Approximate bayesian network classifiers

Dominik Ślęzak, Jakub Wróblewski

Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warsaw, Poland
{slezak,jakubw}@pjwstk.edu.pl

Abstract. Bayesian network (BN) is a directed acyclic graph encoding probabilistic independence statements between variables. BN with decision attribute as a root can be applied to classification of new cases, by synthesis of conditional probabilities propagated along the edges. We consider approximate BNs, which almost keep entropy of a decision table. They have usually less edges than classical BNs. They enable to model and extend the well-known Naive Bayes approach. Experiments show that classifiers based on approximate BNs can be very efficient.

1 Introduction

Bayesian network (BN) is a directed acyclic graph (DAG) designed to encode knowledge about probabilistic conditional independence (PCI) statements between considered variables, within a given probabilistic space [6]. Its expressive power increases while removing the edges, unless it causes a loss of control of exactness of derivable PCI-statements. When mining real-life data, one needs less accurate, approximate criteria of independence. We base such an approximation on the information measure of entropy [4], by letting a reasonably small increase of its quantity during the edge reduction. It leads to approximate BNs corresponding to approximate PCI-statements, introduced in [7].

BN can model the flow of information in decision tables, while reasoning about new cases. Necessary probabilities can be calculated directly from training data, by substituting the foregoing decision values in a loop. One can maximize the product of such probabilities and choose the most probable decision value. This is, actually, an example of the bayesian reasoning approach (cf. [2]).

We analyze how the strategies of choosing the approximation threshold and searching for corresponding approximate BNs can influence the new case classification results. We extract optimal DAGs from data in a very basic way, just to provide a material for simulations. Development of more sophisticated methods is a direction for further research. Some algorithms for learning approximate BNs are proposed in [8]. Various other approaches to extraction of classical BNs (cf. [3]) are worth generalizing onto the approximate case as well.

Although BN-related framework can be regarded as purely probabilistic, let us stress its relationship to the rough set theory [5], by means of correspondence between fundamental notions, like e.g. these of decision reduct and Markov boundary (cf. [7]), as well as between optimization problems concerning extraction of approximate BNs and rough-set-based models from data (cf. [8]).

2 Probabilities in information systems

Following [5], we represent data as information systems – tuples $\mathbb{A} = (U, A)$. Each attribute $a \in A$ is identified with function $a : U \to V_a$, for V_a denoting the set of all possible values on a. Let us assume ordering $A = \langle a_1, \ldots, a_n \rangle$. For any $B \subseteq A$, consider B-information function, which labels objects $u \in U$ with vectors $\langle a_{i_1}(u), \ldots, a_{i_m}(u) \rangle$, where values of $a_{i_j} \in B$, $j = 1, \ldots, m$, occur due to the ordering on A. We denote this function by $B : U \to V_B^U$, where $V_B^U = \{B(u) : u \in U\}$ is the set of all vectors of values on B occurring in $\mathbb{A}$.

Classification problems concern distinguished decisions to be predicted under information provided over conditional attributes. For this purpose, one represents data as a decision table $\mathbb{A} = (U, A \cup \{d\})$, $d \notin A$. One can use various classification methodologies, provided, e.g., by statistical calculus [2]. Occurrence of $v_d \in V_d$ conditioned by $w_B \in V_B^U$, can be expressed as probability

$$P_{\mathbb{A}}(v_d / w_B) = |\{u \in U : B(u) = w_B \wedge d(u) = v_d\}| \,/\, |\{u \in U : B(u) = w_B\}| \quad (1)$$

For a given $\alpha \in [0, 1]$, we say that α-inexact decision rule $(B = w_B) \Rightarrow_\alpha (d = v_d)$ is satisfied iff $P_{\mathbb{A}}(v_d / w_B) \geq \alpha$, i.e., iff for at least $\alpha \cdot 100\%$ of objects $u \in U$ such that $B(u) = w_B$ we have also $d(u) = v_d$. The strength of the rule is provided by prior probability $P_{\mathbb{A}}(w_B) = |\{u \in U : B(u) = w_B\}| \,/\, |U|$. It corresponds to the chance that an object $u \in U$ will satisfy the rule's left side. One can consider such probabilities not only for the case of a distinguished decision attribute at the right side of a rule. In case of bayesian approaches to the new case classification one uses probabilistic rules with decision features involved in their left sides.

