Intelligent Data Analysis 18 (2014) 3–24
DOI 10.3233/IDA-130625
IOS Press

Profit optimizing customer churn prediction with Bayesian network classifiers

Thomas Verbraken[a,*], Wouter Verbeke[a] and Bart Baesens[a,b,c]
[a]*Department of Decision Sciences and Information Management, Katholieke Universiteit Leuven, Leuven, Belgium*
[b]*School of Management, University of Southampton, Southampton, UK*
[c]*Vlerick Leuven-Gent Management School, Leuven, Belgium*

Abstract. Customer churn prediction is becoming an increasingly important business analytics problem for telecom operators. In order to increase the efficiency of customer retention campaigns, churn prediction models need to be accurate as well as compact and interpretable. Although a myriad of techniques for churn prediction has been examined, there has been little attention for the use of Bayesian Network classifiers. This paper investigates the predictive power of a number of Bayesian Network algorithms, ranging from the Naive Bayes classifier to General Bayesian Network classifiers. Furthermore, a feature selection method based on the concept of the Markov Blanket, which is genuinely related to Bayesian Networks, is tested. The performance of the classifiers is evaluated with both the Area under the Receiver Operating Characteristic Curve and the recently introduced Maximum Profit criterion. The Maximum Profit criterion performs an intelligent optimization by targeting this fraction of the customer base which would maximize the profit generated by a retention campaign. The results of the experiments are rigorously tested and indicate that most of the analyzed techniques have a comparable performance. Some methods, however, are more preferred since they lead to compact networks, which enhances the interpretability and comprehensibility of the churn prediction models.

Keywords: Bayesian network classifier, churn prediction, maximum profit criterion

1. Introduction

The number of mobile phone users has increased tremendously during the last decade. At the end of 2010, there will be more than five billion mobile phone users,[1] which is over 70% of the world population. As a result, telecommunication markets are getting saturated, particularly in developed countries, and mobile phone penetration rates are stagnating. Therefore, operators are shifting their focus from attracting new customers to retaining the existing customer base. Moreover, the literature reports that customer retention is profitable because: (1) acquiring a new client is five to six times more costly than retaining an existing customer [5,7,17,52]; (2) long-term customers generate higher profits, tend to be less sensitive to competitive marketing activities, become less costly to serve, and may provide new referrals through positive word-of-mouth, while dissatisfied customers might spread negative word-of mouth [18,29,46,50,53,58,66]; (3) losing customers leads to opportunity costs because of reduced

[*] Corresponding author: Thomas Verbraken, Department of Decision Sciences and Information Management, Katholieke Universiteit Leuven, Naamsestraat 69, B-3000 Leuven, Belgium. E-mail: thomas.verbraken@econ.kuleuven.be.

[1] http://www.eito.com.

sales [54]. A small improvement in customer retention can therefore lead to a significant increase in profit [60]. For successful targeted marketing campaigns, it is crucial that operators are able to identify clients with a high probability to churn in the near future. Next to correctly classifying the future churners, it is important to gain insight in the reasons for customers to be classified as churners. Therefore, telecom operators prefer compact and interpretable models, as it allows them to check whether the model is in line with current domain knowledge. Moreover, it enables operators to recognize potential warning signs for customer churn.

Although a myriad of techniques for churn prediction has been examined, there has been little attention to the use of Bayesian Network (BN) classifiers. This paper will investigate the predictive power of a number of Bayesian Network algorithms, ranging from the Naive Bayes classifier, with very strict independence assumptions, to General Bayesian Network classifiers, which allow more flexibility. The performance of the classifiers is evaluated with both the Area under the Receiver Operating Characteristic Curve and the Maximum Profit criterion and is rigorously statistically tested.

Like most real life data mining problems, also churn prediction involves a large number of attributes. Including irrelevant variables would result in complex and incomprehensible classification models, impeding the interpretability of these classifiers [45]. Hence, feature selection is commonly applied to withhold only those variables with strong explanatory power. Many feature selection algorithms have been proposed in the literature. However, most of these methods perform the selection from a univariate perspective, i.e. they assess a measure of dependence between the attributes and the target variable for each attribute separately. In this study, a feature selection method based on the concept of the Markov Blanket, which is genuinely related to Bayesian Networks, will be analyzed. This method approaches the input selection problem from a multivariate point of view and relies on the concept of *conditional* independence. Especially in the context of some of the Bayesian Network classifiers, this form of feature selection proves to be useful, as will be shown in Section 5. The impact of this variable reduction on classification performance and network complexity is investigated, since, basically, one is looking for the most compact Bayesian network with the highest explanatory power.

For measuring classifier performance and selecting the most appropriate classification method, a variety of performance measures has been used [1,45]. In this study, the well-known Area under the Receiver Operating Characteristic Curve (AUC) is employed, as well as the Maximum Profit (MP) criterion, recently proposed by Verbeke et al. [61]. Instead of measuring performance over the whole output range, as AUC does, the maximum profit criterion performs an intelligent optimization by targeting this fraction of the customer base which would maximize the profit generated by a retention campaign. As such, it is able to indicate the model which maximizes the effectiveness of a retention campaign. The rationale behind this performance measure is that the most important goal for a telecom provider is to optimize its profit.

The remainder of this paper is organized as follows. In Section 2, the general problem of customer churn will be stated. Section 3 will give an overview of the main Bayesian Network classifiers and discuss the algorithms briefly. In Section 4, the experimental setup is described, the data set characteristics are discussed, and tests for statistical significance are clarified. Finally, the results of the experiments are described in Section 5.

2. Customer churn prediction

Many companies and organizations are confronted with customer churn. For instance, wireless telecom operators report annual churn rates up to 40% of their customer base [49]. Customer churn is associated with a direct loss of income and a diversity of supplementary costs, such as for instance the investments to acquire new customers to maintain the level of the customer base. Therefore, reducing customer churn by directing specifically designed marketing campaigns to the customers with the highest probability to attrite, has proven to be profitable to a company [60]. To improve the efficiency of customer retention campaigns, a customer churn prediction model is needed to indicate the customers which are the most likely to churn and should be included in the retention campaign.

Customer churn prediction is a problem for which typically a data mining approach is adopted. Data mining entails the overall process of extracting knowledge from data. Based on historical data a model can be trained to classify customers as future churners or non-churners. Numerous classification techniques have been applied to predict churn, including traditional statistical methods such as logistic regression [9,43,49], non-parametric statistical models like for instance k-nearest neighbor [20], decision trees [44,64], and neural networks [6,34]. An extensive literature review on customer churn prediction modeling can be found in Verbeke et al. [62].

The process of developing a customer churn prediction model consists of several steps. The first step in this process consists of gathering relevant data and selecting candidate explanatory variables. The resulting data set is then cleaned and preprocessed. In the second step a model is built. A modeling technique is selected based on the requirements of the model and the type of data. Input selection is often applied to reduce the number of variables in order to get a consistent, unbiased, and relevant set of explanatory variables. Depending on the number of observations, which can be small in case of new products, a model is trained by cross validation or by splitting the data set in a separate training and test set. The resulting model is then evaluated, typically by comparing the true values of the target variable with the predicted values, but also, if possible, by interpreting the selected variables and the modeled relation with the target variable. A variety of performance measures to evaluate a classification model have been proposed in the literature [1]. As will be discussed in Section 4, in this study both the statistically based area under the receiver operating characteristic curve [38] and the maximum profit criterion, recently proposed by Verbeke et al. [61], will be applied. The latter estimates the profit a telecom operator would make when optimally exploiting the results of a particular classifier. Next, in a third step the model is assessed by a business expert to check whether the model is intuitively correct and in line with business knowledge which requires the induced model to be interpretable [45]. A prototype of the model is then developed, and deployed in the information and communication technology (ICT) architecture. The final step, once a model is implemented that performs satisfactory, consists of regularly reviewing the model in order to asses whether it still performs well. Surely in a highly technological and volatile environment as the telecommunications sector, a continuous evaluation on newly gathered data is of crucial importance.

Because of the third step, where the model is assessed by a business expert to confirm whether the model is intuitively correct, the comprehensibility aspect of customer churn prediction models is of crucial importance. However, comprehensibility of customer churn models thus far only received limited attention in the literature [44,62]. A specific family of classification techniques, which result in comprehensible models but have not been tested rigorously in a customer churn prediction setting before, are the Bayesian Network Classifiers. Two simple Bayesian Network Classifiers, i.e. Naive Bayes and standard Bayesian Networks, have been included in an extensive benchmarking study by Verbeke et al. [61], which compares the performance of a variety of state-of-the-art classification techniques applied on an extensive number of data sets. Their results suggest that Bayesian Network Classifiers form a viable alternative modeling approach for customer churn prediction as they are able to produce compact and interpretable models. The next section will discuss a number of Bayesian Network Classification techniques into detail, which are applied to five real-life telecom churn data sets in Sections 4 and 5.

