Analele Universităţii de Vest, Timişoara
Seria Matematică – Informatică
XLVII, 3, (2009), 101– 115

# Analyzing Markov Random Fields using $t$-cherry junction tree structures

Edith Kovács, Tamás Szántai

*Dedicated to Professor Gheorghe Constantin on the occasion of his 70th anniversar*

**Abstract.** The approximation of the multivariate probability distribution is a popular problem in many fields. In this paper we give a survey of our results related to this problem. Our approach exploits the conditional independences between the variables. We present here some new concepts and our most important results. In the last part there will be given an application of our new results in pattern recognition, namely in recognizing of Parkinson's disease based on voice disorder.

**AMS Subject Classification (2000).** 62-09 ; 62H30 ; 62M40 ; 68T10
**Keywords.** Multivariate probability distribution, Markov Random Field, junction tree, speech recognition

## 1 Introduction

The modeling of multivariate probability distribution is a central problem for many fields. One of the widely used approaches is the copula approach [5] and [11]. Copulas offer a flexible method for combining marginal distributions into a multivariate distribution, capturing various types of dependences. Also the copula approach is useful in quantifying dependence between the random variables involved.

Another method for modeling a multivariate probability distribution uses the Markov Random Field (MRF), which has its roots in statistical physics. The standard stochastic model for spatially distributed systems is the random field over multidimensional lattice $Z^d$. The best-known of these lattice systems is the Ising model, invented in 1925 by E. Ising [10] to help explain ferro magnetism. An MRF encodes in a graph structure the conditional independences. The theory of random fields has several applications such as complex engineering, physics, economics, pattern recognition, biology. In this respect, the finding of convenient means of description of the random field poses an important problem.

In order to approximate the probability distribution associated to an MRF we introduced the $t$-cherry junction tree in [12] and the $k$-th order $t$-cherry junction tree in [17]. This survey presents this new approach and some of our most important results.

The paper is structured as follows. The second part contains a short review of Markov random fields and junction trees. In the third part we present the concept of a special hyper graph structure introduced in [2] called $t$-cherry tree and in [3] called $t$-cherry hyper tree. In the fourth part we give the concept of the $t$-cherry junction tree and present some of our essential results. In the fifth part we show an application of this approach in pattern recognition.

## 2 Markov Random Field and junction tree

Let $G\left(V,E\right)$ be a finite graph, where $V=\{1,\ldots,n\}$ is the set of vertices, and $E$ is the set of edges. We denote by $(i,k)$ the edge connecting the vertices $i$ and $k$.

**Definition 2.1.** *The neighborhood of vertex $k$ is the set $N\left(k\right)=\{i \mid (i,k) \in E\}$ that is the set of vertices connected by edges with the vertex $k$.*

A simple undirected graph can be equivalently defined by its set of vertices V and a neighborhood set $\mathcal{N}=\{N(k) \mid k\in V\}$, which associates to each vertex its neighbors.

**Definition 2.2.** [1] *The couple $(V,\mathcal{N})$ is called topology.*

Let $X=\{X_i\}_{i=1,\ldots,n}$ be a set of discrete finite random variables over the same $(\Omega,\mathcal{A},P)$ probability space. Let $\Lambda_i$ denote the set of values of $X_i$, and $V=\{1,\ldots,n\}$ the set of indices. In physical literature the set $\Lambda_i$ is called phase space, and the index set $V$, sometimes denoted by $S$, is called set of sites.

