ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2006, Vol. 35, No. 2

DATA CLASSIFICATION USING DIRICHLET MIXTURES

Dovilė Rudokaitė-Margelevičienė[1], Henrikas Pranevičius[2], Mindaugas Margelevičius[3,2]

[1]Department of Software Engineering, Kaunas University of Technology
Studentų St. 50, LT-51368 Kaunas, Lithuania
[2]Department of Business Informatics, Kaunas University of Technology
Studentų St. 56, LT-51424 Kaunas
[3]Laboratory of Bioinformatics, Institute of Biotechnology
Graičiūno St. 8,LT-02241 Vilnius

Abstract. In the article, we present a method for data classification that is based on the Dirichlet mixture statistics. An important property of the method is its ability to classify data of any type. To test performance of the method, we implemented it as a stand-alone program and tested it on the three different databases of real data. Receiver operating characteristics of the classification was used to compare the method of Dirichlet mixtures to the other classification methods. The classification results and its performance are discussed in the article. The practical value of this study is that the method based on the complex statistics is implemented as a tool and compiled as a library for further development of machine learning environments.

1. Introduction

"Classification, the separation and naming of appearances, is one of the most basic cultural activities of humanity; it is a fundament for our science and civilization" [1]. These are words written by Frank Hampel, the famous professor of statistics, and they have a philosophical meaning as well. Basically, classification is grouping of data into predefined categories according to their similarity, where similarity in particular can be measured by means of an ordered set of related attributes that logically and physically describe data. In the real world, diversity of data is high and quantities of data to be processed in order to obtain significant results differ from field to field. In addition, data specificity varying over fields of investigations requires specific expertise systems to process them correctly. All this makes human difficult to directly inspect data distributions, to interpret those, to accept hypotheses, or reject them. Therefore, the main goal of data classification is to simplify representation of the real world and to infer rules by which data could successfully be mapped into categories. In other words, classification proposes clearer understanding about data.

In order to make observed data proper for classifycation they are assigned to meaningful categories. Without accurate and systematic specification of categories according to common properties, statistical models applied to data cannot guarantee reliable and mutually comparable results. Developed mathematical and statistical models formally describe distribution of data with respect to common properties, what means that those models can be applied to data to classify them when there is no knowledge about categories data could belong to. However, for valid classification, data and classification in general must meet several essential requirements. For instance, to avoid ambiguity observed data cannot have been assigned to multiple categories, a set of categories must be consistent, data should be predicted to belong to one category, and classification must have conceptual base and logical structure.

Accuracy of classification depends on mathematical models used to classify data, or in other words, it depends on how accurately model-based distribution simulates that of the real world data. There are a lot of methods that can successfully be (have been) used to create formal models. Consequently, space of feasible classification architectures built on these models and tuned for specific problems is hardly practically exhaustible.

One of the recently published classification methods [2] based on theory of graphs assigns data to categories according to a dynamically composed graph. That classifier applied to an image recognition problem is compared to Support Vector Machines [3]. Another publication [4] presents a classifier that is based on computations of multivariate Gaussian density where the parameters of the classifier are optimized by Expectation Maximization algorithm [5]. This model-based clustering has successfully been applied to Magnetic Resonance Imaging for classification and clustering of MRI data. Support Vector

machines (SVM) have extremely become popular in bioinformatics. Various SVM architectures [6-9] are used to classify proteins into the appropriate classes and folds. In addition, SVM architectures are configured with neural networks [10], dynamic programming algorithms [11], and other methods in order to make particular classifier better in performance. The methods mentioned above reveal a small part of their possible usage in classification of the real world data, and they are just a few examples of practical classifiers.

In this article, we present a classification method that is based on computations of Dirichlet mixture density. In mathematical statistics, Dirichlet mixtures are important where multinomial distribution is used to describe frequency characteristics of some data. Dirichlet mixtures and mixtures in general are specific in that mean they can be adapted to classify data whose attributes can logically be grouped to compose complex data structures [12]. For instance, amino acid distribution in multiple protein sequence alignments [13] can expose various biologically important and specific regions. Using the posterior probability distribution estimated from the analysis of multiple sequence analysis, one is able to model biologically important evolutionary processes [14, 15].

The method presented in this article can be used for classification of data of any kind. Though Dirichlet mixture method was used to solve some particular problems [16] and there were derived general Bayes mixture models [17, 18], to our knowledge, a tool based on Dirichlet mixtures has not been developed to classify data of any kind.

2. Methods

2.1. Dirichlet densities and their mixtures

A Dirichlet density g is a probability density function of probability vectors, **p** [12, 16]. Let us introduce some alphabet **A** with cardinality denoted as |**A**|. Then each possible vector **p** corresponds to *a priori* probabilities for distribution of the letters of alphabet **A**. A Dirichlet density is defined by a vector of parameters $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^{|\mathbf{A}|}$ with $\alpha_i > 0$ and is equal to

$$g(\mathbf{p} \mid \boldsymbol{\alpha}) = \frac{\Gamma(|\boldsymbol{\alpha}|)}{\prod_{i=1}^{|\mathbf{A}|}\Gamma(\alpha_i)} \prod_{i=1}^{|\mathbf{A}|} p_i^{\alpha_i - 1}, \quad (1)$$

where $\Gamma(\cdot)$ denotes gamma function [19], $|\boldsymbol{\alpha}| = \sum_{i=1}^{|\mathbf{A}|}\alpha_i$, $p_i \geq 0$ ($i = 1,\ldots,|\mathbf{A}|$), and $\sum_{i=1}^{|\mathbf{A}|} p_i = 1$.

A mixture of Dirichlet densities is a weighted superposition of individual Dirichlet densities that constitute a new probability density function. Each individual density in the mixture is assigned a weight called mixture coefficient, and each individual density is called a component of the mixture. A Dirichlet mixture density φ composed of l components is defined

$$\varphi = \sum_{j=1}^{l} q_j g_j, \quad (2)$$

where g_j are the individual Dirichlet densities with their own set of parameters $\boldsymbol{\alpha}_j = \{\alpha_{ji}\}_{i=1}^{|\mathbf{A}|}$, and q_j are the mixture coefficients for which the sum $\sum_{j=1}^{l} q_j = 1$ holds true. We name the entire set of parameters $\Theta = \left(\{\boldsymbol{\alpha}_j\}_{j=1}^{l}, \{q_j\}_{j=1}^{l}\right)$ a Dirichlet mixture model. Each component in the mixture formally describes a particular probability density defined by parameters, hence their mixture is useful in data classification where data possess the properties the individual components can recognize. A mixture with one component becomes a simple Dirichlet density. The number of components in a mixture is unlimited but the large number of them increases the number of parameters in a model and makes finding of the optimal parameter values difficult.