3. Bayesian network classifiers

3.1. Bayesian networks

A Bayesian network (BN) represents a joint probability distribution over a set of stochastic variables, either discrete or continuous. It is to be considered as a probabilistic white-box model consisting of a qualitative part specifying the conditional (in)dependencies between the variables and a quantitative part specifying the conditional probabilities of the data set variables [51]. Formally, a Bayesian network consists of two parts $B = \langle G, \Theta \rangle$. The first part G is a directed acyclic graph (DAG) consisting of nodes and arcs. The nodes are the variables X_1 to X_n in the data set whereas the arcs indicate direct dependencies between the variables. The graph G then encodes the independence relationships in the domain under investigation. The second part of the network, Θ, represents the conditional probability distributions. It contains a parameter $\theta_{x_i|\Pi_{x_i}} = P_B(x_i|\Pi_{x_i})$ for each possible value x_i of X_i, given each combination of the direct parent variables of X_i, Π_{x_i} of Π_{X_i}, where Π_{X_i} denotes the set of direct parents of X_i in G. The network B then represents the following joint probability distribution:

$$P_B(X_1, \ldots, X_n) = \prod_{i=1}^{n} P_B(X_i|\Pi_{X_i}) = \prod_{i=1}^{n} \theta_{X_i|\Pi_{X_i}}. \tag{1}$$

The first task when learning a Bayesian network is to find the structure G of the network. Once we know the network structure G, the parameters Θ need to be estimated. In general, these two estimation tasks are performed separately. In this paper, we will use the empirical frequencies from the data D to estimate these parameters:

$$\theta_{x_i|\Pi_{x_i}} = \hat{P}_D(x_i|\Pi_{x_i}) \tag{2}$$

It can be shown that these estimates maximize the log likelihood of the network B given the data D. Note that these estimates might be further improved by a smoothing operation [27].

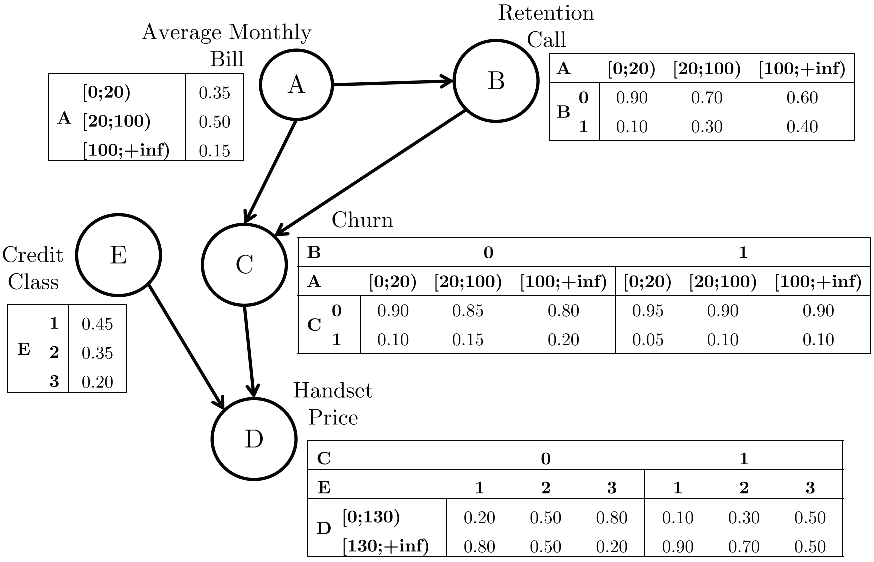
A Bayesian network is essentially a statistical model that makes it feasible to compute the (joint) posterior probability distribution of any subset of unobserved stochastic variables, given that the variables in the complementary subset are observed. This functionality makes it possible to use a Bayesian network as a statistical classifier by applying the winner-takes-all rule to the posterior probability distribution for the (unobserved) class node. The underlying assumption behind the winner-takes-all rule is that all gains and losses are equal. For a discussion of this aspect see, e.g. [22]. A simple example of a Bayesian network classifier is given in Fig. 1. Suppose that, for a particular customer, all variables except C are known and take the following values: $A \in [20; 100)$, $B = 0$, $D \in [0; 130)$ and $E = 1$. The probability that the customer will churn conditional on this information can be calculated as:

$$P\,(C|A, B, D, E) = \frac{P\,(C, A, B, D, E)}{P\,(A, B, D, E)}$$

Thus, reading from Fig. 1 and using Eq. (1) yields:

$$P\,(C = 0, A \in [20; 100), B = 0, D \in [0; 130), E = 1) = 0.85 \cdot 0.5 \cdot 0.7 \cdot 0.2 \cdot 0.45 = 0.0268$$

$$P\,(C = 1, A \in [20; 100), B = 0, D \in [0; 130), E = 1) = 0.15 \cdot 0.5 \cdot 0.7 \cdot 0.1 \cdot 0.45 = 0.0024$$



	A	
A	[0;20)	0.35
	[20;100)	0.50
	[100;+inf)	0.15

A		[0;20)	[20;100)	[100;+inf)
B	0	0.90	0.70	0.60
	1	0.10	0.30	0.40

	E	
E	1	0.45
	2	0.35
	3	0.20

B		0			1		
A		[0;20)	[20;100)	[100;+inf)	[0;20)	[20;100)	[100;+inf)
C	0	0.90	0.85	0.80	0.95	0.90	0.90
	1	0.10	0.15	0.20	0.05	0.10	0.10

C		0			1		
E		1	2	3	1	2	3
D	[0;130)	0.20	0.50	0.80	0.10	0.30	0.50
	[130;+inf)	0.80	0.50	0.20	0.90	0.70	0.50

Fig. 1. Classification with a Bayesian network.

Hence, the conditional probability for churning is:

$$P\left(C = 0|A \in [20;100), B = 0, D \in [0;130), E = 1\right) = \frac{0.0268}{0.0268 + 0.0024} = 0.92$$

$$P\left(C = 1|A \in [20;100), B = 0, D \in [0;130), E = 1\right) = \frac{0.0024}{0.0267 + 0.0024} = 0.08$$

According to the winner-takes-all rule, the customer will be classified as a non-churner. In what follows, several structure learning algorithms for the construction of Bayesian network classifiers will be discussed.

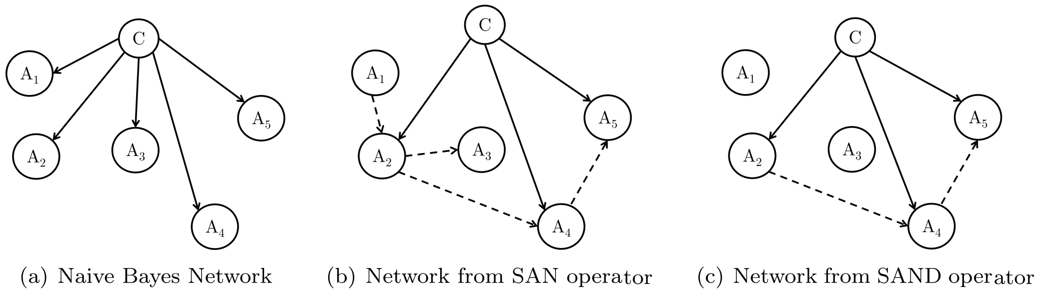
3.2. The Naive Bayes classifier

A simple classifier, which in practice often performs surprisingly well, is the Naive Bayes classifier [22,35,40]. This classifier basically learns the class-conditional probabilities $P(X_i = x_i|C = c_l)$ of each variable X_i given the class label c_l. A new test case $(X_1 = x_1, \ldots, X_n = x_n)$ is then classified by using Bayes' rule to compute the posterior probability of each class c_l given the vector of observed variable values:

$$P(C = c_l|X_1 = x_1, \ldots, X_n) = \frac{P(C = c_l)P(X_1 = x_1, \ldots, X_n = x_n|C = c_l)}{P(X_1 = x_1, \ldots, X_n = x_n)} \tag{3}$$

The simplifying assumption behind the Naive Bayes classifier then assumes that the variables are conditionally independent given the class label. Hence,

$$P(X_1 = x_1, \ldots, X_n = x_n|C = c_l) = \prod_{i=1}^{n} P(X_i = x_i|C = c_l). \tag{4}$$



(a) Naive Bayes Network (b) Network from SAN operator (c) Network from SAND operator

Fig. 2. Examples of augmented Bayesian networks.

This assumption simplifies the estimation of the class-conditional probabilities from the training data. Notice that one does not estimate the denominator in Eq. (3) since it is independent of the class. Instead, one normalizes the nominator term $P(C = c_l)P(X_1 = x_1, \ldots, X_n = x_n|C = c_l)$ to 1 over all classes. Naive Bayes classifiers are easy to construct since the structure is given apriori and no structure learning phase is required. The probabilities $P(X_i = x_i|C = c_l)$ are estimated by using the frequency counts for the discrete variables and a normal or kernel density based method for continuous variables [35]. Figure 2(a) provides a graphical representation of a Naive Bayes classifier.

3.3. Augmented Naive Bayes classifiers

The strength of Naive Bayes classifiers inspired several authors to develop *Augmented* Naive Bayes network classifiers. These are methods based on the Naive Bayes classifier while partially relaxing the independence assumption. The *Selective Naive Bayes* classifier omits certain variables to deal with strong correlation among attributes [39], whereas the *Semi-Naive Bayesian* classifier clusters correlated attributes into pairwise disjount groups [36]. Friedman et al. developed *Tree Augmented Naive Bayes (TAN) classifiers*, an algorithm where every attribute has one and only one additional parent next to the class node [27].

