Annales Univ. Sci. Budapest., Sect. Comp. 36 (2012) 201–215

A PROBABILISTIC CLASSIFICATION METHOD BASED ON CONDITIONAL INDEPENDENCES

Edith Kovács and Tamás Szántai

(Budapest, Hungary)

Communicated by László Lakatos

(Received December 20, 2011; revised January 29, 2012;
accepted March 6, 2012)

Abstract. We introduce a probabilistic classification method which exploits the conditional independences between the features (random variables). This idea is at the heart of the selection of a special type of informative features. The so called t-informative features are selected from the great amount of observed features in order to diminish the uncertainty of the classification variable and in the same time avoid the overfitting of the model. We construct a probabilistic function of the selected t-informative variables which classifies a new entity into one of the classes. We present the efficiency of our algorithm in real-life applications.

1. Introduction

Classification is an important task for decision making under uncertainty. The large number of applications, ranging from the classical ones (such as automatic character recognition and medical diagnosis) to more recent ones in data mining (such as credit scoring, gene selection, credit card transaction analysis) have increased the research activity in this field.

Key words and phrases: Classification, conditional independences, Markov network, t-cherry probability distribution, informative variables.
2010 Mathematics Subject Classification: 05C65, 60C05, 62H30, 68T37.
1998 CR Categories and Descriptors: G3.

Probabilistic classification is a special branch of pattern recognition. Comprehensive books about probabilistic pattern recognition are Devroy, Györfi and Lugosi [8] and Bishop [1]. A good introductory book about reasoning under uncertainty is Pearl [20].

A simple yet powerful model which uses conditional indepences for the classification task is the Naive Bayes model. It assumes that the random variables associated with the features are conditionally independent given the class variable. Although these independence assumptions are often unrealistic, nevertheless this model prove to be robust and effective for classification across a wide range of applications, see Duda et al [9] and Cheesman and Stutz [5].

Probabilistic graphical modeling languages for representing complex domains, learning these representations from data and algorithms for reasoning using these representations can be found in Koller and Friedman [15].

In this paper we propose a new approach for probabilistic classification. In practice it often occurs that the classification variable depends on a plenty of variables, and these variables depend on each other. Redundant variables in the model may cause overfitting which leads to poor generalization (the performance of the model on a new dataset). First one has to reduce the number of variables involved in the classification as much as possible. For example if in Figure 1. a) the nodes correspond to the indices of the random variables X_1, X_2, X_3, X_4, X_5 then the correlation coefficients between the variables may be: $R(X_1, X_5) = 0.85$, $R(X_2, X_5) = 0.69$, $R(X_3, X_5) = 0.90$, $R(X_4, X_5) = 0.78$. In the same time the following relation may exist between the conditional entropies: $H(X_5|X_1, X_2, X_3, X_4) = H(X_5|X_3, X_4) < H(X_5|X_1, X_3)$. Here the conditional entropy $H(X_5|X_1, X_3)$ for example is defined as

$$-\sum P(X_1 = x_1, X_3 = x_3, X_5 = x_5) \log P(X_5 = x_5|X_1 = x_1, X_3 = x_3),$$

where the summation is running over all possible x_1, x_3, x_5 values of the random variables X_1, X_3, X_5. Although X_5 has the largest two correlations with the random variables X_1 and X_3, the entropy of X_5 may be reduced more considerably by conditioning on the random variables X_3, X_4 than conditioning on the random variables X_1, X_3. For example this may happen if X_1 and X_3 has relatively large correlation while X_3 and X_4 are independent or conditionally independent given X_5. If now X_5 is considered as classification variable then it is enough to observe the variables X_3 and X_4, only. In this case we say that X_5 "directly depends" on the variables X_3 and X_4 and "indirectly dependends" on the variables X_1 and X_2. The same can be formulated also as $X_5 \perp X_1|X_3, X_4$ and $X_5 \perp X_2|X_3, X_4$ where "$\perp$" is used for the notation of conditional independence. In the graph of Figure 1. a) the edges represent the direct dependences between X_1, X_2, X_3, X_4, X_5.

This example shows the importance of finding those variables which influence directly the classification variable and which should be included in the

classifier. This task is achieved by an unsupervised learning which discovers the "direct dependence" structure between the variables. Roughly speaking this represents the Markov network underlying the variables. In order to discover the Markov network we fit a probability distribution to the data. It belongs to a special class of probability distributions called t-cherry probability distributions introduced in [16] and [22]. On the base of the best fitting t-cherry distribution we select those variables, on which the classifier depends directly. These variables will be included in the classifier, and then will be trained by supervised learning on the training dataset. Summing up our approach consists of two phases. The first one is an unsupervised learning which aims to reveal the direct dependence structure and the second one is a supervised learning, which learns the classifier from the training dataset.

