International Journal of Approximate Reasoning 51 (2010) 695–717

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

Feature selection for Bayesian network classifiers using the MDL-FS score

Mădălina M. Drugan [a,*], Marco A. Wiering [b]

[a] Department of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
[b] Department of Artificial Intelligence, University of Groningen, 9700 AK Groningen, The Netherlands

ARTICLE INFO

Article history:
Received 12 October 2007
Received in revised form 3 February 2010
Accepted 3 February 2010
Available online 18 February 2010

Keywords:
Feature subset selection
Minimum Description Length
Selective Bayesian classifiers
Tree augumented networks

ABSTRACT

When constructing a Bayesian network classifier from data, the more or less redundant features included in a dataset may bias the classifier and as a consequence may result in a relatively poor classification accuracy. In this paper, we study the problem of selecting appropriate subsets of features for such classifiers. To this end, we propose a new definition of the concept of redundancy in noisy data. For comparing alternative classifiers, we use the Minimum Description Length for Feature Selection (MDL-FS) function that we introduced before. Our function differs from the well-known MDL function in that it captures a classifier's conditional log-likelihood. We show that the MDL-FS function serves to identify redundancy at different levels and is able to eliminate redundant features from different types of classifier. We support our theoretical findings by comparing the feature-selection behaviours of the various functions in a practical setting. Our results indicate that the MDL-FS function is more suited to the task of feature selection than MDL as it often yields classifiers of equal or better performance with significantly fewer attributes.

1. Introduction

Many real-life problems, such as medical diagnosis and troubleshooting of technical equipment, can be viewed as a classification problem, where an instance described by a number of features has to be classified in one of several distinct predefined classes. For many of these classification problems, instances of every-day problem solving are recorded in a dataset. Such a dataset often includes more features, or attributes, of the problem's instances than are strictly necessary for the classification task at hand. When constructing a classifier from the dataset, these more or less redundant features may bias the classifier and as a consequence may result in a relatively poor classification accuracy. By constructing the classifier over just a subset of the features, a less complex classifier is yielded that tends to have a better generalisation performance [1]. Finding a minimum subset of features such that the selective classifier constructed over this subset is optimal for a given performance measure, is known as the *feature subset selection* problem [2–4]. The feature subset selection problem unfortunately is NP-hard in general [5–8].

We begin by providing a new definition of the concept of redundancy of attributes, where the redundancy is viewed within some allowed amount of noise in the data under study. It allows us to study feature selection for different types of Bayesian network classifier more specifically. With our definition we distinguish between different *levels of redundancy* for an attribute. The levels depend on the cardinality of the (sub)sets of other attributes with which the attribute is combined so that the attribute is not useful for the classification task. We will argue that these levels of redundancy provide for relating the problem of feature subset selection to the types of dependence that can be expressed by a Bayesian network classifier. By allowing noise for the various levels, our concept of redundancy provides for studying feature selection in a practical setting.

* Corresponding author.
E-mail address: M.M.Drugan@uu.nl (M.M. Drugan).

0888-613X/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2010.02.001

For constructing a selective classifier, generally a heuristic algorithm [9] is used that searches the space of possible models for classifiers of high quality. Because of its simplicity, its intuitive theoretical foundation and its associated ease of computation, the MDL function and its variants [10] have become quite popular as quality measures for constructing Bayesian networks from data, and in fact for constructing Bayesian network classifiers [11]. The function in essence weighs the complexity of a model against its ability to capture the observed probability distribution. While the MDL function and its variants are accepted as suitable functions for comparing the qualities of alternative Bayesian networks, they are not without criticism when constructing Bayesian network classifiers. The criticism focuses on the observation that the functions capture a joint probability distribution over the variables of a classifier, while it is the conditional distribution over the class variable given the attributes that is of interest for the classification task [11–17].

For comparing the qualities of alternative classifiers, we propose the *Minimum Description Length for Feature Selection* (MDL-FS) function [18]. The MDL-FS function is closely related to the well-known Minimum Description Length (MDL) function. It differs from the MDL function only in that it encodes the conditional probability distribution over the class variable given the various attributes. Upon using the function as a measure for comparing the qualities of Bayesian network classifiers therefore, this conditional distribution has to be learned from the available data. Unfortunately, learning a conditional distribution is generally acknowledged to be hard [19–21], since it does not decompose over the graphical structure of a Bayesian network classifier as does the joint distribution. Our MDL-FS function approximates the conditional distribution by means of an auxiliary Bayesian network which captures the strongest relationships between the attributes. With the function, both the structure of the Bayesian network classifier over all variables involved and the structure of the auxiliary network over the attributes are learned using a less demanding generative method. The conditional log-likelihood of the classifier then is approximated by the difference between the unconditional log-likelihood of the classifier and the log-likelihood of the auxiliary network.

This paper is organised as follows: In Section 2 we provide some background on Bayesian networks and on Bayesian network classifiers more specifically; we further review the MDL function and present our notational conventions. In Section 3 we introduce the problem of feature subset selection and provide a formal definition of the concept of redundancy. We introduce our new MDL-FS function and study its relationship with the MDL function in Section 4. In Section 5, we investigate the feature-selection behaviour of the MDL-FS function in general and we compare it with the behaviour of the MDL function. In Section 6 we study the use of the MDL-FS function in constructing selective Naïve Bayes and TAN classifiers from data. In Section 7 the feature-selection behaviour of the MDL-FS and MDL functions and other state of the art feature selection algorithms are compared in a practical setting. Our results indicate that the MDL-FS function indeed is more suited to the task of feature subset selection than the MDL function or other feature selection algorithms as it yields classifiers of comparably good or even significantly better performance with fewer attributes. The paper ends with our concluding observations and remarks in Section 8.

2. Background

In this section, we provide some preliminaries on Bayesian networks and on Bayesian network classifiers more specifically. We conclude this section with a discussion of the MDL function.

2.1. Bayesian networks and Bayesian network classifiers

We consider a set V of stochastic variables $V_i,\ i = 1, \ldots, n,\ n \geqslant 1$. We use $\Omega(V_i)$ to denote the set of all possible (discrete) values of the variable V_i; for ease of exposition, we assume a total ordering on the set $\Omega(V_i)$ and use v_i^k to denote the kth value of V_i. For any subset of variables $S \subseteq V$, we use $\Omega(S) = \times_{V_i \in S}\Omega(V_i)$ to denote the set of all joint value assignments to S. A *Bayesian network* over V now is a tuple $\mathcal{B} = (G, P)$ where G is a directed acyclic graph and P is a set of conditional probability distributions. In the digraph G, each vertex models a stochastic variable from V. The set of arcs captures probabilistic independence: for a topological sort of the digraph G, that is, for an ordering $V_1, \ldots, V_n$, $n \geqslant 1$, of its variables with $i < j$ for every arc $V_i \to V_j$ in G, we have that any variable V_i is independent of the preceding variables $V_1, \ldots, V_{i-1}$ given its parents in the graphical structure. Associated with the digraph is a set P of probability distributions: for each variable V_i are specified the conditional distributions $P(V_i|p(V_i))$ that describe the influence of the various assignments to the variable's parents $p(V_i)$ on the probabilities of the values of V_i itself. The network defines a unique joint probability distribution $P(V)$ over its variables with

$$P(V) = \prod_{V_i \in V} P(V_i|p(V_i))$$

Note that the thus defined probability distribution factorises over the network's digraph into separate conditional distributions. Bayesian network classifiers are Bayesian networks of restricted topology that are tailored to solving classification problems. In a classification problem, instances described by a number of features have to be classified in one of several distinct predefined classes. We consider to this end a set A of stochastic variables A_i, called *attributes*, that are used to describe the features of the instances. We further have a designated variable C, called the *class variable*, that captures the various possible classes. *Bayesian network classifiers* now are defined over the set of variables $A \cup \{C\}$. Like a Bayesian network in general, they include a graphical structure that captures a probabilistic independence relation among the variables involved, and represent a joint probability distribution that is factorised in terms of this graphical structure.

A *Naive Bayes classifier* over $A \cup \{C\}$ has for its graphical representation a tree-like structure with the variables $A \cup \{C\}$ for its nodes. The class variable is the root of the tree and each attribute has the class variable for its unique parent. The graphical structure of the classifier models the assumption that all attributes $A_i \in A$ are mutually independent given the class variable. The joint probability distribution $P(C,A)$ defined by the classifier now equals $P(C,A) = P(C) \cdot \prod_{A_i \in A} P(A_i|C)$. A *Tree Augmented Network (TAN) classifier* [11] over $A \cup \{C\}$ has for its graphical representation a directed acyclic graph in which the class variable is the unique root and in which each attribute has the class variable and at most one other attribute for its parents. The subgraph induced by the set of attributes, moreover, is a directed tree, termed the *attribute tree* of the classifier. The joint probability distribution that is defined by the classifier equals $P(C,A) = P(C) \cdot \prod_{A_i \in A} P(A_i|p(A_i))$.