In this study, the Augmented Naive Bayes classifiers developed by Sacha [55] are used. This is a family of classifiers where the constraints of the TAN approach are further relaxed: not all attributes need to be dependent on the class node and there does not necessarily need to be an undirected path between two attributes. The algorithms exist of a combination of five basic operators, summarized in Table 1. The measure of dependency between two attributes is defined as follows:

$$I(X,Y) = \begin{cases} \sum_{x,y} p\left(x,y|c\right) \log\left(\dfrac{p\left(x,y|c\right)}{p(x|c)p\left(y|c\right)}\right) & \text{if } X, Y \text{ are dependent on } C \\ \sum_{x,y} p\left(x,y|c\right) \log\left(\dfrac{p\left(x,y|c\right)}{p(x|c)p\left(y\right)}\right) & \text{if } \textit{only } X \text{ is dependent on } C \\ \sum_{x,y} p\left(x,y|c\right) \log\left(\dfrac{p\left(x,y|c\right)}{p(x)p\left(y|c\right)}\right) & \text{if } \textit{only } Y \text{ is dependent on } C \\ \sum_{x,y} p\left(x,y\right) \log\left(\dfrac{p\left(x,y\right)}{p(x)p\left(y\right)}\right) & \text{if } X, Y \text{ are independent from } C \end{cases} \quad (5)$$

Combining different class dependency operators with augmenting methods from Table 1 yields a number of algorithms, listed below as per increasing complexity:

Table 1
Augmented Naive Bayes approach: Different operators

Operator	Description
Class dependency operator	Connects the class node with *all* attributes. This can be used to construct a naive Bayes network.
Selective augmented naive bayes (SAN)	Connects the class node with the attributes it depends on. A greedy search algorithm is used to seek for attributes dependent on the class node. Starting with an empty set, it adds in each step the attribute to the network that optimizes a quality measure. After adding an attribute, the dependencies among *all* other attributes, whether they are connected with the class node or not, are determined by one of the *augmenter* operators.
Selective augmented naive bayes with discarding (SAND)	Connects the class node with attributes it depends on, as the SAN operator does. The difference, however, is that the augmenter will only add connections between attributes dependent on the class node, while other attributes are discarded and are no part of the network in that particular iteration. The difference between a network resulting from the SAN operator and SAND operator is illustrated in Figs 2(b) and (c).
Tree-augmenter	This operator builds the minimum spanning tree among a given set of attributes. The algorithm is based on a method developed by Chow and Liu [16], but differs in the way how the mutual information is calculated. Sacha uses the conditional or unconditional probability of X and Y depending on whether there is an arc between the class node and the attribute (see formula 5).
Forest-augmenter	The forest augmenter is also used to create dependencies between the attributes, but allows for more flexibility. The resulting structure can be a forest consisting of a number of disjoint trees, meaning that there does not need to be a path between every attribute.

- Naive Bayes
- TAN: Tree augmented Naive Bayes
- FAN: Forest augmented Naive Bayes
- STAN: Selective tree augmented Naive Bayes
- STAND: Selective tree augmented Naive Bayes with discarding
- SFAN: Selective forest augmented Naive Bayes
- SFAND: Selective forest augmented Naive Bayes with discarding

The aim of these classifiers is to find a trade-off between the simplicity of the Naive Bayes classifiers (with a limited number of parameters) and the more realistic and complex case of full dependency between the attributes.

Except for Naive Bayes and TAN, all of the above procedures use a search method that requires a quality measure to assess the fitness of a network given the data. In this study, two quality measures proposed by Sacha will be used. The first is the Standard Bayesian (SB) measure, which is proportional to the posterior probability distribution $p(G, \Theta|D)$ and contains a penalty for the network size. This penalty term is a function of the dimension of the network, the latter being defined as the number of free parameters needed to fully specify the joint probability distribution. A derivation of the Standard Bayesian measure can be found in [55]. The second quality measures is the Local Leave-One-Out Cross Validation. Let V_l be the training set D without instance x_l:

$$V_l = D \backslash \{x_l\} \tag{6}$$

The quality measure is then defined as:

$$LOO(G, D) = \sum_{l=1}^{m} \left(p \left(c^{(l)} | \mathbf{a}^{(l)}, V_l, G \right) \right) \tag{7}$$

with m instances in the data set and $p\left(c^{(l)}|\mathbf{a}^{(l)}, V_l, G\right)$ being the probability of c^l conditional on the values of the attributes for instance l, the data set V_l and the structure G. In this paper, the latter two quality measures were combined with the five algorithms defined above, resulting in ten different classifiers.

It is worthwhile mentioning that the Naive Bayes classifier and the ANB classifiers without node discarding do not have the flexibility to remove variables from the network. Hence, even if a variable is completely independent from the target variable, it will still be used by the classifier. As this is clearly undesirable, a feature selection algorithm is carried out as part of the data preprocessing procedure. Although there are many input selection methods available, in this study the Markov Blanket feature selection will be used, since it tackles the problem from a multivariate perspective (see Section 4 for more information). Essentially, it applies the principle of conditional independence, which plays a pivotal role in the theory of Bayesian Networks.

3.4. General Bayesian network classifiers

All the previously discussed methods restrain the structure of the network in order to limit the complexity of the algorithms. Omitting those restrictions extends the search space of allowed networks to all *General Bayesian Networks (GBN)*. Finding the optimal network in such a solution space is known to be an NP-hard problem since the number of DAGs for n variables is superexponential in n [14]. As described by Kotsiantis [37], there are two broad categories of structure learning algorithms. The first consists of heuristic methods, searching through the space of all possible DAGs and measuring the fitness of a DAG with a score metric, whereas the second category comprises constraint based approaches using conditional independence (CI) tests to reveal the structure of the network.

Scoring based methods have been compared with CI-based algorithms by Heckerman et al. [33], leading to the observation that CI based methods perform better on sparse networks. Search-and-score algorithms, on the other hand, work with a broader variety of probabilistic models even though their heuristic nature may inhibit finding the optimal structure.

3.4.1. Search-and-score algorithms

Several algorithms have been proposed in the literature, e.g. [15,33] show that selecting a single DAG using greedy search often leads to accurate predictions. In this analysis, the well-known *K2* algorithm [19] is applied. It seeks for the network with the highest posterior probability given a data set and makes the following four assumptions:

- All attributes are discrete,
- The instances occur independently, given one Bayesian network,
- There are no cases that have variables with missing values,
- The prior probability of the conditional probabilities in the conditional probability tables at each node is uniform.

Given these assumptions, a greedy search algorithm will find the network with the highest score, given the database D. For a detailed discussion, one may refer to [19].

3.4.2. Constraint based algorithms

Constraint based algorithms are also known as Conditional Independence (CI) based methods. They do not use a score but employ conditional independence tests to find the relations between attributes given a data set. In this paper, the *Three Phase Dependency Analysis (TPDA)* algorithm [13] is used, in which the concept of d-separation plays an essential role. It can be shown that if sets of variables X and

Z are d-separated by Y in a directed acyclic graph G, then X is independent of Z conditional on Y in every distribution compatible with G [30,63]. It is precisely this property that will be exploited in the algorithm of Cheng to learn the Bayesian network structure. The algorithm itself consists of four phases. In a first phase, a draft of the network structure is made based on the mutual information between each pair of nodes. The second and third phase then add and remove arcs based on the concept of d-separation and conditional independence tests. Finally, in the fourth phase, the Bayesian network is pruned and its parameters are estimated. The algorithm is described in detail by Cheng [11,12].

3.4.3. Hybrid methods

Also hybrid methods have been developed, combining characteristics of both search-and score and constraint based algorithms. Examples of such techniques are the *Sparse Candidate (SC)* algorithm [28] and the *Max-Min Hill-Climbing learning algorithm (MMHC)* [59].

This latter method finds the parent-children set ($\mathbf{PC}$) of each and every node, and thus determines the skeleton of the Bayesian network. In a first phase, also called the forward phase, nodes selected by a heuristic procedure sequentially enter a candidate PC set ($\mathbf{CPC}$). The set may contain false positives, which are removed in phase II of the algorithm, i.e. the backward phase. The algorithm tests whether any variable in $\mathbf{CPC}$ is conditionally independent on the target variable, given a blocking set $\mathbf{S} \subseteq \mathbf{CPC}$. If such variables are found, they are removed from $\mathbf{CPC}$. As a measure of conditional (in)dependence, the G^2 measure, as described by Kotsiantis [57], is used. This measure is asymptotically following a χ^2 distribution with appropriate degrees of freedom, which allows to calculate a p-value indicating the probability of falsely rejecting the null hypothesis. Conditional independence is assumed when the p-value is less than the significance level α (0.05 and 0.01 in this study). Once the skeleton is determined, a greedy search method is used to direct the edges between the attributes. This is the second step and the search-and-score part of the algorithm, making it a hybrid method. The *BDeu* score [32] has been used in this paper.

4. Experimental setup

The aim of the study is twofold. Firstly, the differences in terms of performance between the Bayesian Network classifiers, described in Section 3, are investigated. Obviously, the aim is to find the algorithm with the most discriminative power, and to determine whether the differences are significant. Secondly, the experiments need to reveal whether the Markov Blanket feature selection has a deteriorating impact on classification performance. Preferably, the input selection would reduce the number of attributes without affecting the predictive performance substantially, since it is supposed to remove variables which are independent from the target, conditional on the variables which are withheld. The remainder of this section will describe the experimental setup.

4.1. Data sets and preprocessing

Four real life and one synthetic data set will be used to evaluate the performance of the different classification techniques. Table 2 summarizes the most important aspects of these data sets. The first data set was obtained directly from a European telecommunication operator, the next three are available at the website of the Center for Customer Relationship Management at Duke University,[2] and the last one is a synthetic data set, available at the UCI repository.[3]

[2] www.fuqua.duke.edu/centers/ccrm/datasets/download.html.

[3] www.sgi.com/tech/mlc/db.

