Order-based genetic algorithms for extraction of approximate bayesian networks from data

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

Abstract

Approximate bayesian nets (a-BN's) *almost* keep the information entropy of data tables and encode knowledge about approximate dependencies between features. Advantages of applying a-BN's to real-life data analysis are discussed. Genetic-based algorithmic framework for extracting a-BN's from data is proposed.

1 Introduction

Bayesian network (BN) is a directed acyclic graph (DAG) designed to encode knowledge about conditional independence statements between considered variables, within a given probabilistic space [7]. Expressive power of such an encoding increases while removing DAG's edges, unless it causes a loss of control of exactness of derivable statements. BN-structures can be also used to model the flow of information while reasoning about new cases by analogy to records gathered in data tables (cf. [10]).

Classical BN's correspond to exact probabilistic independence, which is too accurate while mining real-life data, because of the risk of possible noises or fluctuations. Thus, one needs to generalize fundamental concepts and results concerning bayesian networks, to let them deal with appropriately defined approximate independence statements. The idea of basing such a generalization onto the rough set framework [5] originates in the fact that it provides a wide range of tools for expressing data inconsistency, in particular, those related to rough membership functions [6]. Rough membership decision reducts (cf. [10]) correspond to probabilistic Markov boundaries, crucial for effective construction of BN-models ([7]). Various criteria for the reduction of redundant information by the approximate preserving of rough membership information [9] can be applied to approximate the concepts related to Markov boundaries and bayesian networks.

We focus on approximations dedicated to the information measure of entropy [3, 8], by letting a reasonably small increase of its quantity while reducing edges of a network. It leads to the notion of an approximate bayesian network (a-BN) corresponding to approximate conditional independence statements (cf. [10]). We discuss advantages of dealing with a-BN's while analyzing data. We also provide genetic-based algorithmic tools for efficient extraction of optimal a-BN-based knowledge models from real-life data tables.

Order-based genetic algorithms (o-GA's) are known to be able to deal with complex NP-hard optimization problems efficiently [2, 4]. Exemplary applications of o-GA's to searching for optimal rough-set-based decision reducts were developed in [11]. We show that the same framework can be used in case of approximate decision reducts and Markov boundaries. Furthermore, we discuss possibility of its extension in order to cope with directed acyclic graphs inducing approximate bayesian networks.

2 Elementary notions

Data can be represented as an information system $\mathbb{A} = (U, A)$, where 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 ([5]). Let us write $A = \langle a_1, \ldots, a_n \rangle$ according to some ordering over the set of attributes. For any $B \subseteq A$, one can consider information function $B : U \to V_B^U$, which labels objects $u \in U$ with vectors $B(u) = \langle a_{i_1}(u), \ldots, a_{i_m}(u) \rangle$, where values of successive $a_{i_j} \in B$, $j = 1, \ldots, m$, occur due to the ordering assumed on A. The set $V_B^U = \{B(u) : u \in U\}$ gathers all vectors of values on B supported in $\mathbb{A}$.

Reasoning about data can be stated, e.g., as the classification problem concerning a distinguished decision to be predicted under information provided over the rest of attributes. For this purpose, one represents data as a decision table $\mathbb{A} = (U, A \cup \{d\})$, $d \notin A$. To express conditions→decision dependencies, one can use various methodologies, provided, e.g., by statistical calculus or theory of rough sets [5]. In this paper we focus on probabilities. For instance occurrence of $v_d \in V_d$ conditioned by $w_B \in V_B^U$, can be expressed as probability $P_{\mathbb{A}}(v_d / w_B) =$

$$= \frac{|\{u \in U : B(u) = w_B \wedge d(u) = v_d\}|}{|\{u \in U : B(u) = w_B\}|} \qquad (1)$$

Obviously, one can consider such probabilities not only for the case of a distinguished decision attribute. For a given $\alpha \in [0, 1]$, 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 quantity $P_{\mathbb{A}}(w_B) = |\{u \in U : B(u) = w_B\}| / |U|$. It corresponds to the chance that an object $u \in U$ will satisfy the rule's left side.