**Definition 2.3.** [1] *A random field on $V$ with phase space in $\prod_{i=1}^{n} \Lambda_i$ is a collection $X = \{X_i\}_{i=1,\ldots,n}$ of random variables $X_i$ with values in $\Lambda_i$.*

**Definition 2.4.** [1] *A random field is called Markov random field (MRF) with respect to the neighborhood system $\mathcal{N}$ if $\forall i \in V$, $X_i$ is independent from $X - \{X_j\}_{j \in V \setminus \{N(i) \cup \{i\}\}}$ given $\{X_j\}_{j \in N(i)}$.*

**Definition 2.5.** [18] *For a variable $X_i$ the set $\{X_j\}_{j \in N(i)}$ is called set of informative variable.*

If we are interested in a variable $X_i$ it is enough to study or model the probability distribution of this variable together with its neighborhood. This diminishes the number of variables involved in the model. For example if one is interested in classification then it is useful to find the set of informative variables for the classifying variable. For this purpose one needs to discover the MRF, that in practice is mostly unknown.

As an example see the Markov random field of Figure 1. The informative variables for $X_6$ are $X_1, X_3, X_4$.

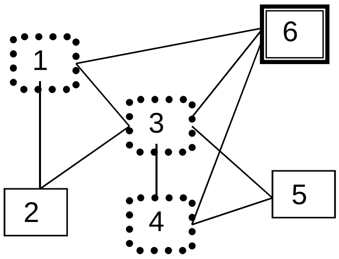

Figure 1: Markov random field over the set of variables with indices in $V = \{1, 2, \ldots, 6\}$

Throughout the paper we will use the following popular notation:

$$\sum_{\mathbf{X}} P(\mathbf{X}) = \sum_{i_1=1}^{m_1} ... \sum_{i_n=1}^{m_n} P\left(X_1 = x_1^{i_1}, ..., X_n = x_n^{i_n}\right).$$

where $x_k^{i_k}, \ i_k = 1, ..., m_k$ are the possible values of the random variable $X_k, \ k = 1, ..., n$. Apply similar notation for products, too.
The junction tree is a special tree with nodes containing random variables:

**Definition 2.6.** [17] *A tree which fulfills the following properties is called junction tree over* $\boldsymbol{X}$*.*

1) *To each node of the tree, a subset of $X$ called cluster and the marginal probability distribution of these variables is associated.*

2) *To each edge connecting two clusters of the tree the subset of $X$, given by the intersection of the connected clusters, and the marginal probability distribution of these variables is associated.*

3) *If two clusters contain a random variable, than all clusters on the path between these two clusters contain this random variable (running intersection property).*

4) *The union of all clusters is $X$.*

The junction tree provides a joint probability distribution of $\mathbf{X}$:

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

where $\mathcal{C}$ is the set of clusters and $\mathcal{S}$ is the set of separators, $P(\mathbf{X}_C)$ and $P(\mathbf{X}_S)$ are the marginal probability distribution associated to the random vector, with variables from the set $C$ and $S$, respectively $\nu_s$ is the number of those clusters which contain all of the variables involved in $S$.

## 3 $t$-cherry hyper tree

For constructing lower and upper bounds on the probability of union or intersection of events there was introduced by J. Bukszár and A. Prékopa [2] the concept of $t$-cherry tree. Later this concept was generalized by J. Bukszár and T. Szántai [3] and called $t$-cherry hyper tree.

**Definition 3.1.** *The recursive construction of the $k$-th order $t$-cherry tree:*

*(i) The complete graph of (k-1) nodes from $V$ represent the smallest $k$-th order $t$-cherry tree.*

*(ii) By connecting a new vertex from $V$, with all vertices of $(k-1)$ dimensional complete subgraph of the existing $k$-th order $t$-cherry tree, we obtain a new $k$-th order $t$-cherry tree.*

*(iii) Each $k$-th order $t$-cherry tree can be obtained from (i) by successive application of (ii).*

**Remark 3.1.** The $k$-th order $t$-cherry tree is a special case of the $k$-uniform hyper graphs introduced by Tomescu [19].