2.2. Classification by Dirichlet mixture

Let us suppose that the letters from an alphabet **A** are multinomial distributed random variables and for each alphabet letter a_i we have the corresponding frequency n_i that matches the number of occurrences for that letter. Then the entire set of frequencies can be denoted as $\mathbf{n} = \{n_i\}_{i=1}^{|\mathbf{A}|}$ and the likelihood of the frequency vector **n** is defined

$$P(\mathbf{n} \mid \mathbf{p}) = \Gamma(|\mathbf{n}| + 1) \prod_{i=1}^{|\mathbf{A}|} \frac{p_i^{n_i}}{\Gamma(n_i + 1)}, \quad (3)$$

where $|\mathbf{n}| = \sum_{i=1}^{|\mathbf{A}|} n_i$ and p_i is an occurrence probability of a letter a_i from the alphabet **A**. There are a number of methods how to estimate p_i and one of them is to use a Dirichlet density or a mixture of those. Assume that the random variables p_i have the Dirichlet density function defined by (1). We see that the probability vector **p** depends on a certain parameter vector $\boldsymbol{\alpha}_j$, what means that the frequency vector likelihood depends on the parameters $\boldsymbol{\alpha}_j$ so that

$$P(\mathbf{n} \mid \boldsymbol{\alpha}_j) = \int_{\mathbf{p} \in \mathcal{P}} P(\mathbf{n} \mid \mathbf{p}) g(\mathbf{p} \mid \boldsymbol{\alpha}_j) d\mathbf{p}, \quad (4)$$

where integral is taken over the entire domain $\mathcal{P}$ of probability vectors **p**. Substituting (1) and (3) into (4) we obtain [16]

$$P(\mathbf{n} \mid \boldsymbol{\alpha}_j) = \frac{\Gamma(|\mathbf{n}|+1)\Gamma(|\boldsymbol{\alpha}_j|)}{\Gamma(|\mathbf{n}| + |\boldsymbol{\alpha}_j|)} \prod_{i=1}^{|\mathbf{A}|} \frac{\Gamma(n_i + \alpha_{j,i})}{\Gamma(n_i + 1)\Gamma(\alpha_{j,i})}. \quad (5)$$

If we assume that **p** conforms a Dirichlet mixture density, then the frequency vector **n** depends on the model parameters Θ and the likelihood of **n** is defined as follows:

$$P(\mathbf{n} \mid \Theta) = \sum_{j=1}^{l} q_j P(\mathbf{n} \mid \boldsymbol{\alpha}_j). \quad (6)$$

One can think of data classification as a process consisting of two stages: training and classification. In

the training stage, probabilities used in frequency distribution (observed data) should be estimated as accurately as possible. In the classification stage, it is supposed those estimated probabilities match distribution of the frequencies observed in the classification set of data best. That means, if in the training stage one have obtained the optimal probabilities, it is very likely that in the classification stage one will be able to classify observed data (not used in the training stage) accurately. However, a question is how to find optimal estimators for probabilities.

Posterior mean estimate and maximum likelihood estimation are a few techniques of hypothesis testing. It is known [20, 21] that posterior mean estimate takes the form

$$\hat{p}_i = \int_{\mathbf{p}\in\mathcal{P}} p_i P(\mathbf{p} \mid \Theta, \mathbf{n}) d\mathbf{p}\,, \tag{7}$$

where $\mathbf{p}$ is a vector with components p_i and p_i is supposed to be a probability having the Dirichlet mixture density function (2). Once we have the Dirichlet mixture, we can expand

$$P(\mathbf{p} \mid \Theta, \mathbf{n}) = \sum_{j=1}^{l} P(\mathbf{p} \mid \boldsymbol{\alpha}_j, \mathbf{n}) P(\boldsymbol{\alpha}_j \mid \mathbf{n}, \Theta), \tag{8}$$

giving

$$\hat{p}_i = \sum_{j=1}^{l} P(\boldsymbol{\alpha}_j \mid \mathbf{n}, \Theta) \int_{\mathbf{p}\in\mathcal{P}} p_i P(\mathbf{p} \mid \boldsymbol{\alpha}_j, \mathbf{n}) d\mathbf{p}\,. \tag{9}$$

It is also known from the theory [22] that the posterior mean estimate in the case of a single Dirichlet density is equal to

$$\hat{p}_i^{\mathrm{s}} = \int_{\mathbf{p}\in\mathcal{P}} p_i P(\mathbf{p} \mid \boldsymbol{\alpha}_1, \mathbf{n}) d\mathbf{p} = \frac{n_i + \alpha_{1,i}}{|\mathbf{n}| + |\boldsymbol{\alpha}_1|}\,, \tag{10}$$

here the parameters $\{\alpha_{1,i}\}$ comprise a single vector $\boldsymbol{\alpha}_1$ (a mixture consists of one Dirichlet density). Using Bayes' rule it is possible to express

$$P(\boldsymbol{\alpha}_j \mid \mathbf{n}, \Theta) = \frac{q_j P(\mathbf{n} \mid \boldsymbol{\alpha}_j)}{P(\mathbf{n} \mid \Theta)}\,. \tag{11}$$

Substituting (10) and (11) into (9), we obtain

$$\hat{p}_i = \frac{1}{P(\mathbf{n} \mid \Theta)} \sum_{j=1}^{l} q_j P(\mathbf{n} \mid \boldsymbol{\alpha}_j) \frac{n_i + \alpha_{j,i}}{|\mathbf{n}| + |\boldsymbol{\alpha}_j|}\,. \tag{12}$$

Now we have the posterior mean estimates for all the probabilities $\{p_i\}$. However, we have not defined yet how to obtain the optimal parameter values from the model Θ. In this place we have to return shortly to the expression of $P(\mathbf{n} \mid \Theta)$. The frequency likelihood depends on all the parameters we should find and on the observed frequency vector $\mathbf{n}$. Before making a certain decision, people often perform many measurements to get enough experimental data. The same is with automata: the more data are used in the training of a machine, the more accurate results one could expect to obtain employing the machine on the control dataset. For instance, a rule-based machine will derive more robust rules from the data collected from many patients, say, with the arrhythmia heart disease, than in the case when the machine will be trained on the data from one patient. Or, one will not be able to reliably classify proteins to their families if a classifier has been trained according to a distribution of amino acids from the proteins one per family. Hence, sufficient amount of data is crucial for development of an accurate classifier.

