Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. IV (2009), No. 1, pp. 65-72

# Bayesian Network Classifier for Medical Data Analysis

Beáta Reiz, Lehel Csató

*Beáta Reiz*
Biological Research Center, Central Labs, Bioinformatics Group
62 Temesvári krt., HU-6701, Szeged, Hungary
E-mail: beareiz@brc.hu
*Lehel Csató*
Babeş Bolyai University, Faculty of Mathematics and Computer Science
1 Kogălniceanu str. RO-400084 Cluj-Napoca, Romania
E-mail: csatol@cs.ubbcluj.ro

**Abstract:** Bayesian networks encode causal relations between variables using probability and graph theory. They can be used both for prediction of an outcome and interpretation of predictions based on the encoded causal relations. In this paper we analyse a tree-like Bayesian network learning algorithm optimised for classification of data and we give solutions to the interpretation and analysis of predictions. The classification of logical – i.e. binary – data arises specifically in the field of medical diagnosis, where we have to predict the survival chance based on different types of medical observations or we must select the most relevant cause corresponding again to a given patient record.
Surgery survival prediction was examined with the algorithm. Bypass surgery survival chance must be computed for a given patient, having a data-set of 66 medical examinations for 313 patients.
**Keywords:** Bayesian networks, classification, medical data analysis, causal discovery.

## 1 Introduction

In this paper we analyse tree-like Bayesian network (BN) implementation for medical data classification. We consider a general case for data attributes, where the observations can be both continuous and discrete, and - general to almost all medical data - missing observations also can occur. We aim to establish causal relationships between variables representing medical examinations. Whilst interested in a good classification performance, we also want to interpret and analyse the predictions in terms of the encoded causal relations.

The database we used consists of 66 medical examinations of 313 people containing both discrete and continuous observations. The task is thus to predict the surgery survival chance based on the available data – the medical examinations [1, 2], and analysis of impact of a specific examination on patient survival. Partial observability characterises the database, the number of missing values is 2413.

Our aim is to predict target variables value – survival – for a particular patient and to obtain the "most relevant" variables affecting the output of the classifier. Of equal interest is to analyse the decisions in terms of encoded relationships between data attributes. This analysis is usually done to provide support for physicians. We encode the dependencies between the class variable and the observations using a tree with root node the class variable. The other attributes are inside the tree with coresponding conditional probability tables "learned" from the data-set. Finding the most appropriate structure is an extremely difficult task. We reduce the complexity of constructing the tree of *immediate causal* relationships between

class variable and observations [3]. A tree-like Bayesian network structure was inferred from the data [4], where the root of the tree is the class variable and remaining nodes are attributes. Direct causal relations between attributes and class variable were revealed in the first phase of the algorithm, constructing a Naive Bayesian network. Attribute-attribute correlations were searched based on Chow-Liu's algorithm in the second phase of the algorithm. In practical situations we also have to face the problem – general to almost all medical data – of missing observations for some patients, meaning incomplete data items; this issue can also be considered in a principled way with a Bayesian network. The paper is organised as follows: next we present the Bayesian networks, then a stochastic algorithm to extract a plausible network sructures from the data, and we also analise experimental results of applying the algorithm to real data-sets.

## 2 Bayesian Networks

Bayesian networks (BNs) [5] are triplets $(V,E,\mathcal{P})$, where $(V,E)$ is a directed acyclic graph (DAG) with nodes $V$, edges $E$, and a set of probability distributions $\mathcal{P}$, called parameters, whose elements are assigned to the nodes of the graph. The nodes represent domain variables and edges mark direct causal relations between these variables.

The network encodes a joint probability distribution function representative to the domain:

$$P(X) = \prod_{i=1}^{n} P(X_i|\text{par}(X_i))$$

where $n$ is the number of domain variables, $X_i$ is a node from the BN and $\text{par}(X_i)$ is the set of $X_i$'s parents. The aciclicity of the graph ensures the product to be finite.

We employed a tree-like representation for the topology of BN in order to increase efficiency in class variable estimation and interpretation. In section 3. we describe this algorithm, where we construct a tree in such a way that the root of the tree will be the class variable and the remaining nodes are attributes. Direct causal relations encoded by the BN are interpreted as the maximum of mutual respective conditional mutual information [6, 7, 8] between nodes. Now we present the necessary information theoretical concepts [9] for our algorithm.