**Definition 3.2.** *The set of vertices of the* $(k-1)$ *dimensional complete subgraph used in step (ii) of Definition 3.1 is called hyper edge of the $k$-th order $t$-cherry tree.*

**Definition 3.3.** *The set of vertices of a hyper edge together with a new vertex is called hypercherry of the $k$-th order $t$-cherry tree.*

The set of hyper edges of the $k$-th order $t$-cherry tree let be denoted by $\varepsilon_{k-1}$. The set of hyper cherries of the $k$-th order $t$-cherry tree let be denoted by $\mathcal{C}_k$. The set of vertices $V$, the set of hyper edges $\varepsilon_{k-1}$and the set of hypercherries $\mathcal{C}_k$ define the $\Delta_k = (V, \varepsilon_{k-1}, \mathcal{C}_k)$ $k$-th order $t$-cherry tree.
Figure 2. illustrates the construction of a $t$-cherry tree.

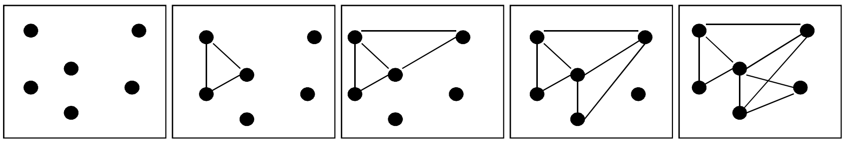

Figure 2: Construction of a 3-rd order $t$-cherry tree

## 4 $t$-cherry junction tree

**Definition 4.1.** *The $k$-th order $t$-cherry junction tree* [17] *can be defined in the following way.*

*1) By using Definition 3.1 we construct a k-th order t-cherry tree over* $V:\ \Delta_k = (V, \epsilon_{k-1}, \mathcal{C}_k)$.

*2) To each hyper cherry* $(\{i_1, \dots, i_{k-1}\}, i_k)$ *we order a cluster set containing the variables* $\left\{X_{i_1}, \dots, X_{i_{k-1}}, X_{i_k}\right\}$.

*3) To each hyper edge* $\{i_1, \dots, i_{k-1}\}$ *we oder a separator set containing the variables* $\{X_{i_1}, \dots,\ X_{i_{k-1}}\}$ *(edge of the junction tree).*

The junction tree provides a joint probability distribution of $\mathbf{X}$:

$$P_k(\mathbf{X}) = \frac{\prod\limits_{\left(X_{i_1},\dots X_{i_k}\right)\in\mathcal{C}} P\left(X_{i_1},\dots X_{i_k}\right)}{\prod\limits_{\left(X_{j_1},\dots,X_{j_{k-1}}\right)\in\mathcal{S}} P\left(X_{j_1},\dots,X_{j_{k-1}}\right)^{\nu_{j_1,\dots,j_{k-1}}-1}}, \tag{4.1}$$

where $\mathcal{C}$ is the set of clusters and $\mathcal{S}$ is the set of separators, $P\left(X_{i_1},\dots X_{i_k}\right)$ and $P\left(X_{j_1},\dots,X_{j_{k-1}}\right)$ are the marginal probability distributions. $\nu_{j_1,\dots,j_{k-1}}$ is the number of those clusters which contain all of the variables $X_{j_1},\dots,$ $X_{j_{k-1}}$.
We emphasize that for $k = 2$ we obtain the so called Chow–Liu dependence tree [4]. The best fitting Chow–Liu dependence tree can be find by the Prim or the Kruskal algorithm.

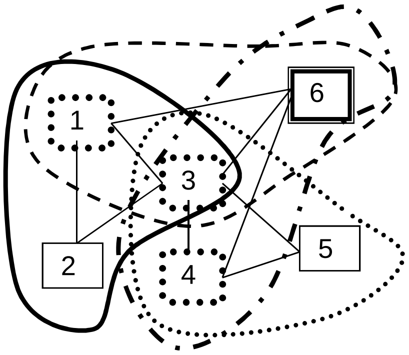

Figure 3: Clusterization of MRF of Figure 1 which leads to the junction tree of Figure 4

