B. Nathwani, *Towards normative expert systems: Part I – the Pathfinder project*, Methods Inf. Med. 31 (1992), pp. 90–105.
[24] F.V. Jensen, S.H. Aldenryd, and K.B. Jensen, *Sensitivity analysis in Bayesian networks*, Lecture Notes Comput. Sci. 946 (1995), pp. 243–250.
[25] F.V. Jensen and T.D. Nielsen, *Bayesian Networks and Decision Graphs*, Springer, New York, 2007.
[26] J.H. Kim and J. Pearl, *CONVINCE: A conversational inference consolidation engine*, IEEE Trans. Syst. Man Cybernet. 17 (1987), pp. 120–132.
[27] U.B. Kjærulff, A.L. Madsen, *Bayesian Networks and Influence Diagrams*, Springer, New York, 2008.
[28] G.D. Kleiter, *Bayesian diagnosis in expert systems*, Art. Int. 54 (1992), pp. 1–32.
[29] K.B. Korb and A.E. Nicholson, *Bayesian Artificial Intelligence*, Chapman and Hall/CRC, London, 2004.
[30] W. Lam and F. Bacchus, *Learning Bayesian belief networks. An approach based on the MDL principle*, Comput. Intell. 10 (1994), pp. 269–293.
[31] R.M. Locksley, N. Killeen, and M.J. Lenardo, *The TNF and TNF receptor superfamilies*, Cell 104 (2001), pp. 487–501.
[32] A. Mittal and A. Kassim, *Bayesian Network Technologies: Applications and Graphical Models*, IGI Publishing, Hershey, New York, 2007.
[33] S. Mendis, P. Puska, and B. Norrving (eds.), *Global Atlas on Cardiovascular Disease Prevention and Control*, WHO Press, Geneva, 2011.
[34] N.A. Obuchowski, *Fundamentals of clinical research for radiologists: ROC analysis*, AJR 184 (2005), pp. 364–372.
[35] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufman, San Mateo, CA, 1988.
[36] J.L. Rodgers and W.A. Nicewander, *Thirteen ways to look at the correlation coefficient*, Amer. Stat. 42 (1988), pp. 59–66.
[37] P. Sebastiani, M.F. Ramoni, V. Nolan, C.T. Baldwin, and M.H. Steinberg, *Genetic dissection and prognostic modeling of overt stroke in sickle cell anemia*, Nature Genet. 37 (2005), pp. 435–440.
[38] J. Tao, Q. Li, C. Zhu, and J. Li, *A hierarchical naive Bayesian network classifier embedded GMM for textural image*, Int. J. Appl. Earth Observation Geoinformation 14 (2012), pp. 139–148.
[39] M.A. Valle, S. Varas and G.A. Ruz, *Job performance prediction in a call center using a naive Bayes classifier*, Expert Syst. Appl. 39 (2012), pp. 9939–9945.

[40] H. Volzke, C. Schmidt, S. Baumeister, T. Ittermann, G. Fung, J. Krafczyk-Korth, W. Hoffmann, M. Schwab, H. Meyer zu Schwabedissen, M. Dorr, S. Felix, W. Lieb, and H. Kroemer, *Personalized cardiovascular medicine: concepts and methodological considerations*, Nat. Rev. Cardiol. 10 (2013), pp. 308–316.
[41] C. Wang, N. Komodakis, and N. Paragios, *Markov Random Field modeling, inference & learning in computer vision & image understanding: A survey*, Comput Vision Image Understanding 117 (2013), pp. 1610–1627.
[42] X. Wu and V. Kumar, *The Top Ten Algorithms in Data Mining*, Chapman and Hall/CRC, Taylor & Francis Group, Boca Raton, FL, 2009, pp. 37–59.

Appendix 1. Basic network algorithms

Here, we briefly describe algorithms of network learning and evidence collection which are necessary for AUC calculation. In our case of the naive topology, these algorithms are much simpler than in general one (for general case, see [25,29]).

In the following algorithms we denote all variables by integer numbers, and the root variable marked as zero. We also assume that all our variables are discrete; values of any continuous variable can be split into several intervals and each interval can be indexed by an integer. So, the ith variable possesses values from 0 to $(S_i - 1)$, where S_i is the number of values the ith variable can take.