2.2. Learning Bayesian network classifiers

Bayesian network classifiers are typically constructed from a dataset in which instances of every-day problem solving have been recorded along with their associated classes. A *labelled instance* is composed of a set of attributes A and an associated class value; it thus is an element of $\Omega(A \cup \{C\})$. For the learning task, we consider a *dataset* D with $N \geqslant 1$ labelled instances over $A \cup \{C\}$. With D, we associate the *counting function* $N_D : \cup_{S \subseteq A \cup \{C\}} \Omega(S) \to \mathbb{N}$ that associates with each value assignment s^k to S, the number of instances in D for which $S = s^k$; for $S = \varnothing$, we take the function value of N_D to be equal to N. The dataset D now induces a joint probability distribution $\widehat{P}_D(C,A)$, termed the *observed* distribution, over $A \cup \{C\}$, with $\widehat{P}_D(c^g, a^k) = N(c^g, a^k)/N$, for all values c^g of C and all value assignments a^k to A. In the sequel, we will omit the subscript D from the counting function N_D and from the observed distribution $\widehat{P}_D$ as long as ambiguity cannot occur. Learning a classifier from the dataset now amounts to selecting a classifier, from among a specific family of classifiers, that approximates the observed data. We assume that there might be noise – by means of any errors that interfere in the relationships between class and attributes – in the dataset D. For a review of the impact of the noise over the class variable and the attributes in a dataset we refer to Zhu and Wu [22]. For comparing alternative classifiers various different quality measures are in use. In this section, we review the measures that we will use throughout the paper. Before doing so, we briefly review the basic concepts of entropy; for a more elaborate introduction, we refer the reader to any textbook on information theory.

2.2.1. Entropy

The concept of *entropy* originates from information theory and describes the expected amount of information that is required to establish the value of a stochastic variable, or set of stochastic variables, to certainty. For an overview of these concepts we refer to Shannon [23]. We consider a set of variables X and a joint distribution P over X. The entropy $H_P(X)$ of X in P is defined as

$$H_P(X) = -\sum_{x^i \in \Omega(X)} P(x^i) \cdot \log P(x^i)$$

where log indicates a logarithm to the base 2 and $0 \cdot \log 0$ is taken to be equal to 0. The entropy function attains its *maximum* value for a uniform probability distribution over X. The larger the set $\Omega(X)$ of possible value assignments to X, the larger the maximum attained is; for a binary variable, for example, the maximum equals 1.00, while for a variable with $|\Omega(X)|$ possible values, the maximum entropy is $\log|\Omega(X)|$. The function further attains its *minimum* value for any degenerate distribution P over X with $P(x^j) = 1$ for some value assignment $x^j \in \Omega(X)$ and $P(x^i) = 0$ for all other assignments $x^i \in \Omega(X),\ i \neq j$. The minimum value equals 0, indicating that there is no uncertainty left as to the true value of X.

We now consider a set of stochastic variables $X \cup Y$ and a joint probability distribution P over $X \cup Y$. The amount of uncertainty as to the true value of X that is expected to remain after observing a value assignment for Y, is captured by the *conditional entropy* $H_P(X|Y)$ of X given Y in P; it is defined as

$$H_P(X|Y) = -\sum_{x^i \in \Omega(X), y^j \in \Omega(Y)} P(x^i, y^j) \cdot \log P(x^i|y^j)$$

The entropy of the set of variables X is never expected to increase by observing a value assignment for the set Y, that is, $H_P(X|Y) \leqslant H_P(X)$, for any (disjoint) sets X, Y; if the sets X and Y are independent in the probability distribution under consideration, then $H_P(X|Y) = H_P(X)$. More in general, for any sets Y, Z with $Z \cap Y = \emptyset$, we have that $H_P(X|Y,Z) \leqslant H_P(X|Y)$.

2.2.2. Quality measures

The main purpose in constructing a Bayesian network is to approximate, as well as possible, the unknown true joint probability distribution P over the variables involved. Upon constructing the network from data, for this purpose only an observed distribution $\widehat{P}$ is available. Alternative networks then are compared by means of a *quality measure* that serves to express how well the represented distribution explains the data. The most commonly used quality measure is the *log-likelihood* measure that assigns to a network, given a particular dataset, a numerical value that is proportional to the probability of the dataset being generated by the joint probability distribution represented by the network. The log-likelihood of a network $\mathcal{B}$ given a dataset D is defined more formally as

$$LL(\mathcal{B}|D) = -N \cdot \sum_{V_i \in V} H_{\widehat{P}}(V_i|p_{\mathcal{B}}(V_i))$$

where V is the set of variables included in the network and $p_{\mathcal{B}}(V_i)$ denotes the set of parents of V_i in the network's digraph.

While for constructing Bayesian networks in general the main purpose is to approximate the true joint distribution, for a Bayesian network classifier it is the conditional probability distribution of the class variable given the attributes that is of importance. Alternative classifiers therefore are to be compared as to their ability to describe the data for the various different classes. The *conditional log-likelihood* of a classifier $\mathcal{C}$ given a dataset D now is defined as

$$CLL(\mathcal{C}|D) = -N \cdot H_{\widehat{P}_{\mathcal{C}}}(C|A)$$

where $\widehat{P}_{\mathcal{C}}$ again denotes the observed joint distribution factorised over the classifier's graphical structure. Since the conditional probability distribution of the class variable given the attributes does not factorise over the graphical structure of a classifier, the conditional log-likelihood measure does not decompose into separate terms as does the unconditional log-likelihood measure. Because of its associated complexity of computation, the conditional log-likelihood measure is not used directly in practice.

An alternative measure that is often used for comparing the qualities of Bayesian network classifiers, is the *classification accuracy* [24]. In essence, we define the classification accuracy of a classifier $\mathcal{C}$ to be the probability of an instance being labelled with its true class value, that is,

$$accuracy(\mathcal{C}) = \sum_{a^k \in \Omega(A)} P(a^k) \cdot accuracy(\mathcal{C}, a^k)$$

where

$$accuracy(\mathcal{C}, a^k) = \begin{cases} 1 & \text{if } \mathcal{C} \text{ returns the true class value for } a^k \\ 0 & \text{otherwise} \end{cases}$$

Note that the joint probability distribution $P(a^k)$ over all possible value assignments to A is readily established from the joint probability distribution over all variables involved:

$$P(a^k) = \sum_{c^g \in \Omega(C)} P(c^g, a^k)$$

Since upon constructing a Bayesian network classifier from data the true joint probability distribution is not known, it is approximated for practical purposes by the observed distribution.

2.2.3. The MDL function

The well-known *Minimum Description Length* (MDL) principle [10] is often employed as the basis for comparing the qualities of Bayesian networks in general. Since in this paper we build upon this principle, we briefly review the, more or less standard, two-parts MDL function.

Let V be a set of stochastic variables as before. Let D be a dataset of N labelled instances over V and let $\widehat{P}(V)$ be the joint probability distribution observed in D. Let $\mathcal{B}$ be a Bayesian network over V. Then, the MDL score of the network with respect to the data is defined as

$$MDL(\mathcal{B}|D) = \frac{\log N}{2} \cdot |\mathcal{B}| - LL(\mathcal{B}|D)$$

where

$$|\mathcal{B}| = \sum_{V_i \in V} (|\Omega(V_i)| - 1) \cdot |\Omega(p_{\mathcal{B}}(V_i))|$$

with $p_{\mathcal{B}}(V_i)$ as before, and where $LL(\mathcal{B}|D)$ is the log-likelihood of the network given the data.

The MDL score of a given network serves to indicate the network's quality with respect to the data under study. The smaller the score, the better the network is. The larger the value of the log-likelihood term, that is, the closer it is to zero, the better the network models the observed probability distribution. As a fully connected network perfectly matches the data, it will have the largest log-likelihood term. Such a network will generally show poor performance, however, as a result of overfitting. The penalty term now counterbalances the effect of the log-likelihood term within the MDL function since it increases in value as a network becomes more densely connected. For a network that is too simple, the values of both the penalty term and the log-likelihood term are rather small. For a network that is too complex, on the other hand, the values of the two terms are both quite large.

