# On application of rough sets to Bayesian network construction from data

**Karina Łuksza**
Faculty of Informatics,
Warsaw University,
Banacha 2, Warsaw 02-097, Poland
k.luksza@students.mimuw.edu.pl

**Hung Son Nguyen**
Institute of Mathematics
Warsaw University,
Banacha 2, Warsaw 02-097, Poland
son@mimuw.edu.pl

## Abstract

This paper presents a method of Bayesian network construction from data. Many technical problems like discretization, attribute ordering and network structure construction are discussed and solved by applying rough set methodology to data analysis. The proposition has been implemented on base of RSES-lib and illustrated by experiments on benchmark data sets. The experimental results are be presented and compared with other Bayesian network methods.

**Keywords:** Bayesian networks, Rough Sets, Classification.

## 1 Introduction

Bayesian networks ([17]) provide an efficient tool for data analysis and classification. They have a structure of directed acyclic graph (DAG) $\mathcal{D} = (V, \overrightarrow{E})$, where $V$ is a set of vertices and $\overrightarrow{E}$ is set of directed edges. The variables are situated in the vertices $V$ of the graph. The edges connecting them denote direct dependencies. Bayesian networks encode the probability distribution of the variables they include. They are designed in such way that the following condition holds: *Every variable $X$ in Bayesian network $\mathcal{D}$ is independent on condition of its parents $\Pi_{\mathcal{D}}(X)$ from its nondescendats $Nd_{\mathcal{D}}(X)$.*

$$P(X|\Pi_{\mathcal{D}}(X)) = P(X|\Pi_{\mathcal{D}}(X), Nd_{\mathcal{D}}(X))$$

This condition implies that the probability distribution represented by Bayesian network is decomposable and can be derived from the following equation:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|\Pi_{\mathcal{D}}(X_i))$$

Bayesian networks can be used in classification tasks. In such case, one variable is distinguished and its state is unknown. The task is to predict the most probable value of this variable, basing on the knowledge learned from the set of examples. Having a case $u = (x_1, \ldots, x_n)$ one wants to specify the most probable decision value $v_d$

$$v(u) = \arg\max_{v_d \in V_d} [prior(v_d)P(x_1, \ldots, x_n|v_d)]$$

where $V_d$ is a set of possible decision values.

First the Bayesian network structure must be constructed and then the probability values $P(X_i|\Pi_{\mathcal{D}}(X_i))$ for each variable $X_i$ must be calculated. The subject of this work was the construction of Bayesian networks from data. In general the problem is known to be NP-hard [1], so efficient and effective heuristics must be used. The method described in this paper is based on Rough Set theory ([15]) and frequency reduct calculation ([16]).

The knowledge derived from real life data may contain some noise information. That's why the approximate, not accurate independencies are to be calculated. The task is to find the simplest structure possible while keeping the specified level of accuracy. Accurate networks with complex structure have worse ability of

generalization. The new cases that did not occur in the sample data may be unclassified. This paper presents the method of entropy approximation introduced in [20]. Then the classification results are compared to those derived on Bayesian networks built with the use of classical statistical method - MDL principle, implemented in [5].

## 2 Basic notions

An information system is a pair $\mathbb{A} = (U, A)$, where $U$ is a set of examples and $A = \langle a_1, \ldots, a_n \rangle$ is a set of attributes. For each attribute $a \in A$ a function $a : U \to V_a$ is specified, where $V_a$ denotes the set of all possible values on $a$.

Given a subset of attributes $B \subseteq A$, $B = \{a_{i_1}, \ldots, a_{i_m}\}$ a $B$-information function $B : U \to V_B^U$ is defined as $B(u) = (a_{i_1}(u), \ldots, a_{i_m}(u))$, where $V_B^U$ is a set of all vector values on B that occur in U.

A decision table is an information system $\mathbb{A} = (U, A \cup \{d\})$, where $d$ is a decision attribute. Let's assume that $V_d = \{1, \ldots, r\}$. Then the $k$-th decision class is the set: $class(k) = \{u \in U : d(u) = k\}$, where $1 \leq k \leq r$.

