Noname manuscript No.
(will be inserted by the editor)

A decision support tool for credit domains: Bayesian Network with a variable selector based on imprecise probabilities

Javier G. Castellano†, Serafín Moral-García†, Carlos J. Mantas†, María D. Benítez§, and Joaquín Abellán †

Received: date / Accepted: date

Abstract A Bayesian Network (BN) is a graphical structure, with associated conditional probability tables. This structure allows us to obtain different knowledge than the one obtained from standard classifiers. With a BN, representing a dataset, we can calculate different probabilities about set of features with respect to other ones. This inference can be more powerful than the one obtained from classifiers. A BN can be built from data and have analytical and diagnostic capabilities that make it very suitable for credit domains. Credit scoring and risk-analysis are fundamental tasks for financial institutions with the aim to avoid important losses. In these tasks and other domains, an excessive number of features can convert a BN in a complex and difficult to interpret model, but a few number of features can represent a loss of information obtained from data. A new method based on imprecise probabilities is presented to select an informative subset of features. Using this new feature selection method we can build a BN that has an excellent adjustment to the data, considering a reduced number of features. Via a set of experiments, it is shown that the adjustment is better than the ones obtained with: no previous variable selection method; and with a similar and successful variable subset selection method based on precise probabilities. Finally, a BN is built with two important characteristics: (i) it represents a better adjustment to the data; and (ii) it has a low complexity (better interpretability) due to the small number

† Javier G. Castellano, Serafín Moral-García, Carlos J. Mantas, and Joaquín Abellán
Department of Computer Science and Artificial Intelligence
University of Granada, Granada, Spain
Tel.: +34-958-240467
Fax: +34-958-243317
E-mail: {fjgc,seramoral,cmantas,jabellan}@decsai.ugr.es

§ María D. Benítez
BANKIA
Central Office, Granada, Spain
E-mail: mbenitez@bankia.com

of important selected features. A practical example about inference on a BN to help on credit risk analysis is also presented.

Keywords Credit domains · Bayesian networks · inference · feature selection · imprecise probabilities

1 Introduction

Nowadays, the analysis of risks is a general important issue for banks and financial institutions, not only because they must measure credit risk[1], but because any small improvement would produce great profits [1]. A lot of effort has been done to build models which allow to predict if a specific applicant for credit is good or not. In [2],[3], [4], [5], [6] or [7], among other works, different approaches have been applied to credit scoring and it has been studied their behaviour as predictors of the goodness of a specific credit.

Bayesian networks (BNs) are very interpretable models, which allows for the *right to an explanation*[2], due to their graphical structures that represent dependence relations among the features of the problem. They have been successfully used to work with credit scoring datasets [8], [9], [10] or [11]. In fact, a BN can be used as a classifier but it is not one of the principal virtues of this model. BNs have different characteristics than the standard classifiers:

- BNs are interpretable probabilistic models, whereas some classifiers perform as black-boxes, such as e.g. some of the most popular and successful models: Support Vector Machines (SVM)[12–14], Random Forest [15] or Artificial Neural Networks (ANN) ([16,17]).
- If the classifier is also interpretable as BNs (for example, a decision tree) we need to know the values of all the features associated with the case to predict (all the values of the antecedent to know the consequent in a rule generated by a decision tree). With a BN we can do inference regardless of the number of observations about the features that we have. Furthermore, BNs are capable of informing about the probability of each value from any feature. These probabilities change when we know the values of any other features. For example, knowing part of the credit applicant data, we will be able to calculate the probability whether the credit is positive or not through inference methods.
- With a BN we can do inference from causes to effects and from effects to causes, whereas with a classifier we can only predict the class variable (causes to effects). Knowing the values of some features, with a BN we can

[1] Since the Basel second accord from 2004, known as Basel II and released by the Basel Committee on Banking Supervision, the supervised financial institutions are required to use internal ratings to measure credit risk

[2] In the United States, credit score has a right to explanation under the Equal Credit Opportunity Act (Regulation B of the Code of Federal Regulations), Title 12, Chapter X, Part 1002. Likewise, in the European Union, the European Union General Data Protection Regulation extends the automated decision-making rights in the 1995 Data Protection Directive to provide the right to an explanation.

find the most probable combination of the rest of the features. For example, let suppose that a credit is negative and the client is under twenty years of age, then, we can find the most probable combination of values for the rest of the client characteristics (features).

As with another data mining tools, the reduction of the number of features can improve the performing and reduce the complexity of a NB. For that aim it is important that the procedure used to select variables could find the most informative features. A higher number of features does not necessarily imply that the learned BN be a better representation of the data available. If we have irrelevant features, the BN could use them and build a model with erroneous relations. Redundant variables will usually deteriorate the goodness of a fitted model. Furthermore, the models including a great number of features become less interpretable because the network is bigger and more complex. For these reasons, it is appropriate to take advantage of a good feature selection algorithm that would remove any irrelevant/redundant variables before learning the network.

On the other hand, a considerable number of mathematical models based on imprecise probabilities have been developed with the goal of representing the information [18], [19], [20]. The use of imprecise probabilities has several advantages. The most important of them might be the suitable management of the little reliable information, when the sample size is not enough or there are noisy data. In particular, it has been developed a imprecise information measure in order to build classification trees [21], which is called *Imprecise Info Gain* (IIG). It has been shown that this measure works specially well on noisy data; on credit scoring data; and for extracting a high number of quality rules from traffic datasets ([3], [4], [22]). For this reasons, we think that the IIG measure could be interesting to be used in a feature subset selection method and to apply it before using BNs on credit data.

In this paper we define a new feature selection method to select a subset of informative features. This method will be based on the IIG measure in a forward way to add features. The new feature subset selection algorithm will be called *Forward Feature Selection based on Imprecise Information Gain* (FFSIIG). The principal aim of the paper is to show that if we build a BN from data using the FFSIIG in a previous step, we obtain a better representation of the data than the BN built with no previous subset feature selection. Moreover, we will also show that the BN build with the features selected by the FFSIIG is also more representative of the data than similar BN built with one of the most used and successful procedures to select variables called *Correlation-Based Feature Selection* (CFS) [23].

Currently, collecting of huge amounts of information is common. The problem arises in how to deal with a very large number of features or variables. The reduction of such number is an important task in any information system. If with a smaller number of variables we are able to correctly represent the data, the information gain can be clearer and better. The use of BNs can allow different ways to extract knowledge from data, but even if the set of variables

is reduced, the results can be easier to understand. The problem here is to find a middle point between number of features and fit to the data. That is the aim of this paper, to build BNs with a reduced number of features but with a good fit to the data, to make inferences as correct as possible. When we use the FFSIIG feature selection method, we are considering a few number of important variables but with an excellent fit to the data

To prove the procedures, we carry out an experimentation where we will consider five very known and used credit scoring datasets. For each one of these datasets, we will learn BNs using different known learning methods of BNs from data after we have selected the features through FFSIIG. Then, we measure how representative are the networks built with respect to the original data through the Kullback-Leibler divergence [24]. It is the more appropriate measure to quantify the distance from a model to data [25–27]. We will compare these divergences with the divergences obtained by applying the same learning algorithms with (i) no previous variable selection method; (ii) using the CFS approach. This experimentation will show that the models learned applying previously our feature selection method represent the data in a better way than the other procedures.

Finally, in this paper, we will present a practical case to illustrate the advantages of BNs in credit risk analysis. In concrete, we will consider an specific learned BN after applying the FFSIIG method and we will show information that we can extract from this model that we can not extract with other systems.

The rest of this paper is structured as follows: In Section 2 we describe the previous knowledge about BNs and CFS algorithm. Section 3 deals with the advantages of BNs versus other systems. In Section 4 we present our new feature selection method. Section 5 presents the experimentation carried out, the results obtained and the comments about the results. Section 6 consists of a detailed example about a specific BN learned in this work. Finally, Section 7 is devoted to the conclusions about this work.

2 Previous knowledge

In this section we shall introduce the needed concepts about Bayesian Networks and the *Correlation-Based Feature Selection* algorithm.