The two-parts MDL function reviewed above is just one of the many forms of the Minimum Description Length principle. An overview and comparison of the various alternative forms is provided by Hansen and Yu [25]. The alternative MDL functions typically are generalisations of the two-parts function. Many of these alternative functions have been designed for other purposes and are hard to implement in the context of learning Bayesian networks from data. A possible exception is the normalised maximum log-likelihood (NML) [26] function for multimodal (discrete) data, which has been tailored to Naive Bayes classifiers [27] and to general Bayesian network classifiers [28]. Just like the two-parts MDL function, moreover, the NML function encodes the joint probability distribution over the class variable and the attributes rather than the conditional distribution. In the remainder of the paper, we will argue that the poor feature-selection behaviour of the two-parts MDL function originates from not using the conditional distribution. Our observations can thus be extended to the NML function and in fact to any form of the MDL function that captures the joint distribution over the variables involved.

3. Feature subset selection

We now define the problem of feature subset selection. We further introduce the concept of redundant attribute which we will use in the sequel for studying the feature-selection behaviour that is induced by various different quality measures.

3.1. The problem of feature subset selection

For our motivating example for doing feature selection in the context of the Bayesian network classifiers, we consider the task of constructing a Bayesian network classifier over a set of attributes A that contains two perfectly correlated attributes A_i and A_j where A_j is an exact copy of A_i. As argued before [29], by including both A_i and A_j in for example a Naive Bayesian classifier, A_i (or A_j alternatively) will have twice the influence of the other attributes, which may strongly bias the performance of the classifier. A possible way to improve the classification performance then is to eliminate one of the attributes A_i and A_j from the set A and to construct the classifier over the reduced set of attributes; the resulting classifier is called a *selective classifier*. Eliminating attributes upon constructing a classifier is commonly known as *feature subset selection*. We define the problem of feature subset selection more formally.

Definition 1. Let A be a set of attributes, let C be a class variable, and let D be a set of labelled instances over $A \cup \{C\}$. Let $\mathcal{M}$ be a specific family of Bayesian network classifiers and let $\mathcal{R}$ be a performance measure on $\mathcal{M}$. The *problem of feature subset selection* for A and D given $\mathcal{M}$ and $\mathcal{R}$ is the problem of finding a minimum subset $S \subseteq A$ such that the selective classifier $\mathcal{C} \in \mathcal{M}$ constructed over S maximises performance on D according to the measure $\mathcal{R}$.

From the definition we have that the problem of feature subset selection is restricted to a specific family of Bayesian network classifiers and to a specific performance measure. Example families of classifiers are the family of Naive Bayesian classifiers and the family of TAN classifiers [11]. Examples of performance measures are the classification accuracy and the conditional log-likelihood.

Our definition of the problem of feature subset selection is related to the first definition proposed by Tsamardinos and Aliferis [6]. In our notation, they define a feature selection problem to be a tuple $\langle D, A \cup \{C\}, alg, \mathcal{R}\rangle$, where D is a dataset over the variables $A \cup \{C\}$, *alg* is the algorithm used to construct the classifier with, and $\mathcal{R}$ is a performance measure. A solution to the problem then is a subset of attributes $S \subseteq A$ such that the selective classifier over S that is constructed using *alg* maximises performance on D given $\mathcal{R}$. There are a number of differences between the two definitions, however. For example we assume in our definition a fixed family of classifiers from among which a model is to be selected. Tsamardinos and Aliferis argue that practitioners would not like to solve a feature-selection task for a fixed family of classifiers and therefore do not restrict their definition. Our main motivation for including a fixed family of classifiers in our definition is that practitioners often are forced to select a model from among a fixed family for computational reasons or for lack of data. Another motivation for restricting our definition to a family of classifiers, is that it provides for studying the feature-selection behaviour of different quality measures in more detail. We further note that we do not specify a particular learning algorithm with our definition of the problem of feature subset selection. Our main motivation for not including a learning algorithm is that we do not want to capture the biases introduced by the heuristics involved into our definition. Defining the problem feature subset selection as a fundamental concept now allows us to study and compare the biases of the various learning algorithms in use.

3.2. The concept of redundancy

Upon constructing a selective classifier, the set of attributes A under study is split into two subsets S and O, with $S \cup O = A$ and $S \cap O = \emptyset$. S is the subset of attributes that are selected to construct the classifier with and O is the subset of attributes that will not be incorporated in the classifier. The attributes included in S are deemed important, whereas the attributes from O are considered to be *redundant* for the classification task. We define our concept of redundancy.

Definition 2. Let $A_i \in A$ be an attribute, let $S \subseteq A \setminus \{A_i\}$ be a subset of attributes, and let C be the class variable as before. Let D be a dataset of labelled instances over $A \cup \{C\}$. We say that A_i is *redundant* for C given S in D, if for every value a_i^k of A_i, for every value c^l of C, and for every value assignment s^j to S such that $N(a_i^k, s^j) > 0$, we have that

$$\frac{N(a_i^k, s^j, c^l)}{N(a_i^k, s^j)} = \frac{N(s^j, c^l)}{N(s^j)}$$

For $|S| = m$, we say that A_i is redundant for C *at level m*. We further say that A_i is *irredundant* for C given S in D if it is not redundant for C given S in D. If, for all subsets S with $|S| = m$, attribute A_i is not redundant for C given S, we say that A_i is irredundant for C *at level m*.

We note that, if an attribute A_i is redundant for C given S in D, we have, in terms of probabilities, that $\widehat{P}(C|A_i,S) = \widehat{P}(C|S)$, that is, the class variable C is independent of A_i given S in the observed distribution. Moreover, if A_i is redundant for C given S and $N(s^j, c^l) > 0$ for all value assignments c^l and s^j, then

$$\frac{N(a_i^k, s^j, c^l)}{N(s^j, c^l)} = \frac{N(a_i^k, s^j)}{N(s^j)}$$

from which we have that $\widehat{P}(A_i|S,C) = \widehat{P}(A_i|S)$.

The following example illustrates our concept of redundancy as well as the different levels at which attributes can be redundant for a class variable.

Example 1. We consider a classification problem with the binary attributes $A = \{A_1, \dots, A_8\}$ and the binary class variable C. The class variable C is defined as $C = (A_4 \oplus A_1) \vee A_2$, where $\oplus$ denotes the XOR operator and $\vee$ the logical OR operator. Among the nine variables involved, there are some logical relationships and some probabilistic independence relationships. The logical relationships among the attributes are $A_6 = A_1 \vee A_2 \vee A_4,\ A_7 \equiv A_5$, and $A_8 \equiv A_2$. For the independence relationships, we have that A_3 is independent of C given A_2; A_3 further is unconditionally dependent of A_2 and of C. Given these relationships, there are 32 possible instances of the variables involved; these instances are shown in Table 1. We now assume that we have a dataset in which each possible instance occurs exactly once. From the dataset, we observe that the attributes A_1 and A_4 both are redundant for the class variable C at level 0; so, for all values $a_1^j \in \Omega(A_1),\ a_4^k \in \Omega(A_4)$ and $c^l \in \Omega(C)$, we have that $N(c^l, a_1^j)/N(a_1^j) = N(c^l)/N$ and $N(c^l, a_4^k)/N(a_4^k) = N(c^l)/N$. A_1 and A_4 are irredundant for C at all higher levels, since for all subsets $S \subseteq A \setminus \{A_1, A_4\}$ there are values $a_1^j,\ a_4^k,\ c^l$, and $s^i \in \Omega(S)$, for which $N(c^l, a_1^j, a_4^k, s^i)/N(a_1^j, a_4^k, s^i) \neq N(c^l, s^i)/N(s^i)$. The attributes A_5 and A_7 are redundant for the class variable at all possible levels, that is, from level 0 to level $|A| - 1 = 7$. The attribute A_2 is irredundant for C at all levels including level 0. A_3 and A_8 are irredundant for C at level 0, but redundant at all higher levels given any subset of attributes that containsA_2. The attribute A_6 is irredundant for C at all levels below level 3; at level 3 and higher, it is redundant for C given any subset of attributes that contains $A_1,\ A_2$ and A_4. We note that the attributes $A_1,\ A_2$ and A_4 serve to completely determine the value of the class variable C. The Bayesian network classifier with the smallest number of attributes giving the highest classification accuracy is shown in Fig. 1.

3.2.1. Related work

We are not the first to define a concept of redundancy in the context of feature subset selection. The various concepts in use [5,6,30–35] differ in whether they address redundancy with respect to the class variable in terms of single attributes or in terms of sets of attributes. Using a concept of redundancy in terms of single attributes involves studying the relationship between the class variable and each attribute separately. Using a concept of redundancy in terms of subsets of attributes involves investigating all possible subsets of attributes and, as a consequence, is much more demanding from a computational point of view. Tsamardinos and Aliferis [6] studied redundancy in terms of sets of attributes and found that the concept does not behave monotonically with respect to taking supersets of attributes, that is, a redundant subset of attributes may become irredundant by including an additional attribute, and vice versa. We have decided, in accordance with this observation, to explicitly distinguish between redundancy at various different levels.