A vector $\mathbf{n}$ corresponds to one observation (each element matches one attribute). Then, a classifier is supposed to process many vectors $\{\mathbf{n}_c\}_{c=1}^{N}$ to have optimized the parameters as accurately as possible. If we assume the set of vectors $\{\mathbf{n}_c\}$ to be independent and identically distributed (iid) random variables, then according to maximum likelihood estimation [21] the set of parameters Θ can be optimized by maximizing the product $\Pi_c P(\mathbf{n}_c \mid \Theta)$. Since the logarithm is a monotonically increasing function, it is possible to minimize the sum of logarithms instead of maximizing the product of raw probabilities:

$$f(\Theta) = -\sum_{c=1}^{N} \log P(\mathbf{n}_c \mid \Theta)\,. \tag{13}$$

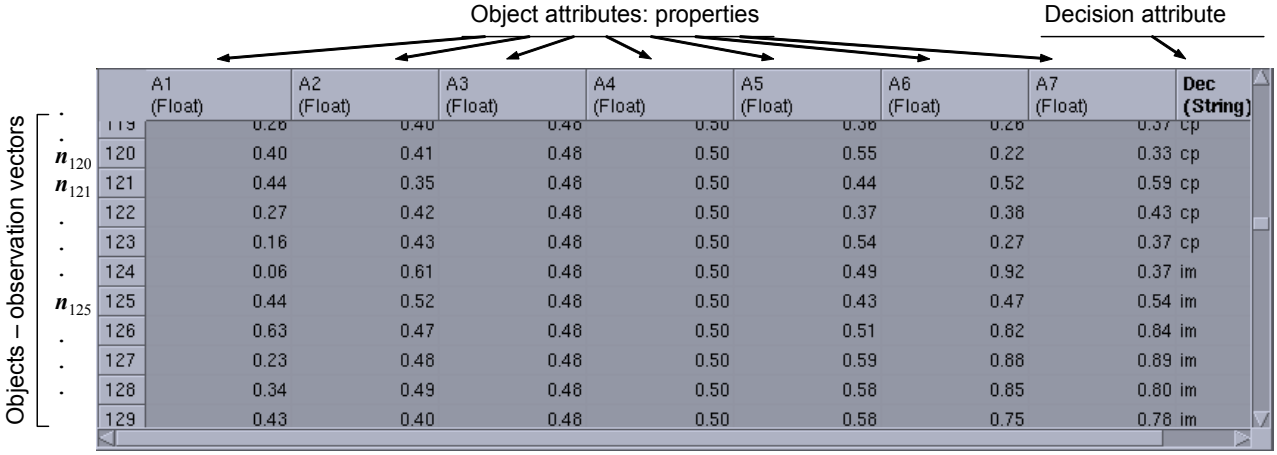
The last expression (13) gives the objective function to be minimized in order to find the optimal parameters for data classification machine.

3. Implementation

To test capabilities of Dirichlet mixtures to classify data of any kind, we have implemented a classifier based on Dirichlet mixture statistics as a stand-alone tool and compiled as a library of classification routines. Since in this article we do not consider the programming aspects of the developed software, we limit ourselves to some implementation features of the tool which we used in this work to test how well the Dirichlet mixture method performs. For implementation of the tool we used the routine library of an integrated data mining system, Rosetta 1.0 [23]. Rosetta encompasses a machine learning computational kernel based on rough sets theory [24, 25] as well as other computational facilities used to process and prepare data for classification with the rules derived in the training stage be more rigorous. Those additional facilities include data discretization, reduction methods, scaling, completing and other algorithms. However, we used the Rosetta library only to represent data in a format Rosetta does, i.e. we used the Rosetta structures to keep data in the internal representation formats.

Rosetta represents data with a table in which each row corresponds to a single observation called an object. Values in the columns (observation values) used to name the attributes of the table characterize and physically describe the objects. Let us denote not an empty set of objects by U and let T be not an empty set of attributes, then an information system (a table) $S = (U, T)$. In order to be capable to classify data, in the training stage there must be given *a priori* information about belonging of the objects to the classification categories called decision classes. One object

can belong to one and only one decision class and that means measurements for the object correspond to a known classification category. If we denote an additional attribute for classification categories, called a decision attribute, by $d \notin T$, then an augmented information system with the decision attribute is called a decision system and has a notation $S = (U, T \cup \{d\})$. An example of a decision system is shown in Figure 1.



	A1 (Float)	A2 (Float)	A3 (Float)	A4 (Float)	A5 (Float)	A6 (Float)	A7 (Float)	Dec (String)
120	0.40	0.41	0.48	0.50	0.55	0.22	0.33	cp
121	0.44	0.35	0.48	0.50	0.44	0.52	0.59	cp
122	0.27	0.42	0.48	0.50	0.37	0.38	0.43	cp
123	0.16	0.43	0.48	0.50	0.54	0.27	0.37	cp
124	0.06	0.61	0.48	0.50	0.49	0.92	0.37	im
125	0.44	0.52	0.48	0.50	0.43	0.47	0.54	im
126	0.63	0.47	0.48	0.50	0.51	0.82	0.84	im
127	0.23	0.48	0.48	0.50	0.59	0.88	0.89	im
128	0.34	0.49	0.48	0.50	0.58	0.85	0.80	im
129	0.43	0.40	0.48	0.50	0.58	0.75	0.78	im

Figure 1. An example of a decision table with the objects enumerated.
Each of the objects is described by the set of global properties called attributes.
The last attribute is a decision attribute that defines a decision class an object belongs to

The source code of the Rosetta system is freely available to use for non-commercial purposes [26]. The code is structural and reusable. Nevertheless, our decision to use this system is not due to its publicity alone but due to successful applications of the system itself and of the methods encapsulated in it as well. Rough sets and the Rosetta system proved to be useful in a number of scientific investigations: in identification and prediction of gastric carcinomas according to microarray gene expression profiles [27], in early diagnosis of coronary artery disease [28], in predicting protein functions [29], and in discovering regulatory binding sites according to gene expression profiles [30] and in the other projects. Therefore, it would be a good idea to compare the classifier of Dirichlet mixtures to the classification method based on rough sets.

For all the datasets we chose to test the Dirichlet mixture classifier on, we compare the classifier with two other classification methods, namely, with classification by rules derived on the basis of rough sets computations and with the Naïve Bayes method [31]. We chose several databases from different scientific fields to check the pronounced feature of the method to classify data of any kind. A question may here arise is how to computationally treat data of any kind where types of attributes may vary from table to table.

Each attribute in a decision table is given a data type (Figure 1). There are three data types an attribute can be of: integer number, floating point number, and string. Floating point numbers and strings are converted to integers so that mapping from those data types is unique. After attribute values are converted, a distribution of values in a decision table is supposed to be a distribution of frequencies that make up frequency vectors $\mathbf{n}_c$ for each of the objects in the table. When negative values for attributes are observed, then each value in a column is linearly transformed so that it acquires a non-negative value. The range of values an attribute with its value can fall in may be very wide. Hence, scaling of values is performed before optimization of Dirichlet mixture parameters takes place. The scaling is necessary because large particular attribute values can negatively affect the optimization process.