In rough set theory, reducts are the minimal subsets of attributes containing a necessary portion of *information* about the set of all attributes [15]. Let us present a generalized definition of reducts with respect to any monotone measure on attribute sets.

**Definition 2.1.** *Let $\mathbb{A} = (U, A)$ be a given information system and let $\mu : \mathcal{P}(A) \to \mathbb{R}^+$ be an arbitrary monotone measure.*

1. *Any $B \subseteq A$ is said to "define $A$ with respect $f$" (thus $B$ $\mu$-defines $A$) if*
$$\mu(B) = \mu(A) \tag{1}$$
2. *$B$ is a $\mu$-reduct for $A$ if it satisfies (1) and none of its proper subsets does it.*

The information reduct is specified by the following function which counts the number of objects discerned by $B$, $\mu_{\mathbb{A}}(B) =$

$$|\{(u, u') \in U : B(u) \neq B(u')\}|$$

The decision oriented reduct is defined by the function $\mu_{\mathbb{A}}(B) =$

$$|\{(u, u') \in U : d(u) \neq d(u') \wedge B(u) \neq B(u')\}|$$

which counts objects with different decision attribute values discerned by $B$.

Those reducts do not handle inconsistences in data. To overcome this problem a frequency decision reducts based on frequencies in data will be used.

To define a proper $\mu_{\mathbb{A}}$ function we have to define a probability function based on frequencies in data first. For every vector value $w_B \in V_B^U$ the probability of its occurrence is defined by the frequency:

$$P(w_B) = \frac{|\{u \in U : B(u) = w_B\}|}{|U|}$$

The conditional probability of the occurrence of decision $v_d$ conditioned on $w_B$ is equal to the frequency

$$P(v_d|w_B) = \frac{|\{u : B(u) = w_B \wedge d(u) = v_d\}|}{|\{u \in U : B(u) = w_B\}|}$$

The problem of attribute reduction stated in rough set framework can be solved with the use of probability functions.

To specify this function we need to define the coefficient:

$$\mu_{d|B}(u) = P(d(u)|B(u))$$

A subset $B \subseteq A$ $\mu$-preserves information if for every object $u \in U$ the following condition holds:

$$\mu_{d|A}(u) = \mu_{d|B}(u) \tag{2}$$

If for none of the proper subsets of $B$ the condition (2) is satisfied, then $B$ is a $\mu$-decision reduct for $A$.

The $\mu_{\mathbb{A}}$ function for $\mu$-decision reduct is defined as:

$$\mu_{\mathbb{A}}(B) = \sum_{u \in U} \mu_{d|B}(u)$$

This function is monotone. Adding new attributes to $B$ increases the overall decision discernibility, so $\mu_{\mathbb{A}}(B) \leq \mu_{\mathbb{A}}(B)$ for $B \subseteq B'$.

### 2.1 Approximate decision reducts

To weaken the condition of reduct accuracy, the concept of $\varepsilon$-approximate $\mu$-reduct was introduced. The $B \subseteq A$ is a $\varepsilon$-approximate decision reduct if

$$\mu_{\mathbb{A}}(B) \geq \mu_{\mathbb{A}}(A)(1-\varepsilon) \tag{3}$$

In our research we used approximation based on the measure of entropy.