Table 1
The example dataset illustrating the concept of redundancy.

C	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	C	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0
1	0	0	1	1	0	1	0	0	1	0	0	1	1	1	1	1	0
1	0	1	1	0	0	1	0	1	1	0	1	1	0	1	1	1	1
1	0	1	1	1	1	1	1	1	1	0	1	1	1	0	1	0	1
1	1	0	1	0	0	1	0	0	1	1	0	1	0	1	1	1	0
0	1	0	0	1	0	1	0	0	0	1	0	0	1	1	1	1	0
1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
1	0	0	0	1	1	1	1	0	1	0	0	0	1	0	1	0	0
1	0	1	1	0	1	1	1	1	1	0	1	1	0	0	1	0	1
1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	1	1	1
1	1	0	0	0	1	1	1	0	1	1	0	0	0	0	1	0	0
0	1	0	1	1	1	1	1	0	0	1	0	1	1	0	1	0	0
1	1	1	1	0	0	1	0	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1

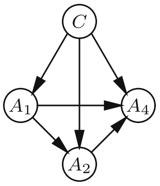

Fig. 1. The optimal Bayesian network classifier for our example dataset.

We compare our concept of redundancy to some of the other concepts that have been defined in terms of sets of attributes. John et al. [31], for example, defined the closely related concepts of relevance and irrelevance. Upon introducing their concepts, they argued that a simple distinction between relevant and irrelevant attributes does not suffice to partition the set of attributes into subsets that provide for studying the differences in feature-selection behaviour of alternative algorithms and quality measures. They therefore introduce two degrees of relevance. In contrast with the concept of John et al., our concept provides for studying redundancy at *all* possible levels $0, \ldots, |A| - 1$ separately. In the sequel we will illustrate the importance of distinguishing between these levels for learning different types of classifier.

Alternative definitions of redundancy have also been proposed by Tsamardinos and Aliferis [6], Koller and Sahami [33] and Liu et al. [34]. Tsamardinos and Aliferis relate the conditional probability of the class variable given a set of attributes to conditional independence and build their concept of redundancy on the associated concept of Markov blanket. Given a dataset of labelled instances, the Markov blanket of the class variable is the minimal set of attributes which, upon value assignment, completely substitutes the influences of the other attributes on the class variable; given its Markov blanket, therefore, the class variable is independent of all other attributes. Tsamardinos and Aliferis now showed that, for any probability distribution that is faithful to a Bayesian network, the Markov blanket of the class variable coincides with the set of strongly relevant attributes; a distribution is said to be faithful to a Bayesian network if all dependences and independences embedded in the distribution can be captured by a graphical structure. They further showed that, in a Bayesian network faithfully modelling the distribution, an attribute A_i is weakly relevant for the class variable C if and only if A_i is not strongly relevant and there is an undirected path from A_i to C. The concept of redundancy defined by Tsamardinos and Aliferis thus in essence is equivalent to that of John et al. for any probability distribution that is faithful to a Bayesian network. For an observed distribution that is not faithful to a Bayesian network, however, the Markov blanket of the class variable may not be unique. Moreover, any of the multiple blankets may then include weakly relevant attributes in addition to strongly relevant ones [36]. Building upon the concept of Markov blanket would then not result in a minimal subset of attributes shielding the influences of the other attributes from the class variable. Finally, Liu et al. [34] use parameters which are learned from the data to discriminate between relevant and irrelevant features.

3.3. The issue of noise

When constructing a selective classifier in practice, the relationship between an attribute A_i and the class variable C, as captured by the available data, is investigated. Using our definition, the attribute can be either redundant or irredundant for the class variable, that is, it can be either (conditionally) independent or dependent of C. The (in)dependences are established from a dataset of instances that are assumed to have been generated from an unknown true probability distribution. As the dataset under study is finite, however, it may not reflect the (in)dependences from the true distribution exactly; the dataset then is said to include *noise* [22]. More specifically, the attribute A_i may be independent of the class variable C in the true distribution, yet appear to be irredundant in the observed distribution, for example, due to chance of observed instances or to the misclassified instances. Attributes that have a very weak dependence of the class variable in the observed distribution therefore, may in fact be independent. To provide for feature subset selection in a practical setting, we introduce the concept of *redundancy within an allowed amount of noise*.

Definition 3. Let $A_i \in A$ be an attribute, let $S \subseteq A \setminus \{A_i\}$ be a subset of attributes, and let C be the class variable as before. Let D be a dataset of N labelled instances over $A \cup \{C\}$. Let $\xi(A_i, C, S, N) > 0$ be a threshold value for the allowed amount of noise. We say that A_i is *redundant* for C given S in D *within the allowed amount of noise* $\xi(A_i, C, S, N)$, if

$$H_{\widehat{P}}(C|S) - H_{\widehat{P}}(C|A_i, S) < \xi(A_i, C, S, N)$$

Otherwise, we say that A_i is irredundant for C given S.

From the above definition, we have that an attribute A_i is said to be redundant for the class variable C given S within some allowed amount of noise ξ, if obtaining a value for A_i serves to reduce the conditional entropy of C given S by at most ξ. Note that if obtaining a value for A_i does not reduce the entropy of C given S at all, that is, if $H_{\widehat{P}}(C|S) - H_{\widehat{P}}(C|A_i, S) = 0$, we have that A_i is simply redundant for C given S. Since the conditional entropy depends on the number of values that the variables involved can adopt, we have defined the allowed amount of noise to be functionally dependent of A_i, C and S. The function is also taken to be dependent of the number of observed instances, since we would like to allow less noise for larger datasets in which the true (in)dependences are better represented. The threshold function may have many different forms; we will return to this observation in subsequent sections where we analyse the feature-selection behaviour of the MDL function.

Note that $H_{\widehat{P}}(C|S) - H_{\widehat{P}}(C|A_i, S) = H_{\widehat{P}}(A_i|S) - H_{\widehat{P}}(A_i|C, S)$. We further call the term $H_{\widehat{P}}(A_i|S) - H_{\widehat{P}}(A_i|C, S) - \xi(A_i, C, S, N)$ the *amount of irredundancy* the attribute A_i has for C given the attribute set S within the allowed noise level $\xi(A_i, C, S, N)$. We observe that a negative amount of irredundancy corresponds to redundancy of the attribute A_i for C given S within $\xi(A_i, C, S, N)$, whereas a positive amount of irredundancy corresponds with irredundancy.

Zhu and Wu [22] experimentally study the relationship between the (in)dependency between attributes and the class variable and the impact of noise over the performance of a classifier. In their study, they make the assumption that attributes are independent of each other given the class variable. They show that the stronger the relationship between an attribute and the class variable the more impact the noise of this attribute has over the classifier. Our definition of noise also captures the relationships between the class variable and the involved attributes. Furthermore, it generalises the above observation by considering also the dependencies between the attributes given the class variable.

By redefining our concept of redundancy, we explicitly provide for handling a limited amount of noise in a dataset under study. Another approach to the problem of insufficient data is to not allow explicitly for noise, but to use heuristic algorithms for establishing redundancy. Such an algorithm for example never studies a larger number of variables at the same time.

4. An MDL-based quality measure for feature subset selection

We build upon the assumption that the relatively poor feature-selection behaviour of the MDL function originates, to at least some extent, from not using the conditional probability distribution. For that we introduce a new quality measure, called *MDL-FS*, that is tailored to feature selection [18]. The MDL-FS function in essence is based upon the same ideas as the MDL function. Like the MDL function, it captures the joint probability distribution $P(C, A)$ over all variables involved in a log-likelihood term. In addition however, it captures the joint probability distribution $P(A)$ over just the attributes. We note that while the joint distribution $P(C, A)$ factorises over the structure of the classifier under study, the distribution $P(A)$ does not; to allow for ease of computation, the function therefore uses an *auxiliary Bayesian network* to factorise $P(A)$. The function now establishes the difference between the log-likelihood of the probability distribution $P(C, A)$ and the log-likelihood of the distribution $P(A)$, and thereby effectively models the conditional probability distribution

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

Informally speaking, by capturing the difference between the two log-likelihood terms, the strengths of the relationships among the attributes themselves are eliminated from the joint distribution. In contrast with the MDL function, therefore, the MDL-FS function will identify and remove attributes that are redundant for the class variable yet strongly related to one or more other attributes.

In Section 4.1, we introduce the MDL-FS function. In Section 4.2, we study the relationships of the conditional log-likelihood term of the function and the conditional distribution of the class variable given the set of attributes; we will argue more specifically that the former may be considered an approximation of the latter.

4.1. The MDL-FS function for feature selection

We formally define the MDL-FS function.