3 Probabilistic decision reducts

Each pair $(B, u) \in \mathcal{P}(A) \times U$ generates approximate decision rule pointing at the $d(u)$-th decision class. It is described by means of the following parameters:

Definition 1. *Let $\mathbb{A} = (U, A \cup \{d\})$, $B \subseteq A$ and $u \in U$ be given. By the accuracy and support coefficients for (B, u) we mean, respectively, quantities*

$$\mu_{d/B}(u) = P_{\mathbb{A}}(d(u)/B(u)) \qquad \mu_B(u) = P_{\mathbb{A}}(B(u)) \quad (2)$$

In the context of the above coefficients, the rough-set-based principle of reduction of redundant information [5] corresponds to the following notion:

Definition 2. *Let $\mathbb{A} = (U, A \cup \{d\})$ be given. $B \subseteq A$ μ-preserves d iff*

$$\forall_{u \in U} \left[\mu_{d/B}(u) = \mu_{d/A}(u)\right] \quad (3)$$

B is a μ-decision reduct iff it satisfies (3) and none of its proper subsets does it.

Property (3) is an example of a probabilistic conditional independence (PCI) statement. Usually, PCI is defined over subsets of variables considered within a discrete product probabilistic space, over all possible configurations of vectors of values. Since we deal with probabilistic distributions derived directly from information systems, let us focus on the following, equivalent [7] definition:

Definition 3. *Let $\mathbb{A} = (U, A)$ and $X, Y, Z \subseteq A$ be given. We say that Y makes X conditionally independent from Z iff*

$$\forall_{u \in U} P_{\mathbb{A}}(X(u)/Y(u)) = P_{\mathbb{A}}(X(u)/(Y \cup Z)(u)) \tag{4}$$

Corollary 1. *Let $\mathbb{A} = (U, A \cup \{d\})$ and $B \subseteq A$ be given. B is a μ-decision reduct iff it is a Markov boundary of d within A, i.e., it is an irreducible subset, which makes d probabilistically independent from the rest of A.*

4 Entropy-based approximations

Each $B \subseteq A$ induces in $\mathbb{A} = (U, A \cup \{d\})$ the bunch of inexact decision rules $B = B(u) \Rightarrow_{\mu_{d/B}(u)} d = d(u)$ for particular objects $u \in U$. One can measure the quality of B in terms of both accuracy and support of such rules.

Definition 4. *Let $\mathbb{A} = (U, A \cup \{d\})$ and $B \subseteq A$ be given. We put*

$$G_{\mathbb{A}}(B) = \sqrt[|U|]{\Pi_{u \in U} \mu_B(u)} \qquad G_{\mathbb{A}}(d/B) = \sqrt[|U|]{\Pi_{u \in U} \mu_{d/B}(u)} \tag{5}$$

$G_{\mathbb{A}}$ corresponds to the measure of information entropy adapted to the rough set, statistical and machine learning methodologies in various forms (cf. [4, 7]).