2.1 Bayesian Networks

The probabilistic nature of Bayesian networks makes them adequate for representing data uncertainty and for efficiently handle uncertain knowledge. A BN [28] is a graphical model which encodes a joint probability distribution, being composed of a qualitative part, a directed acyclic graph which represents the dependencies among the variables, and a quantitative part, a collection of numerical parameters, commonly conditional probability tables.

Formally, let us consider a finite set of discrete random variables $X_1, \ldots, X_m$, each variable taking on values from a finite set, a BN is a pair (G, θ), where:

- G is a directed acyclic graph with a node for each variable of the problem X_i $(i = 1, \ldots, m)$. In this graph, a link represents direct dependence relationships between the variables. For example, $X_j \rightarrow X_i$ indicates that the node X_i directly depends on the node X_j.
- θ is a set of conditional probability distributions. This set contains, for each node, a conditional probability distribution of the variables on which it depends directly, i.e, its parents. If a node has no parents, its distribution is simply the probability distribution of the node.

In this way, let $Pa(X_i)$ denote the set of parents of the variable X_i, where $Pa(X_i) = \{X_j \mid X_j \rightarrow X_i \in G\}$. Therefore, for each variable X_i we have a set of conditional distributions $P(X_i \mid Pa(X_i))$. From these conditional distributions we can recover the joint probability distribution [28]:

$$P(X_1, \ldots, X_m) = \prod_{i=1}^{m} P(X_i \mid Pa(X_i)) \tag{1}$$

The independence relationships which make this decomposition possible are graphically represented, using the *d-separation criterion* (see [28]), through the existence or not of arcs between pairs of variables in G. The d-separation criterion allows us to decide whether a set of variables is independent of another set, given a third set. For example, each variable X_i is independent of its nondescendants known $Pa(X_i)$.

2.2 Learning of Bayesian Networks

A BN can be built manually from an expert but the common practice is to obtain it automatically from a data set. There are also mixed methods to build a Bayesian Network where the network can be learned automatically from data and manually refined by an expert. Therefore, the problem of learning automatically a BN from data is to find the network that, in some sense, best represents the data.

Since a BN is composed of a qualitative part and a quantitative part, we distinguish two types of machine learning:

- *Structural learning*: the learning of the graphical structure (a directed acyclic graph), that is, the qualitative part.
- *Parametric learning*: the learning of the collection of numerical parameters (a conditional probability distribution for each variable), that is, the quantitative part.

Because parametric learning consists of estimating the conditional probabilities given by the structure of the graph using the observed frequencies on the data, we first must learn the topology of the network. The conditional

probabilities can be computed by using a maximum likelihood estimation [29], though it is normally done by using a Bayesian estimator based on the Dirichlet distribution [30].

There are a lot of works on the automatic learning of the BN structures from data and, consequently, many structural learning algorithms have been developed that may be categorized into two general approaches: algorithms based on conditional independence tests, and algorithms based on a metric and a search procedure.

Algorithms based on conditional independence tests (also called constraint-based algorithms)[31] [32] [33] perform a qualitative study of the dependence and independence relationships between the variables. The aim of these methods is to find the network that best match these relationships by using conditional independence tests. The most telling example of this kind of structural learning is the PC algorithm [31] which, starting with a complete graph, first eliminates as many edges as possible, and after it gives direction to the edges. The elimination of edges is guided by the results of some statistical tests of conditional independence applied to the data.

The second type of structural learning algorithms attempt to find a graph that best represents the data by maximizing the selected metric and minimizing the number of arcs. The metric or scoring function is a measure of fit between the graph and the data. There are several proposals based on Bayesian scoring functions, such as BD/BDe metric [30], BDeu metric [34] or K2 [35] and other approaches based on information theory scoring functions, such as entropy [36] or the Minimum Description Length [37]. Furthermore, a search procedure is needed to find the best structures according to the selected metric. Local search methods are commonly used [35] [30] due to the exponentially large size of the search space.

2.3 Inference with Bayesian Networks

Once we have obtained a BN we usually need to determine various probabilities of interest as we get new information or evidence. For example, in a credit scoring problem we want to know the probability of grant a credit given the data of a new client. Thus, we can define the probability propagation or probabilistic inference [28] [38] as the computation needed to obtain the posterior probability of one or several variables (e.g., grant a credit or not) given the values of other variables (e.g., new client data) in the BN.

Cooper [39] proved that exact computation of probabilistic inference for BN is NP-hard, even for a single variable. Therefore, any kind of probabilistic inference has exponential complexity, even approximate inference is NP-hard. For this reason, there are several approaches to probabilistic inference in BNs that can be categorized in exact inference methods [40] [41] [42], [43] and approximate inference methods [44] [45] [46] [47] .

2.4 Correlation-Based Feature Selection

The aim of a feature selection algorithm is to select a subset of variables which can effectively replace the original set of attributes while reducing the unfavorable effects of irrelevant/redundant variables and provides good results or, even better, improves the performance of the selected pattern recognition technique.

The *Correlation-Based Feature selection* (CFS) algorithm [23] is one of the most used method for feature selection [48]. This method was introduced to find subsets of variables hat are highly correlated with the class and uncorrelated with each other. Hence, this algorithm ignores irrelevant features because of their low correlation with the class and discards redundant features because of their high correlation with the remaining variables. This state of the art algorithm has proved to obtain very good results in several domains such as bioinformatics [49], traffic accident analysis [50], network intrusion detection [51] or credit scoring [52].

Other very recent works where the CFS algorithm is used as benchmark are the following ones: [53], [54], [27] and [55].

The CFS algorithm uses a local search method which starts with an empty set of variables and, in each step, selects the feature that, by adding to the subset that we have in that moment, provides the maximum heuristic value for the new set of variables. If there is not a variable that improves the heuristic of the actual subset, then the algorithm stops and returns the actual subset as the best feature subset found. The Figure 1 shows the pseudo-code for this search algorithm.

1. $\chi = \emptyset$
2. $V = \{X_1, \ldots, X_m\}$
3. $I = 0$
4. Stop = False
2. While !*Stop*
   2.1. Select the variable $v \in V \setminus \chi$ that maximizes the heuristic of $\chi \cup v$
   2.2. I_2 = heuristic value of $\chi \cup v$
   2.3. If $I_2 > I$ then $I = I_2$, $\chi = \chi \cup \{v\}$ else Stop = True
3. Return χ

Fig. 1 Search algorithm used by the CFS method.

The evaluation function of this method is based on the next principle [23]: *"Good feature subsets contain features highly correlated with the class, yet uncorrelated with each other"*. Hence, the heuristic evaluates fundamentally two things about a feature subset: The average correlation among no-class features, giving more heuristic value as long as this correlation is lower, and the average correlation among the class and the rest of features, being the heuristic value directly proportional to this average correlation. The heuristic is formalized in the next equation, [56]:

$$Merit_S = \frac{m'\overline{r_{cf}}}{\sqrt{m' + m'(m'-1)\overline{r_{ff}}}} \quad (2)$$

where $Merit_S$ is the heuristic value given to this feature subset, m' the number of variables in the feature, $\overline{r_{cf}}$ is the average correlation among no-class features and the class and $\overline{r_{ff}}$ is the average feature-feature correlation.

The numerator of the equation 2 indicates how relevant are the features of the subset and the denominator indicates the grade of redundancy of the features in the subset.

In [23] the case of that the class variable is numeric is also considered. However, in this work the class variable will be always discrete. In this case CFS discretizes numeric features using the technique given in [57] and, for calculating the correlation among two features, X and Y, CFS algorithm uses the next measure, known as Symmetrical Uncertainty (SU) [23]:

$$SU(X,Y) = 2 \times \frac{H(X) + H(Y) - H(X,Y)}{H(X) + H(Y)} \quad (3)$$

being H the Shannon entropy defined in (4).

3 Advantages of Bayesian Networks versus other systems

A classic model for valuation of a loan to an individual client or a company, is based on the score on various elements (variables or features) of the data obtained on the client (person or company) and the loan type. If the variables are qualitative, each state is assigned a score; and if they are quantitative, each range of values is assigned a score too. The final valuation of the possible transaction is done through the final score obtained, classifying the risk of the loan according to the final intervals of that valuation.