The junction tree of Figure 4 provides the real probability distribution of the MRF of Figure 3 (under the assumption that the probability distribution is positive and the variables associated to those pairs of vertices which are connected in the MRF are contained in at least one of the clusters of the junction tree):

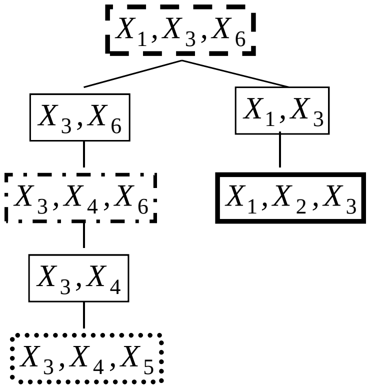

Figure 4: The 3-rd order $t$-cherry junction tree corresponding to the MRF of Figure 3

$$P\left(X_1, X_2, X_3, X_4, X_5; X_6\right) =$$
$$= \frac{P\left(X_1, X_3, X_6\right) P\left(X_3, X_4, X_6\right) P\left(X_1, X_2, X_3\right) P\left(X_3, X_4, X_5\right)}{P\left(X_3, X_6\right) P\left(X_1, X_3\right) P\left(X_3, X_4\right)},$$

In practice the MRF is not known. If the clusters of the junction tree are not containing an edge of the real MRF then the probability distribution associated to the junction tree will be an approximation of the real probability distribution. Therefore we search for the junction tree which gives the best approximation of the real probability distribution.

In order to find the best fitting junction tree we have to minimize the Kullback–Leibler divergence [14] between the approximation and the real probability distribution.

**Theorem 4.1.** [17] *The Kullback- Leibler divergence between the approximation (4.1) and the real distribution $P\left(\mathbf{X}\right)$ is given by:*

$$\begin{aligned} KL\left(P_k\left(\mathbf{X}\right), P\left(\mathbf{X}\right)\right) = -H\left(\mathbf{X}\right) - \Bigg[ \sum_{\left(X_{i_1},\ldots,X_{i_k}\right)\in\mathcal{C}} I\left(X_{i_1},\ldots,X_{i_k}\right) \\ - \sum_{\left(X_{j_1},\ldots,X_{j_{k-1}}\right)\in\mathcal{S}} (\nu_{j_1,\ldots,j_{k-1}}-1) I\left(X_{j_1},\ldots,X_{j_{k-1}}\right) \Bigg] + \sum_{i=1}^{n} H\left(X_i\right), \end{aligned} \tag{4.2}$$

*where $H\left(\mathbf{X}\right)$ denotes the entropy of the random vector $\mathbf{X}$ and $I\left(X_{i_1},\ldots,X_{i_k}\right)$ denotes the information content (see* [6]*) of the random variables $X_{i_1},\ldots,X_{i_k}$.*

It can be seen that in the formula (4.2) the first and the last term does not depend on the junction tree. The difference between the sums of information contents depends on the structure of the junction tree. Therefore it is worthy to introduce the following definition:

**Definition 4.2.** [17] *The difference*

$$\sum_{\left(X_{i_1},\ldots,X_{i_k}\right)\in\mathcal{C}} I\left(X_{i_1},\ldots,X_{i_k}\right) - \sum_{\left(X_{j_1},\ldots,X_{j_{k-1}}\right)\in\mathcal{S}} (\nu_{j_1,\ldots,j_{k-1}}-1) I\left(X_{j_1},\ldots,X_{j_{k-1}}\right)$$

*is called the weight of the junction tree.*

We can see that in order to find a better approximation using junction trees, we have to find the junction tree having the greatest weight.