The database used for network learning is represented by a matrix, each column corresponds to a given variable denoted with the same integer, each row is one data unit, e.g. the data for a given patient. Integers in cells of that matrix correspond to values of respective variables ('-1' stands for missing data). Also, we assume that zero column (responsible for the root variable) has no missing data (if there is any we just ignore the corresponding row in the database). The latter assumption was made because we need learning procedure in order to calculate AUC (Algorithm A1).

Algorithm A1 An 'excluding-by-one' algorithm of AUC calculation

Function:	AUC	
Input:	$\mathcal{B}$	Bayesian network
	$\mathcal{D}$	database with M data lines, first line has index 0
Output:	A	AUC value

1: ALLOCATE(**T**) {ROC table}
2: **for** $i \leftarrow 0$ **to** $(M - 1)$ **do**
3: $\{\mathcal{D}', l\} \leftarrow$ EXCLUDELINE$(\mathcal{D}, i)$ {Exclude ith line from database and store it in l}
4: $\mathcal{B} \leftarrow$ LEARNNETWORK$(\mathcal{B}, \mathcal{D}')$ {Learn network on database with excluded line}
5: $x \leftarrow$ REALROOTEVIDENCE(l)
6: $y \leftarrow$ CALCROOTPROBABILITY$(\mathcal{B}, l)$
7: ADDLINE$(\mathbf{T}, x, y)$
8: SORT(**T**)
9: $A \leftarrow$ CALCAUC(**T**)

But row without evidence on the root variable cannot contribute to AUC value (because missing root value cannot be included to ROC table). However, we would like to emphasize that all other missing data are taken into account and participate in BN learning. It is absolutely irrelevant to ignore it as far as the patient database in the present work (Section 3) has a lot of missing data: for some variables up to 40% patients have no data.

In the case of naive topology network, learning is reduced to direct usage of definition of conditional probability. The learning algorithm is presented in Algorithm A2. We use the structure $\hat{\mathcal{E}}$ that appears to be an array of arrays of integers, as a 'storage' for statistics of given evidences. Every element of this structure is needed for further calculation of value of corresponding cell in the CPTs, so every $\mathcal{E}^i$ (for $i \geq 1$) has $(S_i \cdot S_0)$ cells as it corresponds to the future CPT of the ith variable, $\mathcal{E}^0$ has S_0 cells. At first we allocate memory for this structure (line 1) and make all its elements equal to zero (line 4). Then, we analyze evidence in every data line and store the total sum in the corresponding cells of $\hat{\mathcal{E}}$ (line 6 – for the root variable, line 9 – for others). After that we allocate memory for array of CPTs (line 14), every element of which appears to be a matrix with real number elements, every $\mathcal{P}^i$ (for $i \geq 1$) is $S_i \times S_0$ matrices, $\mathcal{P}^0$ is $1 \times S_0$ matrix. We assume that for every $0 \leq i < S_0, 0 \leq j < S_k,\ 1 \leq k \leq N$ P^k_{ji} corresponds to conditional probability $P(X_k = j \mid X_0 = i)$,

P_i^0 corresponds to $P(X_0 = i)$ for every $0 \le i < S_0$. We obtain values of conditional probabilities by dividing values in cells $\mathcal{E}^i$ by the sum of values of element corresponding to a single row of CPT (normalization probability to 1, lines 16, 19). Finally, we create naive BN using obtained CPTs (line 20). Presented algorithm is precise – it has no any random factors. So, the result of learning is the same for the given network and database for any independent run.

Algorithm A2 Naive network learning algorithm

Function:	LEARNNETWORK	
Input:	$\mathcal{B}$	naive Bayesian network, which contains **X** and **S**:
	$\mathbf{X} = \{X_0, .., X_N\}$	list of network variables, X_0 is root one
	$\mathbf{S} = \{S_0, .., S_N\}$	list of number of values for every variable from **X**
	$\mathcal{D}$	database: $M \times (N+1)$ integer matrix, M – number of data lines, $N+1$ – number of variables
Output:	$\mathcal{B}$	learned naive Bayesian network