Definition 4. Let A be a set of attributes and let C be the class variable as before. Let D be a dataset of N labelled instances over $A \cup \{C\}$. Let $\mathcal{C}$ be a Bayesian network classifier over $A \cup \{C\}$ and let $\mathcal{S}$ be a Bayesian network over A. Then,

$$\text{MDL-FS}(\mathcal{C}, \mathcal{S}|D) = \frac{\log N}{2} \cdot |\mathcal{C}| - CALL(\mathcal{C}, \mathcal{S}|D)$$

where $|\mathcal{C}|$ is as before and

$$CALL(\mathcal{C}, \mathcal{S}|D) = LL(\mathcal{C}|D) - LL(\mathcal{S}|D)$$

with $LL(\mathcal{C}|D)$ as before and

$$LL(\mathcal{S}|D) = -N \cdot \sum_{A_i \in A} H_{\widehat{P}}(A_i|p_{\mathcal{S}}(A_i))$$

where, for each attribute $A_i \in A$, the set $p_{\mathcal{S}}(A_i)$ is the set of parents of A_i in the graphical structure of the network $\mathcal{S}$.

The basic idea underlying the MDL-FS function is the same as that of the MDL function. The MDL-FS function also includes a *penalty term* to capture the length of an encoding of the classifier and a term that indicates the length of an encoding of the observed probability distribution given the classifier. The latter term in essence captures the observed conditional distribution and is called the *conditional auxiliary log-likelihood term* of the MDL-FS function. Like the MDL score, the MDL-FS score of a Bayesian network classifier indicates the classifier's quality with respect to the data under study. The smaller the score, the better the classifier is.

For the conditional auxiliary log-likelihood term, we have that the larger the value of the term $LL(\mathcal{C}|D)$, that is, the closer it is to zero, the better the classifier models the observed joint probability distribution, as we have argued before in Section 2. The term therefore tends to approach zero for classifiers with a complex graphical structure and to be quite small for simpler models. The term $LL(\mathcal{S}|D)$ also attains its minimum value for the simplest Bayesian network and its maximum value for a fully connected model. The maximum value of the conditional auxiliary log-likelihood term therefore is obtained for a fully connected Bayesian network classifier and an empty auxiliary Bayesian network without any arcs. Now, to achieve a small score for a classifier, we basically would like to maximise the conditional auxiliary log-likelihood term of the MDL-FS function. From the above considerations however, we must conclude that we cannot simply maximise the term, as the function does not suggest any reason for using a more complex auxiliary network than the empty one. Yet, using an empty auxiliary network would not meet our purpose of capturing the relationships among the attributes from the joint distribution: for this purpose, a more complex auxiliary network is required. As the MDL-FS function has no control over the complexity of the auxiliary network to be used, in practical applications we propose to set a specific family of auxiliary networks beforehand. Since we would like to eliminate the influence of $P(A)$ from $P(C,A)$, we should select from this family a maximum log-likelihood network. We thus have to maximise the log-likelihood of both the classifier and the auxiliary networks to model a conditional auxiliary log-likelihood term suited for feature subset selection.

Like the MDL function, the MDL-FS function includes a penalty term to counterbalance the effect of the conditional auxiliary likelihood term. From the definition of the MDL-FS function, we observe that this penalty term captures just the complexity of the classifier and not the complexity of the auxiliary network. We have decided not to include a penalty term for the auxiliary network since we are basically interested in the complexity of the classifier only. A difference of penalty terms for the complexities of the classifier and the auxiliary network however, would serve to more evenly counterbalance the difference between the log-likelihoods of the two networks. A quality measure that includes such a difference of penalty terms would amount to taking the difference of the MDL score of the classifier and the MDL score of the auxiliary network. The penalty term for the auxiliary network would then have a negative effect on the feature-selection behaviour of the MDL-FS function in the sense that it would become less selective.

4.2. Comparing the conditional auxiliary log-likelihood with conditional entropy

Upon reviewing the MDL-FS function, we have argued that its conditional auxiliary log-likelihood term in essence models the log-likelihood of the conditional probability distribution of the class variable given the attributes. In this section, we show that for a fully connected classifier and a fully connected auxiliary network, the term indeed models the log-likelihood of the conditional distribution. In practical applications, fully connected classifiers have major disadvantages: in addition to the large number of data required for their construction, these classifiers tend to overfit the available data and to show poor generalisation performance. As a consequence, they are hardly ever used in practice. Our result therefore serves to give a fundamental insight only. We will further argue that for classifiers and auxiliary networks that do not accurately capture all information from the data, the conditional auxiliary log-likelihood term can only be looked upon as an approximation of the log-likelihood of the conditional distribution.

Proposition 1. *Let $\mathcal{C}_{full}$ be a fully connected Bayesian network classifier over $A \cup \{C\}$ and let $\mathcal{S}_{full}$ be a fully connected Bayesian network over A. Then,*

$$CALL(\mathcal{C}_{full}, \mathcal{S}_{full}|D) = -N \cdot H_{\widehat{P}}(C|A_1, \ldots, A_n)$$

Proof. Since the Bayesian network classifier $\mathcal{C}_{full}$ is fully connected, we have that

$$LL(\mathcal{C}_{full}|D) = -N \cdot H_{\widehat{P}}(C, A_1, \ldots, A_n) = -N \cdot (H_{\widehat{P}}(C|A_1, \ldots, A_n) + \cdots + H_{\widehat{P}}(A_n))$$

For the Bayesian network $\mathcal{S}$ moreover, we have that

$$LL(\mathcal{S}_{full}|D) = -N \cdot H_{\widehat{P}}(A_1, A_2, \ldots, A_n) = -N \cdot (H_{\widehat{P}}(A_1|A_2, \ldots, A_n) + \cdots + H_{\widehat{P}}(A_n))$$

For the conditional auxiliary log-likelihood term of the MDL-FS function, we thus find that

$$CALL(\mathcal{C}_{full}, \mathcal{S}_{full}|D) = -N \cdot H_{\widehat{P}}(C|A_1, \ldots, A_n)$$

as stated above. □

From the previous proposition, we have that for fully connected classifiers and fully connected auxiliary networks, the conditional auxiliary log-likelihood term of the MDL-FS function accurately models the log-likelihood of the conditional distribution. As we have argued above, in practical applications classifiers of a simpler complexity than fully connected ones are used. For these classifiers, the previous proposition no longer holds and the conditional auxiliary log-likelihood term of the MDL-FS function may differ from the log-likelihood of the conditional distribution.

4.3. Related work

Kontkanen et al. [37] propose to perform feature subset selection using the supervised marginal log-likelihood (closely related with the conditional log-likelihood). When they evaluate a subset of features, they only consider the case when the attributes are independent of each other given the class variable. Their method is closely related with our method because it learns the joint probability distribution over the class variable and the attribute set. However, unlike our method, they do not use an auxiliary structure to express the joint probability distribution over the attribute set and, as a consequence, this method is too computationally expensive to consider the relationships between attributes given the class variable (for example to learn TANs). Jebara and Jaakkola [38] associate to each attribute a probability value with a maximum entropy discrimination framework. They use regression methods in discriminant functions (closely related with conditional log-likelihood) to perform classification or regression. In the sequel, similarly with Drugan and van der Gaag [18], Guo and Greiner [12] propose a BIC score composed of a conditional log-likelihood term and a penalty score to learn Bayesian network classifiers. Again, unlike our method, they do not use an auxiliary structure, but search for parameters that optimize the conditional log-likelihood score. For their experiments, they set these parameters to the frequencies from the dataset. They experimentally show that their conditional BIC can learn Naive Bayes and TAN classifiers from data.

Bilmes [39], and later Pernkopf and Bilmes [40], uses a measure that prefers arcs between two attributes which have a high conditional mutual information given the class variable but a low un-conditioned mutual information. Such an algorithm has a different behaviour in learning the structure of the Bayesian network classifiers than an algorithm that uses the MDL-FS score since it does not perform feature selection (e.g. the arcs are preferentially added between attributes of the entire set of attributes). Grossman and Domingos [19] also maximise the conditional log-likelihood using the parameters of maximum log-likelihood approximated with a regression algorithm. Burge and Lane [41] learn also discriminative structures by using separate Bayesian network classifiers for each class value. They approximate the conditional log-likelihood for the two value class as the ratio between the log-likelihood of the classifier given one class variable and the classifier given the other class variable and the same classifier.