The objective function (13) to be minimized is continuous and rather complex because: (i) the number of observations is unlimited and the number of frequency vectors can be large, (ii) the objective function is to be computed in a multi-dimensional space where dimensionality of the space depends on the number of parameters of a Dirichlet mixture model. For example, if one defines the Dirichlet mixture model to consist of 20 components and each of them has 20 pseudo frequency parameters $\{\alpha_{ji}\}_{i=1}^{20}$, then there are 420 parameters overall to be optimized. The function has many local minima (see Figure 2 for illustration) and to find the global minimum point is a complicated task.

To optimize the objective function, we used three optimization methods: genetic algorithm [32, 33], the conjugate gradient method, and the Levenberg-Marquardt method [34]. Genetic algorithm is a combinatorial optimization method that best fit for problems where an objective function is discrete or has many local minima. However, as we see later, the other two optimization methods are used not needlessly. There are cases [35] where genetic algorithm converges rather slowly and requires many generations to perform. We decided to implement [36] the convex optimization methods (conjugate gradient and Levenberg-

Marquardt) to learn which of the optimization methods fit our problem best. A weak point of the convex optimization methods is that sometimes their performance in finding the global minimum for the most part depends on how lucky one is in choosing the starting point for the iterative search [34]. Combining combinatorial and convex optimizations can lead to better results.

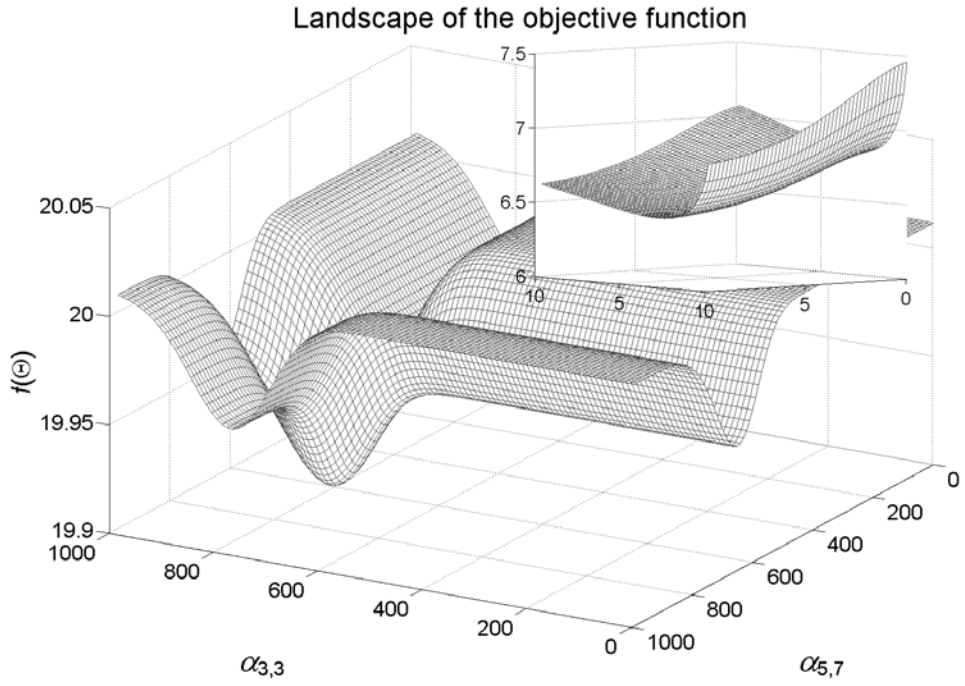


Figure 2. The landscape of the objective function (13) is drawn for a specific problem using a Dirichlet mixture model consisting of 20 components and 7 pseudo frequencies in each. The figure shows how the objective function varies with respect to two arbitrary chosen parameters $\alpha_{3,3}$ and $\alpha_{5,7}$ from the components 3 and 5, respectively, when the other 158 parameters are set to be fixed near a locally optimum point. The inset illustrates the objective function's profile with respect to the same parameters, $\alpha_{3,3}$ and $\alpha_{5,7}$, when they vary in the neighborhood of another locally optimum point moved closer towards the origin

4. Results and discussion

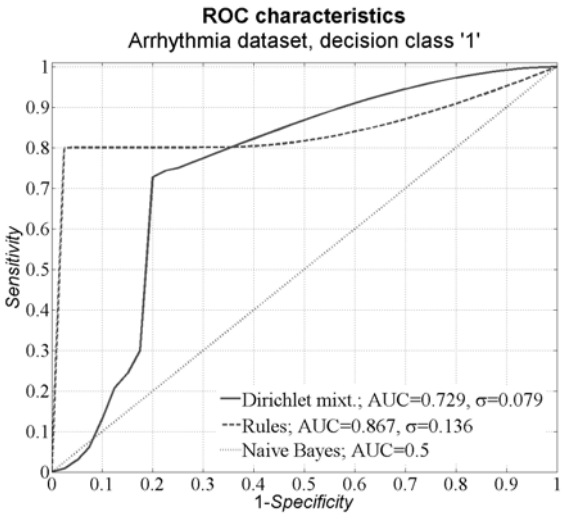
To test the performance of the Dirichlet mixture classifier, we chose three databases from the machine learning database repository of the University of California [37]: the cardiac arrhythmia database, the *E.coli* protein classification database, the database for classification of radar returns from the ionosphere. Before the training of the classifiers, we divided all the datasets into the training and testing datasets so that data in the testing dataset are not used to be in the training dataset. We trained the classifiers on the training dataset and tested them on the testing dataset. The ratios of the sizes of the training and testing datasets for the arrhythmia, *E.coli*, and ionosphere databases were 90:10, 86:14, and 86:14, respectively. In all cases we compiled the testing datasets to have equal percentage of the objects from all the classification categories in a dataset. For example, if there are two classification categories in the ionosphere database, 'g' and 'b', then the testing dataset for this database would contain 14% of the objects from both classification categories 'g' and 'b'.

We assess the performance of the Dirichlet mixture classifier by the ROC characteristic curves [38]. The ROC curves characterize the classification accuracy with a set of points corresponding to different levels of specificity and sensitivity. Specificity indicates how accurately (what percentage) a particular classifier has recognized objects that do not actually belong to a classification category. Sensitivity indicates how accurately a classifier has recognized objects that do belong to the classification category.