Table 2
Summary of data set characteristics

ID	Source	# Obs.	# attr. orig.	# attr. MB.05	# attr. MB.01	% Churn
O1	Operator	47,761	28	21	12	3.69
D1	Duke	12,499	12	12	11	34.67
D2	Duke	99,986	172 (50)	44	42	49.55
D3	Duke	40,000	49	31	26	49.98
UCI	UCI	5,000	12	8	8	14.14

The number of observations or instances in a data set is of great importance. The more observations, the more generally applicable the generated model will be. A large data set also allows to split the data set in a separate training and test set. The training set is used to induce a model, while the test set is only used to evaluate the proposed model. This ensures the performance measure is not biased due to overfitting the model to the test data. As can be seen from the table, the smallest data set has five thousand observations, and the largest up to almost hundred thousand observations. The data sets also differ substantially regarding the number of candidate explanatory variables, in a range from 12 up to 171 (for our analysis this is limited to 50 variables, see below). More attributes do not guarantee a better classification model however. The eventual performance of the model mainly depends on the explanatory power of the variables. Since a large number of attributes heavily increases the computational requirements, and most often only a limited number of variables is effectively valuable to explain the target variable, a feature selection method could be used to limit the available attributes. In the next subsection, a Markov Blanket based algorithm for feature selection is described. Another characteristic of the data set is the class distribution of the target variable, which is usually heavily skewed in a churn prediction setting. Three of the data sets approximately have an even class distribution, which is the result of undersampling (i.e. removing non-churners from the data set). To test the classifiers, the test sets have been adjusted in order to resemble a realistic churn data set.

When working with real life data sets, it is essential to preprocess the data before starting the analysis because of two main reasons [24]: (1) problems with the data (e.g. missing values, irrelevant data, etc.), and (2) preparation for data analysis (e.g. discretization of continuous variables, etc.). To ensure that the results are comparable, all data sets have been preprocessed in the same manner:

- The data set is split into a training set (66% of the instances) and a test set (34% of the instances).
- The missing values are replaced by the median or the mode for continuous or nominal variables respectively. Note that for none of the variables there were more than 5% missing values.
- Coarse variables (such as e.g. ZIP code) are clustered into smaller groups, in order to have a more meaningful contribution to the prediction of churn behavior.
- Since most Bayesian network algorithms only work with discrete variables, all continuous variables are discretized. This is done according to the algorithm of Fayyad and Irani [25].
- A large number of attributes demands high computational requirements for the Markov Blanket feature selection algorithm. Therefore, if necessary, the data sets are treated with a χ^2-filter in order to withhold the 50 most predictive attributes as input for the further analysis.[4]

[4] The choice for the χ^2 filter is based on a robustness check of several univariate procedures. The outcome suggests that most of the univariate methods lead to similar results. The aim of this step is to remove variables which show very limited dependence on the target (note that independence implies conditional independence, the reverse is not true). The number fifty is chosen to ensure that sufficient variables are withheld. If some variables are still redundant, those will be removed by the Markov Blanket feature selection, but the computational cost will have been reduced. Since the MB feature selection does still remove variables from data set D2 (from 50 to 44/42), the number fifty was not set too low.

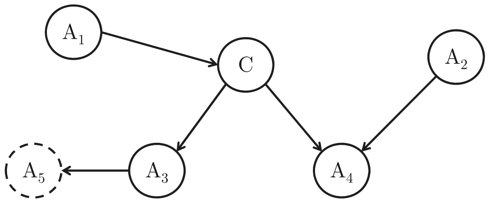


Fig. 3. The Markov Blanket of a classification node.

Once these preprocessing steps are carried out, the data sets are ready for the Markov Blanket feature selection algorithm and the Bayesian Network classifiers, described in the following subsections.

4.2. Markov blanket feature selection

Over the past decades, feature selection has become an essential part in predictive modeling. It aims to deal with three problems [31]: (1) improving the accuracy of predictive algorithms, (2) developing faster and more cost-effective predictors, and (3) gaining insight in the underlying process that generated the data. Several techniques for input selection have been proposed, but many of these rely on univariate testing procedures, i.e. the variables are assessed one by one in terms of dependency on the target variable. For this study, the Markov Blanket (MB) feature selection algorithm of Aliferis et al. [3,4] has been applied. The procedure is based on the concept of the Markov blanket, which also plays a crucial role in Bayesian Networks. As opposed to commonly used univariate methods, this algorithm approaches the task from a multivariate perspective and exploits the concept of conditional independence. The Markov Blanket of a node X, is the union of X's parents, X's children and the parents of X's children. It can be shown that when the values of the variables in the Markov Blanket of the classification node are observed, the posterior probability distribution of the classification node is independent of all other variables (nodes) that are not in the Markov Blanket [41]. Hence, all variables outside the Markov Blanket can be safely discarded because they will have no impact on the classification node and thus will not affect the classification accuracy. In this way, the Markov Blanket results in a natural form of variable selection. E.g. in Fig. 3, node A_1 will be part of the Markov Blanket because it is a parent of C, A_3 and A_4 because they are children and A_2 because it is the parent of one of the children of C. A_5 is not part of the Markov Blanket and can be ignored for classification purposes, since it will not influence C for a fixed value of A_3 (which is known in a classification context).

The Markov Blanket feature selection algorithm has been applied to the data sets at a significance level of 1% and 5%, and will be referred to as MB.01 and MB.05. Note that if attribute selection is performed, it is applied prior to training and testing the classifiers. The feature selection algorithm has been implemented in the *Causal Explorer* package for Matlab [2].

4.3. Bayesian network construction

For all the techniques discussed in Section 3, freely available software implementations have been used. The Naive Bayes classifier, TAN-classifier and the Augmented Naive Bayes classifiers have been implemented by Sacha, and exist in the form of Weka bindings [56], allowing to run the software from within the Weka Workbench [65]. In total, there are ten Augmented Naive Bayes classifiers, as a result of combining two quality measures (Local Leave One Out Cross Validation (LCV_LO) and Standard Bayesian measure (SB)) with five algorithms (itemized in Section 3). The K2 algorithm is directly available in the Weka Workbench. The constraint based algorithm, also called Three Phase Dependency

Table 3
Implementations used in this study

Algorithm	Implementation
Markov blanket feature selection	Causal explorer [2]
Logistic regression	Weka toolbox [65]
(Augmented) naive bayes classifiers	Bayesian network classifier toolbox [56]
K2 algorithm	Weka toolbox [65]
Three phase dependency analysis	Powerpredictor [10]
Max-min hill-climbing	Causal explorer [2] for structure learning and Bayesian net toolbox [47] for inference

Analysis, is available in the application *Powerpredictor*, developed by Cheng [10]. The Max-Min Hill-Climbing algorithm is available for Matlab [2]. For inference in the networks generated by the MMHC algorithm, the Bayesian Net Toolbox for Matlab, developed by Murphy [47], has been used. In Table 3, an overview of all software implementations is given. Note that Logistic Regression and the Naive Bayes Classifier are included in this study as benchmarks for the Bayesian Network classifiers.

4.4. Measuring classifier performance

A variety of performance measures has been used to gauge the strength of different classifiers and to select the appropriate model [1,45]. In this study two measures are reported: the *Area Under the Receiver Operating Characteristic Curve* (AUROC, or briefly AUC), and the *Maximum Profit* (MP) criterion, recently introduced by Verbeke et al. [61].

We tested 16 algorithms on five data sets, in combination with MB.05, MB.01, or without preceding Markov Blanket feature selection. This results in 240 values for AUC and 240 values for MP. Section 4.5 will explain how the statistical significance of differences between methods is analyzed.

4.4.1. Area under the receiver operating characteristic curve

Most classification techniques result in a continuous output. For instance in a customer churn prediction setting, a probability estimate between zero and one of being a churner is produced by the model. Depending on a threshold probability value, a customer will be classified as a churner or a non-churner. The *receiver operating characteristic* (ROC) curve displays the fraction of the identified churners by the model on the Y-axis as a function of one minus the fraction of identified non-churners. These fractions are dependent on the threshold probability. ROC curves provide an indication of the correctness of the predictions of classification models. In order to compare ROC curves of different classifiers regardless of the threshold value and misclassification costs, one often calculates the *area under the receiver operating characteristic curve* (AUROC or AUC). Assume that a classifier produces a score $s = s\left(X\right)$, with X the vector of attributes. For a BN classifier, the score s is equal to the probability estimate $p\left(c = 1|X = x\right)$. Let $f_l(s)$ be the probability density function of the scores s for the classes $l \in \{0, 1\}$, and $F_l(s)$ the corresponding cumulative distribution function. Then, it can be shown that AUC is defined as follows [38]:

$$AUC = \int_{-\infty}^{\infty} F_0(s) f_1(s) ds \tag{8}$$

An intuitive interpretation of the resulting value is that it provides an estimate of the probability that a randomly chosen instance of class one is correctly rated or ranked higher by the classifier than a randomly selected instance of class zero (i.e., the probability that a churner is assigned a higher probability to churn than a non-churner). Note that since a pure random classification model yields an AUC equal to 0.5, a good classifier should result in a value of the AUC much larger than 0.5.

4.4.2. Maximum profit criterion

A second performance measure that will be applied in this study is the MP criterion. In order to compare the performance of different classification models, this measure calculates the maximum profit that can be generated with a retention campaign using the output of a classification model. The profit generated by a retention campaign is a function of the discriminatory power of a classification model, and can be calculated as [49]:

$$\Pi = N\alpha\{[\gamma CLV + \delta(1-\gamma)]\,\beta_0\lambda - \delta - c\} - A \tag{9}$$

With:

- Π = profit generated by a customer retention campaign,
- N = the number of customers in the customer base,
- α = the fraction of the customer base that is targeted in the retention campaign and offered an incentive to stay,
- β_0 = the fraction of all the operator's customers that will churn,
- λ = lift, i.e. how much more the fraction of customers included in the retention campaign is likely to churn than all the operator's customers. The lift indicates the predictive power of a classifier, and is a function of the included fraction of customers α with the highest probabilities to attrite, as indicated by the model. Lift can be calculated as the percentage of churners within the fraction α of customers, divided by β_0. Thus, $\lambda = 1$ means that the model provides essentially no predictive power because the targeted customers are no more likely to churn than the population as a whole.
- δ = the cost of the incentive to the firm when a customer accepts the offer and stays
- γ = the fraction of the targeted would-be churners who decide to remain because of the incentive (i.e. the success rate of the incentive),
- c = the cost of contacting a customer to offer him or her the incentive,
- CLV = the customer lifetime value (i.e., the net present value to the firm if the customer is retained), and
- A = the fixed administrative costs of running the churn management program.