Definition 5. *Let $\mathbb{A} = (U, A)$ and $X \subseteq A$ be given. By entropy of X we mean*

$$H_{\mathbb{A}}(X) = -\sum_{w_X \in V_X^U} P_{\mathbb{A}}(w_X) \log_2 P_{\mathbb{A}}(w_X) \tag{6}$$

By entropy of X conditioned by Y we mean

$$H_{\mathbb{A}}(X/Y) = \begin{cases} H_{\mathbb{A}}(X \cup Y) - H_{\mathbb{A}}(Y) & \text{iff } Y \neq \emptyset \\ H_{\mathbb{A}}(X) & \text{otherwise} \end{cases} \tag{7}$$

Proposition 1. *Let $\mathbb{A} = (U, A \cup \{d\})$ and $B \subseteq A$ be given. We have equalities*

$$H_{\mathbb{A}}(B) = -\log_2 G_{\mathbb{A}}(B) \qquad H_{\mathbb{A}}(d/B) = -\log_2 G_{\mathbb{A}}(d/B) \tag{8}$$

Given the above interpretation of $H_{\mathbb{A}}$, let us focus on the following way of approximate preserving of accuracy under the conditional attribute reduction.

Definition 6. *Let $\varepsilon \in [0, 1)$, $\mathbb{A} = (U, A \cup \{d\})$ and $B \subseteq A$ be given. We say that B ε-approximately μ-preserves d iff $G_{\mathbb{A}}(d/B) \geq (1 - \varepsilon) G_{\mathbb{A}}(d/A)$, i.e., iff*

$$H_{\mathbb{A}}(d/B) + \log_2(1 - \varepsilon) \leq H_{\mathbb{A}}(d/A) \tag{9}$$

We say that B is an ε-approximate μ-decision reduct (ε-approximate Markov boundary) iff it satisfies (9) and none of its proper subsets does it.

Definition 7. *Let $\varepsilon \in [0, 1)$, $\mathbb{A} = (U, A)$ and $X, Y, Z \subseteq A$ be given. We say that Y makes X conditionally ε-approximately independent from Z iff*

$$H_{\mathbb{A}}(X/Y) + \log_2(1 - \varepsilon) \leq H_{\mathbb{A}}(X/Y \cup Z) \tag{10}$$

Such a criterion of *approximate* probabilistic conditional independence is more robust to possible fluctuations in real life data. Moreover, we have equivalence of the notions of independence and 0-approximate independence.

5 Bayesian networks

Bayesian network (BN) has the structure of a directed acyclic graph (DAG) $\mathcal{D} = (A, \overrightarrow{E})$, where $\overrightarrow{E} \subseteq A \times A$. The objective of BN is to encode conditional independence statements involving groups of probabilistic variables corresponding to elements of A, in terms of the following graph-theoretic notion [6]:

Definition 8. *Let DAG $\mathcal{D} = (A, \overrightarrow{E})$ and $X, Y, Z \subseteq A$ be given. We say that Y d-separates X from Z iff any path between any $x \in X \setminus Y$ and any $z \in Z \setminus Y$ comes through: (1) a serial or diverging connection covered by some $y \in Y$,[1] or (2) a converging connection not covered by Y, having no descendant in Y.*[2]

Let us formulate the notion of BN in terms of data analysis:

Definition 9. *Let $\mathbb{A} = (U, A)$ and DAG $\mathcal{D} = (A, \overrightarrow{E})$ be given. We say that $\mathcal{D}$ is a bayesian network for $\mathbb{A}$ iff for any $X, Y, Z \subseteq A$, if Y d-separates X from Z, then Y makes X conditionally independent from Z.*

Theorem 1. *([6]) Let $\mathbb{A} = (U, A)$, $A = \langle a_1, \ldots, a_n \rangle$, be given. Let us assume that for each table $\mathbb{A}_i = (U, \{a_1, \ldots, a_{i-1}\} \cup \{a_i\})$, $i > 1$, a μ-decision reduct B_i is provided. Then we obtain a bayesian network $\mathcal{D} = (A, \overrightarrow{E})$ defined by*

$$\overrightarrow{E} = \bigcup_{i=1}^{n} \{\langle b, a_i \rangle : b \in B_i\} \tag{11}$$

In [7] the following approach to approximation of the notion of BN was proposed:

Definition 10. *Let $\varepsilon \in [0, 1)$, $\mathbb{A} = (U, A)$ and DAG $\mathcal{D} = (A, \overrightarrow{E})$ be given. We say that $\mathcal{D}$ is ε-approximately consistent with $\mathbb{A}$ iff*

$$H_{\mathbb{A}}(\mathcal{D}) + \log(1 - \varepsilon) \leq H_{\mathbb{A}}(A) \tag{12}$$

where $H_{\mathbb{A}}(\mathcal{D}) = \sum_{a \in A} H_{\mathbb{A}}(a / \{b \in A : \langle b, a \rangle \in \overrightarrow{E}\})$.

Condition (12) keeps the aggregate information induced by $\mathcal{D}$-based local conditional distributions *close* to that encoded within the whole of $P_{\mathbb{A}}(A)$.

Definition 11. *Let $\varepsilon \in [0, 1)$, $\mathbb{A} = (U, A)$, $\mathcal{D} = (A, \overrightarrow{E})$ be given. We say that $\mathcal{D}$ is an ε-approximate bayesian network (ε-BN) iff for any $X, Y, Z \subseteq A$, if Y d-separates X from Z, then Y makes X ε-approximately independent from Z.*

The following result generalizes Theorem 1. In particular, any DAG $\mathcal{D}$ built on the basis of μ-decision reducts is 0-approximately consistent with a given $\mathbb{A}$, as well as any 0-approximate bayesian network is a bayesian network.

Theorem 2. *[7] Let $\varepsilon \in [0, 1)$ and $\mathbb{A} = (U, A)$ be given. Each DAG which is ε-approximately consistent with $\mathbb{A}$ is an ε-approximate BN for $\mathbb{A}$.*

[1] Descriptions '*serial*', '*diverging*' and '*converging*' correspond to directions of arrows meeting within a given path, in a given node

[2] We say that b is a *descendant* of a iff there is a directed path from a towards b in $\mathcal{D}$

6 BN-based classification

Bayesian decision models are related to the analysis of approximations of distribution $P_{\mathbb{A}}(A(u)/v_d)$. One can let $u \in U$ be classified as having decision value

$$v = \arg\max_{v_d \in V_d}[prior(v_d)P_{\mathbb{A}}(A(u)/v_d)] \tag{13}$$

for $prior : V_d \to [0,1]$. Let us set up an arbitrary ordering $A = \langle a_1, \ldots, a_n \rangle$ and denote by V_i the set of all values of a_i. We decompose $P_{\mathbb{A}}(A/d)$ by noting that for any supported combination of values $v_d \in V_d$, $v_i \in V_i$, $i = 1, \ldots, n$, one has

$$P_{\mathbb{A}}(v_1, \ldots, v_n/v_d) = \prod_{i=1}^{n} P_{\mathbb{A}}(v_i/v_d, v_1, \ldots, v_{i-1}) \tag{14}$$

Proposition 2. *[7] Let $\mathbb{A} = (U, A \cup \{d\})$, $A = \langle a_1, \ldots, a_n \rangle$, be given. Assume that for each table $\mathbb{A}_i = (U, \{d, a_1, \ldots, a_{i-1}\} \cup \{a_i\})$, $i = 1, \ldots, n$, a μ-decision reduct B_i has been found. For any $u \in U$, decision obtained by (13) equals to*

$$v = \arg\max_{v_d \in V_d} prior(v_d) \prod_{i:\, d \in B_i} P_{\mathbb{A}}(a_i(u)/v_d, (B_i \setminus \{d\})(u)) \tag{15}$$