Depending on the final score, for example, the possible customer loan would be valued in the set C = { *very good, good, medium, weak, bad* }. In many cases, the possible valuation corresponds to a binary variable, with C ={*good, bad*}.

For example, we suppose that the features are $\{X_1, X_2, \cdots, X_{10}\}$ and each feature has a different weight because its importance but the final possible score, obtained adding the ones of each features, is in the interval $[0, 100]$. An example of valuation table is presented in Table 1:

The final position about the value obtained can be expressed using a set of intervals for the *Final Score*:

$[0, 25] \longrightarrow Bad$
$[26, 40] \longrightarrow Weak$
$[41, 65] \longrightarrow Medium$
$[66, 85] \longrightarrow Good$
$[86, 100] \longrightarrow Very\ Good$

Table 1 Example of a scoring table based on 10 features and a Final Score in $[0, 100]$

Feature	Type	Values : score	Current	Score
X_1	Qual.	$a_1 : 10$ $a_2 : 8$ $a_3 : 4$	a_1	10
X_2	Quant.	$X_2 > 30 : 12$ $30 \geq X_2 > 10 : 5$ $10 \geq X_2 > 4 : 2$ $X_2 \leq 4 : 0$	25	5
...				
X_{10}	Quant.	$X_{10} > 300 : 7$ $300 \geq X_{10} > 180 : 3$ $X_{10} \leq 180 : 0$	225	3
				Final Score = 76

In the same way, if we have a decision tree type classifier, using the characteristics (features) of the client, the possible loan is valued following the branches of the tree according to these characteristics, until arriving at a leaf node where we will have the valuation in the set C (as we will see in Figure 7).

In the above situations, a major problem arises when the value of one or several variables on the characteristics of the loan and client are either unknown or mistrusted. Suppose that for a company, we do not know or mistrust (motivated by the results of an audit) the data provided on its profit balance of the last quarter of the year, which is a very important variable for the final analysis (high value in the scoring table). If the value is not known, we cannot obtain a final scoring that gives us the valuation of the transaction. If we distrust this value, the final valuation will have some uncertainty that causes distrust even if the final scoring is good. Likewise, if this variable is very important, it will surely be in the first levels of decision tree and will not allow the final classification or it will be given with distrust.

A BN does not suffer in situations like the one raised above. We can dispense with this variable and express with a probability the final valuation taking into account the rest of the variables that are presented in the model, even if we do not know any other. The idea is that with the available information, in this case with the information of the variables in which we have more confidence or knowledge, we can establish a final valuation that helps us in the decisions. Surely, if the variable that gives us problems is important, the valuations will have a lower level of probability. But it has sense because, in this case, we are working only with the information on the variables of greater confidence.

In general, we can compare BNs with standard systems for valuation of a loan for a customer; or with a classifier when it is used for similar aim. In the section 6 we will illustrate with a example some of these advantages in

the field of credit scoring. The principal advantages can be resuming of the following way:

1. BNs are quite interpretable models, unlike other accuracy procedures of classification that performs as black-boxes (SVM, ANNs, Random Forest, etc). But it is true that with a huge amount of variables in the graphic representation, a BN become less interpretable. Hence the importance of a previous variable selection procedure that does not harm the information from data.
2. This probabilistic graphic model represents in an easy way the dependence relations between the features. However, the majority of the other systems do not let us know the dependence relations so easily.
3. BNs let do inference although we do not know the value of some variables. One of the most interpretable predictors known thus far are decision trees due to the rules which we are able to extract after building the model. Nevertheless, in a decision tree we have to know the values of all antecedent variables to know the consequent value, whereas with a BN we can make inference even when we do not know the real value of some features. With a classic system of a table for credit scoring, the lack of knowledge of some values (even one) does not allow us to obtain a correct score to use in a decision making.
4. With a BN we can extend our knowledge of the system via a risk analysis in two directions: from causes to effects and from effects to causes. We can know the probability of some events or to know the most probable combination of values of some variables knowing other variables, including the information a posteriori with respect a loan. This means that we can do inference from effects to causes and from causes to effects. However, in predictive models it is only possible to predict the class value or determinate the probability that the class variable has a certain value. Moreover, BNs let us make inference not only about the class variable, but any variable of the network.

4 Forward Feature Selection based on Imprecise Info Gain (FFSIIG)

Previously to introduce the new method to select subset of informative variables, we will see the principal differences between the criterion used in the new method, based on imprecise probabilities and general uncertainty measures, with respect to similar one based on classical probabilities.

4.1 Classic Info-Gain criterion vs. Imprecise Info Gain

We assume that we have the class variable C, whose possible values are $\{c_1, \cdots, c_K\}$; and let $\mathcal{D}$ a dataset about the observations of the features $X_1, \ldots, X_m$. The Shannon Entropy [58] about the class variable C is defined as

$$H^{\mathcal{D}}(C) = \sum_{i=1}^{K} p(c_i) \log_2(1/p(c_i)) \tag{4}$$

where $p(c_i)$ is the estimation of the probability of c_i based on the data $\mathcal{D}$ (by computing relative frequencies).

Let now X_j, $(1 \leq j \leq m)$ be a specific feature and suppose that its possible values are $\{x_1, \ldots, x_t\}$. The entropy of C given X_j is given by the following expression:

$$H^{\mathcal{D}}(C|X_j) = \sum_{i=1}^{t} p(x_i) H^{\mathcal{D}_i}(C|X_j = x_i) \tag{5}$$

where $\mathcal{D}_i$ is the partition associated with the value x_i, i.e, is the subset of $\mathcal{D}$ in which $X_j = x_i$, and $p(x_i)$ is the estimation of the probability that $X_j = x_i$ in $\mathcal{D}$, $\forall i = 1, \ldots, t$.

Once we have defined the measures given by 4 and 5, then the classic Info-Gain measure [59] can be defined as follows:

$$IG(C, X_j)^{\mathcal{D}} = H^{\mathcal{D}}(C) - H^{\mathcal{D}}(C|X_j) \tag{6}$$

The Imprecise Info-Gain (IIG) [21] is based on the Imprecise Dirichlet Model [18]. According to this model, for each possible value of class $C = c_i$, an imprecise probability interval is obtained instead of a precise estimation:

$$\left[\frac{n_{c_i}}{N+s}, \frac{n_{c_i}+s}{N+s}\right] \tag{7}$$

being n_{c_i} the number of cases in which $C = c_i$ in the data set, $\forall i = 1, \ldots; K$, s a given parameter of the model; and N the number of instances of the data set. It is clear that the higher value of s is, the bigger the interval is. It is not trivial to decide which is the most appropriate value of s. In [18] the value $s = 1$ is recommended.

This set of probability intervals gives rise to a credal set of probabilities on the variable C, which is defined in [60] as follows:

$$K^{\mathcal{D}}(C) = \{p | p(c_i) \in \left[\frac{n_{c_i}}{N+s}, \frac{n_{c_i}+s}{N+s}\right], \forall i = 1, \ldots, K\} \tag{8}$$

On this credal set, we can apply the maximum of entropy function H^*:

$$H^*(K^{\mathcal{D}}(C)) = \max\{H^{\mathcal{D}}(p) | p \in K^{\mathcal{D}}(C)\} \tag{9}$$

For $s \leq 1$, the procedure to obtain maximum of entropy attains its lower computational cost [61]. This is other important reason to set the value of $s = 1$.

In a similar way we can define for each feature, X_j $(j = 1, \ldots, m)$, the average of the maximum entropy on C generated by X_j:

$$H^*(K^{\mathcal{D}}(C|X_j)) = \sum_i p(X_j = x_i) H^*(K^{\mathcal{D}}(C|X_j = x_i),$$

which are obtained using the values n_{c_r,x_i}, with $r \in \{1, \cdots, K\}$ and $i \in \{1, \cdots, t\}$

With these two measures we can express the Imprecise Info Gain (IIG) [21] of the following way:

$$IIG(C, X_j)^{\mathcal{D}} = H^*(K^{\mathcal{D}}(C)) - H^*(K^{\mathcal{D}}(C|X_j)) \tag{10}$$

To calculate the probability in $K^{\mathcal{D}}(C)$ that gives the maximum entropy, a simple algorithm can be applied, which can be found in [61].