1: ALLOCATE($\hat{\mathcal{E}}$) {an array of integer arrays}
2: **for** $i \leftarrow 0$ **to** N **do**
3: **for** $j \leftarrow 0$ **to** SIZEOF($\mathcal{E}^i$) **do**
4: $E_j^i \leftarrow 0$ {E_j^i is a jth element of $\mathcal{E}^i$}
5: **for** $i \leftarrow 0$ **to** $(M-1)$ **do**
6: $E^0_{\mathcal{D}_{i0}} \leftarrow (E^0_{\mathcal{D}_{i0}} + 1)$
7: **for** $j \leftarrow 1$ **to** N **do**
8: **if** $\mathcal{D}_{ij} > (-1)$ **then**
9: $E^j_{(S_j \cdot \mathcal{D}_{i0} + \mathcal{D}_{ij})} \leftarrow (E^j_{(S_j \cdot \mathcal{D}_{i0} + \mathcal{D}_{ij})} + 1)$
10: **else**
11: **if** $\mathcal{D}_{ij} = (-1)$ **then**
12: **for** $l \leftarrow 0$ **to** $S_j - 1$ **do**
13: $E^j_{(S_j \cdot \mathcal{D}_{i0} + l)} \leftarrow (E^j_{(S_j \cdot \mathcal{D}_{i0} + l)} + 1/S_j)$
14: ALLOCATE($\hat{\mathcal{P}}$) {an array of CPTs}
15: **for** $i \leftarrow 0$ **to** $S_0 - 1$ **do**
16: $P_i^0 = E_i^0 / M$ {$P(X_0 = i)$}
17: **for** $k \leftarrow 0$ **to** N **do**
18: **for all** $i, j:$ $0 \le i < S_0,\ 0 \le j < S_k$ **do**
19: $$P^k_{ji} = E^k_{(i \cdot S_k + j)} \cdot \left(\sum_{l=(i \cdot S_k)}^{(i+1) \cdot S_k + j} E^k_l \right)^{-1} \quad \{P(X_k = j | X_0 = i)\}$$
20: $\mathcal{B} \leftarrow$ CREATENAIVENETWORK($\mathbf{X}, \hat{\mathcal{P}}$)

Considering collect evidence procedure (for the root variable – the only one we need for AUC calculation) with naive network (Algorithm A3), we use CPT representation in potential form (see, for example, [25]). So, due to native topology, the presented algorithm is simplified to merging (multiplying) all potentials (those allocated in line 1) of the network (lines 4, 5) (with taking given evidence into account – line 3) with further normalization to 1 (line 6).

Description of used functions can be found in Table A1. The function SETEVIDENCE (line 3) needs some explanation: any single evidence is taken into account while calculating probability of the root variable. Within the framework of the language of potentials it means merging of potential $\mathbf{P^i}$ of CPT corresponding to the considered variable X_i with potential $\mathbf{Z^i}$ of variable X_i of the form

$$\forall j,\ 0 \le j < S_i:\ Z_j^i = \begin{cases} 0, & j \ne d_i \\ 1, & j = d_i, \end{cases}$$

Algorithm A3 Collect evidence for root variable in naive network

Function:	CalcRootProbability	
Input:	$\mathcal{B}$	learned naive Bayesian network with variables $\mathbf{X} = \{X_0, .., X_N\}$
	d	data line with evidences (may contain missing values)
Output:	$\mathbf{P_0}$	Probabilities (for different values) of root variable

1: Allocate(**P**, **Z**) {Temporary potentials}
2: **for** $i \leftarrow 1$ **to** N **do**
3: **Z** $\leftarrow$ SetEvidence(X_i, d_i) {Set evidence form data line **d** to variable X_i and store it in temporary potential **Z**}
4: **P** $\leftarrow$ Merge(**P**, **Z**)
5: **P** $\leftarrow$ Merge(**P**, CPT(X_0)) {Merge temporary potential with CPT of root variable}
6: **P** $\leftarrow$ Normalize(**P**)
7: $\mathbf{P_0} \leftarrow$ Probability(**P**) {Take desired probabilities from temporary potential}

where d_i is the value of the ith element of given data line **d**. With such definition of the $\mathbf{Z^i}$ procedure SetEvidence(X_i, d_i) is equal to Merge(CPT(X_i), $\mathbf{Z^i}$), where CPT(X_i) returns potential from CPT of variable X_i. After merging temporary potential **P** with potential of the root variable CPT(X_0) we obtain a potential of one variable X_0. After normalization to 1, this potential contains desired probabilities of the root variable, so we only need to extract them (line 7).

Table A1. Description of functions used in algorithms for the naive network topology.