Although all models considered in the paper are of probabilistic nature, we refer them also to the rough set framework, especially while dealing with optimization problems of extraction from data. In the theory of rough sets probabilities take the form of rough membership functions. They were introduced in [6] in purpose of measuring degrees of inclusion of indiscernibility classes $[u]_B = \{u' \in U : B(u) = B(u')\}$ into concepts $X \subseteq U$ being approximated in terms of $B \subseteq A$. Yet another interpretation can be obtained by noticing that 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*

$$\begin{array}{rcl} \mu_{d/B}(u) & = & P_{\mathbb{A}}(d(u)/B(u)) \\ \mu_B(u) & = & P_{\mathbb{A}}(B(u)) \end{array} \qquad (2)$$

General rough-set-based principle of reduction of redundant conditional information takes here the following form:

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

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

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, the notion of PCI is defined over mutually disjoint subsets of variables considered within some 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 [10] definition:

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

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

Corollary 1 *Let $\mathbb{A} = (U, A \cup \{d\})$ and $B \subseteq A$ be given. (1) B μ-preserves d, iff it makes d conditionally independent on $A \setminus B$, in terms of $P_{\mathbb{A}}$. (2) 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 independent on the rest of A.*

3 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 the BN-based methodology is to encode conditional independence statements involving groups of probabilistic variables corresponding to elements of A, in terms of the following graph-theoretic notion [7]:

Definition 4 *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 node in $X \setminus Y$ and any node in $Z \setminus Y$ comes through: (1) a serial or diverging connection covered by some element of Y,*[1] *or (2) a converging connection not covered by Y, having no descendant in Y.*[2]

Let us formulate the notion of a bayesian network in terms of data analysis:

Definition 5 *Let $\mathbb{A} = (U, A)$ and DAG $\mathcal{D} = (A, \overrightarrow{E})$ be given. We say that $\mathcal{D}$ is a Bayesian net for $\mathbb{A}$, iff for any $X, Y, Z \subseteq A$, if Y d-separates X from Z, then Y makes X conditionally independent on Z.*

Theorem 1 *([7]) 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 DAG $\mathcal{D} = (A, \overrightarrow{E})$, where*

$$\overrightarrow{E} = \bigcup_{i=1}^{n} \{\langle b, a_i \rangle : b \in B_i\} \tag{5}$$

is a bayesian network for $\mathbb{A}$.

One can combine the bayesian network techniques with the rough set framework (see [10] for further references). Bayesian decision models deal with the rules with decision situated at their left side. They are related to the analysis of distribution $P_{\mathbb{A}}(A(u)/v_d)$, letting $u \in U$ be classified as having decision value

$$v = \arg\max_{v_d \in V_d} [prior(v_d) P_{\mathbb{A}}(A(u)/v_d)] \tag{6}$$

where $prior : V_d \to [0, 1]$ expresses prior knowledge about probability of decision values. 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

$$P_{\mathbb{A}}(v_1, .., v_n/v_d) = \prod_{i=1}^{n} P_{\mathbb{A}}(v_i/v_d, v_1, .., v_{i-1}) \tag{7}$$

The attribute reduction can be now related to sub-component distributions:

Proposition 1 *([10]) Let $\mathbb{A} = (U, A \cup \{d\})$, $A = \langle a_1, \ldots, a_n \rangle$, 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 μ-decision reduct B_i has been found. Then, for any given $u \in U$, the decision value calculated by (6) is equal to*

$$v = \arg\max_{v_d \in V_d} prior(v_d) P_{\mathbb{A}}(\mathcal{D}(u)/v_d) \tag{8}$$

where $P_{\mathbb{A}}(\mathcal{D}(u)/v_d) =$

$$= \prod_{i:\, d \in B_i} P_{\mathbb{A}}(a_i(u)/v_d, (B_i \setminus \{d\})(u)) \tag{9}$$

If applied to new cases, the above equivalence can be considered only over the vectors of values occurring in data. In case of combinations not included in $V_A^U = \{A(u) : u \in U\}$, it remains to trust into the generalization ability of the classification model related to formula (8). Thus, it is worth searching for ensembles of μ-decision reducts $B_i \subseteq \{d, a_1, \ldots, a_{i-1}\}$ of possibly law overall cardinality, because they provide more robust probabilities in (9).