We will use the following notations: $X$ and $Y$ are random variables defined on probability spaces $\Omega_X$ respective $\Omega_Y$ with corresponding distribution functions $p(x)$ respective $p(y)$. We use their joint and conditional probability functions, denoted with $p(x,y)$ and $p(x|y)$ respectively.

Information theory offers us numerical characterisation of uncertainty in domain variables. Uncertainty is measured using the information entropy of the respective variable. Information entropy can be understood as the average minimal message length that should be sent on a channel to encode the message and is defined as follows:

$$H(X) = -\sum_{x\in\Omega_X} p(x)\log p(x)$$

Mutual information is the quantity of information two random variables contain about each other, defined as:

$$I(X,Y) = \sum_{x\in\Omega_X}\sum_{y\in\Omega_Y} p(x,y)\ \log\frac{p(x,y)}{p(x)p(y)} \tag{1}$$

In next section we present a two-phase tree-like Bayesian network structure learning algorithm. The algorithm consists of an extension of Naive Bayesian structure learning algorithm with inner structure learning for finding causal relations between attributes.

# 3 Network topology learning

Bayesian network classification consists of emphasising the node corresponding to the class variable during inference. As an optimisation of the learning process and inference we construct the network topology such a way to optimise the efficiency of prediction and data attribute impact estimation on target variable. We set out the class variable and in the first phase we're searching for direct dependencies between attributes and class variable, this way constructing a Naive Bayesian network.

Naive Bayes classifiers [10, 11] are widely used in classification problems. They are called Naive because of the independence assumption of the attributes. Although this is strong assumption when facing real data-sets, the Naive Bayes classification is a powerful tool for its simplicity and often gives convenient results.

During the Naive Bayesian network learning process direct dependencies between class variable and attributes has to be find. Dependency relations are interpreted as class variable specifiers, so an edge from $X$ to class variable $Y$ means that variable $X$ has information about class variable $Y$. Mutual information between class variable and attributes, conditioning on attributes already placed between direct dependencies of class variable, gives the amount of new information the respective attribute has regarding the class variable [12].

Considering the problem this solutions means that we choose some medical examinations which we place in the network and exclude the rest of the attributes. This is a strong restriction considering that some examinations are replaced by others in different hospitals. The second phase of the algorithm consists of applying Chow-Liu algorithm [13] to learn the inner structure of network and reveal attribute-attribute correlations.

The Naive Bayesian network is formed of class variable $Y$ respective variables $X$ directly linked to the class variable. Our next task is to place the excluded attributes in the Bayesian network. We use mutual information maximisation to discover the causal relations between attributes from the network and excluded attributes. Class variable could be ignored. Mutual information maximisation is enough in this case, because dependency now has the meaning of replaceability. We are searching for the excluded attributes that carry almost the same information about class variable as the attributes already placed in the network. Before presenting the algorithm, we introduce the notations used in the following:

| | | | |
|---|---|---|---|
| $\boldsymbol{X}$ | set of attributes not *yet* placed in the net | $\boldsymbol{Z}$ | set of attributes in the net |
| $X$ | one attribute from $\boldsymbol{X}$ | $Z_i$ | an element from $\boldsymbol{Z}$ |
| $Y$ | the class variable | $I(X,Y\|\boldsymbol{Z})$ | conditional mutual information of |
| $I(X,Y)$ | mutual information of $X$ and $Y$ | | $X$ and $Y$ given $\boldsymbol{Z}$ |

Our algorithm introduces a threshold parameter – denoted with $\alpha_1$ – which is the minimum "information" required when putting a new attribute in the network during the Naive Bayesian structure learning. This parameter controls the number of direct connections between class variable and attributes. The algorithm is presented in algorithm. 1.

The threshold parameter $\alpha_1$ assures the selection of relevant attributes respect to the class variable, controlling the number of direct causal relations of class variable and attributes. The result is a tree-like Bayesian network as in Figure 2, where the root of the tree is the class variable, and the other nodes are attribute variables. The orientation of edges is from parent to the child, this way minimising the modification of network parameters during a learning step.