List of used functions	Description of used functions
AddLine(**T**, x, y)	Add line to ROC-table **T** containing from known evidence x and calculated probability y
Allocate(T)	Allocate memory for structure T
AUC($\mathcal{B}, \mathcal{D}$)	Calculate AUC for network $\mathcal{B}$ using network $\mathcal{D}$. Returns a real number
CalcAUC(**T**)	Calculate AUC from sorted table **T**. Returns real number
CalcRootProbability($\mathcal{B}, l$)	Calculate probability of root variable (of 0th value, for definiteness) of network $\mathcal{B}$ using evidences form data line l (ignoring evidence for root variable). Returns a real number. Discussed in detail in present article for naive networks
CPT(X)	Returns potential of CPT of variable X
CreateNaiveNetwork(**X**, $\hat{\mathcal{P}}$)	Create naive network from list of variables **X**, where X_0 is root variable, with CPTs $\hat{\mathcal{P}}$. Returns naive network
EvNum($\mathcal{D}, X_i, j$)	Calculate number of evidences of value j of variable X_i in database $\mathcal{D}$
ExcludeLine($\mathcal{D}, i$)	Exclude ith line from database $\mathcal{D}$. Returns new database and excluded line
ExcludeVariable($\mathcal{B}, i$)	Exclude ith variable from network $\mathcal{B}$. Returns new network
LearnNetwork($\mathcal{B}, \mathcal{D}$)	Learn network $\mathcal{B}$ using database $\mathcal{D}$. Returns learned network. Discussed in detail in present article for naive networks
Max(**b**)	Find maximum element of array **b**. Returns the value of that element and its index
Merge(**X**, **Y**)	Merge potential **X** with potential **Y**
Normalize(**P**)	Normalize potential **P** so sum of its value becomes 1
OptimizeNaiveNetwork($\mathcal{B}, \mathcal{D}$)	Optimize naive network $\mathcal{B}$ by the number of variables using database $\mathcal{D}$. Returns an optimized naive network
Probability(**P**)	Extract probabilities from normalized potential **P**. Returns an array of real numbers

(*Continued*)

Table A1. Continued

List of used functions	Description of used functions
REALROOTEVIDENCE(l)	Extract known evidence of root variable from data line l. Returns an integer number
SENSITIVITYANALYSIS($\mathcal{B}, \mathcal{D}$)	Provide a 'weighted' SA for naive network $\mathcal{B}$, using database $\mathcal{D}$. Returns an array of real numbers – scores for every variable.
SETEVIDENCE(X, x)	Set evidence with index x for variable X. Returns corresponding potential for variable X
SIZEOF(**X**)	Returns size of array **X**
SORT(**T**)	Sort table **T** for AUC calculating by column with calculated probabilities
VALNUMBER(X)	Returns number of values for variable X

Appendix 2. List of clinical characteristics

Table A2. Clinical characteristics of the involved patients.

Index	N of case
Sex (male/female)	755/438
Age, years[a]	61.4±11.72
Smokers	458
Unstable angina/Non ST MI	767
Acute MI with ST-segment elevation	426
History of MI	384
History of CAD	806
Heart failure	651
History of stroke	108
Diabetes mellitus	156

[a]Mean age±Standard Error of mean.

Appendix 3. List of the network variables

Table A3. List of common variables for *net-06* and *net-18* after optimization by maximal AUC.

Variable	Possible values	Percentage of presence in database (%)
Age	Intervals 28..53..61..71..82	100
History of MI	Yes/no	100
Increase of angina attacks severity at the time of admission to the hospital	Yes/no	55
Systolic blood pressure	Intervals 85..120..140..180	67
Prior hypolipidemic therapy	Yes/no	100
Use of angiotensin II receptor blocker within 10 days of hospitalization	Yes/no	100
Use of non-cardiovascular drugs within 10 days of hospitalization	Yes/no	100
Use of nitrates within 10 days of hospitalization	Yes/no	100
Use of spironolactone within 10 days of hospitalization	Yes/no	100
Daily consumption of fruits or vegetables	Yes/no	100
Uric acid level	Intervals 39..420..934	75
Atherogenic index	Intervals 0.2..4.5..6..36.9	74
Sinoatrial or atrioventricular conduction disorders	Yes/no	100
ST segment elevation at the time of admission to the hospital	Yes/no	45
E/*A* ratio	Intervals 0.1..1..1.5..5	41
Ejection fraction (%)	Intervals 16..30..45..55..85	54
End diastolic volume	Intervals 1.8..2.2..3.2..3.5..3.7..4.2 (male), 1.8..2.4..3.3..3.5..3.8..4.9 (female)	70
Polymorphism gene PROC C(-1654)T	CC/CT/TT	96
Polymorphism TNF gene A(-308)G	AA/AG/GG	91

Table A4. Variables with the best SA results for *net-06*.