For a given information system $\mathbb{A}$, the entropy of attributes $B \subseteq A$ is defined as:

$$H_{\mathbb{A}}(B) = -\sum_{w_B \in V_B^U} P_{\mathbb{A}}(w_B) \log_2 P_{\mathbb{A}}(w_B)$$

In decision tables $\mathbb{A} = (U, A \cup \{d\})$ the conditional entropy of $d$ conditioned on $B \subseteq A$ is defined as

$$\begin{aligned} H_{\mathbb{A}}(d|B) &= H_{\mathbb{A}}(B \cup \{d\}) - H_{\mathbb{A}}(B) \\ &= \sum_{w_B \in V_B^U} P_{\mathbb{A}}(w_B) \cdot H_{w_B}(d) \end{aligned}$$

where

$$H_{w_B}(d) = -\sum_{v_d \in V_d} P(v_d|w_B) \log_2 P(v_d|w_B)$$

**Definition 2.2.** *[20, 21] Let $\varepsilon \in [0,1)$, decision table $A = (U, A \cup \{d\})$ and $B \subseteq A$ be given. We say that $B$ is $(H,\varepsilon)$-approximate decision reduct if*

$$H_{\mathbb{A}}(d|B) + \log_2(1-\varepsilon) \leq H_{\mathbb{A}}(d|A) \tag{4}$$

*and none of its proper subsets satisfies this condition.*

The condition (4) is equivalent to (3). Having the $\mu_{\mathbb{A}}$ function defined as: $\mu_{\mathbb{A}}(d|B) = \sqrt[|U|]{\prod_{u \in U} \mu_{d|B}(u)}$ one can check that

$$H_{\mathbb{A}}(d|B) = -log_2 \mu_{\mathbb{A}}(d|B)$$

as it was shown in [1].

## 3 Bayesian Networks

The Bayesian classification rule in decision tables is as follows:

$$v(u) = \arg\max_{v_d \in V_d} [prior(v_d) P_{\mathbb{A}}(A(u)|v_d)]$$

Bayesian networks provide a decomposition of the factor $P(A(u)|v_d)$. The chain rule implies that:

$$P_{\mathbb{A}}(v_1, \ldots, v_n|v_d) = \prod_{i=1}^{n} P_{\mathbb{A}}(v_i|v_d, v_1, \ldots, v_{i-1})$$

**Proposition 3.1.** *[20] Let $\mathbb{A} = (U, A \cup \{d\})$ and $A = \langle a_1, \ldots, a_n \rangle$ be given. Let us assume that for each table $\mathbb{A}_i = (U, \{d, a_1, \ldots, a_{i-1}\} \cup \{a_i\})$, $i = 1, \ldots, n$ a $\mu$-decision reduct $B_i$ has been found. Then, for any given $u \in U$ the decision value calculated by (3) is equal to:*

$$\begin{aligned} v(u) \quad = \quad & \arg\max_{v_d \in V_d} prior(v_d) \cdot \\ & \prod_{i:d \in B_i} P_{\mathbb{A}}(a_i(u)|v_d, B_i(u) \setminus \{d\}) \end{aligned}$$

The complete proof in [20].

**Theorem 3.1.** *[17] Let a decision table $\mathbb{A} = (A, U \cup \{d\})$ and an ordered set of attributes $\langle a_1, \ldots, a_n \rangle$ be given. Let's assume that for each decision table*

$$\mathbb{A}_i = (U, \{d, a_1, \ldots, a_{i-1}\} \cup \{a_i\})$$

*a $\mu$-decision reduct $B_i$ was found. Then DAG $\mathcal{D} = (A \cup \{d\}, \overrightarrow{E})$ is Bayesian network, where $\overrightarrow{E} = \bigcup_{i=1}^{n} \{\langle b, a_i \rangle : b \in B_i\}$.*

The probability distribution tables are defined by the appropriate functions $P_{\mathbb{A}_i}$.

The Bayesian network with minimal number of edges is supposed to have the best generalization ability. The optimization problem is to find such initial ordering of attributes $\langle a_1, \ldots, a_n \rangle$ and then to choose such reducts $B_i$ on $\mathbb{A}_i$ decision tables that the number of edges $\sum_{i=1}^{n} |B_i|$ would be minimal. The polynomial time solution is not known, as the problem is NP-hard [20].