5. The feature-selection behaviour of the MDL-FS and MDL functions in general

We begin by studying the feature-selection behaviour of the MDL-FS function for complete Bayesian network classifiers and auxiliary structures, to review in an informal way some of its general properties. Recall that fully connected networks perfectly model the data and are of maximum log-likelihood for that data but, for pragmatical reasons, are seldom used in practice; we will substantiate the reviewed properties in the subsequent sections for more commonly used classifiers. In this section, we study the ability of the MDL and MDL-FS functions to identify and eliminate redundant attributes at different levels. We will argue that the MDL function tends to eliminate from a Bayesian network classifier only attributes that are redundant at level 0 for the class variable as well as for the other attributes. The MDL-FS function overcomes this drawback by comparing the strength of the relationship between an attribute and its parents in the classifier with the strength of the relationship between an attribute and its parents in the auxiliary structure. We show that MDL-FS tends to eliminate from fully connected Bayesian network classifiers the attributes that are redundant for the class variable at the highest level within an allowed amount of noise determined by the penalty term by using a fully connected auxiliary network whereas MDL tends to not eliminate these attributes. We find that the level of the eliminated redundant attributes with the MDL-FS score depends on the complexity of the auxiliary network: with a fully connected auxiliary network, it eliminates redundant attributes at level $|A| - 1$, whereas with an empty auxiliary network, MDL-FS eliminates attributes redundant at level 0 for the class variable and for the other variables from the attributes set.

5.1. The feature-selection behaviour of the MDL function

In this section, we investigate the feature-selection behaviour of the MDL function. We will show that the MDL function is able to identify and eliminate only attributes that are redundant at level 0 for the class variable as well as for the other attributes. Before presenting this result, we observe that, to allow for comparing the MDL and/or MDL-FS scores of two classifiers, they both need to capture a joint probability distribution over the same set of variables. When comparing the score of a selective classifier with the score of a classifier that includes more attributes therefore, we look upon the selective classifier as being extended with the deleted attributes by means of nodes without any incident arcs.

Proposition 2. *Let $\mathcal{C}$ be a Bayesian network classifier and let $A_i \in A$ be an attribute in $\mathcal{C}$ with the set of parents $p_{\mathcal{C}}(A_i)$ and the set of children $c_{\mathcal{C}}(A_i)$. From $\mathcal{C}$, we construct the selective classifier $\mathcal{C}^-$ by deleting the incident arcs of A_i. Then,*

$$\mathrm{MDL}(\mathcal{C}^-|D) < \mathrm{MDL}(\mathcal{C}|D)$$

if and only if

$$\left[H_{\widehat{P}}(A_i) - H_{\widehat{P}}(A_i|p_{\mathcal{C}}(A_i)) - \frac{\log N}{2 \cdot N} \cdot (|\Omega(A_i)| - 1) \cdot (|\Omega(p_{\mathcal{C}}(A_i))| - 1)\right] + \sum_{A_k \in c_{\mathcal{C}}(A_i)} \Big[H_{\widehat{P}}(A_i|p_{\mathcal{C}^-}(A_k)) - H_{\widehat{P}}(A_i|A_k, p_{\mathcal{C}^-}(A_k))$$
$$- \frac{\log N}{2 \cdot N} \cdot (|\Omega(A_k)| - 1) \cdot |\Omega(p_{\mathcal{C}^-}(A_k))| \cdot (|\Omega(A_i)| - 1)\Big] < 0$$

Proof. We begin by observing that, since the two classifiers differ only in the incident arcs for the attribute A_i, the difference of their MDL scores pertains to just A_i and its parents and children. To investigate the difference of the two scores, we now study the differences of their log-likelihood terms and of their penalty terms separately. The difference of the log-likelihood terms for the two classifiers equals

$$LL(\mathcal{C}|D) - LL(\mathcal{C}^-|D) = -N \cdot (H_{\hat{P}}(A_i|p_{\mathcal{C}}(A_i)) - H_{\hat{P}}(A_i)) - N \cdot \sum_{A_k \in c_{\mathcal{C}}(A_i)} (H_{\hat{P}}(A_k|p_{\mathcal{C}}(A_k)) - H_{\hat{P}}(A_k|p_{\mathcal{C}^-}(A_k)))$$

The difference of the penalty terms of the two classifiers equals

$$\log N/2 \cdot (|\mathcal{C}| - |\mathcal{C}^-|) = \log N/2 \cdot \left[(|\Omega(A_i)| - 1) \cdot (|\Omega(p_{\mathcal{C}}(A_i))| - 1) + \sum_{A_k \in c_{\mathcal{C}}(A_i)} (|\Omega(A_k)| - 1) \cdot |\Omega(p_{\mathcal{C}^-}(A_k))| \cdot (|\Omega(A_i)| - 1)\right]$$

Note that the difference of the two penalty terms is positive since the classifier $\mathcal{C}$ is more complex than the selective classifier $\mathcal{C}^-$. Using

$$\text{MDL}(\mathcal{C}|D) - \text{MDL}(\mathcal{C}^-|D) = \log N/2 \cdot (|\mathcal{C}| - |\mathcal{C}^-|) - (LL(\mathcal{C}|D) - LL(\mathcal{C}^-|D))$$

and

$$\begin{aligned} H_{\hat{P}}(A_k|p_{\mathcal{C}}(A_k)) - H_{\hat{P}}(A_k|p_{\mathcal{C}^-}(A_k)) &= H_{\hat{P}}(A_k, A_i, p_{\mathcal{C}^-}(A_k)) - H_{\hat{P}}(A_i, p_{\mathcal{C}^-}(A_k)) + H_{\hat{P}}(A_k, p_{\mathcal{C}^-}(A_k)) - H_{\hat{P}}(p_{\mathcal{C}^-}(A_k)) \\ &= H_{\hat{P}}(A_i|A_k, p_{\mathcal{C}^-}(A_k)) - H_{\hat{P}}(A_i|p_{\mathcal{C}^-}(A_k)) \end{aligned}$$

for each attribute $A_k \in c_{\mathcal{C}}(A_i)$, where $p_{\mathcal{C}}(A_k) = \{A_i\} \cup p_{\mathcal{C}^-}(A_k)$, we now straightforwardly obtain the proposition's inequality. □

By definition we have that the MDL function prefers the selective classifier $\mathcal{C}^-$ over $\mathcal{C}$ if and only if $\mathcal{C}^-$ has a smaller MDL score than $\mathcal{C}$, that is, if and only if $\text{MDL}(\mathcal{C}^-|D) < \text{MDL}(\mathcal{C}|D)$ for the dataset D under consideration. The difference $\log N/2 \cdot (|\mathcal{C}| - |\mathcal{C}^-|)$ of the penalty terms for the classifier $\mathcal{C}$ and the selective classifier $\mathcal{C}^-$ is greater than zero since $\mathcal{C}$ has a complexer structure than $\mathcal{C}^-$. The difference $LL(\mathcal{C}|D) - LL(\mathcal{C}^-|D)$ of the two log-likelihood terms also is greater than 0 because the classifier $\mathcal{C}$ captures the observed joint probability distribution at least as accurately as the selective classifier $\mathcal{C}^-$. The proposition now indicates under which condition the additional complexity of $\mathcal{C}$ is no longer counterbalanced by its increased log-likelihood.

We study the condition stated in the proposition in some closer detail. We observe that the condition basically pertains to the strengths of the relationships of the attribute A_i with the other attributes. Informally speaking, the stronger the relationships of A_i with its neighbouring attributes in $\mathcal{C}$, the more the observation of a value assignment to these attributes can contribute to resolving the uncertainty as to the value of A_i. The stronger the relationships of A_i with its neighbouring attributes, therefore, the more likely the inequality stated in the proposition does not hold and the full classifier is preferred over the selective one. The next corollary now quantifies the maximal amount of irredundancy the attribute can have with its neighbours before it is effectively removed by the MDL function.

Corollary 1. *Let $\mathcal{C}$, $\mathcal{C}^-$ and A_i be as in Proposition 2. The selective classifier $\mathcal{C}^-$ is preferred over the full classifier $\mathcal{C}$ if only if*

- *the attribute A_i is redundant at level 0 for the variables in its set of parents $p_{\mathcal{C}}(A_i)$ in $\mathcal{C}$ within the allowed amount of noise $\xi(A_i, p_{\mathcal{C}}(A_i), \emptyset, N) < \log N/(2 \cdot N) \cdot (|\Omega(A_i)| - 1) \cdot (|\Omega(p_{\mathcal{C}}(A_i))| - 1)$; and*
- *the attribute A_i is redundant for each child attribute $A_k \in c_{\mathcal{C}}(A_i)$ given A_k's other parents in $\mathcal{C}$ within the allowed amount of noise $\xi(A_i, A_k, \emptyset, N) < \log N/(2 \cdot N) \cdot (|\Omega(A_k)| - 1) \cdot |\Omega(p_{\mathcal{C}^-}(A_k))| \cdot (|\Omega(A_i)| - 1)$.*

From the property stated in the corollary, we conclude that, upon feature selection, an attribute is removed by the MDL function *only if* it is redundant at level 0 for *all other variables*, within an amount of noise that is dependent of the structure of the classifier. For Naive Bayes classifiers, the function will thus serve to remove attributes that are redundant for the class variable C at level 0, since in such a restricted classifier the attributes are assumed to be mutually independent given C. For more complex Bayesian network classifiers, however, attributes that are redundant for the class variable at various levels will not be removed unless these attributes are redundant for all other attributes as well. We conclude that the MDL function is not very well suited for the task of identifying and removing attributes that are redundant for the class variable.