The way of classifying objects in Proposition 1 corresponds to the network construction described in Theorem 1, if applied to $\mathbb{A} = (U, A \cup \{d\})$, for d at the first position of the ordering over attributes $A \cup \{d\}$. We obtain a scheme of the network-based bayesian classification, where conditional probabilities are propagated along the DAG structure, beginning with decision as the root. In particular, we obtain a visual interpretation of the fact of exclusion from (9) those μ-decision reducts B_i, which do not contain d – this is because corresponding nodes a_i are d-separated from the root by the others.

[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 $\mathcal{D}$

4 Entropy-based approximations

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

Definition 6 *Let $\mathbb{A} = (U, A \cup \{d\})$ and $B \subseteq A$ be given. We put*

$$\begin{array}{rcl} G_{\mathbb{A}}(B) & = & \sqrt[|U|]{\Pi_{u \in U} \mu_B(u)} \\ G_{\mathbb{A}}(d/B) & = & \sqrt[|U|]{\Pi_{u \in U} \mu_{d/B}(u)} \end{array} \tag{10}$$

$G_{\mathbb{A}}$ corresponds to the measure of information entropy adapted to the rough set, statistical and machine learning methodologies in various forms (cf. [3, 8, 9]).

Definition 7 *Let $\mathbb{A} = (U, A)$ and $X, Y \subseteq A$ be given. By entropy of X we mean*

$$H_{\mathbb{A}}(X) = - \sum_{w_X \in V_X^U} P_{\mathbb{A}}(w_X) \log_2 P_{\mathbb{A}}(w_X) \tag{11}$$

By entropy of X conditioned by Y we mean

$$H_{\mathbb{A}}(X/Y) = H_{\mathbb{A}}(X \cup Y) - H_{\mathbb{A}}(Y) \tag{12}$$

Proposition 2 *Let $\mathbb{A} = (U, A \cup \{d\})$ and $B \subseteq A$ be given. We have equalities*

$$\begin{array}{rcl} H_{\mathbb{A}}(B) & = & -\log_2 G_{\mathbb{A}}(B) \\ H_{\mathbb{A}}(d/B) & = & -\log_2 G_{\mathbb{A}}(d/B) \end{array} \tag{13}$$

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 8 *Let $\varepsilon \in [0, 1)$, $\mathbb{A} = (U, A \cup \{d\})$ and $B \subseteq A$ be given. We say that B (H, ε)-approximately μ-preserves d, iff*

$$H_{\mathbb{A}}(d/B) + \log_2(1 - \varepsilon) \leq H_{\mathbb{A}}(d/A) \tag{14}$$

B is an (H, ε)-approximate μ-decision reduct, iff it satisfies (14) and none of its proper subsets does it.

The above definition can be treated as providing the notion of an *approximate* Markov boundary, i.e. irreducible subset of features $B \subseteq A$ which makes d *almost* independent on the rest of A.

Definition 9 *Let $\varepsilon \in [0, 1)$, $\mathbb{A} = (U, A)$ and $X, Y, Z \subseteq A$ be given. We say that Y makes X conditionally (H, ε)-approximately independent on Z, iff*

$$H_{\mathbb{A}}(X/Y) + \log_2(1 - \varepsilon) \leq H_{\mathbb{A}}(X/Y \cup Z) \tag{15}$$

Such a criterion of *approximate* probabilistic conditional independence is more robust with respect to possible noises or fluctuations in real life data. On the other hand, we have obvious equivalence of the notions of independence and $(H, 0)$-approximate independence. We can continue this approximation methodology also for bayesian networks:

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 (H, ε)-approximately consistent with $\mathbb{A}$, iff*

$$H_{\mathbb{A}}(\mathcal{D}) + \log(1 - \varepsilon) \leq H_{\mathbb{A}}(A) \tag{16}$$

where

$$H_{\mathbb{A}}(\mathcal{D}) = \sum_{a \in A} H_{\mathbb{A}}(a/\{b \in A : \langle b, a \rangle \in \overrightarrow{E}\}) \tag{17}$$

where we put $H_{\mathbb{A}}(a/\emptyset) = H_{\mathbb{A}}(a)$.