**Definition 4.3.** *A junction tree containing $k$ variables in its largest cluster is called $k$-width junction tree.*

We denote by $\mathcal{F}_k$ the family of $k$-width junction trees.
We proved in [17] that there exists a $k$-th order $t$-cherry junction tree giving the best approximation among the $k$- width junction trees:

**Theorem 4.2.** *In the class $\mathcal{F}_k$ the $k$-th order $t$-cherry junction tree giving the smallest Kullback-Leibler's divergence provides the best approximation of a given $P\left(\mathbf{X}\right)$ probability distribution.*

To prove this theorem we first transformed a general $k$-width junction tree into a $k$-th order $t$-cherry junction tree (see Algorithm 2 of [17]). Then we proved that the probability distribution associated to the $k$-th order $t$-cherry junction tree obtained by Algorithm 2 gives at least as good approximation

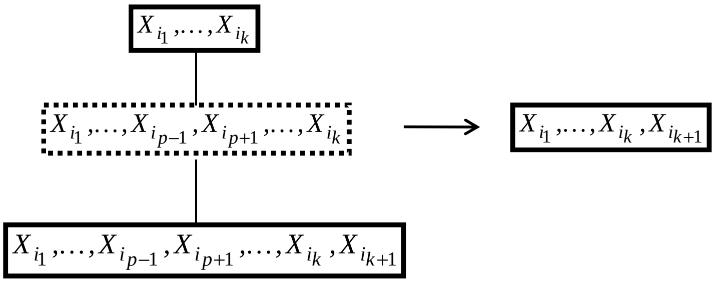
$X_{i_1},\ldots,X_{i_k}$
$X_{i_1},\ldots,X_{i_{p-1}},X_{i_{p+1}},\ldots,X_{i_k}$
$X_{i_1},\ldots,X_{i_{p-1}},X_{i_{p+1}},\ldots,X_{i_k},X_{i_{k+1}}$
$X_{i_1},\ldots,X_{i_k},X_{i_{k+1}}$

Figure 5: The transformation of a simple $k$-th order $t$-cherry junction tree into a cluster which contains $k+1$ variables

of the real probability distribution as the probability distribution associated to the $k$-width junction tree does.

Another important result of our paper [17] is a method for improving $k$-th order $t$-cherry junction trees into $(k+1)$-th order $t$-cherry junction trees, using just the $k$-dimensional information contents.

First we give an algorithm for building a *(k+1)*-th order $t$-cherry junction tree from a $k$-th order $t$-cherry junction tree:

**Algorithm 4.1.**

Step 1. We fix a k-variable cluster in the $k$-th order $t$ -cherry junction tree, and call it root.

Step 2. We transform the root and one of the clusters connected to it in a $(k+1)$- variable cluster (Figure 4.)

Step 3. Using Algorithm 2 of paper [17], we transform the *(k+1)*-width junction tree, obtained in Step 2, into a *(k+1)*-th order $t$-cherry junction tree.

**Theorem 4.3.** [17] *The (k+1)-width junction tree obtained in Step 2 of Algorithm 4.1 gives at least as good approximation of* $P(\mathbf{X})$ *as the k-th order t-cherry junction tree did.*

Applying Theorem 4.2 for the family $\mathcal{F}_{k+1}$ it follows that the $(k+1)$-th order $t$-cherry junction tree obtained in step 3 provides a better approximation of the real probability distribution.

For random vectors containing great number of random variables (more than 50) the construction of higher order (5th order)$t$-cherry junction tree needs too much CPU time and memory. So a new problem arises, is it possible starting from a lower order (3rd order) $t$-cherry junction tree to improve the approximation without calculating all of the 4th and 5th order information contents? The basic idea of the method introduced in [13] was first to fit a 3rd order $t$-cherry junction tree. Then we ,,cut" the branches and fit to the variables contained in the branches higher order $t$-cherry junction trees. Finally we assemble the improved branches into a 5th width junction tree. In [13] was proved that this procedure provides a better approximation of real distribution.