5.2. The feature-subsection behaviour of the MDL-FS function

In this section, we study the feature-selection behaviour of the MDL-FS function in detail. Before doing so, we relate the function to the MDL function and show under which conditions the two functions exhibit the same behaviour.

We have argued in Section 4 that the MDL-FS function is closely related to the MDL function and differs from this function mainly in that it captures, in addition to the joint probability distribution over the set of all variables, also the joint distribution over just the attributes. We now show that upon comparing classifiers over the same set of variables, the two functions exhibit the same preference behaviour as long as the MDL-FS function uses auxiliary networks that have the same log-likelihood given the data.

Proposition 3. *Let $\mathcal{C}$ and $\mathcal{C}'$ be two Bayesian network classifiers over $A \cup \{C\}$. Let $\mathcal{S}$ and $\mathcal{S}'$ be two Bayesian networks over A with $LL(\mathcal{S}|D) = LL(\mathcal{S}'|D)$. Then,*

$$\mathrm{MDL}(\mathcal{C}|D) - \mathrm{MDL}(\mathcal{C}'|D) = \text{MDL-FS}(\mathcal{C},\mathcal{S}|D) - \text{MDL-FS}(\mathcal{C}',\mathcal{S}'|D)$$

Proof. The property stated in the proposition follows directly from the definitions of the two functions. □

The condition described in the previous proposition hardly ever occurs in a practical setting. Especially in view of feature selection, will it hardly ever be the case that classifiers are compared using (different) auxiliary networks of the same log-likelihood. The importance of the proposition therefore lies mainly in the observation that, with a fixed auxiliary network over a fixed set of attributes, the MDL-FS function will always prefer the same classifier as the MDL function. More specifically, the two functions will exhibit the same preference behaviour if the MDL-FS function uses an empty auxiliary network.

We now turn to the feature-selection behaviour of the MDL-FS function in a more practical setting where classifiers are compared using auxiliary networks of possibly different log-likelihood. To informally review some of the function's properties, we begin by studying the MDL-FS score of a Bayesian network classifier $\mathcal{C}$ over $A \cup \{C\}$ and an auxiliary Bayesian network $\mathcal{S}$ over A. We rewrite this score as a sum of terms for the class variable and for each attribute A_i separately:

$$\begin{aligned}\text{MDL-FS}(\mathcal{C},\mathcal{S}|D) = N \cdot [H_{\widehat{P}}(C) + \log N/(2 \cdot N) \cdot (|\Omega(C)| - 1)] + N \cdot \sum_{A_i \in A} [H_{\widehat{P}}(A_i|p_{\mathcal{C}}(A_i)) - H_{\widehat{P}}(A_i|p_{\mathcal{S}}(A_i))\\ + \log N/(2 \cdot N) \cdot (|\Omega(A_i)| - 1) \cdot |\Omega(p_{\mathcal{C}}(A_i))|]\end{aligned}$$

where $p_{\mathcal{C}}(A_i)$ and $p_{\mathcal{S}}(A_i)$ are the sets of parents of A_i in the networks $\mathcal{C}$ and $\mathcal{S}$, respectively. We observe that strong relationships between the attribute A_i and its parents in the classifier $\mathcal{C}$, that is, $H_{\widehat{P}}(A_i|p_{\mathcal{C}}(A_i))$ going to 0, would decrease the score, while strong relationships between A_i and its parents in $\mathcal{S}$, that is, $H_{\widehat{P}}(A_i|p_{\mathcal{S}}(A_i))$ going to 0, would increase the score. In view of feature subset selection, therefore, the stronger the relationships of the attribute A_i with its parents in the classifier and the weaker the relationships with its parents in the auxiliary network, the less likely the attribute is to be removed. To study the differences in strength of these relationships in more detail, we express the difference $H_{\widehat{P}}(A_i|p_{\mathcal{S}}(A_i)) - H_{\widehat{P}}(A_i|p_{\mathcal{C}}(A_i))$ in terms of the amounts of irredundancy that the attribute A_i has with its parents in the two networks. Let $p_{\mathcal{C}\cap\mathcal{S}}(A_i) = p_{\mathcal{C}}(A_i) \cap p_{\mathcal{S}}(A_i)$ be the set of parents of A_i in both the classifier and the auxiliary network. Then,

$$H_{\widehat{P}}(A_i|p_{\mathcal{S}}(A_i)) - H_{\widehat{P}}(A_i|p_{\mathcal{C}}(A_i)) = (H_{\widehat{P}}(A_i|p_{\mathcal{C}\cap\mathcal{S}}(A_i)) - H_{\widehat{P}}(A_i|p_{\mathcal{C}}(A_i))) - (H_{\widehat{P}}(A_i|p_{\mathcal{C}\cap\mathcal{S}}(A_i)) - H_{\widehat{P}}(A_i|p_{\mathcal{S}}(A_i)))$$

The two terms capturing the amount of irredundancy for attribute A_i both are positive. The amount of irredundancy $H_{\widehat{P}}(A_i|p_{\mathcal{C}\cap\mathcal{S}}(A_i)) - H_{\widehat{P}}(A_i|p_{\mathcal{C}}(A_i))$ describes how "far" the attribute A_i is from being redundant for the set of variables $p_{\mathcal{C}}(A_i) \setminus p_{\mathcal{S}}(A_i)$ given $p_{\mathcal{C}\cap\mathcal{S}}(A_i)$; note that the set $p_{\mathcal{C}}(A_i) \setminus p_{\mathcal{S}}(A_i)$ includes the class variable and all attributes that are parents of A_i in $\mathcal{C}$ but not in $\mathcal{S}$. The closer to 0 this term is, the larger the MDL-FS score will be and the more likely the attribute will be removed upon feature subset selection. The term $H_{\widehat{P}}(A_i|p_{\mathcal{C}\cap\mathcal{S}}(A_i)) - H_{\widehat{P}}(A_i|p_{\mathcal{S}}(A_i))$, on the other hand, indicates how "far" the attribute A_i is from being redundant for the set of variables $p_{\mathcal{S}}(A_i) \setminus p_{\mathcal{C}}(A_i)$ given $p_{\mathcal{C}\cap\mathcal{S}}(A_i))$; note that the set $p_{\mathcal{S}}(A_i) \setminus p_{\mathcal{C}}(A_i)$ includes all attributes that are parents of A_i in $\mathcal{S}$ but not in $\mathcal{C}$. The closer to 0 this term is, the smaller the MDL-FS score will be and the less likely the attribute is to be removed. We conclude that the difference $H_{\widehat{P}}(A_i|p_{\mathcal{S}}(A_i)) - H_{\widehat{P}}(A_i|p_{\mathcal{C}}(A_i))$ represents the amount of irredundancy of the attribute A_i for the class variable and its exclusive parent attributes in the classifier given its parent attributes in the auxiliary network.

We now begin by studying the feature-selection behaviour of the MDL-FS function for complete Bayesian network classifiers using complete auxiliary networks. To pertain to the same joint proposal distributions like the full classifier and auxiliary network, we again extend the selective networks with the deleted attributes by means of nodes without incident arcs.