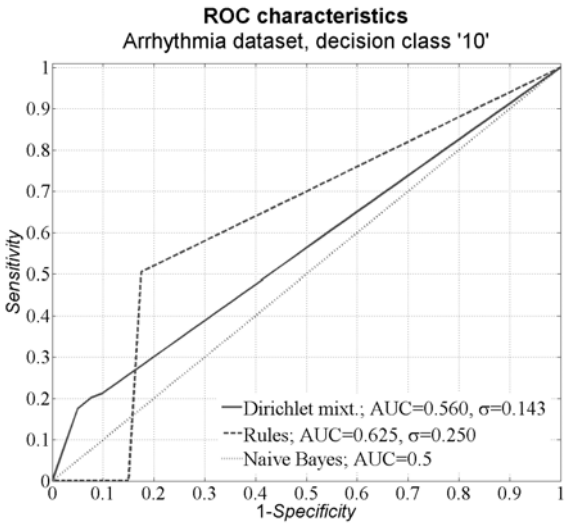
To examine how the Dirichlet mixture classifier compares to the other classification methods, we draw the ROC curves for the other two methods: the classification by rules generated by rough sets computations and the Naïve Bayes method. There were used the same datasets as for the classification by the Dirichlet mixtures. Before classifying the data by the rules, the data were applied the discretization and reduction algorithms under the rough sets theory to make the rules more generalized. The Naïve Bayes method was also fed the data after discretization because doing so led us to the better classification results. There are depicted the best ROC curves obtained for the both classification methods, the classification by rules and the Naïve Bayes classifier. No discretization and reduction were performed in the case of classification by the Dirichlet mixtures.

4.1. Arrhythmia dataset

The data [39] are compiled to distinguish between the presence and absence of cardiac arrhythmia. The classification categories cover 16 distinct levels of cardiac arrhythmia, the first of which refers to "normal". The data comprise 279 attributes: age of patients, sex, height, weight, number of heart beats per minute, and other specific information. The data are collected form the 452 patients (number of objects). There are missing attribute values, namely 0.33%.



a)



b)

Figure 3. The classification performance illustrated by the ROC characteristic curves for the Dirichlet mixture (Dirichlet mixt.), for the rules (Rules), and for the Naïve Bayes method characterizes the predictions obtained for the Arrhythmia dataset for the two classification categories, '1' (a) and '10' (b). The area under the ROC curve (AUC) expresses the classification performance in one value. The parameter σ denotes standard deviation of the AUC computation

We trained the Dirichlet mixture classifier with the conjugate gradient method, the Levenberg-Marquardt method, and the genetic algorithm. However, the reasonable results were obtained by genetic algorithm while the other two methods failed to converge or stuck in a local minimum. We noticed from the analysis of the landscape of the objective function that the surface of the function is rather flat and it most likely happened to be a reason why the convex optimization methods failed.

We tried various compositions for the Dirichlet mixture model and learned that for this dataset a simple Dirichlet classifier performs best. So, one mixture component and 279 pseudo frequency parameters $\{\alpha_{1,i}\}_{i=1}^{279}$ comprised the Dirichlet mixture model. We ran 200 genetic algorithm's generations and applied the even-odd crossover algorithm and the flip mutation algorithm for genes. The other parameters we chose: crossover probability, 0.9, mutation probability, 0.1, population size, 60, number of individuals to be replaced in each generation, 9. The mutation probability is relatively high because of we tried to simulate rapid mutations and to reduce number of generations required for convergence of the algorithm. We ran more than 2000 generations with the mutation probability set to 0.01 as well and obtained similar results.

From the ROC curves (Figure 3) one can realize that for the two classification categories ('1' and '10') the classification by the rules and the classification by the Dirichlet mixture differ slightly. The Naïve Bayes method was unable to classify this dataset: the area under the ROC curve (AUC) is equal to 0.5 indicating a random classification. The classification by the rules was superior to the dirichlet mixture classifier a little, though the accuracy of the Dirichlet mixture classifier for the categories '1' and '10' is 75% and 20%, respectively; while that of the classification by the rules for the same categories is 17% and 0% altogether. The overall accuracy of the Dirichlet mixture classifier is 47.5% while that for the rules is 10%. Such numbers can be explained by the fact that the predictions made by the Dirichlet mixture classifier were comparable to each other and this reduces the reliability of the results. On the other hand, ROC curves are drawn by changing a threshold value a classifier's output must exceed to treat it as a prediction, and this could explain why the performance of the classifications by the Dirichlet mixture and by the rules with respect to ROC analysis is found to be similar.

There are 245 objects from the classification category '1' and 50 objects from the classification category '10'. There are more data from the category '1' and it has influenced the results obtained. However, the overall accuracy is not high but we can conclude that this dataset is complicated for the automated machine learning methods. The authors of this dataset state [39] that the voting feature interval method they used in the 10-fold cross-validation [40] procedure attained classification accuracy of 62%, which is not high. Furthermore, they remarked

frequent discrepancies between the expert's decisions and the classifier's predictions.

4.2. E.coli dataset

The dataset contains data of proteins functioning in bacterium *E.coli* and groups them according to the cellular localization sites [41]. Eight different classification groups, or categories, correspond to the cellular mediums the proteins function in; those are cytoplasm (cp), inner membrane with no signal sequence (im), periplasm (pm), inner membrane with uncleavable signal sequence (imU), non-lipoprotein outer membrane (om), lipoprotein outer membrane (omL), lipoprotein inner membrane (imL), inner membrane with cleavable signal sequence (imS). The sequence analysis scores as outputs from the signal sequence recognition methods contribute 7 attributes for the data. Number of the objects (protein information vectors) in the dataset is 336. There are no missing attribute values.

As in the previous case, we tried three optimization methods: the conjugate gradient method, the Levenberg-Marquardt method, and the genetic algorithm. In contrast to the arrhythmia dataset, the Levenberg-Marquardt method performed best though the genetic algorithm did approximately well. We used various models for the Dirichlet mixture and the one performing quite well was of 20 components each with 7 pseudo frequency parameters. Probably the optimized model obtained by the genetic algorithm could have been more precise resulting in more accurate classification in case we would have used more than several thousands of the genetic algorithm generations.