This paper consists of five sections. The second section of the paper gives a short mathematical background of Markov networks and t-cherry junction trees. In the third section we define the concept of t-informative variables and define our probabilistic classifier. In the fourth section we present numerical experiments for six datasets from the UCI Repository of Machine Learning [11]. The last section lists some conclusions of the present paper.

2. Markov network, t-cherry junction tree

Let $X = \{X_i\}_{i=1,\ldots,d}$ be a set of discrete finite random variables over the same probability space. Let Λ_i denote the set of values of X_i, and let $V = \{1, \ldots, d\}$ be the set of indices. The Markov network links the joint probability distribution to a graph based on the conditional independence structure between the random variables.

First we give a short review of the concepts on graphs that are used in this paper. Let $V = \{1, \ldots, d\}$ be a set of vertices and Γ a set of subsets of V called *set of hyperedges*. A *hypergraph* consists of a set V of vertices and a set Γ of hyperedges. We denote a hyperedge by C_i, where C_i is a subset of V. Two vertices in the same hyperedge are connected, which means the hyperedge of a hyperhraph is a complete graph (a clique) on the set of vertices contained in it.

The *acyclic hypergraph* is a special type of hypergraphs which fulfills the following requirements:

- Neither of the edges of Γ is a subset of another edge.
- There exists a numbering of edges for which the *running intersection property* is fullfiled: $\forall j \geq 2 \quad \exists\, i < j : \; C_i \supset C_j \cap (C_1 \cup \ldots \cup C_{j-1})$.

(Other formulation is that for all hyperedges C_i and C_j with $i < j-1$, $C_i \cap C_j \subset C_s$ for all $s, i < s < j$.)

Let $S_j = C_j \cap (C_1 \cup \ldots \cup C_{j-1})$, for $j > 1$ and $S_1 = \phi$. Let $R_j = C_j \backslash S_j$. We say that S_j *separates* R_j from $(C_1 \cup \ldots \cup C_{j-1}) \backslash S_j$, and call S_j separator set or shortly separator.

Now we link these concepts to the terminology of junction trees.

The junction tree is a special tree stucture which is equivalent to the connected acyclic hypergraphs [19]. The nodes of the tree correspond to the hyperedges of the connected acyclic hypergraph and are called clusters, the edges of the tree correspond to the separator sets and called separators. The set of all clusters is denoted by $\mathcal{C}$, the set of all separators is denoted by $\mathcal{S}$. The junction tree with the largest cluster containing k variables is called *k-width junction tree*.

An important relation between graphs and hypergraphs is given in [19]. A hypergraph is acyclic if and only if it can be considered to be the set of cliques of a triangulated graph (a graph is triangulated if every cycle of length greater than 4 has a chord). In Figure 1 we show an example.

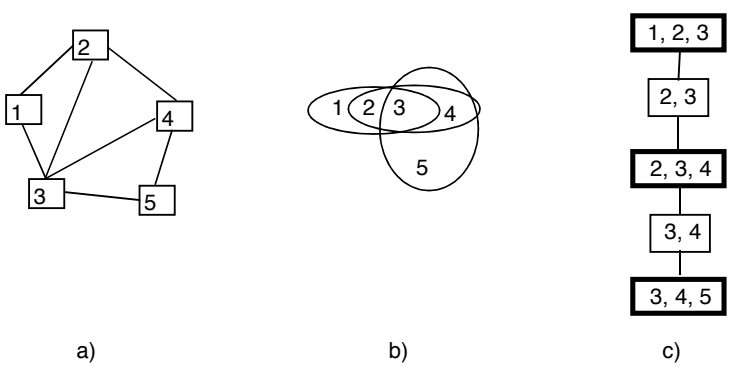

Figure 1. a) triangulated graph, b) the corresponding acyclic hypergraph, c) the corresponding junction tree

Now we link a graph to the joint probability distribution. We consider the random vector $\mathbf{X} = (X_1, \ldots, X_d)^T$, with the set of indices $V = \{1, \ldots, d\}$. We denote the random vector $(X_{i_1}, \ldots, X_{i_k})^T$ by $\mathbf{X}_A$ and the set of random variables $\{X_{i_1}, \ldots, X_{i_k}\}$ with $\{i_1, \ldots, i_k\} = A$ by X_A. The graph structure associated to a Markov network consists of the set of vertices V and the set of edges $E = \{(i,j) \,|\, i, j \in V\}$. We say that the Markov network has