Condition (16) 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 $\varepsilon \in [0, 1)$, $\mathbb{A} = (U, A)$ and DAG $\mathcal{D} = (A, \overrightarrow{E})$ be given. We say that $\mathcal{D}$ is an (H, ε)-approximate 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 (H, ε)-approximately independent on Z.*

The following result generalizes Theorem 1 from [7]. In particular, any DAG $\mathcal{D}$ built on the basis of μ-decision reducts is $(H, 0)$-approximately consistent with a given $\mathbb{A}$, as well as any $(H, 0)$-approximate bayesian network is a bayesian network.

Theorem 2 *([10]) Let $\varepsilon \in [0, 1)$ and $\mathbb{A} = (U, A)$ be given. Each DAG which is (H, ε)-approximately consistent with $\mathbb{A}$ is an (H, ε)-approximate bayesian network for $\mathbb{A}$.*

5 Related optimization problems

One can formulate a number of optimization problems concerning extraction of approximate Markov boundaries, as well as approximate bayesian networks from data. Let us start with an observation that Theorems 1 and 2 provide us with possibility of searching for appropriate DAG structures in two phases:

1. Find an ordering $A = \langle a_1, \ldots, a_n \rangle$ of attributes which is potentially best for an optimal bayesian network.
2. Along this ordering, find optimal (approximate) μ-decision reducts for decision tables $\mathbb{A}_i = (U, \{d, a_1, \ldots, a_{i-1}\} \cup \{a_i\})$.

In purpose of searching for exact, not approximate BN's, one usually begins with some heuristics extracting initial (partial) ordering and then apply further tools to extract locally optimal boundaries (cf. [1]). BN can be regarded as optimal in terms of the law of encoding of conditional independence statements in Definition 5 and/or performance of DAG-based classification scheme (8). In both cases the following task is worth considering:

Problem 1 *For any $\mathbb{A} = (U, A)$, find such ordering $A = \langle a_1, \ldots, a_n \rangle$ that μ-decision reducts B_i for $\mathbb{A}_i$ provide minimal $\sum_{i=2}^{n} |B_i|$.*

Even if the appropriate ordering is given, the task of finding minimal μ-decision reducts remains NP-hard (cf. [10]). An advantage is then that one can search for subsets B_i, $i = 2, \ldots, n$, in parallel – given the initial ordering, optimal solutions of reduction problems for decision tables $\mathbb{A}_i$ are completely independent on each other.

Problem 2 *Let $\varepsilon \in [0, 1)$ be given. For any $\mathbb{A} = (U, A)$, find DAG $\mathcal{D} = (A, \overrightarrow{E})$ (H, ε)-approximately consistent with $\mathbb{A}$, which corresponds to minimal $|\overrightarrow{E}|$.*

In this case we have still the following result:

Theorem 3 *[10] The problem of finding minimal (H, ε)-approximate μ-decision reduct is NP-hard, for any constant $\varepsilon \in [0, 1)$.*

Moreover, given $A = \langle a_1, \ldots, a_n \rangle$, appropriate subsets B_i can be here derived as (H, ε_i)-approximate μ-decision reducts, where the only requirement for local approximation thresholds $\varepsilon_i \in [0, 1)$ is the following:

$$(1 - \varepsilon_2) \cdot \ldots \cdot (1 - \varepsilon_n) \geq 1 - \varepsilon \tag{18}$$

Thus, solutions of local reduction problems are not independent, because we have to search, in some sense, for both optimal ordering and distribution of approximation thresholds, possibly during the second phase of the network construction – the edge reduction.

6 Genetic algorithms

One cannot expect fast and reliable tools for solving problems defined in the previous section in a deterministic way. Genetic algorithms (GA's) provide flexible and universal methodology, able to be combined with approximate but fast heuristics which are known for considered tasks [2, 4]. Heuristics are designed and tuned up with respect to each specified problem, and often are more efficient than a simple GA. On the other hand, they usually cannot avoid local optima. The advantages of both genetic and heuristic approaches can be exploited by hybrid algorithms – nondeterministic, problem-oriented heuristics controlled by the GA-based framework.