Unlike Info-Gain, the value of IIG can be negative. This is an important characteristic that makes different both criteria. The application of the maximum entropy measure in fundamental for this result.

4.2 The new method

The IIG criterion explained in the above section is the basis of the new procedure to select variables that we present here. The procedure consists in a search method with a heuristic based on the IIG. The search method applied in this algorithm will be the same as in CFS, then we need to explain our metric evaluation for feature subsets selection.

Let suppose we have a dataset $\mathcal{D}$. We can define an Imprecise Info Gain for a set of features $\chi = \{X'_1, \ldots, X'_{m'}\}$, $m \geq m'$, as follows:

$$IIG(C, \chi)^{\mathcal{D}} = H^*(K^{\mathcal{D}}(C)) - H^*(K^{\mathcal{D}}(C|\chi)) \tag{11}$$

where H^* is the maximum of entropy defined in (9) and

$$H^*(K^{\mathcal{D}}(C|\chi)) = \sum_{j=1}^{w} P^{\mathcal{D}}(\chi = \chi_j) H^{\mathcal{D}_j}(C|\chi = \chi_j) \tag{12}$$

being χ_j an array of values of the m'-dimensional variable $(X'_1, \ldots, X'_{m'})$; $\mathcal{D}_j$ the partition generated by χ_j and w the number of possible combinations of values in χ, i.e. $w = |X'_1||X'_2| \ldots |X'_{m'}|$, where with $|.|$ is expressed the cardinal or number of possible states of a variable.

What we measure with this heuristic is the gain in information via the IIG criterion, that we have with respect to the class variable using a specific set of features. The Algorithm of the FFSIIG is expressed in the Figure 2, which is similar to the one expressed in Figure 1, but considering that now the gain in information, via the IIG, can be negative for a set of variables. Here the maximum gain in information via the IIG criterion is stored as G_{IIG}.

1. $\chi = \emptyset$
2. $V = \{X_1, \ldots, X_m\}$
3. $G_{IIG} = 0$
4. Stop = False
2. While !*Stop*
   2.1. Let v' the variable $v \in V \setminus \chi$ that maximizes the value $IIG(C|\chi \cup v)$
   2.2. Let $G'_{IIG} = IIG(C|\chi \cup v')$
   2.3. If $G'_{IIG} < 0$ or $G'_{IIG} \leq G_{IIG}$ Stop=True
   2.4. Else $G_{IIG} = G'_{IIG}$ and $\chi = \chi \cup \{v'\}$
3. Return χ

Fig. 2 Algorithm used by the FFSIIG procedure.

5 Experimentation

For our experimentation, we have used *Weka* software [62] on Java 1.5. We have added the necessary implementation of the new method for select variables FFSIIG presented here. We have also used the Elvira System [63] to work with BNs.

A brief description of the datasets used in the experimentation can be found in Table 2, where "N" is the number of instances in the data sets, column "N_Good" is the number of instances labeled as good/positive, column "N_Bad" is the number of cases classified as bad/negative and column "N_Features" is the number of features or attribute variables. All the datasets have two states for the class variable.

Table 2 Dataset description.

Dataset	N	N_Good	N_Bad	N_features
Australian	690	307	383	14
German	1000	700	300	20
Japanese	690	307	383	15
Polish	240	128	112	30
UCSD	2435	1836	599	38

The Australian, German and Japanese datasets were obtained from the *UCI repository of machine learning* [64] and they are related with credit scoring. The Polish data set [65] is about companies bankruptcy forecast. The UCSD data set is a reduced version of a very large database used in the 2007 Data Mining Contest of the University of California, San Diego and is related with residence refinance predictions.

To work with BN, the continuous variables have been discretized using the procedure of [57], which is implemented in the Weka software [62]. Using this approach, in some cases, a few continuous variables were discretized into a single state, which is equivalent to irrelevant features and those variables had to be removed.

For each one of these discretized datasets we have applied two feature selectors: CFS and FFSIIG. The CFS algorithm was already implemented in Weka. However, for the FSSIIG algorithm, we have implemented the function evaluation using Weka structures for evaluators in the package *attributeSelection* (we remember that the search method is the same for both algorithms). The IDM parameter was set to $s = 1$ because the reason remarked in previous sections.

After the preprocessing stage, we have used the Elvira System [63] to build BNs via different methods. Three different approaches were taken into account to learn the structure of the BNs: (1) the score-based K2 algorithm [35]; (2) a local search approach with the BDeu metric [34]; (3) the PC algorithm [31], chosen because it uses conditional independence tests instead of a metric and a search procedure. The performance measure used is the Kullback-Leibler divergence [24] as the distance between the joint probability distributions associated with a candidate network and with the available data set. This measure is accepted as a standard measure of error in the Bayesian networks literature [30] [37] [28].

The Table 3 shows, for each dataset, the number of features of the original database, as well as the number of features that selects the CFS method and the number of features selected by the FFSIIG algorithm. In this table we do not take into consideration the class variable.

Table 3 Number of features selected by each method.

Dataset	Original_features	CFS_Features	FFSIIG_features
Australian	14	7	4
German	20	5	5
Japanese	15	7	4
Polish	30	9	4
UCSD	38	7	4

The Tables 4, 5 and 6 show, for each learning approach and for each dataset, the Kullback-Leibler divergences of the learned BNs from the original datasets and the BNs built from the datasets obtained after applying CFS and FFSIIG. The differences among the methods can be seen in a more clear way in Figures 3, 4 and 5.

s

Table 4 Kullback-Leibler divergence to the original data with the BNs learned with the K2 algorithm.

	Australian	German	Japanese	Polish	UCSD
original	2.3255	1.1034	3.0129	9.4337	11.1422
CFS	0.8247	0.1826	0.8907	3.3809	1.1054
FSSIIG	0.5994	0.2308	0.5994	1.1233	0.5376

Table 5 Kullback-Leibler divergence to the original data with the BNs learned with local search+BDeu.

	Australian	German	Japanese	Polish	UCSD
original	2.1121	0.8701	2.7373	9.2547	10.8284
CFS	0.7064	0.1404	0.7249	3.3818	1.0435
FSSIIG	0.5682	0.1445	0.5682	1.007	0.5293

Table 6 Kullback-Leibler divergence to the original data with the BNs learned with the PC algorithm.

	Australian	German	Japanese	Polish	UCSD
original	1.8896	0.9075	2.4759	5.6511	9.5844
CFS	0.6828	0.1676	0.6571	1.7771	1.0185
FSSIIG	0.5607	0.1729	0.5682	0.7584	0.5397

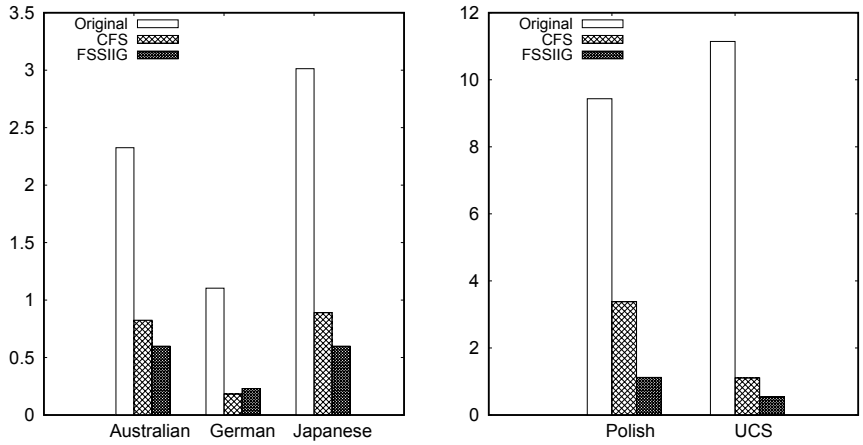

Fig. 3 Kullback-Leibler divergence to the original data with the BNs learned with the K2 algorithm.