### 3.1 Approximate Bayesian Networks

The Bayesian networks defined in the theorem (3.1) might be too rigorous in real life applications. In this work the so-called $(H,\varepsilon)$-approximate Bayesian networks [19] are considered. Those networks satisfy the following condition: Given a positive value $\varepsilon \in [0,1)$

the overall Bayesian net entropy is at most $-\log_2(1-\varepsilon)$ higher than the entropy of the decision table:

$$H_{\mathbb{A}}(\mathcal{D}) + \log_2(1-\varepsilon) \leq H_{\mathbb{A}}(A \cup \{d\})$$

where the entropy of Bayesian network $\mathcal{D}$ is equal to:

$$\begin{aligned} H_{\mathbb{A}}(\mathcal{D}) &= \sum_{i=0}^{n} H_{\mathbb{A}}(a_i|\Pi_{\mathcal{D}}(a_i)) \\ &= \sum_{i=0}^{n} H_{\mathbb{A}}(a_i|B_i) \end{aligned}$$

where $a_0$ denotes $d$ for easier notation reasons.

**Definition 3.1.** *Let $\varepsilon \in [0,1)$, decision table $\mathbb{A} = (U, A \cup \{d\})$ and an ordered set of attributes $A = \langle a_1, \ldots, a_n \rangle$ be given. We say that a set of reducts $B = \langle B_i, \ldots, B_n \rangle$ on $\mathbb{A}_i$ decision is $(H, \varepsilon)$- approximately consistent with $\mathbb{A}$ if*

$$\sum_{i=1}^{n} H_{\mathbb{A}}(a_i|B_i) + \log_2(1-\varepsilon) \leq H_{\mathbb{A}}(A|d)$$

In this work, the Bayesian networks were constructed with a specified value of $\varepsilon$ defining the accuracy of every reduct, not an entire network. Given such $\varepsilon \in [0,1)$ one can estimate the accuracy of the Bayesian network. For every component $H_{\mathbb{A}}(a_i|B_i)$, $i = 0, 1, \ldots, n$ the inequality

$$H_{\mathbb{A}}(a_i|B_i) + \log_2(1-\varepsilon) \leq H_{\mathbb{A}}(a_i|A_i)$$

holds. This implies that

$$\begin{aligned} H_{\mathbb{A}}(\mathcal{D}) &= \sum_{a_i \in A} H_{\mathbb{A}}(a_i|B_i) \\ &\leq \sum_{i=0}^{n} H_{\mathbb{A}}(a_i|A_i) - \log_2(1-\varepsilon)^n \end{aligned}$$

For any decision table $\mathbb{A} = (U, A \cup \{d\})$ the identity

$$\begin{aligned} H_{\mathbb{A}}(A \cup \{d\}) &= H_{\mathbb{A}}(A \cup \{a_0\}) \\ &= \sum_{i=0}^{n} H_{\mathbb{A}}(a_i|A_i) \end{aligned}$$

holds, implying that

$$H_{\mathbb{A}}(\mathcal{D}) + \log_2(1-\varepsilon)^n \leq H_{\mathbb{A}}(A \cup \{d\})$$

$\mathcal{D}$ is an $(H, \varepsilon_{\mathcal{D}})$-approximate Bayesian network where:

$$\varepsilon_{\mathcal{D}} = -\sum_{i=1}^{n} \binom{n}{i} (-\varepsilon)^i$$

as

$$(1-\varepsilon)^n = 1 + \sum_{i=1}^{n} \binom{n}{i} (-\varepsilon)^i$$

### 3.2 Multi Bayesian Networks

Multi Bayesian networks were introduced in [8], [7]. Multi Bayesian network is a tuple $\langle prior(d), P_{\mathcal{D}_1}, \ldots, P_{\mathcal{D}_r} \rangle$, where $prior(d)$ is a prior probability distribution of a decision variable $d$ and $\mathcal{D}_1, \ldots, \mathcal{D}_r$ are local Bayesian networks for each of the decision classes. The independent networks are build for every decision class. Let us denote a decision table for $i$-th decison class as $\mathbb{A}^i = (U_i, A \cup \{d\})$, where $U_i = class(i)$. The classification is done with:

$$v = \arg\max_{v_d \in V_d} [prior(v_d) P_{\mathcal{D}_i}(a_1, \ldots, a_n)]$$

which is equivalent to:

$$v = \arg\max_{i=1,\ldots,r} P_{\mathbb{A}}(i) \prod_{i=1}^{r} P_{\mathbb{A}^i}(a_j|B_i^j(u))$$

where $B_i^j$ is a $\mu$-decision reduct on decision table $\mathbb{A}_i^j = (U_i, \{a_1, \ldots, a_{i-1}\} \cup \{a_i\})$ for $i = 1, \ldots, n$ and $j = 1, \ldots, r$.

Multi Bayesian networks are believed to be more accurate than normal Bayesian networks in case where there are many decision classes. In our research they turned out to be better than normal Bayesian networks.

## 4 Methods

An application of the rough set theory to building Bayesian networks from data needs some data preprocessing.

### 4.1 Discretization

The discussed method of constructing networks does not handle continuous data, so first a discretization must be performed. In

this work the method based on rough set theory was used. This method was described in [12], [13], [14]. The goal is to find such set of cuts on $V_a$

$$D_a = \{(a, c_1^a), (a, c_2^a), \ldots, (a, c_k^a)\}$$

where $l_a < c_1^a < c_2^a < \ldots < c_k^a < r_a$, $l_a$ is the smallest and $r_a$ is the biggest value in $V_a$, defining the partition on $V_a$ into sub-intervals

$$V_a = [l_a, c_1^a) \cup [c_1^a, c_2^a) \cup \ldots \cup [c_k^a, r_a)$$

The number of intervals should be minimalized, while keeping the acceptable level of information loss.

The other well known discretization method introduced by Fayyad and Irani in [4] is based on the measure of entropy. It was used in [5].

A special category of discretization methods was invented specially for Bayesian networks, i.e. [6], [11]. Optimal cuts are found during a network construction. The statistical scoring function like MDL [9] or BE [2] evaluates the goodness of the network with the discretized values.

### 4.2 Attribute ordering

The next step was to find such initial ordering of attributes that would provide the best classification results. We used the method described in [3]. Diaz and Corchado introduce the measure of significance of each attribute. According to that measure, the attributes with less significance should come first. This ordering let us discover as many relations with decision attribute on the left side as possible. The measure is defined as follows: given a decision table $\mathbb{A} = (U, A \cup \{d\})$ and a set of attributes $B \subseteq A$. We measure the quality of approximation $d$ by $B$ as:

$$\gamma_{\mathbb{A}}(d|B) = \frac{POS_{\mathbb{A}}(d|B)}{|U|}$$

where $POS_{\mathbb{A}}(d|B)$ is a positive region of $B$ in $\mathbb{A}$. The measure of significance of attribute $a \in A$ conditioned on $d$ is a value:

$$\eta_{\mathbb{A}}(d|a) = \gamma_{\mathbb{A}}(d|A) - \gamma_{\mathbb{A}}(d|A \setminus \{a\})$$

This measure does not perform well when there are many attributes and proportionally little examples. To overcome this problem, another method was introduced. An array $n \times n$ is constructed, where the value in $i$-th row and $j$-th column is equal to $\eta_{\mathbb{A}}(a_i|a_j)$. It expresses the dependency level of $a_i$ on $a_j$. The measure of significance of attribute $a_i$ is described by the equation:

$$v_i = \sum_{j \neq i} \eta_{\mathbb{A}}(a_i|a_j)\eta_{\mathbb{A}}(a_j|a_i)$$

More detailed desciption of that measure can be found in [3].