- the *pairwise Markov (PM)* property if $\forall i, j \in V$, i not connected to j implies that X_i and X_j are conditionally independent given all the other random variables;

- the *local Markov (LM)* property if $\forall i \in V$, and $Ne(i)$ the neighbourhood of node i in the graph (the nodes connected with i) X_i is conditionally independent from all X_j, $j \notin Ne(i)$, given $X_k, k \in Ne(i)$;

- the *global Markov (GM)* property if $\forall A, B, C \subset V$ when C separates A and B in graphtheoretical sense then $\mathbf{X}_A$ and $\mathbf{X}_B$ are conditionally independent given $\mathbf{X}_C$, which means in terms of probabilities that

$$P(\mathbf{X}_{A\cup B\cup C}) = \frac{P(\mathbf{X}_{A\cup C})\, P(\mathbf{X}_{B\cup C})}{P(\mathbf{X}_C)};$$

- the *factorization (F)* property if $\mathcal{C}$ denotes the set of cliques of the graph (maximum complete graphs) then there exist positive functions $\psi_C(\mathbf{X}_C)$ such that

$$P(\mathbf{X}_V) = \prod_{C\in\mathcal{C}} \psi_C(\mathbf{X}_C).$$

The following implication is well known [13]: $F \Rightarrow GM \Rightarrow LM \Rightarrow PM$. The Hammersley-Clifford theorem states that if the random vector $\mathbf{X}$ takes on all possible combinations of the values of random variables $X_1, \ldots, X_d$ with positive probability (positivity condition) then $PM \Longrightarrow F$. The positivity is a very strong condition and as the authors claim in [13]: "The positivity condition is mathematically convenient; But it hardly seems necessary". In this paper we focus on Markov networks characterized by the global Markov property.

The concept of *junction tree probability distribution* is related to the junction tree graph and to the global Markov property. A junction tree probability distribution is defined as a fraction of products of marginal probability distributions as follows:

$$P(\mathbf{X}) = \frac{\prod_{C\in\mathcal{C}} P(\mathbf{X}_C)}{\prod_{S\in\mathcal{S}} [P(\mathbf{X}_S)]^{\nu_S - 1}},$$

where $\mathcal{C}$ is the set of clusters of the junction tree, $\mathcal{S}$ is the set of separators, ν_S is the number of those clusters which contain the separator S.

Example 2.1. The probability distribution corresponding to the example in Figure 1 is:

$$P(\mathbf{X}) = \frac{P\left(\mathbf{X}_{\{1,2,3\}}\right) P\left(\mathbf{X}_{\{2,3,4\}}\right) P\left(\mathbf{X}_{\{3,4,5\}}\right)}{P\left(\mathbf{X}_{\{2,3\}}\right) P\left(\mathbf{X}_{\{3,4\}}\right)} =$$

$$= \frac{P(X_1, X_2, X_3)\, P(X_2, X_3, X_4)\, P(X_3, X_4, X_5)}{P(X_2, X_3)\, P(X_3, X_4)}.$$

In our paper [21] we introduced a special kind of k-width junction tree, called k-th order "t-cherry junction tree"[1] in order to approximate a joint probability distribution. The k-th order t-cherry junction tree probability distribution is associated with the k-th order t-cherry tree, introduced in [3], [4].

Definition 2.1. The recursive construction of the *k-th order t-cherry tree*:

(i) A complete graph of k vertices from V represents the smallest k-th order t-cherry tree. This set of k vertices is called the first hypercherry.

(ii) By connecting a new vertex $i_k \in V$ with all $\{i_1, \ldots, i_{k-1}\}$ vertices of a $(k-1)$- dimensional complete subgraph of the existing k-th order t-cherry tree, we obtain a new k-th order t-cherry tree. $\{\{i_k\}\{i_1, \ldots, i_{k-1}\}\}$ is called k-th order hypercherry. $V := V \setminus \{i_k\}$.

(iii) A k-th order t-cherry tree can be obtained from (i) by successive application of (ii) till V becomes empty.

Remark 2.1. The k-th order t-cherry tree is a special triangulated graph therefore a junction tree structure is associated with it.