The algorithm above is deterministic in sense that it generates the same network for the same data all the time. We introduce importance sampling [14] in order to avoid the determinism of the algorithm in case of selection from equal information quantities. The distribution used for sampling is based on mutual information. It has the maximum where the mutual information is maximal.

We used two functions during the tests. The first function – denoted $f_1$ – is the conversion of the

**Algorithm 1** Tree-like Bayesian network structure learning.

place the class variable $Y$ in the network
$\mathbf{Z} = \emptyset$
{Naive Bayesian structure learning}
**while** $\mathrm{I}(X, Y|\mathbf{Z}) \geq \alpha_1$ **do**
  $\hat{X} = \underset{\mathbf{X}}{\operatorname{argmax}}\, \mathrm{I}(X, Y|\mathbf{Z})$
  place $\hat{X}$ in the network
  $\mathbf{X} = \mathbf{X} - \{\hat{X}\}$
  $\mathbf{Z} = \mathbf{Z} \cup \{\hat{X}\}$
**end while**
{Inner structure learning}
**while** $\mathbf{X} \neq \emptyset$ **do**
  $\left[\hat{X}, \hat{Z}\right] = \underset{X_i, Z_j}{\operatorname{argmax}}\, \mathrm{I}(X_i, Z_j)$
  place edge between $\hat{X}$ and $\hat{Z}$
  $\mathbf{X} = \mathbf{X} - \{\hat{X}\}$
  $\mathbf{Z} = \mathbf{Z} \cup \{\hat{X}\}$
**end while**

Figure 1: Tree-like Bayesian network structure learning.

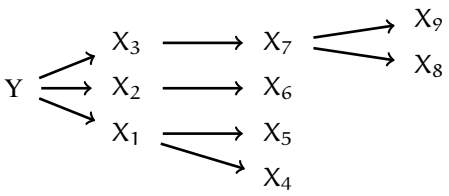



Figure 2: Structure of a BN

mutual information to a distribution function:

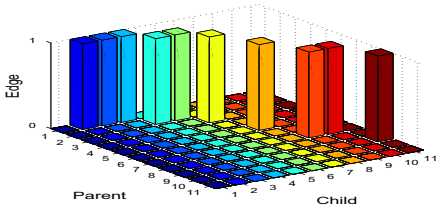
$$\mathrm{f}_1(X) = \frac{I(X,Y)}{\sum_{X' \in \boldsymbol{X}} I(X',Y)} \tag{2}$$

Figure 3(b). illustrates what edges are inferred when using importance sampling with function $f_1$ from artificial data. The generator network for the data is presented on Figure 3(a). On the horizontal plane is the adjacency matrix of the graph and the non-zero columns represent the edges. Figure 3(a). points the edges of the generator network, and Figure 3(b). shows the frequency of learned edges during 300 tests.

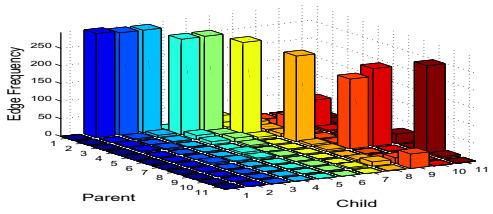
The second function – denoted $f_2$ – uses the exponentiation of the mutual information. It has a $\beta$ parameter which can be understood as a temperature parameter and it controls the constructed distribution function. The higher this parameter is the higher is the probability of selecting the maximum mutual information. On lower values of $\beta$ the probabilities of selecting an attribute becomes closer to the uniform distribution.

$$f_2(X) = \frac{\exp(\beta \cdot \mathrm{f}_1(X))}{\sum_{X' \in \boldsymbol{X}} \exp(\beta \cdot \mathrm{f}_1(X))} \tag{3}$$

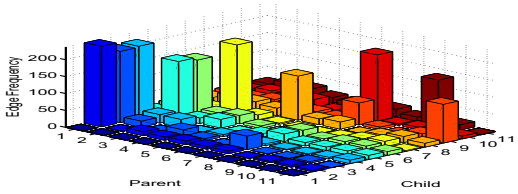
Figure 3. shows the histogram of learned edges using the presented approaches and also the generator network topology of data on Figure 3(a). In each graph on the horizontal plane is the adjacency matrix of the network topology, and the vertical columns represent the histogram edges. We consider the first