5.1 Comments on the results

In a first overview of the results, we can observe that for the three learning methods that we have considered, the BNs represent the data much better when we apply a previous feature selection (CFS or FFSIIG). In fact, for all datasets, the KL divergence to the original data is considerably higher for the BNs learned with all the features. Therefore, we can observe that the

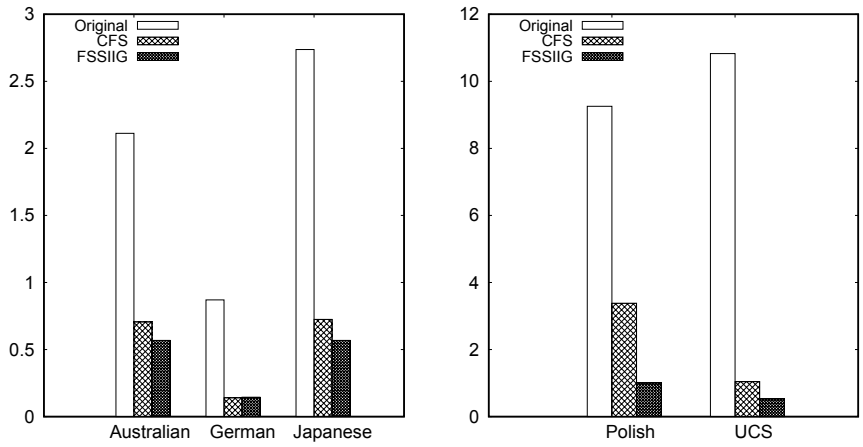

Fig. 4 Kullback-Leibler divergence to the original data with the BNs learned with local search+BDeu.

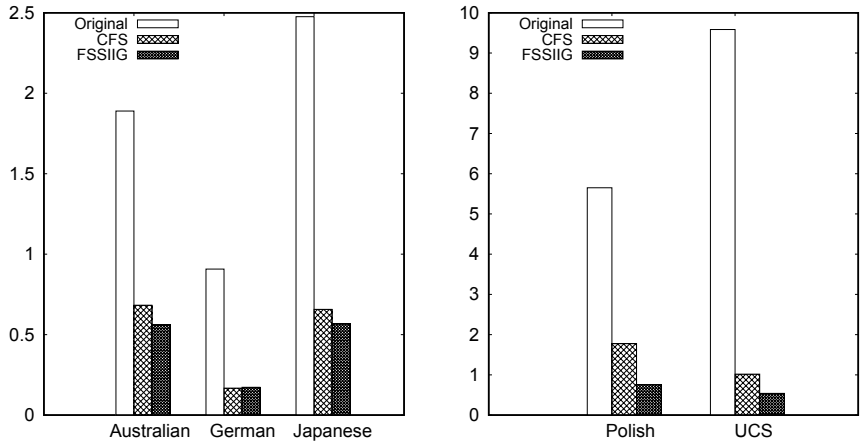

Fig. 5 Kullback-Leibler divergence to the original data with the BNs learned with the PC algorithm.

redundant and/or irrelevant features give rise to a less representative model of the data.

Comparing the two best methods, the KL divergence is lower of the BNs learned with the features selected by FFSIIG than the divergence of the BNs learned after selecting features by applying CFS for all datasets, except for the German database where the CFS obtains better results. So, in general the results obtained with FFSIIG method are better than the obtained with CFS. FFSIIG is better in 4 of 5 datasets. Moreover, for the German dataset, where the KL divergence is lower for the CFS, the difference is not as significant as in the rest of datasets, where the KL divergence for the CFS is considerably higher than the ones obtained with the FFSIIG.

The difference in the German dataset is slightly better when the BN is learned with the K2 algorithm, which obtains the worst results for this dataset than the local serach+BDeu approach and the PC algorithm. These last two learning approaches obtain better results in terms of KL divergence and the differences are difficult to appreciate visually in Figures 4 and 5 for the German dataset. Therefore, we believe that the differences are not significant if we also take into account that the subset of features obtained by CFS and FFSIIG methods is very similar for this particular dataset.

With the experimentation carried out, we can conclude that: (i) a previous variable selection method can improve notably the adjustment on the data when we build a BN; (ii) the new method to select subsets of variables, FFSIIG, obtains BNs that have a better adjustment than the ones obtained by the known method of CFS; and (iii) the BNs obtained with the FFSIIG, as a previous step, are more simple (explicative) because they are built with a lower number of features than the ones built using previously the CFS.

6 A practical example

In this section we want to show an example of the utility of BNs in credit domains that cannot be obtained using other systems. We shall work with the German dataset applying the FFSIIG algorithm in a previous step. After using FFSIIG, we shall work only with 4 features and the class variable. The meaning of the possible values of these five variables is shown in Table 7. We shall consider the BN learned using the PC algorithm. In Figure 6 we can see the graph of this BN.

Table 7 Description of the values of the variables in the Bayesian Network.

feature	value	meaning
credit_history	0	The client has no taken credits or he has paid back all credit duly
	1	The client has paid back all credits at this bank duly
	2	The client has paid all existing credits duly until now
	3	The client has ever delayed paying off some credit in the past
	4	The account is critical or the client has other credit existing at other bank
credit_amount	0	The amount of credit to loan at this client is lower than 3913.5
	1	The amount of credit to loan at this client is higher than 3913.5
other_payment_plans	0	Payment thorough a bank
	1	Payment thorough stores
	2	None
checking_status	0	The client has a checking account less than one day ago
	1	The client has a client account between 1 and 200 days
	2	The client has a client account more than 200 days ago
	3	The client has no checking account
class	0	The client has paid back the credit
	1	The client has not paid back the credit

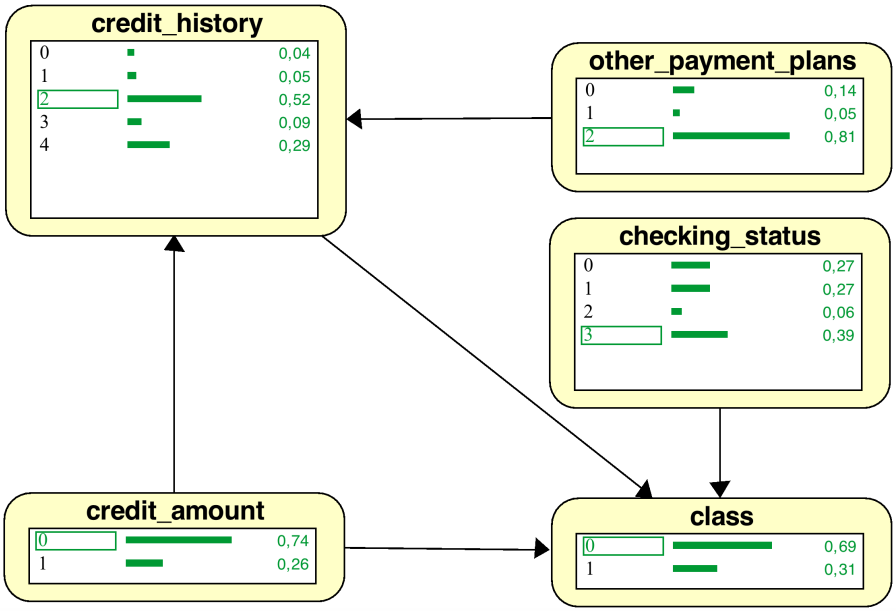

Fig. 6 Bayesian network learned with German database after applying FSSIIG thorough PC algorithm.

According to this graph, the variable *other_payment_plans* does not depend on any other variable. The same situation appears with *credit_amount* and *checking_status*. *Credit_history* depends directly on *other_payment_plans* and on *credit_amount*. Here, the class variable depends directly on *credit_history*, *credit_amount* and *checking_status*.

As we said in Section 2, besides to the graph, a BN contains for each node, a conditional probability of the node given its parents. What we can see in Figure 6 is the probability distribution of each node. These probability distributions represent the prior probability for each value of each variable i.e. the probability before any observation.

Examples of inference via this model can be described as follows:

(1) **Analysis from causes to effects**. It is obvious that this BN is more interpretable than other systems to express the information like some known classifiers (as SVM, ANN or Random Forest). Thus, for an economist, a BN model is more useful in the sense that it can provide information about the dependence relation among the variables, which can be a relevant information for the expert.
When we compare the model based on BN with one based on one of the most interpretable models for classification, as are decision trees, we can find some problems in particular situations. Let consider the decision tree generated by C4.5 algorithm when it is applied with German database selecting before the same variables than the BN. This tree in shown in Figure 7.