As an application was regarded a randomly generated 50 dimensional discrete probability distribution. The underlying Markov network was not known. First we constructed a 3-rd order $t$-cherry junction tree, using the information content of the second and third order marginals. This was done in a greedy way in order to maximize the weight of the junction tree (see Definition 4.2). The K-L divergence between the approximation, belonging to the 3-rd order $t$-cherry junction tree, and the true probability distribution is 6.72643. The 3-rd order $t$-cherry junction tree was partitioned into 6 branches then the branches were improved by 4-th order $t$-cherry junction trees using the "branch cutting and refitting" procedure. After this procedure the K-L divergence became 5.10920. We also improved the branches by 5-th order $t$-cherry junction trees. In this case the K-L divergence became 4.61357. The calculation of all of the five variable information contents should need enormous amount of necessary CPU time. One can see that the new ,,branch cutting and refitting" procedure produced low K-L divergence values with relatively small number of information content calculations.

## 5 Application of $t$-cherry junction trees in pattern recognition

Pattern recognition aims to classify data (patterns) based either on a priori knowledge or on statistical information extracted from the data. In the paper [18] we introduced a pattern recognition method which uses $t$-cherry junction tree approach. This made also possible the selection of the so called 'informative' features. The classifier was achieved by supervised learning. The classification used Bayes decision rule.

We estimated the underlying multi-variate probability distribution, by the exploitation of the conditional independences between the variables (features). This was achieved by fitting to the training data a $t$-cherry junction tree (see [12], [17]).

Let us introduce some notations and assumptions, see [7].
Let $(\mathbf{X}, Y)$ be an $R^n \times \{1, \ldots, M\}$ valued random vector. A classifier is constructed on the basis of a training set $(\mathbf{x}^1, y^1)^T \ldots (\mathbf{x}^m, y^m)^T$ and is denoted by $g_m$. For a given $\mathbf{x} \in R^n$ the value $y \in \{1, \ldots, M\}$ of $Y$ is guessed by $g_m \left( \mathbf{x}; (\mathbf{x}^1, y^1)^T, \ldots, (\mathbf{x}^m, y^m)^T \right)$.
So the classifier $g_n$ is a function:

$$g_m : R^n \times \{R^n \times \{1, \ldots, M\}\} \longrightarrow \{1, \ldots, M\}.$$

The construction of $g_m$ in this way is called supervised learning.
We assume that $(\mathbf{X}^1, Y^1)^T, \ldots, (\mathbf{X}^m, Y^m)^T$ is a sequence of independent identically distributed random vectors with the same distribution as $(\mathbf{X}, Y)^T$.
On the basis of the constructed $t$-cherry junction tree we select the informative features by the following algorithm.
**Algorithm 5.1.** Selection of the informative features.

1. Give as input the training data set in discretized form.
2. From the empirical probability distribution determine all the $k$-th order marginals (it is favorable if $k$ is not to large (less than 7, say).
3. Find the heaviest weighted tree, in sense of Definition 4.2.
4. Output : the set of clusters $\mathcal{C}$ and the set of separators $\mathcal{S}$.
5. Select those clusters which contain the variable $Y$.
6. Select those variables which occur in the clusters selected in step 5 as informative variables.

Let be denoted by $\mathbf{X}_{inf}$ the random vector of those random variables which are selected in step 6.
The joint probability distribution of the random vector $(\mathbf{X}_{\text{inf}}, Y)^T$ can be expressed using those marginals only, which correspond to the clusters that contain $Y$.
For the classifier $g_m$ we give a formula which depends on the probability distribution of $(\mathbf{X}_{inf}, Y)$. This way for classifying a new $m$-dimensional vector we use the informative features only.
In the following we present a practical application of our method in pattern recognition.
Voice disorders can be premonitory of different diseases. This observation makes possible new ways of investigations and diagnosis. The idea to check up our method in discovering the connection between voice disorders and

having Parkinson's disease (PD) came from [15]: "Research has shown that approximately 90% of people with PD exhibit some form of vocal impairment [9], [16]. Vocal impairment may also be one of the earliest indicators for the onset of the illness [8]".