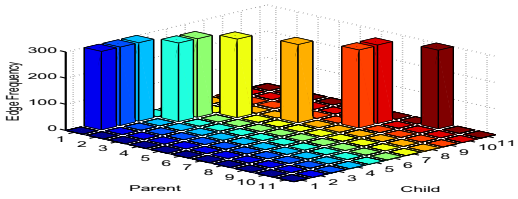

(a) Generator network topology




(b) Learned topologies with function $f_1$




(c) Learned topologies with function $f_2$, $\beta = 5$




(d) Learned topologies with function $f_2$, $\beta = 20$

Figure 3: Generator network and histogram of BN edges

attribute from the network as the class variable, so it is the root of the constructed tree. This means, there's no arc from any attribute to the class variable, hence it's column is 0 at each point. Figure 3(a). represents the generator network topology for the data, hence each edge appears once. The other graphs – on figures 3(b), 3(c), and 3(d) – represent the frequency of edges in the inferred network topologies.

Although there is a randomness introduced with importance sampling in our first approach, the generated structure is relatively stable through the iterations when learning with function $f_1$. The direct causal relations between the class variable and attributes are almost the same during the 300 tests simulations, differences can be observed only in the causal relations between the attributes, more precisely the differences are on the third level of the tree. There in approximately 100 cases – out of 300 – a single edge is placed differently compared to the generator network.

A bit more unstable structure (Figure 3(c)) can be observed when learning with function $f_2$ with parameter $\beta = 5$. This is due to the fact, that the separation between lower and higher values of mutual information is more sensitive for lower values of $\beta$. Figure 3(d). represents the learned structures for $\beta = 20$. One can see that in this case the structure is fully stable, which assures the former statement.

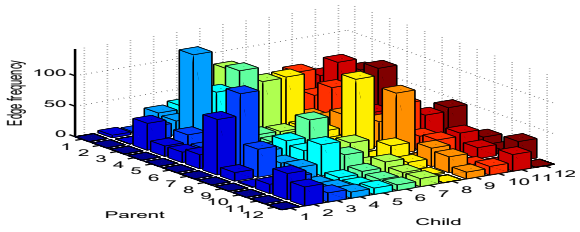
In next sections we will analyse the convergence of the learning process and the usage of the constructed network.

# 4 Results

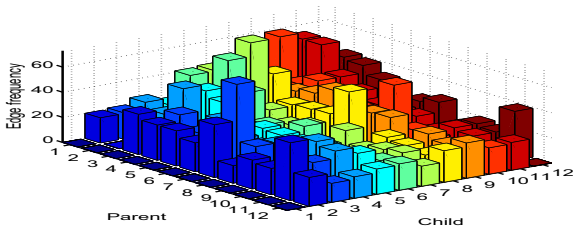
In this section we will present the learned topologies in case of real data. To fully settle the bypass problem, we have to perform the binarisation of data. This is due to the very low number of data samples in bypass database considering conditional probability distribution function estimation.

We made 300 test of the algorithm on the fully specified attributes from the database. Figure 4. presents the results of these testings. One can observe a high order of uncertainty when learning with function $f_2$ with $\beta = 5$. This is reduced by the increase of $\beta$ to 20. Function $f_1$ highlights almost the same dependency relations as function $f_2$ with $\beta = 20$, but there is an annoying level of uncertainty in these relations.

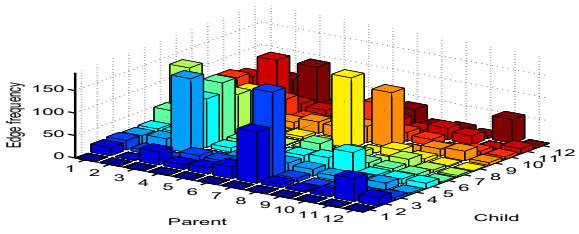
Further analysis of the algorithm meant to check stability of it on real data-set. For this reason Leave-One-Out method on attributes was used: we eliminated one attribute from the data, and tested the algorithm on the remaining ones. The results are showed on Figure 5., where Figure 5(a)., Figure 5(c).