Variable	Possible values	Percentage of presence in database (%)	SA score
History of MI	Yes/no	100	47.10
Life style	Sedentary/moderately/ active	100	40.04
Use of spironolactone within 10 days of hospitalization	Yes/no	100	35.79
Heart failure	Yes/no	100	32.35
Use of spironolactone	Yes/no	100	30.66
Certified disability	1th degree/2th degree/ 3th degree/no	100	29.35
Age	Intervals 28..53..61..71..82	100	27.42
Polymorphism TNF gene A(-308)G	AA/AG/GG	91	27.25
Coronary artery disease	Yes/no	100	26.30
Education	Higher/specialized secondary/ secondary	100	23.29

Table A5. Variables with the best SA results for *net-18*.

Variable	Possible values	Percentage of presence in database (%)	SA score
History of MI	Yes/no	100	65.35
Baseline diagnosis	Unstable angina/non ST MI/acute MI with ST-segment elevation	100	60.58
Sleeplessness	Yes/no	100	47.37
Polymorphism TNF gene A(-308)G	AA/AG/GG	91	46.41
Diabetes mellitus	Mild type 2 diabetes/moderate/severe/no	100	44.86
Heart failure	Yes/no	100	44.30
Weakness	Yes/no	100	40.09
Use of nitrates within 10 days of hospitalization	Yes/no	100	38.17
Manifestation of CAD	Angina/MI	67	34.94
Certified disability	1th degree/2th degree/3th degree/no	100	34.55

Table A6. List of variables for *net-06* after minimization.

Variable	Possible values	Percentage of presence in database (%)	SA score
History of MI	Yes/no	100	47.10
Life style	Sedentary/moderately/active	100	40.04
Use of spironolactone within 10 days of hospitalization	Yes/no	100	35.79
Polymorphism TNF gene A(-308)G	AA/AG/GG	91	27.25
Rest angina at the time of admission to the hospital	Yes/no	55	18.12
Ejection fraction (%)	Intervals 16..30..45..55..85	54	17.86
Smoking experience (years)	Intervals 0..25..35..60	54	17.17
Base infero-lateral segment	Normal/hypokinesis/dyskinesis/akinesis	48	13.94
Daily consumption of fruits or vegetables	Yes/no	100	11.86
ST segment elevation at the time of admission to the hospital	Yes/no	45	11.31
New ischemic ECG sign within hospitalization	Yes/no	100	10.24
Weight changes	Yes/no	100	9.92
Weakness	Yes/no	100	9.23
Necessity of insulin	Yes/no	100	8.55
Mid antero-lateral segment	Normal/hypokinesis/dyskinesis/akinesis	48	4.95
Apex lateral segment	Normal/hypokinesis/dyskinesis/akinesis	48	4.84
Use of angiotensin II receptor blocker within 10 days of hospitalization	Yes/no	100	0.91

Table A7. List of variables for *net-18* after minimization.

Variable	Possible values	Percentage of presence in database (%)	SA score
History of MI	Yes/no	100	65.35
Baseline diagnosis	Unstable angina/Non ST MI/Acute MI with ST-segment elevation	100	60.58
Sleeplessness	Yes/no	100	47.37
Polymorphism TNF gene A(-308)G	AA/AG/GG	91	46.41
Diabetes mellitus	Mild type 2 diabetes/moderate/severe/no	100	44.86
Use of spironolactone within 10 days of hospitalization	Yes/no	100	31.39
Thigh circumference	Intervals 66..95..101..107..163	98	30.85
Parent's alcohol abuse	Yes/no	75	30.39
Prior hypolipidemic therapy	Yes/no	100	28.46
Ventricular septal thickness	Intervals 5..10..12..17	72	27.55
Increase of angina attacks severity at the time of admission to the hospital	Yes/no	100	26.30
Thiazides	Yes/no	100	25.97
Age	Intervals 28..53..61..71..82	100	24.47
Base infero-septum segment	Normal/Hypokinesis/Dyskinesis/Akinesis	54	18.13
History of stroke	Yes/no	100	17.11
Recurrence of MI within hospitalization	Yes/no	100	8.55