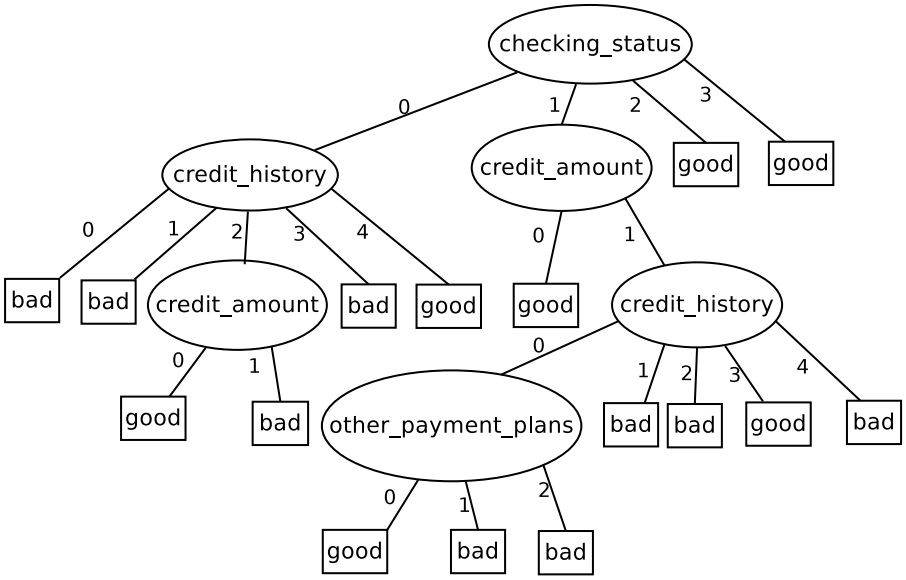

Fig. 7 Decision Tree generated by C4.5 algorithm when it is applied with German database selecting before the same variables than the Bayesian network

Suppose that we know that *checking_status* has the value 1, which is equivalent to the value $0 <= X < 200$ in the tree, that *credit_amount* has the value 1 ($(3913.5 - inf)$) and that *credit_history* has the value 0 (no credits/all paid). If we do not know the value of the variable *other_payment_plans* we can not know if the credit is *good* or *bad*. This situation represents an important inconvenient for an expert in credit scoring because not always it is possible to know all the data about a credit.
Nevertheless, with a BN it is possible to determinate the probability that the credit is *good* or not. In fact, The Figure 8 shows the probability distributions of each one of the variables of the BN (in particular, the class variable) once we have these observations.

(2) **Analysis from effects to causes**. BNs allow us to do inference in both directions, from effects to causes and viceversa, whereas in other predictors it is only possible to predict the class value. For instance, suppose now that we only know that *checking_status* has the value 3 (no checking), in particular the value for the feature *credit_amount* is not known. If we do inference with this observation we obtain the probability distributions for each variable, which we can see in Figure 9. According with these probability distributions, in this case the BN would predict that the credit would be given. As we can observe, the decision tree of Figure 7 also would predict that the credit would be *good*. BNs and decision trees have carried out forward inference in this case. However, we could not know nothing about the *credit_amount* through the tree, unless we knew previously the value

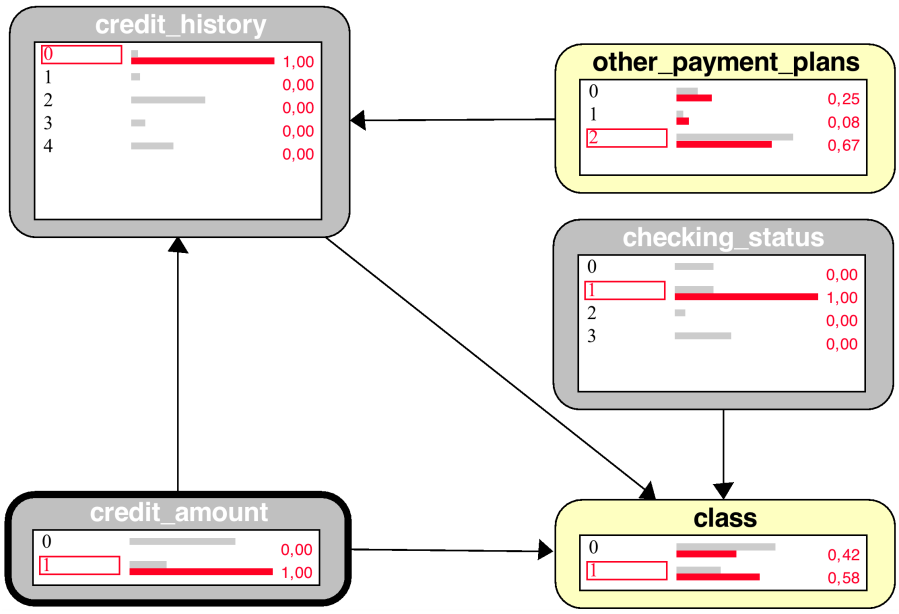

Fig. 8 Probability distributions of the variables of the Bayesian network once we know that *checking_status* = 1 that *credit_history* = 0 and that *credit_amount* = 1

of this variable. This may be problematic for a credit analyst because it may be crucial to have an estimation about the borrowed amount.
The previous problem (not to have an estimation of the value for *credit_amount* when the tree of Figure 7 is used) is solved with BNs. If we know that there is no checking (*checking_status* = 3) and the credit is *good* (*class* = 0) we can do inference with the BN in order to obtain the probability distributions of the rest of variables, among them, *credit_amount*, as we can see in Figure 10. It would be an important advantage for a credit analyst because he could have an estimation about the credit amount that he should give to a client. The credit amount is an essential issue since small amounts can relieve the losses if the credit is bad.

7 Conclusions

In this paper we have emphasized on the capabilities of the Bayesian Networks when they are applied on credit domains. We have presented a method to select features that can be combined with any method to build a BN. The method is based on imprecise probabilities and uncertainty measures on general theories to represent the information; and it selects in a forward procedure a subset of informative features. This method has been called *Forward Feature Selection based on Imprecise Information Gain* (FFSIIG). The principal advantages of this combination of FFSIIG + BN is that the final model expresses a reduction

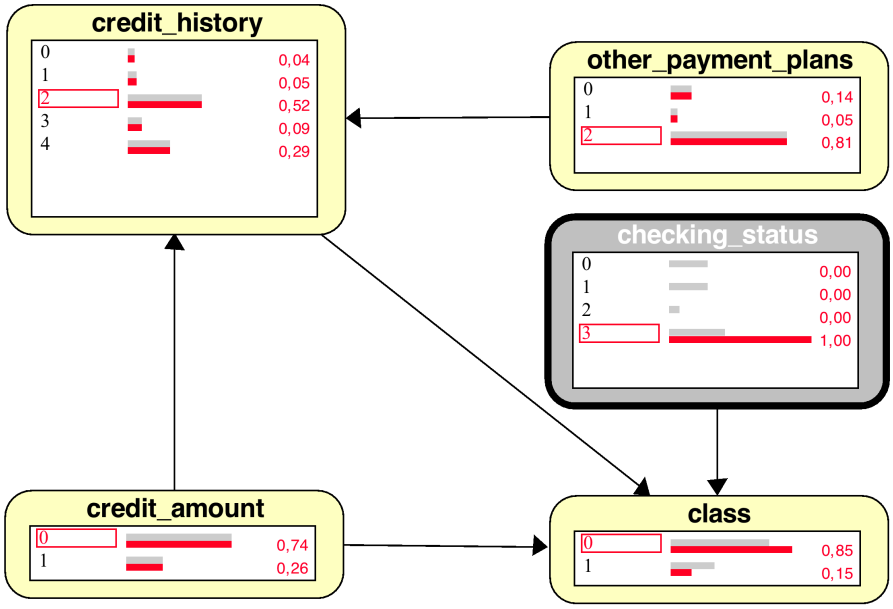

Fig. 9 Probability distributions of the variables of the Bayesian network once we know that *checking_status* = 3