(a) Histogram of edges with function $f_1$




(b) Histogram of edges with function $f_2$, $\beta = 5$




(c) Histogram of edges with function $f_2$, $\beta = 20$

Figure 4: Learned topologies of fully specified attributes from the database

and Figure 5(e). depicts the histogram of learned edges, and on Figure 5(b)., Figure 5(d). and Figure 5(f). the most frequent learned edges are drawn.

Figure 5(a). and Figure 5(b). depicts the histogram of learned edges respective the most frequent edges learned with function $f_2$, $\beta = 20$ for all fully specified attributes from the bypass database. One can see, that there are two crucial attributes in the database, namely the thirrd and seventh, which are central players in attribute dependence relations. Next sub-figures – Figure 5(c). and Figure 5(d). - elimination of 3rd. attribute, respective Figure 5(e). and Figure 5(e). - elimination of 3rd. attribute – depicts the learned BNs when eliminating these attributes.

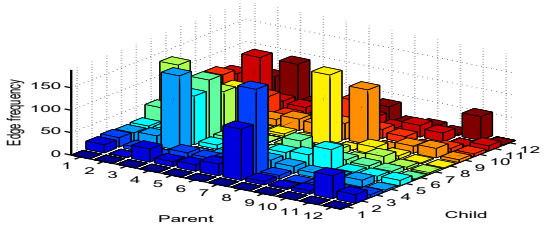
One can see that most of the dependence relations are stable, although there is a reorganisation between dependencies, when eliminating a crucial attribute. The most observable instability in dependence relations is that the edge $3 \rightarrow 12$ becomes an edge $2 \rightarrow 12$ when eliminating either the third attribute or the seventh attribute. But when analysing the dependency of $2 \rightarrow 12$, respective $3 \rightarrow 12$, one can see that their is not so much significant difference between the frequency of the two edges when learning on all attributes.

## 5 Conclusions

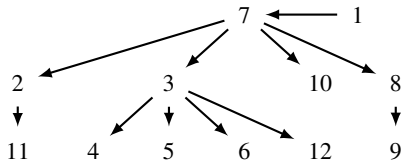
In this paper we presented a tree-like Bayesian network classifier algorithm developed for medical decision making problems and a stochastic algorithm to find the most appropriate structure of the network. We tried two functions for eliminating determinism from the algorithm, with the two functions $f_1$ and $f_2$, defined with eq. 2, respective eq. 3. Learned topologies with the presented algorithm and functions were presented both for artificial and real data. In this section we will present results considering the inference, and compare them with logistic regression and SVM.

Table 1. shows the results of efficiency of the presented algorithms compared to logistic regression. Comparing the first and second approach we described above the results are surprising. Although the high order of uncertainty in some cases, the results of efficiency are similar for all cases.

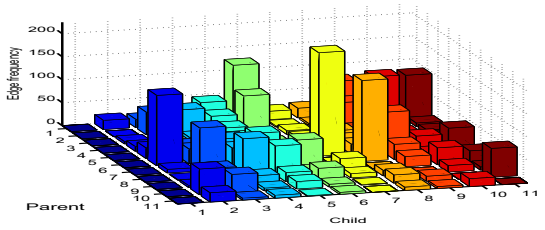
One can observe that the tree-like Bayesian networks constructed with the presented algorithm perform better than logistic regression, but Support Vector Machines with linear kernel obtain higher prediction accuracy than BN-s. It has to be mentioned, that although better accuracy on SVM, they don't allow interpretation of predictions, while BNs do, and interpretation in case of tree-like Bayesian networks,