Definition 2.2. The *k-th order t-cherry junction tree* is defined in the following way:

- By using Definition 2.1 we construct a k-th order t-cherry tree over V.
- To each hypercherry there a cluster of the junction tree is assigned.
- To each hypercherry of the form $\{\{i_k\}\{i_1, \ldots, i_{k-1}\}\}$ we assign the set $\{i_1, \ldots, i_{k-1}\}$ and call it separator set.

We denote by $\mathcal{C}_{\text{ch}}$, and $\mathcal{S}_{\text{ch}}$, the set of clusters and separators of the t-cherry junction tree.

[1] t-cherry tree is the name of the special graph structure behind this junction tree.

Definition 2.3. If the indices of the random vector $\mathbf{X}^T = (X_1, \ldots, X_d)$ are assigned to a t-cherry junction tree structure then the probability distribution is called *t-cherry junction tree probability distribution* (shortly t-cherry pd) and is given by:

$$P_{\text{t-ch}}(\mathbf{X}) = \frac{\prod\limits_{C \in \mathcal{C}_{\text{ch}}} P(\mathbf{X}_C)}{\prod\limits_{S \in \mathcal{S}_{\text{ch}}} (P(\mathbf{X}_S))^{\nu_s - 1}},$$

where the probability distributions involved in the above formula are marginal probability distributions of $P(\mathbf{X})$.

Example 2.1 shows a 3-rd order t-cherry junction tree probability distribution.

3. The classification problem in terms of *t*-cherry junction trees

First we list some notations and assumptions introduced in [8].

Let $(\mathbf{X}, Y)^T$ be an $R^d \times \{1, \ldots, M\}$ valued random vector. A classifier is constructed on the basis of a training set containing n vectors $\left(\mathbf{x}^1, y^1\right), \ldots, (\mathbf{x}^n, y^n)$ and is denoted by g_n. For a given $\mathbf{x} \in R^d$ the value $y \in \{1, \ldots, M\}$ of Y is guessed by $g_n\left(\mathbf{x}; \left(\mathbf{x}^1, y^1\right), \ldots, (\mathbf{x}^n, y^n)\right)$.

So the classifier g_n is a function:

$$g_n : R^d \times \left\{R^d \times \{1, \ldots, M\}\right\} \longrightarrow \{1, \ldots, M\}.$$

The construction of g_n in this way is called supervised learning.

We assume that $\left(\mathbf{X}^1, Y^1\right)^T, \ldots, (\mathbf{X}^n, Y^n)^T$ is a sequence of independent identically distributed random vectors having the same distribution as $(\mathbf{X}, Y)^T$.

The training dataset may be the result of experimental observations (meteorological data, ECG data, ...) The y^i values can be obtained through measurements or through an expert who filled out the y^i values after having the realizations of the $\mathbf{x}^i$ vectors.

The performance of the classifier g_n is measured by the probability of error occurrence:

$$L_n = L(g_n) = P\left\{g_n\left(\mathbf{X}; \left(\mathbf{x}^1, y^1\right), \ldots, (\mathbf{x}^n, y^n)\right) \neq Y\right\}.$$

The best possible classifier is defined by:

$$g^* = \arg\min_{g:R^d \to \{1,\ldots,M\}} P\left(g\left(\mathbf{X}\right) \neq Y\right).$$

We note that g^* depends on the probability distribution of the random vector $\left(\mathbf{X}, Y\right)^T$ which usually is unknown.

In the approach presented here we exploit some of the conditional independences to construct a good approximation of the probability distribution of $\left(\mathbf{X}, Y\right)^T$. In order to do this we search for the best fitting k-th order t-cherry probability distribution. At first sight the search space may seem to be very special, however in [21] we proved that finding the best fitting k-th order t-cherry probability distribution is the best choice among all k-width junction tree probability distributions.

We search for the best-fitting probability distribution to the training data by minimizing the Kullback-Leibler divergence ([18]):

$$KL = \sum_{\mathbf{x},y} P\left(\left(\mathbf{X}, Y\right)\right) \log_2 \frac{P\left(\left(\mathbf{X}, Y\right)\right)}{P_{app}\left(\left(\mathbf{X}, Y\right)\right)}.$$

From now on the random variable Y will be denoted also by X^{d+1} for notational convenience.

The method for finding a k-th order t-cherry junction tree using this criterion is presented in [21] and [22].

On the basis of the t-cherry junction tree obtained by minimizing the K-L divergence we define a special type of informative variables in terms of the t-cherry junction tree.