The way of classifying objects in Proposition 2 corresponds to the DAG construction in Theorem 1, if applied to $\mathbb{A} = (U, A \cup \{d\})$, for d at the first position of the ordering over $A \cup \{d\}$. We obtain a scheme of the bayesian classification, where conditional probabilities are propagated along the DAG structure, beginning with decision as the root. In particular, we obtain an interpretation of the *Naive Bayes* approach (cf. [2]), formulated in terms of the following principle:

$$v = \arg\max_{v_d \in V_d} prior(v_d) \prod_{a \in A} P_{\mathbb{A}}(a(u)/v_d) \tag{16}$$

$\mathcal{D}_0 = (A \cup \{d\}, \overrightarrow{E}_0)$ corresponding to (16) is given by the following set of edges:

$$\overrightarrow{E}_0 = \bigcup_{a \in A} \{\langle d, a \rangle\} \tag{17}$$

If $\mathcal{D}_0$ is BN for $\mathbb{A} = (U, A \cup \{d\})$, then the performance of (16) is the same as in case of (13) and (15). This is because in $\mathcal{D}_0$ any pair $a, b \in A$ is d-separated by d, so – according to Definition 9 – d makes them independent from each other.

7 Related optimization problems

BN can be regarded as optimal in terms of the law of encoding of PCI-statements and/or performance of DAG-based classification scheme (15). In both cases ε-approximately consistent $\mathcal{D} = (A \cup \{d\}, \overrightarrow{E})$, which minimize quantity of $Q_1(\mathcal{D}) = |\overrightarrow{E}|$, are worth finding. Let us consider the following exemplary measures as well:

$$Q_2(\mathcal{D}) = \sum_{a \in A} |V^U_{\pi(a)}| \quad Q_2^d(\mathcal{D}) = \frac{\sum_{a \in \delta(d)} |V^U_{\pi(a)}|}{|\delta(d)|} \quad Q_3(\mathcal{D}) = \sum_{a \in A} H_{\mathbb{A}}(\pi(a)) \tag{18}$$

where $\pi(a) = \{b \in A \cup \{d\} : \langle b, a \rangle \in \overrightarrow{E}\}$ and $\delta(a) = \{b \in A \cup \{d\} : \langle b, a \rangle \in \overrightarrow{E}\}$ denote, respectively, the sets of parents and children of node $a \in A \cup \{d\}$ in $\mathcal{D}$.

Q_2 counts all distinct premises of inexact decision rules, which may occur while using classification scheme (15). Q_2^d takes into account only these rules, which directly participate to the classification process. Q_3 is partially correlated with Q_2 but it is more flexible with respect to the rule supports.[3]

To search for classical BNs, one can begin with extracting initial (partial) ordering and then search for locally optimal Markov boundaries [3]. One can also search for (approximate) BNs over the space of DAGs or, as proposed in [8], apply order-based genetic algorithm to work on permutations of nodes. The following result explains, how one can construct ε-BNs by basing on orderings.

Proposition 3. *Let $\varepsilon \in [0,1)$, $\mathbb{A} = (U, A)$, $A = \langle a_1, \ldots, a_n \rangle$ and $\overrightarrow{E} = \{\langle a_i, a_j \rangle : 1 \leq i < j \leq n\}$ be given. Set up an ordering over $\overrightarrow{E}$. Consider the following steps:* (i) *Take the first $e \in \overrightarrow{E}$,* (ii) *Check whether $\mathcal{D} = (A, \overrightarrow{E} \setminus \{e\})$ is ε-approximately consistent with $\mathbb{A}$,* (iii) *If it is, remove e from $\overrightarrow{E}$,* (iv) *Repeat* (i)-(iii) *for foregoing elements of $\overrightarrow{E}$. DAG obtained at the end is an irreducible ε-BN for $\mathbb{A}$.*

Let us skip discussion about complexity of searching for BNs [4] and focus on simulations showing what should be optimized to get the most efficient classifiers.