The dataset was created by Max Little of the University of Oxford, in collaboration with the National Center for Voice and Speech, Denver, Colorado, who recorded the speech signals, and provided by UCI Machine Learning Repository on the internet: http://archive.ics.uci.edu/ml/datasets/Parkinsons.

This dataset is composed of a range of biomedical voice measurements from 195 voice recording of 31 people, 23 with Parkinson's disease (PD). The main aim of the data is to discriminate healthy people from those with PD.

The 23 features were:

MDVP: Fo(Hz) - Average vocal fundamental frequency - $X_1$

MDVP: Fhi(Hz) - Maximum vocal fundamental frequency - $X_2$;

MDVP: Flo(Hz) - Minimum vocal fundamental frequency - $X_3$;

MDVP: Jitter(%) - $X_4$,

MDVP:Jitter(Abs)- $X_5$,

MDVP:RAP - $X_6$,

MDVP:PPQ - $X_7$,

Jitter:DDP - Several measures of variation in fundamental frequency - $X_8$

MDVP:Shimmer - $X_9$,

MDVP: Shimmer(dB) - $X_{10}$,

Shimmer:APQ3 - $X_{11}$,

Shimmer:APQ5 - $X_{12}$,

MDVP:APQ - $X_{13}$,

Shimmer:DDA - Several measures of variation in amplitude- $X_{14}$

NHR,HNR Two measures of ratio of noise to tonal components in the voice status - $X_{15}$, $X_{16}$

Health status of the subject (one) - Parkinson's, (zero) - healthy - $X_{17} = Y$

RPDE,D2 - Two nonlinear dynamical complexity measures - $X_{18}, X_{19}$

DFA - Signal fractal scaling exponent -$X_{20}$

spread1,spread2,PPE - Three nonlinear measures of fundamental frequency variation $X_{21}, X_{22}, X_{23}$.

MDVP stands for denoting measures introduced in the Multi Dimensional Voice Program by Kay Pentax, which became a standard in voice analysis.

We choose randomly from the data a test set which contains 10 recordings of healthy people and 20 from ill ones. The rest of 165 vectors remained the training data set. The 5-th order *t*-cherry junction tree has the following clusters:

$$
\begin{array}{rrrrr}
(X_1, & X_2, & X_3, & X_{10}, & X_{16}) \\
(X_1, & X_3, & X_{10}, & X_{16}, & X_{20}) \\
(X_1, & X_6, & X_{10}, & X_{16}, & X_{20}) \\
(X_1, & X_{10}, & X_{16}, & X_{19}, & X_{20}) \\
(X_1, & X_{10}, & X_{17}, & X_{19}, & X_{20}) \\
(X_4, & X_6, & X_8, & X_{15}, & X_{16}) \\
(X_4, & X_6, & X_{10}, & X_{15}, & X_{16}) \\
(X_4, & X_7, & X_{10}, & X_{13}, & X_{16}) \\
(X_4, & X_{10}, & X_{13}, & X_{15}, & X_{16}) \\
(X_5, & X_{16}, & X_{19}, & X_{21}, & X_{22}) \\
(X_6, & X_{10}, & X_{15}, & X_{16}, & X_{23}) \\
(X_6, & X_{10}, & X_{16}, & X_{20}, & X_{23}) \\
(X_9, & X_{10}, & X_{11}, & X_{12}, & X_{13}) \\
(X_9, & X_{10}, & X_{11}, & X_{13}, & X_{14}) \\
(X_{10}, & X_{11}, & X_{12}, & X_{13}, & X_{16}) \\
(X_{10}, & X_{12}, & X_{13}, & X_{15}, & X_{16}) \\
(X_{10}, & X_{16}, & X_{19}, & X_{20}, & X_{21}) \\
(X_{16}, & X_{19}, & X_{20}, & X_{21}, & X_{22}) \\
(X_{18}, & X_{19}, & X_{20}, & X_{21}, & X_{22})
\end{array}
$$

Only the 5-th cluster is informative $(X_1, X_{10}, X_{17} = Y, X_{19}, X_{20})$. The informative features are: Fo, shimmer, DFA, spread1. Using the probability distribution $P\left(X_1, X_{10}, X_{19}, X_{20}, Y\right)$, we obtained a 93% correct classification performance for the classifier. The authors of paper [15] got 91.4 % correct classification performance using 10 features and a Kernel support vector machine.