Definition 3.1. The set of variables

$$X_{t\text{-info}} = \left\{ \bigcup_{C \in \mathcal{C}_{ch}} X_C \mid d+1 \in C \right\}$$

are called t-informative variables for X_{d+1}.

The t-informative variables are the variables contained in those clusters which contain the classification variable X_{d+1}. Because of the running intersection property these clusters form a junction tree.

Algorithm 3.1. Selection of the t-informative features (unsupervised learning).

1. Give as input the training dataset in discretized form. (A method for discretization was proposed in paper [17].)

2. From the discretized probability distribution determine all the k-th order marginals (it is favorable if k is not too large).

3. Find the heaviest weighted tree, in the sense (see [21] and [22]):

$$\sum_{(X_{i_1},\ldots,X_{i_c})\in\mathcal{C}_{ch}} I\left(X_{i_1},\ldots,X_{i_c}\right) - \sum_{(X_{j_1},\ldots,X_{j_s})\in\mathcal{S}_{ch}} \left(\nu_{j_1,\ldots,j_s}-1\right) I\left(X_{j_1},\ldots,X_{j_s}\right) \rightarrow \max,$$

where $I\left(X_{i_1},\ldots,X_{i_c}\right)$ is the information content of $P\left(X_{i_1},\ldots,X_{i_c}\right)$ (see [7]).

4. Output: the set of clusters $\mathcal{C}$ and the set of separators $\mathcal{S}$.

5. Select those clusters which contain the variable $X_{d+1}(=Y)$.

6. Select those variables which occur in the clusters selected in step 5 as t-informative variables.

In Figure 1 the t-informative clusters for the variable X_4 are (X_2, X_3, X_4) and (X_3, X_4, X_5). The set of t-informative variables are X_2, X_3 and X_5.

The joint probability distribution of the random vector $\left(\mathbf{X}_{t\text{-info}}, X_{d+1}\right)^T$ will be denoted by $P_{t\text{-info}}\left(\left(\mathbf{X}_{t\text{-info}}; X_{d+1}\right)\right)$. Then we have

$$P_{t\text{-info}}\left(\mathbf{X}_{t\text{-info}}; X_{d+1}\right) = \frac{\prod\limits_{(i_1,\ldots,i_c)\subset\mathcal{C}_{t\text{-info}}} P\left(X_{i_1},\ldots,X_{i_c}\right)}{\prod\limits_{(i_1,\ldots,i_s)\subset\mathcal{S}_{t\text{-info}}} P\left(X_{j_1},\ldots,X_{j_s}\right)^{\left(\nu_{j_1,\ldots,j_s}-1\right)}}, \tag{3.1}$$

where $\nu_{j_1,\ldots,j_s} = \#\left\{\{X_{j_1},\ldots,X_{j_s}\}\subset C \mid C\in\mathcal{C}_{t\text{-info}}\right\}$.

For example for the t-cherry tree of Figure 1 and for the classification variable X_4:

$$P_{t\text{-info}}\left(X_2, X_3, X_5; X_4\right) = \frac{P\left(X_2, X_3, X_4\right) P\left(X_3, X_4, X_5\right)}{P\left(X_3, X_4\right)},$$

Remark 3.1. If the number of selected t-informative variables is not too large then one can use their joint marginal probability distribution. This can be determined from the probability distribution underlying the training dataset. Else it is useful to apply formula (3.1).

Now we are going to define the concept of the *conditional t-informative classifier*.

In the following formulas we will refer to $P_{t\text{-info}}\left((\mathbf{X}_{t\text{-info}}, X_{d+1})\right)$ as $P_{t\text{-info}}\left(X_{i_1}, \ldots, \ X_{i_r}, Y\right)$. This probability is the corresponding marginal probability distribution determined from the training dataset or its approximation given by (3.1).

In order to introduce the classifier first we introduce some notations.

We suppose that we have a set of training vectors : $\{(\mathbf{x}^t, y^t)\}_{t\in\{1,\ldots,n\}}$. Usually $n > \frac{4}{5}N$, where N is the total number of the available observation vectors. These can be selected randomly from the given dataset at the beginnig of the analysis. The remaining data constitute the test dataset.

It may happen that a vector from the test dataset does not occur in the training dataset. Hence we define the following set of marginal probability distributions.