Genetic algorithm performs over a finite state space S. Assume that a function: $f : S \rightarrow R_+$ is given. Our goal is to find x_o: $f(x_o) = max\{f(x)\colon x \in S\}$. Elements of S are *individuals*. We treat a value of the function f as ability to survive in the environment (*fitness*), and we simulate the process of evolution by creating foregoing populations obtained by application of *genetic operators* to relatively strong individuals (see [2, 4, 11] for details). The result of evolution is the best individual x_{max} which is assumed to be nearly as good as the global optimum x_0.

In the *order-based genetic algorithm* (o-GA, [2]) a chromosome is a permutation. There are various methods of recombination (crossing-over) considered in literature. For instance, MOX (*Modified Order Crossover* [11]) operator affects two parent chromosomes and replaces them by two children. First, we choose one gene in the first parent's chromosome at random. This gene will be the end of a matching section, starting at the beginning of chromosome. Identical matching section is marked on the second parent's chromosome. Then, we leave the matching sections unchanged, but the rest of genes of the first chromosome is set up in the order of appearance in the second chromosome.

In case of a *hybrid algorithm*, its heuristic part is launched in order to calculate fitness value of a chromosome. The heuristic procedure is used to obtain actual results, while genetic part of algorithm optimizes the parameters of heuristic part. Fitness value of a control sequence depends on the result of heuristic algorithm. As an example, let us present the hybrid algorithm for searching for approximate (H, ε)-approximate μ-decision reducts, calculated for specified settings of parameter $\varepsilon \in [0, 1)$. It is a generalization of the algorithm developed in [11] for searching for classical rough-set-based decision reducts: (1) Let $B = A$ be the set of all attributes and let $A = \langle a_1, \dots, a_n\rangle$ be the ordering consistent with input permutation. Let $\varepsilon \in [0, 1)$ be given; (2) For $i = 1$ to $n = |A|$ repeat: Let $B \leftarrow B \setminus \{a_i\}$; If B does not μ-preserve d (H, ε)-approximately, then $B \leftarrow B \cup \{a_i\}$.

Proposition 3 *The result of the above algorithm is an (H, ε)-approximate μ-decision reduct. Moreover, for any (H, ε)-approximate μ-decision reduct B there exists a permutation such that B is the result.*

The above properties of the considered algorithm assure that it corresponds to a mapping of the set of possible orderings (permutations) onto the set of reducts. Given a criterion of reduct optimization, there exists a permutation which leads to an optimal reduct. We use the order-based genetic algorithm to generate the proper order. To calculate the function of fitness for a given permutation we have to perform one run of the algorithm and set up the fitness as opposite to the length of the reduct found.

7 Optimal approximate networks

As mentioned before, the problem of searching for optimal a-BN structures can be divided into two subproblems: How to choose initial, superfluous structure of DAG and how to reduce it to obtain minimal net, keeping approximate consistency with data. We suggest to design the order-based genetic algorithm operating at the level of the first above subproblem and to relate calculation of fitness for particular chromosomes to the second one.

Let us assume that ordering $A = \langle a_1, \dots, a_n\rangle$ corresponds to a specified permutation. Let us initiate the structure of $\mathcal{D} = (A, \overrightarrow{E})$ by

$$\overrightarrow{E} = \bigcup_{i=1}^{n-1} \bigcup_{j=i+1}^{n} \{\langle a_i, a_j\rangle\} \tag{19}$$

Calculation of the fitness function should correspond to a strategy of choosing the order of the edge reduction trials. Let $m = \frac{n(n-1)}{2}$, $n = |A|$, for a given $\mathbb{A} = (U, A)$. Let permutation τ over A be given. A general scheme of the reduction strategy is the following:

1. Let $\mathcal{D}$ be an initial complete DAG and let $\overrightarrow{E} = \langle e_1 \dots e_m\rangle$ be any ordering of edges of $\mathcal{D}$, consistent with τ.