## References

[1] **P Bremaud**, *Markov chains, Gibbs Fields, Monte Carlo Simulation, and Queues*, Springer, USA, 1999.

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

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

[4] **Chow C.K. and C.N. Liu**, Approximating Discrete Probability Distribution with Dependence Tree, *IEEE Transactions on Informational Theory*, **14**, (1968), 462-467.

[5] **Constantin Gh. and I. Istratescu**, *Elements of Probabilistic Analysis*, Kluwer Academic Publisher, Dordrecht, 1989.

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

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

[8] **Duffy J. R.**, *Motor Speech Disorders: Substrates Differential Diagnosis, and Management*, Elsevier, St Louis, 2005.

[9] **R. Iansek Ho A. K. C. Marigliani**, Speech impairment in a large sample of patients with Parkinson's disease, *Behav. Neurol.*, **11**, (1998), 131–137.

[10] **Ising E.**, *Zeitschrift Physik*, **31**, (1925), 253.

[11] **Kovács E.**, The use of copulas in the study of certain transforms of random variables with applications in finance, *in: Proceedings of Conference ICTIAMI, Alba Julia, 2005, Part B*, (2005), 129–138.

[12] **Kovács E. and T. Szántai**, On the approximation of discrete multivariate probability distribution using the new concept of $t$-cherry junction tree, *in: Proceedings of the IFIP/IIASA/GAMM Workshop on Copyng with Uncertainty, IIASA, Laxenburg,* (2007.), accepted

[13] **Kovács E. and T. Szántai**, Some improvements of $t$-cherry junction trees, *In: Iantovics B., Enachescu C. and Filip F G (ed.) First International Conference on Complexity and Intelligence of the Artificial and Natural Complex Systems. Medical Applications of the Complex Systems. Biomedical Computing.: CANS 2008. Tg. Mures, Romania, 2008.11.08-2008.11.10., Los Alamitos: IEEE Computer Society*, (2009), 117–129.

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

[15] **M. A. Little P. E. McSharry**, Suitability of Dysphonia Measurements for Telemonitoring of Parkinson's Disease, *IEEE Transactions on Biomedical Engineering*, **56** (4), (2009), 1015–1022.

[16] **J.A. Logemann, H.B. Fisher, and B. Boshes and Blonsky**, Frequency and co-ocurrence of vocal-tract disfunctions in speech of a large sample of patients with Parkinson's disease, *J. Speech, Hear. Disord.*, **43**, (1978), 47–57.

[17] **and Kovács E. Szántai T.**, Hypergraphs as means of discovering the dependence structure of a discrete multivariate probability distribution, *Annals of Operations Research*, (2009.), accepted

[18] **T. Szántai and Kovács**, Pattern recognition using $t$-cherry junction tree structures, *In: Proceedings of International Symposium on Understanding Intelligent and Complex Systems, Petru Maior University, Targu Mures on 22-23 October 2009,* (2009.), to appear

[19] **Tomescu I.**, Hypertrees and Bonferroni inequalities, *J. Combin. Theory*, **41**, (1986), 209–217.

Edith Kovács

Department of Methodology
Budapest College of Management
Villányi u. 11–13.
Budapest
Hungary
E-mail: kovacs.edith@avf.hu

Tamás Szántai
Institute of Mathematics
Budapest University of Technology and Economics
Műegyetem rkp. 3.
Budapest
Hungary
E-mail: szantai@math.bme.hu

Received: 22.06.2009
Accepted: 15.07.2009