Let

$$\begin{aligned}\mathcal{P}^{i,k}_{t\text{-info}}\left(x^t_{i_1}, \ldots, x^t_{i_r}\right) \quad = \quad & \Big\{P_{t\text{-info}}\left(X_{j_1} = x^t_{j_1}, \ldots, X_{j_k} = x^t_{j_k}, Y = i\right) \mid \\ & \{X_{j_1}, \ldots, X_{j_k}\} \subset X_{t\text{-info}}, \\ & P_{t\text{-info}}\left(X_{j_1} = x^t_{j_1}, \ldots, X_{j_k} = x^t_{j_k}, Y = i\right) > 0\Big\}\end{aligned}$$

and

$$k^* = \max\left\{k \mid k \in \{1, \ldots, r\}\,;\ \mathcal{P}^{i,k}_{t\text{-info}}\left(x^t_{i_1}, \ldots, x^t_{i_r}\right) \neq \phi\right\}.$$

$\mathcal{P}^{i,k^*}_{t\text{-info}}$ is in fact the set of the largest order marginals of $\mathcal{P}^{i,k}_{t\text{-info}}\left(x^t_{i_1}, \ldots, x^t_{i_r}\right)$ which are positive for a given realization $\left(x^t_{i_1}, \ldots, x^t_{i_r}\right)$.

We define the following discrete random variable:

$$\gamma^{k^*}_{x^t_{i_1}, \ldots, x^t_{i_r}} : \quad \begin{pmatrix} 1 & \ldots & i & \ldots & M \\ p_1 & \ldots & p_i & \ldots & p_M \end{pmatrix}$$

where

$$p_i = \frac{\sum\limits_{\mathcal{P}^{i,k^*}_{t\text{-info}}\left(x^t_{i_1}, \ldots, x^t_{i_r}\right)} P_{t\text{-info}}\left(X_{j_1} = x^t_{j_1}, \ldots, X_{j_{k^*}} = x^t_{j_{k^*}}, Y = i\right)}{\sum\limits_{\mathcal{P}^{i,k^*}_{t\text{-info}}\left(x^t_{i_1}, \ldots, x^t_{i_r}\right)} P_{t\text{-info}}\left(X_{j_1} = x^t_{j_1}, \ldots, X_{j_{k^*}} = x^t_{j_{k^*}}\right)}, i = 1, \ldots, M.$$

Remark 3.2. It is easy to see that $\sum\limits_{i=1}^{M} p_i = 1$.

Definition 3.2. The classifier $g^{(r)}_{n,t\text{-info}}$ is defined in the following way:

$$g^{(r)}_{n,t\text{-info}} : R^d \times \{R^r \times \{1, \ldots, M\}\} \longrightarrow \{1, \ldots, M\}, r < d$$

$$g^{(r)}_{n,t\text{-info}}\left(x_1, \ldots, x_d\right) = \arg\max_{i\in\{1,\ldots,M\}} P\left(\gamma^{k^*}_{x_{i_1}, \ldots, x_{i_r}} = i\right).$$

The classifier defined in this way is obtained by supervised learning.

Remark 3.3. One can see from this definition that for classifying a new d-dimensional vector we use a number of r t-informative features only.

We call the classifier defined in Definition 3.2 *conditional t-informative classifier*.

Definition 3.3. The performance of the classifier $g^{(r)}_{n,t\text{-info}}$ is measured by the conditional probability of error:

$$L_n = L\left(g^{(r)}_{n,t\text{-info}}(\mathbf{X})\right) = P\left(g^{(r)}_{n,t\text{-info}}(\mathbf{X}) \neq Y \mid \left(\mathbf{x}^1, y^1\right), \ldots, \left(\mathbf{x}^n, y^n\right)\right).$$

In practice the probability of error can be approximated by the relative frequency of errors in the set of test data vectors. The probability of accuracy is defined by $1 - L_n$.

4. Numerical experiments with the UCI Machine Learning Repository

In this section we present numerical experiments according to six classification problems. We used datasets from the Repository of Machine Learning Databases maintained by the University of California at Irvine [11].

The accuracy of our approach was estimated by randomly selecting a subset of the dataset, called training set. The training set consisted of 80% of the observations. The remaining part of the observations was used as a testing dataset. On the testing dataset we counted the ratio of the right classifications when applying our method. This procedure was repeated 20 times with randomly selected training sets. The average accuracy of our procedure was obtained as the mean value of the ratios of right classifications, obtained for each experiment. In the following we give brief description of the datasets. We compare our results with those published in [2], [10] and [14].