### 4.3 Reduct selection

The algorithm of Bayesian network construction consists of $n$ steps, where $n$ is a number of attributes. On every step a $(H, \varepsilon)$-approximate $\mu$-decision reduct $B$ on decision table $\mathbb{A}_i = (U, \{d, a_1, \ldots, a_{i-1}\} \cup \{a_i\})$ is assessed. We used a simple method to find the set of reducts. We permuted the set of attributes $\{d, a_1, \ldots, a_{i-1}\}$ $p$ times. Starting with the empty set of attributes $B_i$ we have searched every permutation from left to right adding the successive attributes to $B_i$ until the condition $H(a_i|B_i) + \log_2(1 - \varepsilon) \leq H(a_i|A_i)$ was satisfied. After having searched $p$ random pemutations, we had a result set of at most $p$ reducts. The one one that would provide the best network performance had to be chosen.

The two opposite methods of choosing reducts $B_i$ on a set of attributes $A_i = \{d, a_1, \ldots, a_{i-1}\} \cup \{a_i\}$ were tested. They were based on the value of entropy difference: $H_{\mathbb{A}}(a_i|B_i) - H_{\mathbb{A}}(a_i|A_i)$

- reduct with minimal entropy difference
- reduct with maximal entropy difference

### 4.4 Prior probabilities

In the next step we came across two problems:

1. Unclassified objects. This problem occurred when the classified object contained a new combination of values, that

the Bayesian network could not generalize on.

2. Not enough support to specify the probability values for $P(a_i|B_i)$.

To solve them we used *smoothing*, the method of combining prior marginal probabilities $P_{\mathbb{A}}(a_i)$ with $P(a_i|B_i)$, described in [5]. The result probability had the form:

$$\widetilde{P}_{\mathbb{A}}(d(u)|B(u)) = \frac{N^0}{|U|+N^0}P_{\mathbb{A}}(d(u)) + \frac{|U|}{N_0+|U|}P_{\mathbb{A}}(d(u)|B(u)) \quad (5)$$

where $N^0$ is a parameter chosen due to experiments. The larger $N_0$, the higher importance of prior probability.

## 5 Experimental Results

The experiments performed on approximate Bayesian networks and multi Bayesian networks were compared to the results from [5], were both: a Bayesian network and multi Bayesian network were constructed. The networks from [5] were build with the use of MDL-principle and Fayyad and Irani discretization method [4].

The same datasets from UCI database were used and analogous experiments were performed. On larger datasets (chess, letter, mofn-3-7-10, satimage, segment, shuttle-small and waveform-21) train and test experiments were performed, on other cross validation tests with 5 folds. The value $N^0$ from equation (5) was set to 5, as it was in in [5].

Eleven values of reduct accuracy $\varepsilon$ were tested, which are: 0.0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9.

While choosing reducts as described in Section 4.3, the number of permutations $p$ was set to 100.

Because of a random nature of the algorithm, all the experiments were repeated 5 times, the average on results was provided.

We also built networks omitting the step of attribute ordering, to check the usefulness of the chosen method.

The experiments were performed with the use of the system based on RSESlib [18] implemented as the part of [10] work. This application includes all the discussed methods of data preprocessing. It lets the user to specify all parameters needed while Bayesian networks construction, which are: $\varepsilon$, number of reducts, reduct choose method, whether include smoothing or not and the type of a network. One can specify a classification test of two types: train and test and cross validation. It provides a convenient user interface and displays networks that were built.

The Figure 1 presents a Bayesian network build on iris dataset.

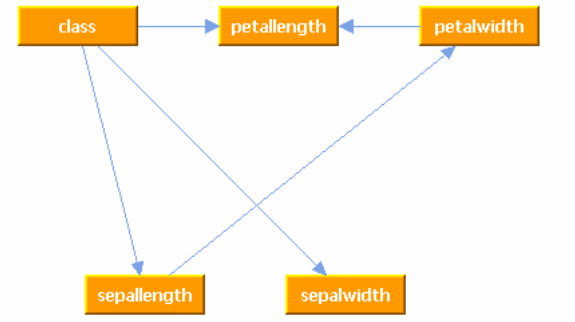


Figure 1: Bayesian network for iris

The best results that were achieved are shown in the table (5) and compared to those from [5].