Both the costs and the profits generated by a retention campaign are a function of the fraction α of included customers. Optimizing this fraction leads to the maximum profit that can be determined using the lift curve:

$$\text{MP} = \max_{\alpha}(\Pi) \tag{10}$$

Many studies on customer churn prediction modeling calculate the top-decile lift to compare the performance of classification models, i.e. the lift when including the ten percent of customers with the highest predicted probabilities to attrite. However, setting $\alpha = 10\%$ is a purely arbitrary choice, and including ten percent of the customers generally leads to suboptimal profits and model selection, as shown by Verbeke et al. [61]. Since the ultimate goal of a company when setting up a customer retention campaign is to minimize the costs associated with customer churn, it is logical to evaluate and select a customer churn prediction model by using the maximum profit that can be generated as a performance measure. Whereas the AUC measures the overall performance of a model, the MP criterion evaluates the prediction model at the optimal fraction of clients to include in a retention campaign. To calculate the MP, the values of the parameters CLV, γ, δ, and c in Equation 9 are taken equal to respectively €200, 0.30, €10 and €1 based on values reported in the literature [8,49] and information from telecom operators.

Table 4

Results of simulations evaluated with AUC and maximum profit criterion. The average rank (AR) which is highest is underlined and bold, ARs which are not significantly different at a significance level of 5% are in bold. Techniques different at a 1% significance level are in italic script, whereas techniques significantly different at 5% but not at 1% are in normal script

Technique	O1	D1	D2	D3	UCI	AR	O1	D1	D2	D3	UCI	AR	
Log. Regr.	0.75	0.76	0.66	0.65	0.86	**8.60**	0.12	0.17	0.01	0.00	4.84	**7.60**	No feature selection
Naive Bayes	0.72	0.77	0.59	0.61	0.83	12.20	0.14	0.17	0.00	0.01	4.22	**8.80**	
TAN	0.75	0.75	0.65	0.65	0.88	**8.60**	0.16	0.11	0.01	−0.00	4.78	**8.70**	
FAN-SB	0.75	0.76	0.66	0.65	0.87	**6.60**	0.16	0.15	0.01	−0.00	4.95	**7.60**	
FAN-LCV_LO	0.75	0.76	0.66	0.65	0.88	**6.40**	0.16	0.14	0.01	0.00	4.78	**6.90**	
SFAN-SB	0.75	0.76	0.65	0.65	0.87	**8.10**	0.16	0.16	0.01	−0.00	4.93	**5.90**	
SFAN-LCV_LO	0.75	0.76	0.66	0.65	0.88	**3.20**	0.15	0.16	0.01	0.00	4.76	**8.70**	
SFAND-SB	0.75	0.76	0.65	0.65	0.87	**7.90**	0.16	0.16	0.01	−0.00	4.93	**6.50**	
SFAND-LCV_LO	0.75	0.76	0.66	0.65	0.88	**3.20**	0.15	0.16	0.01	0.00	4.76	**7.70**	
STAN-SB	0.72	0.76	0.65	0.64	0.87	12.60	0.12	0.17	0.01	0.00	5.02	**5.60**	
STAN-LCV_LO	0.73	0.75	0.66	0.65	0.88	**9.40**	0.13	0.11	0.01	0.00	4.76	**10.90**	
STAND-SB	0.75	0.75	0.65	0.65	0.87	**8.60**	0.16	0.02	0.01	−0.00	4.94	**7.60**	
STAND-LCV_LO	0.75	0.75	0.66	0.65	0.88	**6.80**	0.15	0.02	0.01	−0.00	4.76	**10.70**	
K2	0.73	0.76	0.66	0.64	0.88	**7.20**	0.11	0.16	0.01	−0.00	4.90	**8.80**	
TPDA	0.67	0.76	0.62	0.63	0.83	*13.60*	0.02	0.15	0.00	0.00	4.75	**12.60**	
MMHC	0.66	0.76	0.62	0.62	0.85	13.00	0.02	0.16	0.00	0.00	4.78	**11.40**	
Log. Regr.	0.75	0.76	0.66	0.65	0.86	**8.40**	0.16	0.17	0.01	0.00	4.81	**6.40**	MB.05 feature selection
Naive Bayes	0.73	0.77	0.60	0.62	0.85	**12.00**	0.16	0.16	0.00	0.01	4.37	**7.00**	
TAN	0.75	0.75	0.65	0.65	0.88	**8.80**	0.14	0.11	0.01	−0.00	4.94	**10.20**	
FAN-SB	0.75	0.76	0.66	0.65	0.88	**6.40**	0.14	0.15	0.01	0.00	5.02	**7.00**	
FAN-LCV_LO	0.75	0.76	0.66	0.65	0.88	**6.60**	0.14	0.14	0.01	0.00	4.94	**8.40**	
SFAN-SB	0.75	0.76	0.65	0.65	0.88	**8.00**	0.14	0.16	0.01	0.00	5.02	**6.20**	
SFAN-LCV_LO	0.75	0.76	0.66	0.65	0.88	**5.20**	0.15	0.16	0.01	0.00	4.94	**6.80**	
SFAND-SB	0.75	0.76	0.65	0.65	0.88	**7.60**	0.14	0.16	0.01	0.00	5.02	**6.00**	
SFAND-LCV_LO	0.75	0.76	0.66	0.65	0.88	**3.60**	0.15	0.16	0.01	0.00	4.94	**7.60**	
STAN-SB	0.71	0.75	0.65	0.65	0.87	12.60	0.09	0.16	0.01	−0.00	5.06	**9.00**	
STAN-LCV_LO	0.75	0.75	0.66	0.65	0.88	**7.60**	0.15	0.11	0.01	−0.00	4.94	**10.20**	
STAND-SB	0.75	0.75	0.65	0.65	0.88	**8.00**	0.14	0.02	0.01	−0.00	4.94	**10.40**	
STAND-LCV_LO	0.75	0.75	0.66	0.65	0.88	**5.40**	0.15	0.02	0.01	−0.00	4.94	**9.80**	
K2	0.74	0.76	0.66	0.64	0.88	**9.80**	0.09	0.16	0.01	−0.00	4.98	**9.00**	
TPDA	0.67	0.75	0.61	0.64	0.83	*15.20*	0.02	0.16	0.00	0.00	4.75	**12.00**	
MMHC	0.64	0.76	0.62	0.62	0.88	**10.80**	0.02	0.16	0.00	−0.00	5.04	**10.00**	
Log. Regr.	0.74	0.76	0.66	0.65	0.86	**9.40**	0.14	0.16	0.01	0.00	4.81	**8.20**	MB.01 feature selection
Naive Bayes	0.73	0.76	0.60	0.63	0.85	**12.00**	0.13	0.17	0.00	0.01	4.37	**8.60**	
TAN	0.74	0.75	0.66	0.65	0.88	**8.40**	0.15	0.11	0.01	−0.00	4.94	**9.60**	
FAN-SB	0.74	0.76	0.66	0.65	0.88	**6.80**	0.15	0.15	0.01	0.00	5.02	**6.80**	
FAN-LCV_LO	0.74	0.76	0.66	0.65	0.88	**6.60**	0.15	0.14	0.01	0.00	4.94	**7.60**	
SFAN-SB	0.74	0.76	0.65	0.65	0.88	**8.40**	0.15	0.16	0.01	0.00	5.02	**3.00**	
SFAN-LCV_LO	0.75	0.76	0.66	0.65	0.88	**3.40**	0.13	0.16	0.01	0.00	4.94	**8.40**	
SFAND-SB	0.74	0.76	0.65	0.65	0.88	**8.00**	0.15	0.16	0.01	0.00	5.02	**3.40**	
SFAND-LCV_LO	0.75	0.76	0.66	0.65	0.88	**3.20**	0.13	0.16	0.01	0.00	4.94	**8.80**	
STAN-SB	0.72	0.75	0.65	0.64	0.87	12.80	0.10	0.17	0.01	−0.00	5.06	**8.40**	
STAN-LCV_LO	0.74	0.75	0.66	0.65	0.88	**8.00**	0.12	0.11	0.01	−0.00	4.94	**11.20**	
STAND-SB	0.74	0.75	0.65	0.65	0.88	**9.40**	0.15	0.02	0.01	−0.00	4.94	**9.20**	
STAND-LCV_LO	0.75	0.75	0.66	0.65	0.88	**5.40**	0.13	0.03	0.01	−0.00	4.94	**10.60**	
K2	0.74	0.76	0.66	0.64	0.88	**9.40**	0.13	0.15	0.01	−0.00	4.98	**8.60**	
TPDA	0.68	0.76	0.61	0.63	0.83	*14.00*	0.01	0.15	−0.00	0.00	4.75	13.00	
MMHC	0.67	0.76	0.61	0.62	0.88	**10.80**	0.00	0.16	0.00	−0.00	5.04	**10.60**	
			AUC						Maximum profit				

4.5. Testing statistical significance

The aim of this study is to investigate how classification performance is affected by two specific factors, the type of Bayesian Network classifier and the use of Markov Blanket feature selection. A procedure described in Demšar [21] is followed to statistically test the results of the benchmarking experiments and contrast the levels of the factors. In a first step of this procedure the non-parametric Friedman test [26] is performed to check whether differences in performance are due to chance. The Friedman statistic is defined as:

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \tag{11}$$

with R_j the average rank of algorithm $j = 1, 2, \ldots, k$ over N data sets. Under the null hypothesis that no significant differences exist, the Friedman statistic is distributed according to χ_F^2 with $k - 1$ degrees of freedom, at least when N and k are big enough (e.g. Lehman and D'Abrera [42] give $k \cdot N > 30$ as criterion). This requirement is fulfilled in this study when comparing different classifiers ($N = 5 \cdot 3 = 15$ and $k = 16$) and when analyzing the impact of feature selection ($N = 5 \cdot 16 = 80$ and $k = 3$).