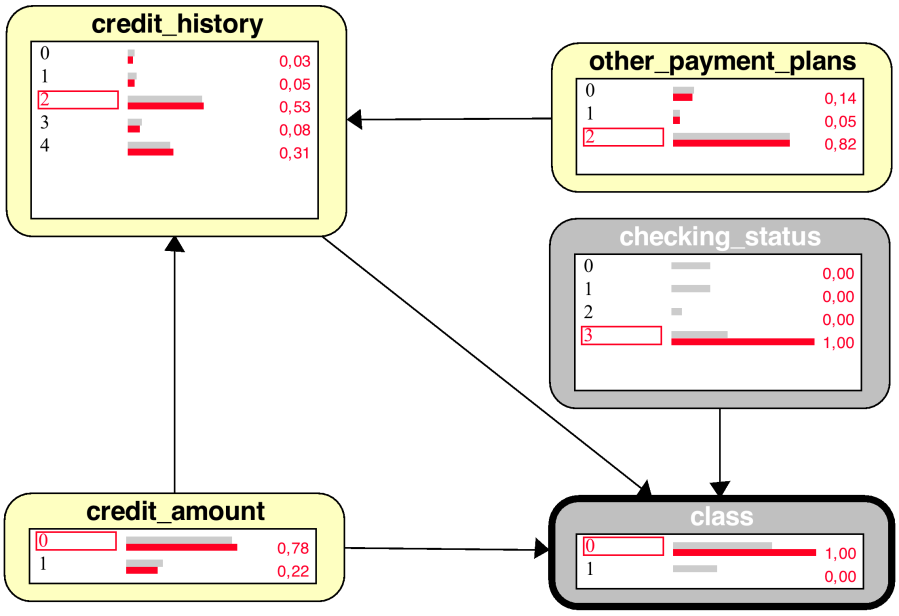

Fig. 10 Probability distributions of the variables of the Bayesian network once we know that *checking_status* = 3 and that *class* = 0

of the complexity of the initial problem, because a few number of features are considered; and represents a better fit on data.

Our proposal has been compared, via experiments on known datasets about credit scoring, with: (i) BNs build without a previous step of variable selection; and (ii) BNs build with a previous step of variable selection carried out by a similar model than the one of our proposal but considering a very known variable selection method based on precise probabilities and classical measures of information. The results show that our proposal, in general, obtains a better adjustment on the data that the methods used to compare.

The final conclusion of the results of this paper is that we propose a method to make inference in credit domains that reduce the complexity of the problem and can be considered as a better representation of the data available.

Acknowledgements This work has been supported by the Spanish "Ministerio de Economía y Competitividad" and by "Fondo Europeo de Desarrollo Regional" (FEDER) under Project TEC2015-69496-R.

References

1. Hand, D.J., Henley, W.E.: Statistical classification methods in consumer credit scoring: a review. Journal of the Royal Statistical Society: Series A (Statistics in Society) **160**(3), 523–541 (1997). DOI 10.1111/j.1467-985X.1997.00078.x
2. Xia, Y., Liu, C., Li, Y., Liu, N.: A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring. Expert Systems with Applications **78**, 225 – 241 (2017). DOI 10.1016/j.eswa.2017.02.017
3. Abellán, J., Mantas, C.J.: Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring. Expert Systems with Applications **41**(8), 3825 – 3830 (2014). DOI 10.1016/j.eswa.2013.12.003
4. Abellán, J., Castellano, J.G.: A comparative study on base classifiers in ensemble methods for credit scoring. Expert Systems with Applications **73**, 1 – 10 (2017). DOI 10.1016/j.eswa.2016.12.020
5. Ala'raj, M., Abbod, M.F.: A new hybrid ensemble credit scoring model based on classifiers consensus system approach. Expert Systems with Applications **64**, 36 – 55 (2016). DOI 10.1016/j.eswa.2016.07.017
6. Xiao, H., Xiao, Z., Wang, Y.: Ensemble classification based on supervised clustering for credit scoring. Applied Soft Computing **43**(C), 73–86 (2016). DOI 10.1016/j.asoc.2016.02.022
7. García, V., Marqués, A.I., Sánchez, J.S.: An insight into the experimental design for credit risk and corporate bankruptcy prediction systems. Journal of Intelligent Information Systems **44**(1), 159–189 (2015). DOI 10.1007/s10844-014-0333-4
8. Baesens, B., Egmont-Petersen, M., Castelo, R., Vanthienen, J.: Learning bayesian network classifiers for credit scoring using markov chain monte carlo search. In: 16th International Conference on Pattern Recognition, vol. 3, pp. 49–52. IEEE (2002). DOI 10.1109/ICPR.2002.1047792
9. Leong, C.K.: Credit risk scoring with bayesian network models. Computational Economics **47**(3), 423–446 (2016). DOI 10.1007/s10614-015-9505-8
10. Maes, S., Tuyls, K., Vanschoenwinkel, B., Manderick, B.: Credit card fraud detection using bayesian and neural networks. In: Proceedings of the 1st international NAISO congress on neuro fuzzy technologies, pp. 261–270 (2002)
11. Zhuang, Y., Xu, Z., Tang, Y.: A credit scoring model based on bayesian network and mutual information. In: Web Information System and Application Conference (WISA), 2015 12th, pp. 281–286. IEEE (2015). DOI 10.1109/WISA.2015.31

12. K. B. Schebesch, R.S.: Support vector machines for classifying and describing credit applicants: Detecting typical and critical regions. The Journal of the Operational Research Society **56**(9), 1082–1088 (2005). URL http://www.jstor.org/stable/4102201
13. Pai, P.F., Tan, Y.S., Hsu, M.F.: Credit rating analysis by the decision-tree support vector machine with ensemble strategies. International Journal of Fuzzy Systems **17**(4), 521–530 (2015). DOI 10.1007/s40815-015-0063-y
14. Tomczak, J.M., Zieba, M.: Classification restricted boltzmann machine for comprehensible credit scoring model. Expert Syst. Appl. **42**(4), 1789–1796 (2015). DOI 10.1016/j.eswa.2014.10.016
15. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001). DOI 10.1023/A:1010933404324
16. West, D.: Neural network credit scoring models. Computers and Operations Research **27**(11-12), 1131–1152 (2000). DOI 10.1016/S0305-0548(99)00149-5
17. Zhao, Z., Xu, S., Kang, B.H., Kabir, M.M.J., Liu, Y., Wasinger, R.: Investigation and improvement of multi-layer perceptron neural networks for credit scoring. Expert Systems with Applications **42**(7), 3508 – 3516 (2015). DOI 10.1016/j.eswa.2014.12.006
18. Walley, P.: Inferences from multinomial data; learning about a bag of marbles (with discussion). Journal of the Royal Statistical Society. Series B (Methodological) **58**(1), 3–57 (1996). DOI 10.2307/2346164
19. Wang, Y.: Imprecise probabilities based on generalized intervals for system reliability assessment. International Journal of Reliability and Safety **4**(4), 319–342 (2010). DOI 10.1504/IJRS.2010.035572
20. Weichselberger, K.: The theory of interval-probability as a unifying concept for uncertainty. International Journal of Approximate Reasoning **24**(2–3), 149 – 170 (2000). DOI 10.1016/S0888-613X(00)00032-3
21. Abellán, J., Moral, S.: Building classification trees using the total uncertainty criterion. International Journal of Intelligent Systems **18**(12), 1215–1225 (2003). DOI 10.1002/int.10143
22. Abellán, J., López, G., Garach, L., Castellano, J.G.: Extraction of decision rules via imprecise probabilities. International Journal of General Systems. **In press**(0), 1–19 (2017). DOI 10.1080/03081079.2017.1312359
23. Hall, M.A., Smith, L.A.: Feature subset selection: a correlation based filter approach. In: International Conference on Neural Information Processing and Intelligent Information Systems, pp. 855–858. Springer (1997). URL http://hdl.handle.net/10289/1515
24. Kullback, S.: Probability densities with given marginals. The Annals of Mathematical Statistics **39**(4), 1236–1243 (1968). DOI 10.1214/aoms/1177698249
25. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT press (2009)
26. van Engelen, R.A.: Approximating bayesian belief networks by arc removal. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(8), 916–920 (1997). DOI 10.1109/34.608295
27. Jiang, L., Kong, G., Li, C.: Wrapper framework for test-cost-sensitive feature selection. IEEE Transactions on Systems, Man, and Cybernetics: Systems pp. 1–10 (2019). DOI 10.1109/TSMC.2019.2904662
28. Pearl, J.: Probabilistic reasoning in intelligent systems. Morgan Kaufmann, San Mateo, CA (1988)
29. Buntine, W.: A guide to the literature on learning probabilistic networks from data. IEEE Transactions on Knowledge and Data Engineering **8**(2), 195–210 (1996). DOI 10.1109/69.494161
30. Heckerman, D., Geiger, D., Chickering, D.M.: Learning bayesian networks: The combination of knowledge and statistical data. Machine Learning **20**(3), 197–243 (1995). DOI 10.1023/A:1022623210503
31. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search, *Lecture Notes in Statistics*, vol. 81. MIT press (1993). DOI 10.1007/978-1-4612-2748-9
32. Cheng, J., Greiner, R., Kelly, J., Bell, D., Liu, W.: Learning bayesian networks from data: An information-theory based approach. Artificial Intelligence **137**(1), 43 – 90 (2002). DOI 10.1016/S0004-3702(02)00191-1