2. For $i = 1$ to m repeat: Let $\overrightarrow{E} \leftarrow \overrightarrow{E} \setminus \{e_i\}$; If $\mathcal{D}$ is not (H, ε)-approximately consistent with $\mathbb{A}$, then $\overrightarrow{E} \leftarrow \overrightarrow{E} \cup \{e_i\}$.

Such an approach requires, obviously, an additional procedure for choosing orderings $\overrightarrow{E} = \langle e_1 \dots e_m\rangle$ for particular permutations encoded by chromosomes. One can, e.g., apply *greedy* heuristics checking, at each step of reduction, deletion of which remaining edge causes the smallest increase of entropy $H_{\mathbb{A}}(\mathcal{D})$. In general, one has then no guarantee that for a given information system $\mathbb{A} = (U, A)$ and approximation threshold $\varepsilon \in [0, 1)$ there exists permutation over A providing at outcome of the considered algorithm the solution of Problem 2 defined in Section 5. Still, the following holds:

Proposition 4 *The result of the above algorithm is always a DAG $\mathcal{D} = (A, \overrightarrow{E})$, which is (H, ε)-approximately consistent with $\mathbb{A}$. Moreover, it is irreducible, i.e. deletion of any edge (or subset of edges) from $\overrightarrow{E}$ makes criterion of (H, ε)-approximate consistency not satisfied any more.*

It assures that by application of the proposed method to calculation of fitness function we map chromosomes into local minima of function $f(\mathcal{D}) = |\overrightarrow{E}|$, defined over the space of all DAG's $\mathcal{D} = (A, \overrightarrow{E})$ being (H, ε)-approximately consistent with the given information system $\mathbb{A} = (U, A)$.

Obviously, this is a very initial step of analysis of performance of such genetic algorithms. Perhaps the most important challenge is to analyze behavior of genetic operators, like e.g. MOX presented in Section 6, over permutations. Soundness of their performance must be in this case expressed in terms of partial orderings over the sets of vertices. It may depend on the way of setting the strategy of choosing edges to be reduced as well. Further study in this area is required.

8 Conclusions

Introduced notion of an entropy-based approximate bayesian network reflects the need of dealing with approximate independence statements in case of the real life data analysis. Presented results provide the framework for the efficient extraction and application of approximate BN-models to the data classification and description tasks. Further development of methodology for applying ordered-based genetic algorithms to searching for optimal (H, ε)-approximate bayesian networks for information systems and decision tables is required. Such a methodology leads to improvement of representation and classification capabilities of network structures *approximately* consistent with observed data.

Acknowledgements

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

References

[1] Buntine, W.: A guide to the literature on learning probabilistic networks from data. IEEE Transactions on Knowledge and Data Engineering (1996).

[2] Davis, L. (ed.): Handbook of Genetic Algorithms. Van Nostrand Reinhold (1991).

[3] Duentsch, I., Gediga, G.: Uncertainty measures of rough set prediction. Artificial Intelligence 106 (1998) pp. 77–107.

[4] Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag (1994).

[5] Pawlak, Z.: Rough sets – Theoretical aspects of reasoning about data. Kluwer Academic Publishers (1991).

[6] Pawlak, Z., Skowron, A.: Rough membership functions. In: R.R. Yaeger, M. Fedrizzi, J. Kacprzyk (eds), Advances in the Dempster Shafer Theory of Evidence. John Wiley & Sons, Inc. (1994) pp. 251–271.

[7] Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1988).

[8] Rissanen, J.: Minimum-description-length principle. In: S. Kotz, N.L. Johnson (eds), Encyclopedia of Statistical Sciences. John Wiley & Sons (1985) pp. 523–527.

[9] Ślęzak, D.: Approximate reducts in decision tables. In: Proc. of IPMU'96. Granada, Spain (1996) 3, pp. 1159–1164.

[10] 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.

[11] Wróblewski, J.: Theoretical Foundations of Order-Based Genetic Algorithms. Fundamenta Informaticae 28/3-4, IOS Press (1996) pp. 423–430.