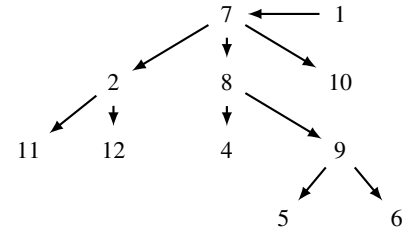

(a) Histogram of edges - all attributes



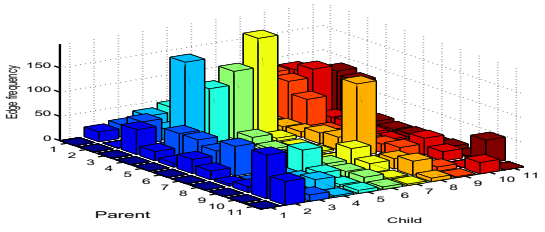

(b) Most frequent edges - all attributes



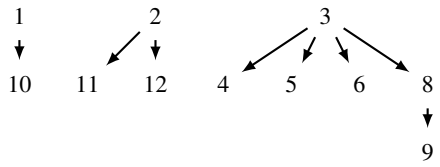

(c) Histogram of edges - LOO 3. attribute




(d) Most frequent edges - LOO 3. attribute




(e) Histogram of edges - LOO 7. attribute




(f) Most frequent edges - LOO 7. attribute

Figure 5: Leave-One-Out results on attributes

| **Method** | **Accuracy** |
|---|---|
| Bayesian network - $f_1$ | 74.69% |
| Bayesian network - $f_2$, $\beta = 1$ | 74.25% |
| Bayesian network - $f_2$, $\beta = 5$ | 74.64% |
| Bayesian network - $f_2$, $\beta = 20$ | 75.71% |
| Logistic regression | 63.50% |
| SVM with linear kernel | 89.84% |

Table 1: Efficiency of presented algorithms

constructed as above, can be done efficiently.

As for a summaryy of results it has to be mentioned that the learned structure by the algorithm is generally stable; the interpretation of the results is possible and partial observability is not a problem in case of prediction and interpretation.

## Acknowledgements

We acknowledge the program committee of the International Conference on Computers, Communication and Control 2008 for the recommendation and also thank for the problem description and the medical database to Béla Vizvári from Department of Operations Research, Eötvös Loránd University, Budapest. This work was partially supported by the Romanian Ministry of Education and Research through grant 11-039/2007.

## Bibliography

[1] Zs. Csizmadia, P.L.Hammer, B. Vizvári. Generation of artificial attributes for data analysis. Rutcor Research Report RRR 42-2004, Rutgers Center for Operations Research, Rutgers University, 2004.

[2] Zs. Csizmadia, B. Vizvári. Methods for the analysis of large real-valued medical databases by logical analysis of data. Rutcor Research Report RRR 42-2004, Rutgers Center for Operations Research, Rutgers University, 2004.

[3] Judea Pearl. *Causality: Modeling, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2000.

[4] Beáta Reiz, Lehel Csató. Tree-like bayesian network classifiers for surgery survival chance prediction. In *Proceedings of International Conference on Computers, Communications and Control*, Vol. III, pp. 470-474, 2008.

[5] Kevin P. Murphy. Learning bayes net structure from sparse data sets. Technical report, Comp. Sci. Div., UC Berkeley, 2001.

[6] Jie Cheng, David A. Bell, and Weiru Liu. An algorithm for bayesian belief network construction from data, 1997.

[7] Jie Cheng, David A. Bell, and Weiru Liu. Learning belief networks from data: An information theory based approach. In *CIKM*, pages 325–331, 1997.

[8] Mieczyslaw A. Klopotek. Mining bayesian network structure for large sets of variables. In *ISMIS*, pages 114–122, 2002.

[9] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.

[10] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.

[11] David Heckerman and Christopher Meek. Models and selection criteria for regression and classification. Technical Report MSR-TR-97-08, Microsoft Research, 1997.

[12] F. Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, November 2004.

[13] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, 14(3):462–467, 1968.

[14] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

**Lehel Csató** obtained his BSc and MSc degrees at the Babeş–Bolyai University, Cluj–Napoca, and his PhD degree from the Neural Computing Research Group at the University of Aston, the United Kingdom. He was interested in the applications of machine learning techniques, specifically to apply non-parametric methods in Bayesian inference. His thesis investigated methods to approximate solutions of Bayesian regression using stochastic Gaussian processes, centring on sparse solutions that approximate the Gaussian processes. He is teaching at the Babes-Bolyai University and he is interested in applications of Bayesian techniques in modern data processing and probabilistic methods in robotics. He is heading the "Data Mining Research Group" at the same university. Web–page: `http://www.cs.ubbcluj.ro/~csatol`

**Beáta Reiz** obtained her BSc and MSc degrees at the Babeş–Bolyai University, Cluj–Napoca. Currently she is a PhD student in Hungary, University of Szeged and she is working in the Bioinformatics group of the Biological Research Center, from Szeged under the supervision of Sándor Pongor and János Csirik.