33. de Campos, L.M., Huete, J.F.: A new approach for learning belief networks using independence criteria. International Journal of Approximate Reasoning **24**(1), 11 – 37 (2000). DOI 10.1016/S0888-613X(99)00042-0
34. Buntine, W.: Theory refinement on bayesian networks. In: Uncertainty Proceedings 1991, pp. 52 – 60. Morgan Kaufmann, San Francisco (CA) (1991). DOI 10.1016/B978-1-55860-203-8.50010-3
35. Cooper, G.F., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Machine Learning **9**(4), 309–347 (1992). DOI 10.1007/BF00994110
36. Chow, C., Liu, C.: Approximating discrete probability distributions with dependence trees. EEE Transactions on Information Theory **14**(3), 462–467 (1968). DOI 10.1109/TIT.1968.1054142
37. Lam, W., Bacchus, F.: Learning bayesian belief networks: An approach based on the mdl principle. Computational Intelligence **10**(3), 269–293 (1994). DOI 10.1111/j.1467-8640.1994.tb00166.x
38. Shenoy, P.P., Shafer, G.: Readings in uncertain reasoning. chap. Axioms for Probability and Belief-function Propagation, pp. 575–610. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1990). URL http://dl.acm.org/citation.cfm?id=84628.85353
39. Cooper, G.F.: The computational complexity of probabilistic inference using bayesian belief networks (research note). Artificial Intelligence **42**(2-3), 393–405 (1990). DOI 10.1016/0004-3702(90)90060-D
40. Pearl, J.: Fusion, propagation, and structuring in belief networks. Artificial Intelligence **29**(3), 241 – 288 (1986). DOI 10.1016/0004-3702(86)90072-X
41. Pearl, J.: Distributed revision of composite beliefs. Artificial Intelligence **33**(2), 173 – 215 (1987). DOI 10.1016/0004-3702(87)90034-8
42. Jensen, F., Lauritzen, S., Olesen, K.: Bayesian updating in causal probabilistic networks by local computations. Computational Statistics Quarterly **4**, 269–282 (1990)
43. Shafer, G.R., Shenoy, P.P.: Probability propagation. Annals of Mathematics and Artificial Intelligence **2**(1), 327–351 (1990). DOI 10.1007/BF01531015
44. Castillo, E., Gutiérrez, J.M., Hadi, A.S.: Expert systems and probabilistic network models. Monographs in Computer Science. Springer New York (1997). DOI 10.1007/978-1-4612-2270-5
45. Madsen, A.L., Jensen, F.V.: Lazy propagation: A junction tree inference algorithm based on lazy evaluation. Artificial Intelligence **113**(1), 203 – 245 (1999). DOI 10.1016/S0004-3702(99)00062-4
46. Kohlas, J., Shenoy, P.P.: Computation in valuation algebras. In: Handbook of defeasible reasoning and uncertainty management systems, vol. 5, chap. 2, pp. 5–39. Springer (2000). DOI 10.1007/978-94-017-1737-3_2
47. Cano, A., Moral, S., Salmerón, A.: Lazy evaluation in penniless propagation over join trees. Networks **39**(4), 175–185 (2002). DOI 10.1002/net.10024
48. Park, C.H., Kim, S.B.: Sequential random k-nearest neighbor feature selection for high-dimensional data. Expert Systems with Applications **42**(5), 2336 – 2342 (2015). DOI 10.1016/j.eswa.2014.10.044
49. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics **23**(19), 2507 (2007). DOI 10.1093/bioinformatics/btm344
50. Durduran, S.S.: A decision making system to automatic recognize of traffic accidents on the basis of a {GIS} platform. Expert Systems with Applications **37**(12), 7729 – 7736 (2010). DOI 10.1016/j.eswa.2010.04.068
51. Koc, L., Mazzuchi, T.A., Sarkani, S.: A network intrusion detection system based on a hidden naïve bayes multiclass classifier. Expert Systems with Applications **39**(18), 13492 – 13500 (2012). DOI 10.1016/j.eswa.2012.07.009
52. Dellepiane, U., Di Marcantonio, M., Laghi, E., Renzi, S.: Bankruptcy prediction using support vector machines and feature selection during the recent financial crisis. International Journal of Economics and Finance **7**(8), 182–195 (2015). DOI 10.5539/ijef.v7n8p182
53. Hancer, E., Xue, B., Zhang, M.: Differential evolution for filter feature selection based on information theory and feature ranking. Knowledge-Based Systems **140**, 103 – 119 (2018). DOI https://doi.org/10.1016/j.knosys.2017.10.028
54. Sanghani, G., Kotecha, K.: Incremental personalized e-mail spam filter using novel tfdcr feature selection with dynamic feature update. Expert Systems with Applications **115**, 287 – 299 (2019). DOI https://doi.org/10.1016/j.eswa.2018.07.049

55. Tallón-Ballesteros, A.J., Riquelme, J.C., Ruiz, R.: Semi-wrapper feature subset selector for feed-forward neural networks: Applications to binary and multi-class classification problems. Neurocomputing **353**, 28 – 44 (2019). DOI https://doi.org/10.1016/j.neucom.2018.05.133. Recent Advancements in Hybrid Artificial Intelligence Systems
56. Ghiselli, E.E.: Theory of psychological measurement, *McGraw-Hill series in psychology*, vol. 13. McGraw-Hill New York (1964)
57. Fayyad, U., Irani, K.: Multi-valued interval discretization of continuous-valued attributes for classification learning. In: Proceeding of the 13th International joint Conference on Artificial Inteligence, pp. 1022–1027. Morgan Kaufmann (1993)
58. Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal **27**(3), 379–423 (1948). DOI 10.1002/j.1538-7305.1948.tb01338.x
59. Quinlan, J.R.: Induction of decision trees. Machine Learning **1**(1), 81–106 (1986). DOI 10.1023/A:1022643204877
60. Abellán, J.: Uncertainty measures on probability intervals from the imprecise dirichlet model. International Journal of General Systems **35**(5), 509–528 (2006). DOI 10.1080/03081070600687643
61. Mantas, C.J., Abellán, J.: Analysis and extension of decision trees based on imprecise probabilities: Application on noisy data. Expert Systems with Applications **41**(5), 2514 – 2525 (2014). DOI 10.1016/j.eswa.2013.09.050
62. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, second edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2005)
63. Elvira Consortium: Elvira: An environment for probabilistic graphical models. In: J. Gámez, A. Salmerón (eds.) Proceedings of the 1st European Workshop on Probabilistics Graphical Models (PGM 2002), pp. 222–230. Cuenca, Spain (2002)
64. Lichman, M.: UCI machine learning repository (2013). URL http://archive.ics.uci.edu/ml
65. Pietruszkiewicz, W.: Dynamical systems and nonlinear kalman filtering applied in classification. In: Cybernetic Intelligent Systems, 2008. CIS 2008. 7th IEEE International Conference on, pp. 1–6 (2008). DOI 10.1109/UKRICIS.2008.4798948