Multi Bayesian network with the reduct selection method based on the principle of minimal entropy difference, turned out to be the best classifier among all the classifiers tested. However the difference between two reduct selection methods was very little for multi Bayesian networks. Simple Bayesian networks were more sensitive to the reduct selection method. In that case the minimal entropy difference again gained better classification accuracy results.

The choice of reduct accuracy $\varepsilon$ had much less effect on the multi Bayesian networks than on the simple ones.

Compared to Bayesian networks built with the MDL-principle, approximate networks yielded better results. They did not han-

Table 1: Classification results

| | Dataset | BN from [5] | | Approx BN | | Approx M-BN | |
|---|---|---|---|---|---|---|---|
| | | **BN** | **M-BN** | min ent | max ent | min ent | max ent |
| 1 | australian | 86.23 | **86.52** | 86.44 | 84.49 | 86.05 | 86.19 |
| 2 | breast | 96.92 | 96.63 | **97.26** | 95.53 | 97.15 | **97.26** |
| 3 | chess | 95.59 | **96.34** | 94.78 | 94.73 | 95.91 | 96.1 |
| 4 | cleve | 81.39 | 81.76 | **83.04** | 77.3 | 82.54 | 82.54 |
| 5 | corral | 97.60 | **99.20** | 97.67 | 97.66 | 97.88 | 98.05 |
| 6 | crx | 85.60 | 86.37 | **86.96** | 84.66 | 85.16 | 85.31 |
| 7 | diabetes | 75.39 | **76.04** | 63.19 | 66.9 | 72.33 | 72.49 |
| 8 | flare | 82.74 | **82.65** | 72.71 | 72.73 | 73.71 | 75.99 |
| 9 | german | **72.30** | 72.20 | 65.2 | 68.85 | 70.53 | 70.23 |
| 10 | glass | 55.57 | 65.91 | 63.95 | 62.33 | **67.9** | 66.86 |
| 11 | glass2 | 75.49 | 77.90 | 79.15 | 79.15 | **82.6** | 81.98 |
| 12 | heart | 82.22 | 83.70 | 83.7 | 79.81 | 82.13 | **84.72** |
| 13 | hepatitis | **91.25** | 90.00 | 80.31 | 77.5 | 90 | 90.63 |
| 14 | iris | 94.00 | 93.33 | **95.33** | 94.33 | 94.67 | **95.33** |
| 15 | letter | 75.02 | 80.10 | 86.67 | 83.92 | **89.28** | 87.15 |
| 16 | lymphography | 75.03 | **79.75** | 72.79 | 57.12 | 79.35 | 78.85 |
| 17 | mofn-3-7-10 | 85.94 | 86.43 | 93.07 | 78.52 | 94.73 | **96.07** |
| 18 | pima | 75.00 | **76.30** | 63.02 | 66.99 | 73.4 | 72.78 |
| 19 | satimage | 59.20 | 77.10 | 81.72 | 81.14 | **86.54** | **86.54** |
| 20 | segment | **93.51** | 90.26 | 88.18 | 76.34 | 91.82 | 90.65 |
| 21 | shuttle-small | 99.17 | 98.97 | 87.49 | 94.52 | **99.64** | **99.64** |
| 22 | soybean-large | 58.54 | 87.01 | 91.2 | 90.8 | **91.42** | 91.02 |
| 23 | vehicle | 61.00 | 64.20 | 63.49 | 63.96 | **72.29** | 69.13 |
| 24 | vote | 94.94 | 90.11 | **95.63** | 94.08 | 94.48 | 93.33 |
| 25 | waveform-21 | 69.45 | 76.85 | 79.51 | 69.63 | 79.37 | **79.79** |
| | **avg** | 80.76 | 83.83 | 82.1 | 79.72 | **85.24** | 85.15 |

dle some of the datasets (flare, german, diabetes, pima) giving low classification accuracy results much below the expected average of known results. But on the rest of datasets approximate Bayesian networks scored so well that they managed to have an overall average classification accuracies higher than those of [5].