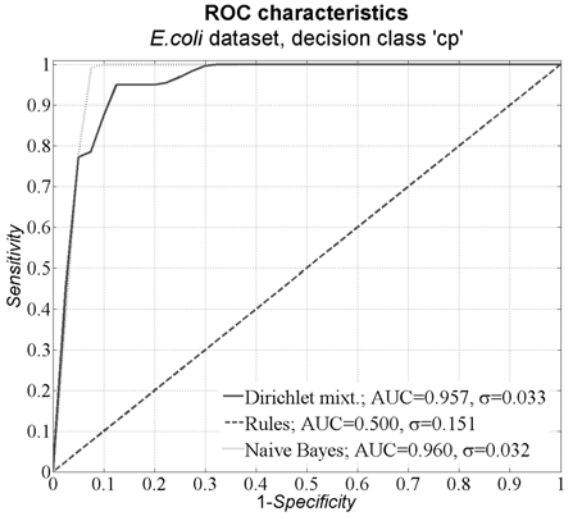
The Levenberg-Marquardt method requires the Jacobian to be explicitly provided with the objective function. The partial derivatives of the objective function (13) with respect to the parameters α_{ji} of the Dirichlet mixture are defined (see [16] for the derivation):

$$\frac{\partial f(\Theta)}{\partial \alpha_{j,i}} = -\sum_{c=1}^{N} P(\boldsymbol{\alpha}_j \mid \mathbf{n}_c, \Theta)\left(\Psi(|\boldsymbol{\alpha}_j|) - \right.$$
$$\left. - \Psi(|\mathbf{n}_c| + |\boldsymbol{\alpha}_j|) + \Psi(n_{c,i} + \alpha_{j,i}) - \Psi(\alpha_{j,i})\right). \quad (14)$$

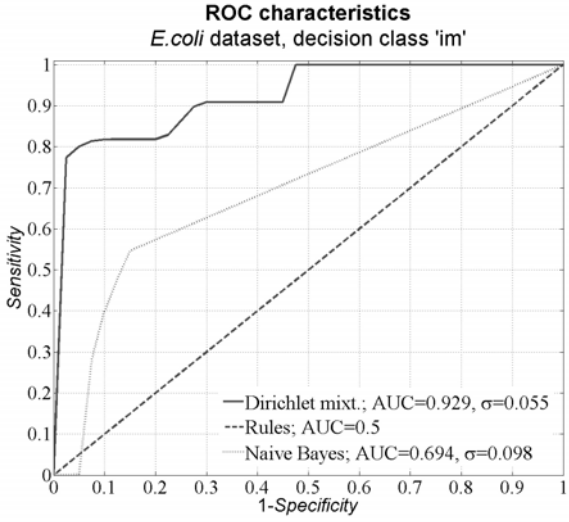
Here we used a notation $\Psi(z) = \Gamma'(z) / \Gamma(z)$ which is the digamma function of argument z. To compute the derivatives with respect to q_j, we introduce variable Q_j so that $q_j = Q_j / |Q|$ where $|Q| = \Sigma_j Q_j$. The substitution of q_j is to ensure the mixture coefficients q_j sum to one. Computing the partial derivatives with respect to Q_j ensures us that the required constraints will be met and the coefficients q_j will be unambiguously solved:

$$\frac{\partial f(\Theta)}{\partial Q_j} = \frac{N}{|Q|} - \frac{\sum_{c=1}^{N} P(\boldsymbol{\alpha}_j \mid \mathbf{n}_c, \Theta)}{Q_j}. \quad (15)$$

One can observe that computing of the Jacobian matrix by (14) and (15) is a rather complicated and time-consuming process if there are many frequency vectors given and the Dirichlet mixture model consists of many components each with a number of parameters. One also knows that the approximated Jacobian matrix can be calculated by the finite difference method [42]. In the Levenberg-Marquardt optimization, we alternatively used both the explicitly given and approximated Jacobian computations. Interestingly, we found that employing the approximated Jacobian computations resulted in the optimized parameters whose utility gave us the most accurate classification. In all our studies with the Dirichlet mixtures this was the case. Not surprisingly, successful replacement of the Jacobian by the approximated computations in large problems does not contradict the theory [34].



a)



b)

Figure 4. The classification performance for the two decision classes of the *E.coli* dataset. The classification performance by the ROC curves for the classification categories 'cp' (a) and 'im' (b) is depicted for the Dirichlet mixture classifier (Dirichlet mixt.), for the rule classifier (Rules), and for the Naïve Bayes classifier. AUC stands for the area under the ROC curve and the parameter σ expresses standard deviation of the AUC computation

The results of the Dirichlet mixture classifier seen in Figures 4 and 5 are obtained using the Levenberg-Marquardt optimization with the approximated Jacobian computations. The Dirichlet mixture classifier was superior to the other two classifiers, the rule and Naïve Bayes classifiers. Despite the Naïve Bayes classifier performed similarly to the Dirichlet mixture classifier for the category 'cp', it did much worse for the other categories. Actually, the Naïve Bayes method all the data objects assigned to the same classification category 'cp', and the accuracy for the other categories did not exceed 0%. Most of the data objects in the test dataset fell in the category 'cp', namely 143, the numbers of the data objects in the other categories are: 77 from 'im', 52 from 'pp', 35 from 'imU', 20 from 'om', 5 from 'omL', 2 from 'imL', 2 from 'imS'. Probably the Naïve Bayes classifier overestimated itself in the training process and did not manage to recognize the objects from the other categories. Its overall classification accuracy is far from high, 44.4%. The rule classifier failed for this dataset and the ROC curves for all the categories resemble random classification. The overall accuracy of the rule classifier is not competing, 35.6%.

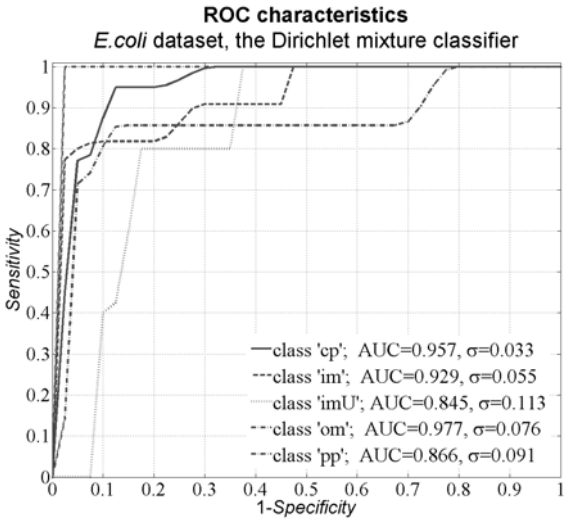


Figure 5. The classification performance by the ROC curves of the Dirichlet mixture classifier for all the decision classes of the *E.coli* dataset used to be in the testing procedure

The Dirichlet mixture classifier was able to classify the data objects near evenly accurate and it achieved the accuracy of 80% in total. The ROC curves of the Dirichlet mixture classifier for all the categories (Figure 5) show strong discriminative power in classifying the *E.coli* protein data objects. The authors of the dataset published they arrived at the accuracy of as high as 81% [41] using the probabilistic model developed specially for this dataset.