If the null hypothesis is rejected by the Friedman test, we proceed by performing the post-hoc Nemenyi test [48] to compare all classifiers to each other. Two classifiers yield significantly different results if their average ranks differ by at least the critical difference equal to:

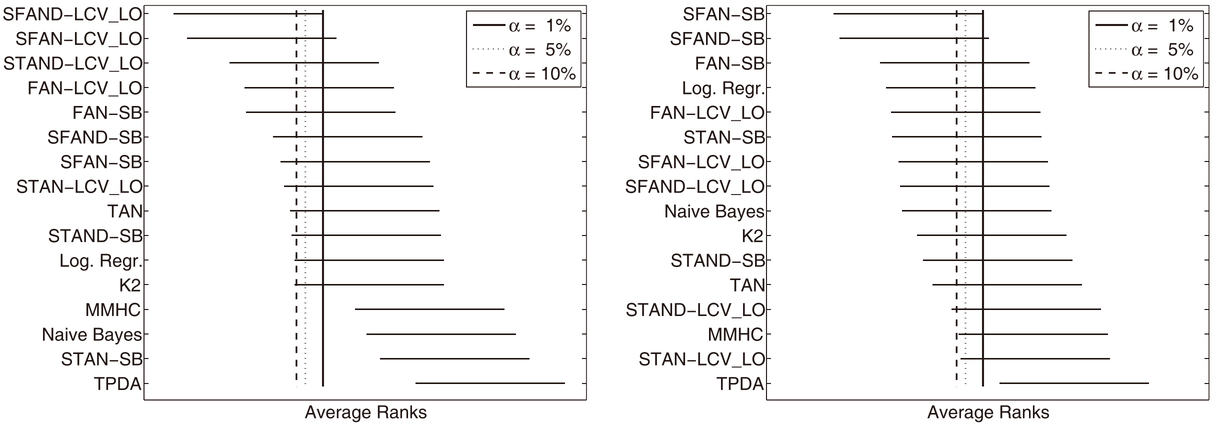
$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{12}$$

with critical values q_α based on the Studentized range statistic divided by $\sqrt{2}$. To compare all classifiers with the best performing classifier the Bonferroni-Dunn test [23] is applied, which is similar to post-hoc Nemenyi but adjusts the confidence level in order to control the family-wise error for making $k - 1$ instead of $k(k - 1)/2$ comparisons.

5. Discussion of results

The results of the experiments are reported in Table 1. The left panel shows the AUC whereas the right panel shows the maximum profit criterion (MP), for each classifier-feature selection combination. The table displays the resulting measure for the five data sets discussed in Section 4. Moreover, the column labeled *AR* reports the average rank of the classifier over all five data sets. To give an indication about the significance of the results, the following notational convention has been adopted. The score or average rank of the best performing classifier is always underlined. The average rank of a classifier is bold if the performance is not significantly different from the best performing technique at a significance level of 5%. Techniques which are different at a 1% significance level are in italic script, whereas a classifier differing at a 5% but not at a 1% level is in normal script. The Bonferroni-Dunn test (see Section 4.5) has been used to compute the statistical significance indicated in Table 4.

In what follows, the impact of MB feature selection and the performance of the Bayesian Network classifiers will be analyzed. Next to AUC, the Maximum Profit criterion has been used to measure classification performance. The main conclusions will be based on the MP, since it gives a more accurate assessment of classification performance in a practical churn setting. After all, an operator is mostly interested in the fraction of the client base which will maximize his profit, i.e. those customers with a high churn probability. Typically, only a small proportion of all customers will be included in a retention campaign.



(a) AUC (b) Maximum profit

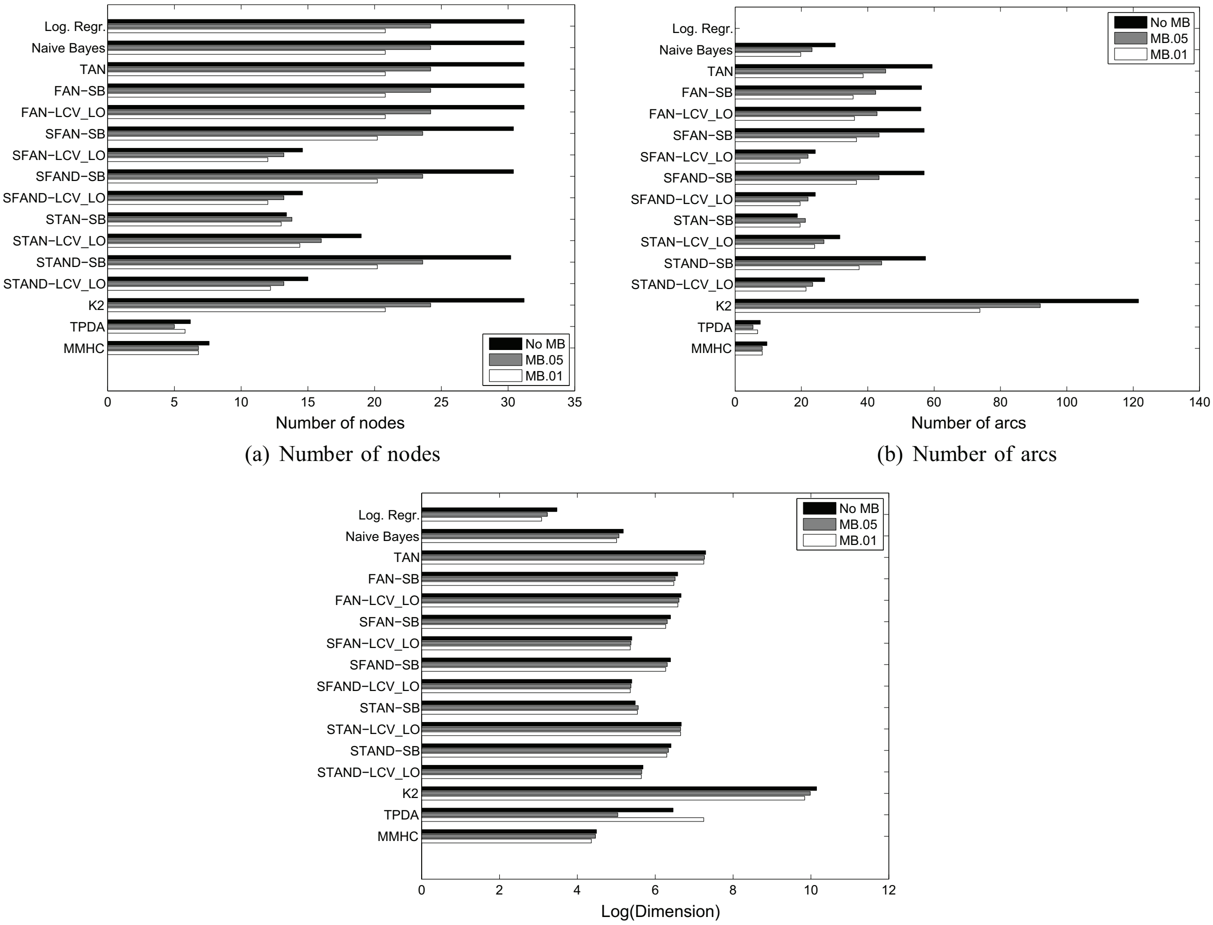
Fig. 4. Post-hoc Nemenyi test at a significance level of 1%. The dotted and dashed line indicate the cutoff at a significance level of 5% and 10% respectively.

5.1. Classification performance

To analyze the impact of feature selection on classification performance, a Friedman test has been employed. When using AUC as performance measure, the p-value is 0.24, for MP it equals 0.85, both unambiguously implying that the feature selection does not significantly affect performance. This is positive, as it gives the opportunity to reduce the number of variables without a significant decrease in the predictive power of the resulting models. The complexity of the resulting models will be discussed in the next subsection.

The Friedman test is also applied to investigate whether the differences among classifiers are significant. The p-value with regards to the AUC measure is $6.53{\cdot}10^{-13}$ and for MP it equals $5.77{\cdot}10^{-4}$, indicating that the differences among classifiers are statistically significant. Hence, a post-hoc Nemenyi test is performed. The outcome of this test is graphically illustrated for both performance metrics in Figs 4(a) and (b). The left end of the line segments indicate the average rank whereas the length is equal to the critical distance at a 1% level of significance, enabling the reader to compare techniques among each other, not only with the best method.[5] The vertical full line gives the cutoff for the 1% level, the dotted and dashed line for the 5% and 10% level whereas the critical distance is equal to 6.75, 5.96, and 5.56 for the the 1%, 5%, and 10% significance levels respectively. One can observe that AUC and MP lead to different rankings, although most of the techniques are not significantly different. AUC is more discriminative as opposed to MP, following from the fact that AUC is measuring classifier performance over the whole output range, whereas MP only looks at the optimal fraction of clients to include in a retention campaign and assesses prediction performance at this specific point. Thus, when using MP to evaluate the classifiers for practical purposes, i.e. application in a real life churn setting, it is likely that none of the Bayesian Networks significantly outperforms the others, except for TPDA. This is an interesting conclusion, since it indicates that General Bayesian Network methods (except for TPDA) could be used, which are preferable as they lead to more compact and interpretable networks. It is also remarkable that the BN classifiers are not able to outperform logistic regression, a straightforward and fast technique.