The tests examining the usefulness of the chosen attribute ordering method were performed. The classification tests were performed on the same datasets with the initial ordering of the attributes. The goal was to check does the ordering can deteriorate the classification accuracy. In fact for Bayesian networks, on 18 of 25 datasets average results were better for ordered attributes. Multi Bayesian networks scored better with ordering on 16 datasets.

## 6 Conclusions

The implementation of approximate Bayesian and the tests performed proved, that they are good classifiers. In the comparison to the networks built with the use of the MDL-principle, they performed well, and turned out to have the better average accuracy value. Multi Bayesian networks were more stable and accurate than normal Bayesian networks. They yielded better classification results on most of the datasets. The attribute ordering method did not ensure the best classification results on all datasets. In feature work the more adequate method should be found.

**Acknowledgements**

The research has been partially supported by the grant 3T11C00226 from Ministry of Scientific Research and Information Technology of the Republic of Poland. This paper was

written on the basis of Master Dissertation of the first author [10].

## References


[1] D. Chickering, D. Geiger, and D. E. Heckerman. Learning Bayesian Networks is NP-Hard. Technical report, MSR-TR-94-17, November 1994.

[2] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, pages 309 – 347, 1992.

[3] F. Diaz and J. Corchado. Rough sets for detecting dependence relationships and bayesian networks construction. In *Proc. of IPMU2000*, Madrit, Spain, 2000.

[4] U. Fayyad and K. Irani. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. *Machine Learning, 13th IJCAI*, 2.

[5] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian Network Classifiers. *Machine Learning 29*.

[6] N. Friedman and M. Goldszmidt. Discretizing Continous Attributes While Learning Bayesian Networks. In *Proc. of International Conference on Machine Learning*, pages 157– 165, 1996.

[7] D. Geiger and D. Heckerman. Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence 82*.

[8] D. Heckerman. *Probabilistic Similarity Networks*. MA : MIT Press, Cambridge, 1991.

[9] W. Lam and F. Bacchus. Learning Bayesian belief networks An approach based on the MDL principle. *Computational Intelligence*, (10):269–293, 1994.

[10] K. Łuksza. Metody klasyfikacji za pomoca sieci bayesowskich. Master's thesis, Warsaw University, 2005.

[11] S. Monti and G. Cooper. Multivariate Discretization Method for Learning Bayesian Networks from Mixed Data. In *Proc. of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, pages 404–413, 1998.

[12] H. Nguyen and A. Skowron. Quantization of real value attributes. In *Proc. of Second Joint Annual Conf. on Information Sciences*, pages 34–37, Wrightsville Beach, North Carolina, 1995.

[13] H. S. Nguyen and S. H. Nguyen. Some efficient algorithms for rough set methods. In *Proc. of the Sixth International Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 1451–1456, Granada, Spain, July 1-5 1996.

[14] S. H. Nguyen. *Discretization of real value attributes. Boolean reasoning approach.* PhD thesis, Warsaw Uniwersity, 1997.

[15] Z. Pawlak. *Rough sets - theoretical aspects of reasoning about data*. Kluwer Academic Publishers, Dordrecht, 1991.

[16] Z. Pawlak and A. Skowron. Rough membership functions. *Advances in the Dempster-Shafer theory of evidence*, pages 251–271, 1994.

[17] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[18] RSES. http://logic.mimuw.edu.pl/∼rses/.

[19] D. Ślęzak. Foundations of Entropy Based Bayesian Networks: Theoretical Results and Rough Set Based Extraction from Data. In *Proc. of IPMU'00*, pages 248–255, Madrit, Spain, 2000.

[20] D. Ślęzak. Approximate Bayesian networks. *Technologies for constructing intelligent systems: tools*, pages 313–325, 2002.

[21] D. Ślęzak. Approximate entropy reducts. *Fundam. Inf.*, 53(3,4):365–390, 2002.