1. *Australian Credit Card* (ACC): The dataset, submitted to the repository by Quinlan, consists of 690 records of Master Card applicants, 307 of which are classified as positive and 383 as negative. While 37 records have some missing data, they were not omitted from our analysis. Each record consists of 15 features, which corresponds to random variables.

It is very interesting that "this problem detect a degree 1 pattern consisting of the ninth attribute alone, which correctly predicts the outcome in 85.1% of the cases" (see [2]). Using our conditional t-informative (CI) approach we got similar result. We also found, that by adding features to the t-informative features the prediction accuracy does not improve what is in accordance with the results in [2].

2. *Wisconsin Breast Cancer* (WBC): The dataset, compiled by Mangasarian and Benett, is widely used in machine learning community for comparing machine learning algorithms. Each observation describes a cytological test specified by nine numerical attributes, which were obtained from 31 attributes. Currently the dataset consists of 699 observations; out of which 683 are complete (have no missing data). We use just these records in our experiment. The classification task is "malign" or "benign".

3. *Congressional Voting* (CV): The dataset, contributed by Schlimmer, includes votes for each of US House of Representative Congressmen in the 98th Congress, second session of 1984, on 16 key issues identified by Congressional Quarterly Almanac (CQA, Volume XL). The dataset contains 435 data (267 Democrats and 168 Republicans). The task is to identify the party membership on the bases of the result of the 16 votes. We use the whole dataset, despite of Boros et al. who removed from the dataset of observations corresponding to six congressmen who did not vote on most of the issues.

4. *Diabetes* (D): This dataset, compiled by the national Institute of Diabetes and digestive and Kidney disease, was contributed to the repository by Sigilito. The dataset consists of 768 complete observations, about 2/3 of which correspond to patients showing signs of diabetes and the rest exhibiting no such signs. Each patient is described by 8 numerical attributes.

5. *Cleveland Heart Disease* (CHD): The Cleveland Clinic Foundation heart disease dataset, contributed to the repository by Detrano, contains 303 observations, 165 of which describe healthy people and 138 sick ones. Each patient is described by 13 attributes.

6. *German Data* (GD): The dataset was provided by Prof. Hans Hofmann, Institut für Statistik und Ökonometrie Universität Hamburg. It contains categorical/symbolic attributes. Each customer is characterized by 24 attributes. The dataset consists of 1000 observations.

In Table 1 the correct prediction rates and their standard deviations obtained by CtIC (Conditional t-Informative Classifier), by LAD (Logical Analysis of Data) ([2]) and by GA (Genetic Algorithm) ([10]) are given. Paper [2] has

no results according to the German Data dataset and paper [10] has no results according to the Wisconsin Breast Cancer and Congressional Voting datasets. The average prediction rates were calculated over 20 runs for CtIC and LAD, and over 10 runs for GA. The k values given according to CtIC mean the order of the t-cherry tree applied in the construction of the set of t-informative variables.

	CtIC	k	LAD	GA
ACC	**86.6% (2.5)**	3	85.5% (2.6)	86.3% (0.8)
WBC	95.9% (1.5)	2	**97.2% (1.3)**	–
CV	**96.7% (1.9)**	3	96.6% (2.1)	–
D	**75.4% (3.9)**	2	72.2% (4.3)	74.1% (0.5)
CHD	**84.6% (5.5)**	3	83.8% (6.0)	82.3% (7.1)
GD	**73.5% (3.1)**	3	–	72.9% (0.8)

Table 1. Correct prediction rates (in percent) on the test problems

For the German Data dataset other approaches using SVM combined with Grid, F-score and GA (see [14]) found 76.0%, 77.5% and 77.9% classification accuracies with standard deviations 3.86, 4.03 and 3.97, respectively.

These experiments highlight that our Conditional t-Informative Classifier is competitive with the well-established classifiers published earlier.

5. Conclusions

In this paper we introduced a classification method which by using some of the conditional independences between the features diminishes the number of informative variables which have to be taken into consideration. Unlike other approaches exploiting the conditional independeces, like for example K2 ([6]), which discovers the Bayesian network, our approach does not require the ordering of the variables (which usually needs collaboration with experts).

Our approach uses just the k-th order marginals obtained from the training data. Using their information content we were able to fit the t-cherry probability distribution to our data. Based on this concept we determined the t-informative variables which were included in the classifier. In this way we reduced the dimensionality of the problem in order to avoid the phenomenon of overfitting. Discovering the dependence structure makes also possible to obtain informative variables for more than one classification variables in the same time. This allows the efficient structuring of databases, too.