4.3. Ionosphere dataset

This dataset is for classification of radar returns from the ionosphere where the targets were free electrons in the ionosphere [43]. This is a binary classification task and there are two classification categories, 'g' and 'b'. "Good" radar returns (category 'g') are those showing evidence of some type of structure in the ionosphere. "Bad" returns (category 'b') are those that do not; their signals pass through the ionosphere. Received signals were processed using an autocorrelation function whose arguments were the pulse numbers and comprise 34 attributes for classification. The number of the objects (radar returns) in the dataset is 351. There are no missing attribute values.

As in the case of the Arrhythmia dataset, the classification by the Dirichlet mixture was most accurate when using the parameters optimized by the genetic algorithm. Interestingly, the similar results were obtained with the maximum likelihood (13) estimate and the posterior mean estimate (12) suggesting that both techniques can successfully be used for estimation of the parameters for the likelihood expressions.

We found that the Dirichlet mixture model consisting of the 32 components each with the 34 pseudo frequency parameters $\{\alpha_{ji}\}_{i=1}^{34}$ furnished us with the best classification results. To train the Dirichlet mixture classifier, we ran 200 genetic algorithm's generations, applied the even-odd crossover algorithm and the flip mutation algorithm for genes. The other parameters were as follows: crossover probability, 0.9, mutation probability, 0.1, population size, 60, the number of individuals to be replaced in each generation, 9. We also tried various genetic configurations by changing scaling schemes for the fitness function (no scaling, linear scaling, power law scaling, etc.), the individual replacement schemes (the parent, worst, best, or other individuals), the population housekeeping strategies (how many populations to keep, overlapping or in parallel) [33] and discovered that for this problem the sigma truncation scaling, replacement of the worst individuals, management of the overlapping populations gave us the most reasonable optimization.

The Dirichlet mixture classifier outperformed the other two classifiers (The Naïve Bayes method was incorrect for this dataset and we omitted it from the further analysis). The overall accuracy for the Dirichlet mixture and rule classifiers are 76% and 28%, respectively. The Dirichlet mixture classifier correctly predicted all 32 data objects from category 'g' (100%) and did properly for 6 ones from category 'b' having 18 data objects in total (33.3%). The rules did not predict the objects to belong to category 'g' at all; the ROC curve (Figure 6) indicates the poor classification with the curve segment up to the level of specificity of 0.5. The accuracy of the rule classifier for category 'b' was 77.8%; the classifier correctly predicted 14 objects out of 18. Conversely, in the case of classification by the rules, predictions were made either for category 'b' or there were no predictions made at all. Therefore, the specificity of the rules for the dataset is low and the ROC curves demonstrate the poorer

performance with respect to that of the Dirichlet mixture classifier.

The results (Figure 6) suggest that the Dirichlet mixture classifier is sensitive to the amount of data used in the training dataset. Category 'g' contained most of the data objects, namely 225, and the Dirichlet mixture classifier performed for this category without errors, however it was not so accurate in classifying the objects from category 'b' which enclosed 126 data objects. On the other hand, the area under the ROC curve (Figure 6) for this class is the same as that for class 'b', showing that the predictions for class 'b' could be more reliable and accurate if, for classification, one chooses the threshold value gained from the ROC analysis.

The authors of the dataset affirm [43] they reached an accuracy of 96% using the neural network architecture built particularly for this dataset.

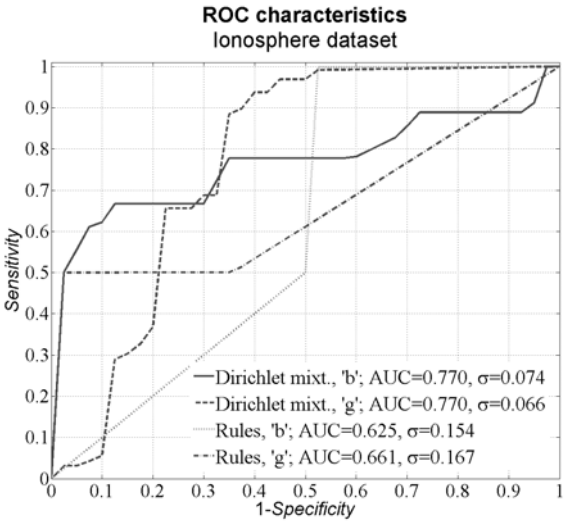


Figure 6. The classification performance by the ROC curves of the Dirichlet mixture classifier and the rule classifier for all the decision classes of the Ionosphere dataset. The parameter σ means the standard deviation of the area under the ROC curve (AUC) computation

5. Conclusions

By its attempt to formally describe real-world data and, at the same moment, to simplify and to clarify data representation of the real world, classification has necessarily become a very important field in many scientific investigations and industrial activities. However, there is no accurate universal classification model developed to fit any classification problem. In this article, we have proposed a method based on the Dirichlet mixture statistics that is aimed at wide range of various problems. To show usefulness of this method to classify data from the diverse fields of human activity, we chose three different databases to test the classifiers on. After the classifiers were applied on the medical, biological, and physical databases, it was observed by the ROC analysis that the Dirichlet mixture classifier outperformed the other two classifiers we had chosen to compare the Dirichlet mixture classifier with. For all three datasets used, neither the classification by rules in the context of the rough sets theory nor the Naïve Bayes classifier could compete in accuracy with the Dirichlet mixtures. To judge the Dirichlet mixtures classifier against the originally built classification architectures (published by the authors of the datasets), we contrasted the accuracies obtained by the methods. The Dirichlet mixtures performed almost identically with the classifier constructed for the biological dataset. For the other two datasets, the mixtures did not attain accuracy as high as that of the originally developed classifiers. However, we were unable to accomplish ROC analysis for the original classifiers (we did not have data), which would tell more about the reliability and the performance of the classifiers. It should be noted as well, that the proportions and the distribution of the training and testing datasets used in our tests necessarily differed from those used in the original studies, what made up a bias of several percents of accuracy. On the other hand, we did not accomplish cross-validation procedure that would have helped us to more precisely asses the overall accuracy of the Dirichlet mixture classifier.

By applying the Dirichlet mixture classifier on the three different datasets, we simulated practicability of the classification method for data of any kind. The feasibility of the Dirichlet mixture classifier extends to configuration of the classification models that would consist of any number of Dirichlet components each with a defined number of pseudo frequency parameters. This option enables to tune the Dirichlet mixture for a specific problem as this was demonstrated in the text. We implemented the Dirichlet mixture method as a computational tool and also compiled as a library for further development of machine learning environments. The latter increases the practical value of the method and this study.

References

[1] **F. Hampel**. Some thoughts about classification. *In: K. Jajuga, A. Sokolowski, H.-H. Bock (eds.). Classification, Clustering, and Data Analysis. Recent advances and applications. Springer*, 2002, 5-26.