8 Experimental results

We analyzed several known benchmark data tables available at [1]. Experiments were performed on relatively large data sets, all of them equipped with the testing table and, in general, discrete values of attributes (`DNA splices` data set was considered in its preprocessed version[5]). For a given $\varepsilon \in [0,1)$, ε-BNs were created by using a method described in Proposition 3. The obtained networks were then applied to classify the testing table, by using techniques described in Section 6. The aim of our experiments was to learn how to choose the most suitable optimization measures and the best levels of ε.

For each considered $\varepsilon \in [0,1)$ we generated randomly 50 DAGs with decision as a root. Then we collected the classification rates for the testing table. Average classification rates of 10 DAGs being the best with respect to each optimization measure were calculated.[6] During initial calculations the most interesting results were obtained for the approximation thresholds near to that corresponding to the DAG-based interpretation of Naive Bayes, i.e.: [7]

$$\varepsilon_0 = 1 - 2^{H_{\mathbb{A}}(A/d) - \sum_{a \in A} H_{\mathbb{A}}(a/d)} \tag{19}$$

Let us parameterize interval $[0,1)$ as follows:

$$[0,1) = \{\varepsilon_\alpha, \alpha \in \mathbb{R}\} \quad \text{where} \quad \varepsilon_\alpha = 1 - (1 - \varepsilon_0)^{(1-\alpha)} \tag{20}$$

[3] One could consider Q_3^d, defined analogously to Q_2^d, as well as other measures.
[4] We would like to refer the reader to [7] and [8] for further details.
[5] It consists of 20 out of original 60 conditional attributes, each of them with 4 values.
[6] Besides functions defined by (18), we considered also other possibilities.
[7] Indeed, $\varepsilon_0 \in [0,1)$ defined by (19) is the minimal approximation threshold, for which DAG $\mathcal{D}_0 = (A \cup \{d\}, \overrightarrow{E}_0)$, defined by (17), is ε_0-approximately consistent with $\mathbb{A}$.

Results for ε_α-BNs with various levels of $\alpha \in \mathbb{R}$ are shown in Table 1. In majority of cases either Q_2^d or Q_3 turn out to be the best choice.

α	Q_1	Q_2	Q_2^d	Q_3
0.0025	76.15%	76.28%	76.30%	76.28%
0.04	76.32%	77.26%	76.65%	76.27%
0.09	74.99%	75.42%	75.03%	75.56%

α	Q_1	Q_2	Q_2^d	Q_3
0.0008	95.54%	95.50%	95.66%	95.54%
0.0027	95.65%	95.43%	95.43%	95.67%
0.0064	94.32%	94.69%	95.07%	94.09%

Table 1. Average rates of the proper classification. Left: `letter`. Right: `DNA splices`.

Fig. 1 shows correlation between values of these functions (calculated on the training data) and final results of correct classification of the test objects.

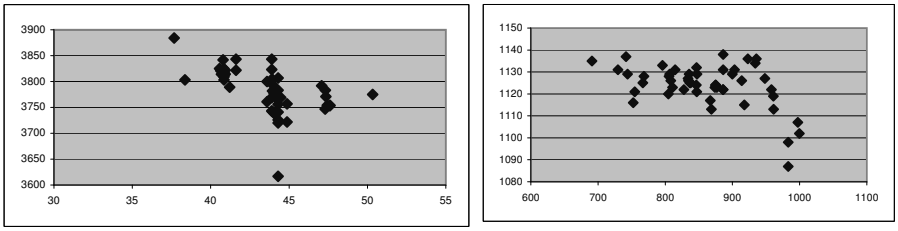

Fig. 1. Values of quality measure (horizontal) and classification results (vertical) for `letter` (left, measure Q_3) and `DNA splices` (right, measure Q_2^d) databases.

Fig. 2 shows that ε-BNs being optimized by using Q_2^d and Q_3 are significantly more efficient than the average.