References

[1] **Bishop, C.,** *Pattern Recognition and Machine Learning*, Springer, Berlin, 2006.

[2] **Boros, E., P.L. Hammer, T. Ibaraki and A. Kogan,** An Implementation of Logical Analysis of Data, *IEEE Transactions on Konowledge and Data Engineering* **12** (2000), 292–306.

[3] **Bukszár, J. and A. Prékopa,** Probability Bounds with Cherry Trees, *Mathematics of Operational Research* **26** (2001), 174–192.

[4] **Bukszár, J. and T. Szántai,** Probability Bounds given by hypercherry trees, *Optimization Methods and Software*, **17** (2002), 409–422.

[5] **Cheeseman, P. and J. Stutz,** Bayesian Classification (AutoClass): Theory and Results, In Fayyad U., Piatesky-Shapiro G., Smyth P. and Uthuruasmy R., eds *Recent Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA, 1995, pp 153–180.

[6] **Cooper, G.F. and E. Herskovits,** A Bayesian Method for the Induction of Probabilistic Networks from Data, *Machine Learning*, **9** (1992), 309–347.

[7] **Cover, T.M. and J.A. Thomas,** *Elements of Information Theory*, Wiley Interscience, New York, 1991.

[8] **Devroye, L., L. Györfi and G. Lugosi,** *A Probabilistic Theory of Pattern Recognition*, Springer, New York, 1996.

[9] **Duda, R.O., P.E. Hart and D.G. Stork** *Pattern Classification*, John Wiley & Sons, New York, 2000.

[10] **Eggermont, J., J. Kok and W. Kosters,** Genetic Programming for Data Classification: Partitioning the Search Space, in: *Proceedings of the 2004 Symposium on Applied Computing* (ACM SAC'04)

[11] **Frank, A. and A. Asuncion,** *UCI Machine Learning Repository* [`http://archive.ics.uci.edu/ml`]. Irvine, CA: University of California, School of Information and Computer Science, (2010).

[12] **Guyon, I., N. Matic and V. Vapnik,** Discovering informative patterns and data cleaning, *Advances in Knowledge Discovery and Data Mining,* AAAI, USA, (1996), 181–203.

[13] **Hammersley, J.M.,** *Markov fields on finite graphs and lattices*, not published, (1971)

[14] **Huang, Cheng-Lung, Mu-Chen Chen and Chieh-Jen Wang,** Credit scoring with a data mining approach based on support vector machines, *Expert Systems with Applications* **33** (2007) 847–856.

[15] **Koller, D. and N. Friedman,** *Probabilistic Graphical Models*, MIT Press, Cambridge, Massachusetts, London, 2009.

[16] **Kovács, E. and T. Szántai,** On the approximation of discrete multivariate probability distribution using the new concept of t-cherry junction tree, in: K. Marti, Y. Ermoliev and M. Makowski(Eds.) *Proceedings of the IFIP/IIASA/GAMM-Workshop on "Coping with Uncertainty"* (IIASA, Laxenburg, 2007), Lecture Notes in Economics and Mathematical Systems **633**, *Coping with Uncertainty: Robust Solutions*, Springer, Heidelberg, 2008, pp. 39–56.

[17] **Kovács, E. and T. Szántai,** Multivariate copula expressed by lower dimensional copulas, arXiv:1009.2898v1.

[18] **Kullback, S.,** *Information Theory and Statistics*, Wiley and Sons, New York, 1959.

[19] **Lauritzen, S. L. and D.J. Spiegelhalter,** Local Computations with Probabilites on Graphical Structures and their Application to Expert Systems, *J.R. Statist. Soc. B*, **50** (1988), 157–227.

[20] **Pearl, J.,** *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, 1988.

[21] **Szántai, T. and E. Kovács,** Hypergraphs as a mean of discovering the dependence structure of a discrete multivariate probability distribution, in: *Proc. Conference Applied Mathematical Programming and Modelling* (APMOD), Bratislava, 27-31 May 2008, *Annals of Operations Research*, **193** (2012), 71–90.

[22] **Szántai, T. and E. Kovács,** Discovering a junction tree behind a Markov network by a greedy algorithm, arXiv:1104.2762.

E. Kovács
ÁVF Budapest College of Management
H-1114 Budapest, Villányi út 11-13.
Hungary
kovacs.edith@avf.hu

T. Szántai
Department of Differential Equations
Institute of Mathematics
Budapest University of Technology and Economics
H-1111 Budapest, Egry József utca 1
Hungary
szantai@math.bme.hu