[5] Note that the Nemenyi test is designed to allow comparisons between each pair of techniques, i.e. to make $k(k-1)/2$ comparisons.



(a) Number of nodes

(b) Number of arcs

(c) Logarithm of the network dimension

Fig. 5. Complexity of the Bayesian networks.

5.2. *Complexity and interpretability of the Bayesian networks*

The discussion above focussed on the performance of the classifiers, which is only one aspect of a good churn prediction model. Also the complexity and interpretability of the resulting classifiers are key properties, since comprehensibility of the model is important as discussed in Section 2. Bayesian networks are appealing to practitioners, as they give an intuitive insight in the factors driving churn behavior and the dependencies among those factors. This applies less to the Naive Bayes classifier and the Augmented Naive Bayes classifiers, since they include, by nature, all or many variables and prohibit general network structures. General Bayesian networks, on the other hand, are more flexible and typically use less variables. As a result, Markov Blanket feature selection will be more useful in combination with (Augmented) Naive Bayes classifiers, since these do not contain a mechanism to get rid of redundant variables.

Figure 5 illustrates the network complexity by showing the network dimension and the number of nodes and arcs for each algorithm-feature selection combination, averaged over the data sets. The dimension of a Bayesian Network is defined as the number of free parameters needed to fully specify the

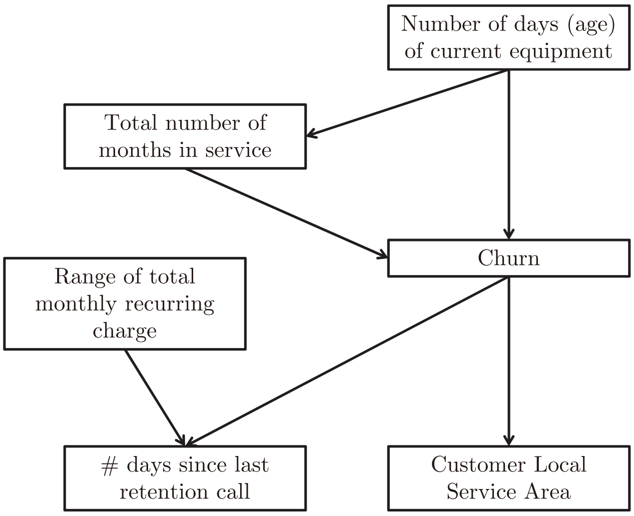

Number of days (age) of current equipment
Total number of months in service
Churn
Range of total monthly recurring charge
# days since last retention call
Customer Local Service Area

Fig. 6. Bayesian network for data set D1, created with MMHC without prior feature selection.

joint probability distribution encoded by the network, and is calculated as:

$$DIM = \sum_{i=1}^{n} (r_i - 1) \cdot q_i \tag{13}$$

with r_i being the cardinality of variable X_i and:

$$q_i = \prod_{X_j \in \Pi_{X_i}} r_j \tag{14}$$

with Π_{X_i} the direct parent set for node X_i. For logistic regression, which is included in the study as a benchmark algorithm, the number of nodes is equal to the number of attributes, the dimension (or number of free parameters) equals the number of attributes plus one, and the number of arcs is meaningless and therefore discarded for this algorithm.

The graphs show that feature selection reduces the number of nodes, arcs and dimension, as one would expect. The reduction of nodes and arcs is more substantial for the (Augmented) Naive Bayes networks, as mentioned before. An exception could be noticed for the TPDA algorithm where the complexity again increases for MB.01 after having dropped for MB.05. This could be attributed to the fact that the MB feature selection algorithm excluded a variable which had large explanatory power within the TPDA framework and in order to offset this loss, the algorithm has to include more variables than before. This case illustrates that one should be careful when applying feature selection prior to the use of a GBN algorithm. This category of methods already reduces the number of parameters by itself, so reducing the variables prior to training the classifier might be redundant at best or even worsen the result.

Moreover, one can observe that the Augmented Naive Bayes classifiers, which relaxed the TAN assumption, are able to reduce the number of nodes and arcs compared to TAN, without a loss in predictive power. Nevertheless, the networks are still too complex to be easily interpretable. The GBN algorithms reduce the complexity even more and contain around 5 to 7 variables, with the only exception of K2, creating very dense networks.[6] Figure 6 shows the network created for data set D2 by the MMHC algorithm (without prior input selection). D2 is a real life data set from an operator and contains a high number of variables. Nonetheless, the algorithm is able to withhold only 5 attributes to predict churn behavior. When looking at the network, it is very important to realize that the arcs do not necessarily imply causality, but they should rather be interpreted as correlation between the variables. For this network, one can observe that for instance the age of the current handset is correlated with the number of months in service and with churn behavior. The former relation could be explained by the fact that many operators offer a new mobile when signing a contract, whereas the latter could point to a motive for changing mobile phone operator, i.e. a promotional action at another operator. Such relations could be helpful for a company to identify red flags or warning signs for churn behavior. Moreover, it allows to check whether a churn prediction model is in line with current domain knowledge, increasing the credibility and applicability of those models.

6. Conclusions

Customer churn is becoming an increasingly important business analytics problem for telecom operators. In order to increase the efficiency of customer retention campaigns, operators employ customer churn prediction models based on data mining techniques. These prediction models need to be accurate as well as compact and interpretable. This study investigates whether Bayesian Network techniques are appropriate for customer churn prediction.

In this paper, classification performance is measured with the Area under the Receiver Operating Characteristic Curve (AUC) and the Maximum Profit (MP) criterion. The results show that both performance measures lead to a different ranking of classification algorithms, even though not always statistically significant, and that AUC is more discriminative than the MP criterion. Whereas AUC measures performance over the whole customer base, the MP criterion only focusses on the optimal fraction of customers in order to maximize the effectiveness of a retention campaign. In a real life context, the MP criterion is preferred as it will maximize the profit for a telecom operator and therefore, the main conclusions will be based on this performance measure.

The results of the experiments show that Bayesian Network classifiers are not able to outperform traditional logistic regression. However, their contribution lies in the fact that they offer a very intuitive insight into the dependencies among the explanatory variables. The study indicates that Augmented Naive Bayes methods do not lead to compact networks, whereas General Bayesian Network algorithms result in simple and interpretable networks. This may aid practitioners in understanding the drivers behind churn behavior and in identifying warning signs for customer churn. Moreover, the Max-Min Hill-Climbing (MMHC) algorithm was able to create a compact and comprehensible model without a statistically significant loss in classification performance, as compared to logistic regression and Augmented Naive Bayes techniques.

Furthermore, the impact of Markov Blanket (MB) feature selection was tested. The outcome indicates that a reduction of variables as a result of MB feature selection does not decrease the performance significantly. Especially for Augmented Naive Bayes networks it proves to be useful, as it decreases the number of attributes substantially. For General Bayesian Network classifiers on the other hand, MB feature selection is discouraged as the General Bayesian Network algorithms themselves already limit the network complexity, making a prior input selection redundant.

[6] In the K2 algorithm, it is possible to limit the number of parents for each node. As the goal was to test this algorithm as a GBN, this restriction was not imposed, resulting in very dense networks.

Acknowledgements

We extend our gratitude to the Flemish Research Council for financial support (FWO postdoctoral research grant, Odysseus grant B.0915.09), and the National Bank of Belgium (NBB/10/006).

References

[1] S. Ali and K.A. Smith, On learning algorithm selection for classification, *Applied Soft Computing* **6**(2) (2006), 119–138.

[2] C.F. Aliferis, I. Tsamardinos and A. Statnikov, Causal explorer: Causal probabilistic network learning toolkit for biomedical discovery, http://www.dsl-lab.org/causal_explorer, 2003.

[3] C.F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani and X.D. Koutsoukos, Local causal and markov blanket induction for causal discovery and feature selection for classification Part I: Algorithms and empirical evaluation, *The Journal of Machine Learning Research* **11** (2010), 171–234.

[4] C.F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani and X.D. Koutsoukos, Local causal and markov blanket induction for causal discovery and feature selection for classification part II: Analysis and extensions, *The Journal of Machine Learning Research* **11** (2010), 235–284.

[5] A.D. Athanassopoulos, Customer satisfaction cues to support market segmentation and explain switching behavior, *Journal of Business Research* **47**(3) (2000), 191–207.

[6] W.H. Au, K.C.C. Chan and X. Yao, A novel evolutionary data mining algorithm with applications to churn prediction, *IEEE Transactions on Evolutionary Computation* **7**(6) (2003), 532–545.

[7] C.B. Bhattacharya, When customers are members: Customer retention in paid membership contexts, *Journal of the Academy of Marketing Science* **26**(1) (1998), 31–44.

[8] J. Burez and D. Van den Poel, CRM at a pay–TV company: Using analytical models to reduce customer attrition by targeted marketing for subscription services, *Expert Systems with Applications* **32** (2007), 277–288.