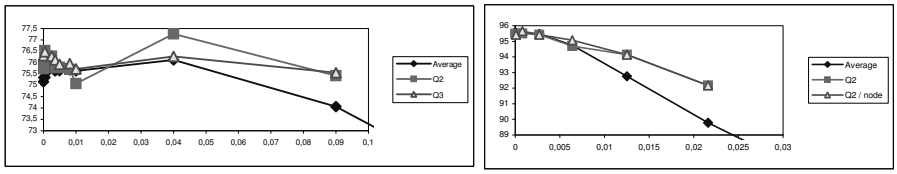

Fig. 2. Classification results on `letter` (left) and `DNA splices` (right) databases for different α values (horizontal): average and for minima of measures Q_3, Q_2^d.

Some regularities repeat for all tables we have worked on: By setting $\alpha = 0$ we obtain fair ε_α-BNs, similar to ε_0-BN related to Naive Bayes. The most efficient α values are usually between 0.0005 and 0.005. After reaching some threshold (depending on data) the average performance decreases but its diversification increases. It may lead to obtaining very good ε_α-BNs for suboptimal $\alpha \in \mathbb{R}$.

In Fig. 3 the best results found in our experiments are collected. The classification rate of Naive Bayes method is significantly exceeded, not only for optimal, but often even for average (random-ordered) case. Classification result for `DNA splices` is one of the best ever obtained. It is also interesting to observe that relatively small change of ε_0-BN may dramatically improve classification rate. Fig. 4 illustrates the case of improvement from 74.8% to 80.2%.

Table	obj.×attr. (test ob.)	Naive	Best BN
`letter`	15000×17 (5000)	74.8%	80.2%
`DNA spl.`	2000×21 (1186)	95.6%	96.2%
`optdigits`	3823×65 (1797)	82.0%	82.2%
`soybean`	307×36 (376)	64.0%	78.9%

Fig. 3. The best classification results obtained during experiments.

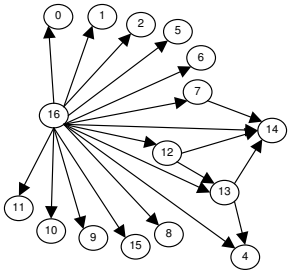

Fig. 4. The best network found for `letter` data (classification rate: 80.2%). Attributes are numbered starting with 0; attribute 16 is a decision.

9 Conclusions

We considered approximate BNs, almost keeping entropy of a decision table. Experiments confirmed potential efficiency of classifiers based on such BNs, depending on the choice of the approximation parameter and the strategy of searching for optimal graphs. We extracted optimal DAGs from data in a very basic way, just to provide a material for simulations. Development of more sophisticated methods is a direction for further research. Some algorithms for learning approximate BNs are proposed in [8]. Various other approaches to extraction of classical BNs (cf. [3]) are worth generalizing onto the approximate case as well.

Acknowledgements: Supported by Polish National Committee for Scientific Research (KBN) grant No. $8T11C02519$.

References

1. Bay, S.D.: The UCI Machine Learning Repository, `http://www.ics.uci.edu/ml`
2. Box, G.E.P., Tiao, G.C.: Bayesian Inference in Statistical Analysis. Wiley (1992).
3. Buntine, W.: A guide to the literature on learning probabilistic networks from data. IEEE Transactions on Knowledge and Data Engineering (1996).
4. Kapur, J.N., Kesavan, H.K.: Entropy Optimization Principles with Applications. Academic Press (1992).
5. Pawlak, Z.: Rough sets – Theoretical aspects of reasoning about data. Kluwer Academic Publishers (1991).
6. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1988).
7. Ślęzak, D.: Approximate Bayesian networks. In: B. Bouchon-Meunier, J. Gutierrez-Rios, L. Magdalena, R.R. Yager (eds), Technologies for Contructing Intelligent Systems 2: Tools. Springer-Verlag (2002) pp. 313–326.
8. Ślęzak, D., Wróblewski, J.: Order-based genetic algorithms for extraction of approximate bayesian networks from data. In: Proc. of IPMU'2002. France (2002).