[2] C. K. Eveland, D. A. Socolinsky, C. E. Priebe, *et al.* A hierarchical methodology for class detection problems with skewed priors. *Journal of Classification*, 22(1), 2005, 17-48.

[3] V. N. Vapnik. The Nature of Statistical Learning Theory. *Springer*, 2nd ed., 1999.

[4] R. Wehrens, L. M. C. Buydens, C. Fraley, *et al.* Model-based clustering for image segmentation and large datasets via sampling. *Journal of Classification*, 21(2), 2004, 231-253.

[5] P. Dempster, N. M. Laird, D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1), 1977, 1-38.

[6] S. Han, B. Lee, S. T. Yu, *et al.* Fold recognition by combining profile-profile alignment and support vector machine. *Bioinformatics*, 21(11), 2005, 2667-2673.

[7] **J. Cheng, P. Baldi**. A machine learning information retrieval approach to protein fold recognition. *Bioinformatics*, 2006 *Mar* 17, *to be published*.

[8] **C. Leslie, R. Kuang**. Fast String Kernels using Inexact Matching for Protein Sequences. *Journal of Machine Learning Research*, 5, 2004, 1435-1455.

[9] **H. Rangwala, G. Karypis**. Profile-based direct kernels for remote homology detection and fold recognition. *Bioinformatics*, 21(23), 2005, 4239-4247.

[10] **H. Ding, I. Dubchak**. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17(4), 2001, 349-358.

[11] **H. Saigo, J. P. Vert, N. Ueda, *et al.*** Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11), 2004, 1682-1689.

[12] **T. J. Santner, D. E. Duffy**. The statistical analysis of discrete data. *Springer-Verlag*, 1989.

[13] **T. Lassmann, E. L. Sonnhammer**. Kalign – an accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics*, 6, 2005, 298.

[14] **R. Hughey, K. Karplus, A. Krogh**. SAM: Sequence Alignment and Modeling software system, version 3. *Technical Report UCSC-CRL*-99-11, *University of California*, 1999.

[15] **K. Karplus, B. Hu**. Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set. *Bioinformatics*, 17(8), 2001, 713-720.

[16] **K. Sjolander, K. Karplus, M. Brown, *et al.*** Dirichlet mixtures: a method for improving detection of weak but significant protein sequence homology. *CABIOS*, 12(4), 1996, 327-345.

[17] **N. Lartillot, H. Philippe**. A bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. *Mol. Biol. Evol.*, 21(6), 2004, 1095-1109.

[18] **S. R. Waterhouse, D. MacKay, A. J. Robinson**. Bayesian methods for mixtures of experts. *Advances in Neural Information Processing Systems*, 8, *MIT Press*, 1996, 351-357.

[19] **G. Arfken**. Mathematical methods for physicists. *Academic Press*, 3rd ed., 1985.

[20] **W. J. Ewens, G. R. Grant**. Statistical methods in bioinformatics. *Springer*, 2001.

[21] **H. Jeffreys**. Theory of Probability. *Oxford University Press*, 3rd ed., 1998.

[22] **R. Durbin, S. R. Eddy, A. Krogh, *et al.*** Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. *Cambridge University Press*, 1999.

[23] **A. Ohrn, J. Komorowski, A. Skowron, *et al.*** The design and implementation of a knowledge discovery toolkit based on rough sets: The ROSETTA system. *In: L. Polkowski, A. Skowron (eds.). Rough Sets in Knowledge Discovery* 2: *Applications, Case Studies and Software Systems. Studies in Fuzziness and Soft Computing*, 19, *Physica-Verlag*, 1998, 376-399.

[24] **L. Polkowski**. Advances in soft computing: Rough sets. *Physica-Verlag*, 2002.

[25] **L. Polkowski, A. Skowron**. Rough Sets in Knowledge Discovery 1: Methodology and Applications. *Studies in Fuzziness and Soft Computing*, *Springer-Verlag*, 1998.

[26] **A. Ohrn**. The ROSETTA C++ library. *SourceForge web resource: http://rosetta.sourceforge.net*, 2000.

[27] **K. G. Norsett, A. Laegreid, H. Midelfart, *et al.*** Gene expression based classification of gastric carcinoma. *Cancer Lett.*, 210(2), 2004, 227-237.

[28] **A. K. Dubey**. Using rough sets, neural networks, and logistic regression to predict compliance with cholesterol guidelines goals in patients with coronary artery disease. *AMIA Annu Symp Proc.*, 2003, 834.

[29] **T. R. Hvidsten, A. Laegreid, J. Komorowski**. Learning rule-based models of biological process from gene expression time profiles using gene ontology. *Bioinformatics*, 19(9), 2003, 1116-1123.

[30] **T. R. Hvidsten, B. Wilczynski, A. Kryshtafovych, *et al.*** Discovering regulatory binding-site modules using rule-based learning. *Genome Res.*, 15(6), 2005, 856-866.

[31] **D. Ripley**. Pattern recognition and neural networks. *Cambridge University Press*, 1996.

[32] **J. Hromkovic**. Algorithmics for Hard Problems. *Springer*, 2nd ed., 2002.

[33] **D. E. Goldberg**. Genetic Algorithms in Search, Optimization, and Machine Learning. *Addison-Wesley*, 1989.

[34] **J. Nocedal, S. J. Wright**. Numerical optimization. *Springer*, 2000.

[35] **Y. Bykov**. Time-Predefined and Trajectory-Based Search: Single and Multiobjective Approaches to Exam Timetabling. *Phd Thesis, Nottingham*, 2003.

[36] **W. H. Press, B. P. Flannery, S. A. Teukolsky, *et al.*** Numerical Recipes in C: The Art of Scientific Computing. *Cambridge University Press*, 1992.

[37] **J. Newman, S. Hettich, C. L. Blake, *et al.*** UCI Repository of machine learning databases. *Web resource: http://www.ics.uci.edu/~mlearn/MLRepository.html*, 1998.

[38] **J. Hanley, B. McNeil**. The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology*, 143(1), 1982, 29-36.

[39] **H. A. Güvenir, B. Acar, G. Demiröz, *et al.*** A supervised machine learning algorithm for arrhythmia analysis. *Proceedings of the Computers in Cardiology Conference*, 24, 1997, 433-436.

[40] **B. Efron, R. J. Tibshirani**. An introduction to the bootstrap. *Chapman & Hall/CRC*, 1994.

[**41**] **P. Horton, K. Nakai**. A probablistic classification system for predicting the cellular localization sites of proteins. *Intelligent Systems in Molecular Biology*, 1996, 109-115.

[42] **H. Levy, F. Lessman**. Finite Difference Equations. *Dover*, 1992.

[43] **V. G. Sigillito, S. P. Wing, L. V. Hutton, *et al.*** Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10(3), 1989, 262-266.

Received March 2006.