[9] J. Burez and D. Van den Poel, Handling class imbalance in customer churn prediction, *Expert Systems with Applications* **36**(3) (2009), 4626–4636.

[10] J. Cheng, Powerpredictor system, http://www.cs.ualberta.ca/jcheng/bnpp.htm, 2000.

[11] J. Cheng and R. Greiner, Comparing bayesian network classifiers, in: *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, Stockholm, Sweden, (1999), 101–108.

[12] J. Cheng, D.A. Bell and W. Liu, An algorithm for bayesian belief network construction from data, in: *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics (AI and STAT)*, Fort Lauderdale, Florida, U.S., (1997), 83–90.

[13] J. Cheng, R. Greiner, J. Kelly, W. Liu and D. Bell, Learning bayesian networks from data: An information-theory based approach, *Artificial Intelligence* **137** (2002), 43–90.

[14] D.M. Chickering, Learning bayesian networks is np-complete, in: *Learning from Data: Artificial Intelligence and Statistics*, D.E. Holmes and C.J. Lakhmi, eds, Springer-Verlag, 1996, pp. 121–130.

[15] D.M. Chickering, Optimal structure identification with greedy search, *Journal of Machine Learning Research* **3** (2002), 507–554.

[16] C.K. Chow and C.N. Liu, Approximating discrete probability distributions with dependence trees, *IEEE Transactions on Information Theory* **14** (1968), 462–467.

[17] M. Colgate and P.J. Danaher, Implementing a customer relationship strategy: The assymetric impact of poor versus excellent execution, *Journal of the Academy of Marketing Science* **28**(3) (2000), 375–387.

[18] M. Colgate, K. Stewart and R. Kinsella, Customer defection: A study of the student market in Ireland, *International Journal of Bank Marketing* **14**(3) (1996), 23–29.

[19] G.F Cooper and E. Herskovits, A bayesian method for the induction of probabilistic networks from data, *Machine Learning* **9** (1992), 309–347.

[20] P. Datta, B. Masand, D.R. Mani and B. Li, Automated cellular modeling and prediction on a large scale, *Artificial Intelligence Review* **14** (2000), 485–502.

[21] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research* **7** (2006), 1–30.

[22] R.O. Duda and P.E. Hart, Pattern classification and scene analysis, John Wiley, New York, 1973.

[23] O.J. Dunn, Multiple comparisons among means, *Journal of the American Statistical Association* **56** (1961), 52–64.

[24] A. Famili, W.M. Shen, R. Weber and E. Simoudis, Data pre-processing and intelligent data analysis, *International Journal on Intelligent Data Analysis* **1**(1) (1997), 1–2.

[25] U.M. Fayyad and K.B. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, in:

Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI), Chambéry, France, 1993. Morgan Kaufmann, 1022–1029.

[26] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, *Annals of Mathematical Statistics* **11** (1940), 86–92.

[27] N. Friedman, D. Geiger and M. Goldszmidt, Bayesian network classifiers, *Machine Learning* **29** (1997), 131–163.

[28] N. Friedman, I. Nachman and D. Peer, Learning bayesian network structure from massive datasets: The sparse candidate algorithm, in: *Fifteenth Conference on Uncertainty in Artificial Intelligence* (1999), 206–215.

[29] J. Ganesh, M.J. Arnold and K.E. Reynolds, Understanding the customer base of service providers: An examination of the differences between switchers and stayers, *Journal of Marketing* **64**(3) (2000), 65–87.

[30] D. Geiger, T.S. Verma and J. Pearl, Identifying independence in bayesian networks, *Networks* **20**(5) (1990), 507–534.

[31] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, *The Journal of Machine Learning Research* **3** (2003), 1157–1182.

[32] D. Heckerman, D. Geiger and D. Chickering, Learning bayesian networks: The combination of knowledge and statitical data, *Machine Learning* **20** (1995), 194–243.

[33] D. Heckerman, C. Meek and G. Cooper, A bayesian approach to causal discovery, in: *Computation, Causation and Discovery*, C.N. Glymour and G.F. Cooper, eds, MIT Press, 1999, pp. 141–165.

[34] S.Y. Hung, D.C. Yen and H.Y. Wang, Applying data mining to telecom churn management, *Expert Systems with Applications* **31** (2006), 515–524.

[35] G.H. John and P. Langley, Estimating continuous distributions in bayesian classifiers. in: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI)*, Montreal, Québec, Canada, Morgan Kaufmann, (1995), 338–345.

[36] I. Kononenko, Semi-naive bayesian classifier, in: *Proceedings Sixth European Working Session on Learning*, Y. Kodratoff, ed., Berlin: Springer Verlag, 1991, pp. 206–219.

[37] S.B. Kotsiantis, Supervised machine learning: A review of classification techniques, *Informatica* **31**(3) (2007), 249–268.

[38] W.J. Krzanowski and D.J. Hand, *ROC curves for continuous data*. CRC/Chapman and Hall, (2009).

[39] P. Langley and S. Sage, Induction of selective bayesian classifiers, in: *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, D. Poole and R. Lopez de Mantaras, eds, Advances in Data Mining in Marketing, San Francisco CA: Morgan Kaufmann, (1994), 399–406

[40] P. Langley, W. Iba and K. Thompson, An analysis of bayesian classifiers, in: *The Tenth National Conference on Artificial Intelligence* San Jose, AAAI Press, (1992), 223–228.

[41] S.L. Lauritzen, Graphical models, Oxford: Clarendon Press, 1996.

[42] E.L. Lehman and H.J.M. D'Abrera, Nonparametrics-statistical methods based on ranks, *Calif: Holden & Day*, San Franciso, (1975), 123–141.

[43] A. Lemmens and C. Croux, Bagging and boosting classification trees to predict churn, *Journal of Marketing Research* **43**(2) (2006), 276–286.

[44] E. Lima, C. Mues and B. Baesens, Domain knowledge integration in data mining using decision tables: Case studies in churn prediction, *Journal of the Operational Research Society* **60**(8) (2009), 1096–1106.

[45] D. Martens, J. Vanthienen, W. Verbeke and B. Baesens, Performance of classification models from a user perspective, *Accepted for publication in Decision Support Systems* (2010).

[46] R.W. Mizerski, An attribution explanation of the disproportionate influence of unfavourable information, *Journal of Consumer Research* (9 December 1982), 301–310.

[47] K. Murphy, The bayes net matlab toolbox, http://code.google.com/p/bnt/, 2001.

[48] P.B. Nemenyi, *Distribution-free multiple comparisons*, PhD thesis, Princeton University, 1963.

[49] S. Neslin, S. Gupta, W. Kamakura, J. Lu and C. Mason, Detection defection: Measuring and understanding the predictive accuracy of customer churn models, *Journal of Marketing Research* **43**(2) (2006), 204–211.

[50] M. Paulin, J. Perrien, R.J. Ferguson, A.M.A. Salazar and L.M. Seruya, Relational norms and client retention: External effectiveness of commercial banking in Canada and Mexico, *International Journal of Bank Marketing* **16**(1) (1998), 24–31.

[51] J. Pearl, Probabilistic reasoning in Intelligent Systems: networks for plausible inference, Morgan Kaufmann, 1988.

[52] E. Rasmusson, Complaints can build relationships, *Sales and Marketing Management* **151**(9) (1999), 89–90.

[53] F.F. Reichheld, Learning from customer defections, *Harvard Business Review* **74**(2) (1996), 56–69.

[54] R.T. Rust and A.J. Zahorik, Customer satisfaction, customer retention and market share, *Journal of Retailing* **69**(2) (1993), 193–215.

[55] J.P. Sacha, *New Synthesis of Bayesian Network Classifiers and Cardiac SPECT Image Interpretation*, PhD thesis, University of Toledo, 1999.

[56] J.P. Sacha, Bayesian network classifier toolbox, http://jbnc.sourceforge.net/#jBNC-WEKA, 1999.

[57] P. Spirtes, C.N. Glymour and R. Scheines, Causation, prediction and search, The MIT Press, 2000.

[58] D. Stum and A. Thiry, Building customer loyalty, *Training and Development Journal* **45**(4) (1991), 34–36.

[59] I. Tsamardinos, L.E. Brown and C.F. Aliferis, The max-min hill-climbing bayesian netowrk structure learning algorithm, *Machine Learning* **65**(1) (2006), 31–78.

[60] D. Van den Poel and B. Larivière, Customer attrition analysis for financial services using proportional hazard models, *European Journal of Operational Research* **157**(1) (2004), 196–217.

[61] W. Verbeke, K. Dejaeger, D. Martens, J. Hur and B. Baesens, New insights into churn prediction in the telecommunication sector: A profit driven data mining approach, *European Journal of Operational Research*, In press, 2011.

[62] W. Verbeke, D. Martens, C. Mues and B. Baesens, Building comprehensible customer churn prediction models with advanced rule induction techniques, *Expert Systems with Applications* **38** (2011), 2354–2364.

[63] T. Verma and J. Pearl, Causal networks: Semantics and expressiveness, in: *Proceedings of the Fourth Workshop on Uncertainty in Artificial Intelligence*, Mountain View, CA, U.S., (1988), 352–359.

[64] C.P. Wei and I.T. Chiu, Turning telecommunications call details to churn prediction: A data mining approach, *Expert Systems with Applications* **23** (2002), 103–112.

[65] I.H. Witten and E. Frank, *Data mining: practical machine learning tools and techniques with Java implementations*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[66] V.A. Zeithaml, L.L. Berry and A. Parasuraman, The behavioural consequences of service quality, *Journal of Marketing* **60**(2) (1996), 31–46.