Proposition 4. *Let $\mathcal{C}$ and $\mathcal{C}^-$ as in Proposition 2. In addition, let $\mathcal{S}$ be an auxiliary network over the attributes A and let $p_{\mathcal{S}}(A_i)$ be the set of parents and $c_{\mathcal{S}}(A_i)$ be the set of children of A_i in $\mathcal{S}$. Let $\mathcal{S}^-$ be the selective auxiliary network that is obtained from $\mathcal{S}$ by deleting the incident arcs of A_i. Then,*

$$\text{MDL-FS}(\mathcal{C}^-,\mathcal{S}^-|D) < \text{MDL-FS}(\mathcal{C},\mathcal{S}|D)$$

if and only if

$$\begin{aligned}&[H_{\widehat{P}}(A_i|p_{\mathcal{S}}(A_i)) - H_{\widehat{P}}(A_i|p_{\mathcal{C}}(A_i)) - \log N/2 \cdot (|\Omega(A_i)| - 1) \cdot (|\Omega(p_{\mathcal{C}}(A_i))| - 1)] + \sum_{A_k \in c_{\mathcal{C}}(A_i)} [H_{\widehat{P}}(A_i|p_{\mathcal{C}^-}(A_k)) - H_{\widehat{P}}(A_i|A_k, p_{\mathcal{C}^-}(A_k))\\ &\quad - \log N/(2 \cdot N) \cdot (|\Omega(A_k)| - 1) \cdot |\Omega(p_{\mathcal{C}^-}(A_k))| \cdot (|\Omega(A_i)| - 1)]\\ &\quad < \sum_{A'_k \in c_{\mathcal{S}}(A_i)} [H_{\widehat{P}}(A_i|p_{\mathcal{S}^-}(A'_k)) - H_{\widehat{P}}(A_i|A'_k, p_{\mathcal{S}^-}(A'_k))]\end{aligned}$$

where $p_{\mathcal{C}^-}(A_k)$, with $p_{\mathcal{C}}(A_k) = \{A_i\} \cup p_{\mathcal{C}^-}(A_k)$, and $p_{\mathcal{S}^-}(A_k)$, with $p_{\mathcal{S}}(A'_k) = \{A_i\} \cup p_{\mathcal{S}^-}(A'_k)$, are the parents sets of A_k and A'_k in $\mathcal{C}^-$ and $\mathcal{S}^-$ graphical structures, respectively.

Proof. These proofs are similar with the one from Proposition 2. To investigate the difference between the two MDL-FS scores, we study the differences of the log-likelihood terms of the classifiers, the log-likelihood terms of the auxiliary networks and of the penalty terms separately. Since we modify only locally the Bayesian structures, the difference in the MDL-FS score will be reflected only by the locally modified parts. The difference of the two log-likelihood terms of the classifiers and of the auxiliary networks equals

$$LL(\mathcal{S}|D) - LL(\mathcal{S}^-|D) = -N \cdot (H_{\widehat{P}}(A_i|p_{\mathcal{S}}(A_i)) - H_{\widehat{P}}(A_i)) - N \cdot \sum_{A'_k \in c_{\mathcal{S}}(A_i)} (H_{\widehat{P}}(A'_k|p_{\mathcal{S}}(A'_k)) - H_{\widehat{P}}(A'_k|p_{\mathcal{S}^-}(A'_k)))$$

We now have

$$\text{MDL-FS}(\mathcal{C}, \mathcal{S}|D) - \text{MDL-FS}(\mathcal{C}^-, \mathcal{S}^-|D) = \log N/2 \cdot (|\mathcal{C}| - |\mathcal{C}^-|) - (CALL(\mathcal{C}, \mathcal{S}|D) - CALL(\mathcal{C}^-, \mathcal{S}^-|D))$$

from which directly results the proposition's inequality by substituting each child $A'_k \in c_{\mathcal{S}}(A_i)$, where $p_{\mathcal{S}}(A'_k) = \{A_i\} \cup p_{\mathcal{S}^-}(A'_k)$, in the equation

$$H_{\widehat{P}}(A'_k|p_{\mathcal{S}}(A'_k)) - H_{\widehat{P}}(A'_k|p_{\mathcal{S}^-}(A'_k)) = H_{\widehat{P}}(A_i|A'_k, p_{\mathcal{S}^-}(A'_k)) - H_{\widehat{P}}(A_i|p_{\mathcal{S}^-}(A'_k))$$

and each child $A_k \in c_{\mathcal{C}}(A_i)$. □

By definition we have that the MDL-FS function prefers $\mathcal{C}$ over $\mathcal{C}^-$ if and only if $\mathcal{C}$ has a smaller MDL-FS score than $\mathcal{C}^-$, that is, if and only if $\text{MDL-FS}(\mathcal{C}, \mathcal{S}|D) - \text{MDL-FS}(\mathcal{C}^-, \mathcal{S}^-|D) < 0$ for the dataset D. Again the difference between the penalty terms is greater than zero because the full Bayesian network classifier is more complex than the selective one. When A_i does not have any children in $\mathcal{C}$, the difference between the two conditional auxiliary log-likelihoods is equal with the A_i's term in the conditional auxiliary log-likelihood of the fully connected classifier, $N \cdot (H_{\widehat{P}}(A_i|p_{\mathcal{C}}(A_i)) - H_{\widehat{P}}(A_i|p_{\mathcal{S}}(A_i)))$. When A_i has children (e.g. A_k) in $\mathcal{C}$, we add a term that indicates the amount of irredundancy of A_i for A_k given A_k's parents in $\mathcal{C}$ except for A_i; when A_i has children (e.g. A'_k) in $\mathcal{S}$, we subtract a term that indicates the amount of irredundancy of A_i for A'_k given A'_k's parents in $\mathcal{S}$ except for A_i. Thus, the difference of conditional auxiliary log-likelihoods increases with the strength of the relationships between the attribute and the other variables in the classifier, and decreases with its strength of the relationship between the attribute and the other attributes in the auxiliary network.

Informally speaking, the stronger one or more of the relationships between A_i and the other variables $A \setminus \{A_i\}$ in the classifier are and the weaker the relationships of A_i in the auxiliary network are, the more the full classifier is preferred over the selective classifier with the MDL-FS function. It is interesting to note that the feature-selection behaviour of the MDL-FS function can be derived directly from the function itself as we have presented in the first part of this section. Thus only the parents sets of an attribute in the classifier and in the auxiliary structure determine the amount of (ir)redundancy an attribute has. The children of an attribute compare the amount of irredundancy the attribute has for them given the other parents of these children in the classifier with the amount of irredundancy the attribute has for their children given the other parents of these children in the auxiliary structure. As a consequence, the feature selection properties in removing or not a redundant attribute with the MDL-FS function is correlated to the complexity of the parents set of the attribute in the classifier and in the auxiliary structure, and not to the attribute's children sets.

From these observations, we conclude that *the MDL-FS function is more suited for the task of feature selection since it can serve to identify and remove redundant attributes at various levels*. Whereas with the MDL function we can eliminate only the attributes that are redundant at level 0 for all other variables from the dataset, $\{C\} \cup A \setminus \{A_i\}$, and thus it can be only used for Naive Bayes classifiers, the MDL-FS function can be used to eliminate redundant attributes at various levels from more complex classifiers. In the following, we practically illustrate the use of the MDL-FS score in reducing redundant attributes at various levels from Bayesian network classifiers of interest. It should be noted that *the level of redundancy for which the MDL-FS function reduces attributes depends on the complexity of the auxiliary structure*. We also note that the MDL-FS function eliminates only redundant attributes at level 0 for the class variable and for the attributes when an empty network is used for the auxiliary structure. To illustrate that, in fact, the MDL-FS function, unlike the MDL score, removes attributes redundant at level $|A| - 1$ from full Bayesian network classifiers with a complete Bayesian auxiliary structure, upon feature selection, we consider again the classification problem from Example 1.

Example 2. Let us consider the dataset D from Example 1 by copying it 128 times – then $N = 4096$ – and $A = \{A_1, \ldots, A_8\}$. Let's consider a complete Bayesian network classifier $\mathcal{C}$ and a selective one $\mathcal{C}^-$ as before. Let's consider a complete Bayesian network $\mathcal{S}$ and a selective one $\mathcal{S}^-$ as before. Suppose that $A_i \equiv A_5$, where A_5 is redundant for C at all levels. Since $H_{\widehat{P}}(A_5|C, A \setminus \{A_5\}) = H_{\widehat{P}}(A_5|A \setminus \{A_5\})$, from the above proposition, the selective Bayesian classifier $\mathcal{C}^-$ is preferred to the full Bayesian classifier $\mathcal{C}$ when we use MDL-FS. Thus, A_5 is correctly removed from the classifier.

When we use MDL, since $A_5 \equiv A_7$ and, then, $H_{\widehat{P}}(A_5|C, A \setminus \{A_5\}) = H_{\widehat{P}}(A_5|A_7) = 0$, the full Bayesian classifier $\mathcal{C}$ is preferred to the selective Bayesian classifier $\mathcal{C}^-$ because $-H_{\widehat{P}}(A_5) = -1 < -\log(4096)/(2 \cdot 4096) \cdot (|\Omega(A_5)| - 1) \cdot (2^8 - 1) \approx -0.37$. Although A_5 is redundant for C at all levels, we have that A_5 is wrongly kept in the classifier. Furthermore, an arc between A_5 and A_7 will be always considered by any algorithm for constructing Bayesian network classifiers by maximising the log-likelihood term, because $H_{\widehat{P}}(A_5|A_7,S) = 0$, for any subset of attributes $S \subseteq A \setminus \{A_5, A_7\}$; thus, an arc between A_5 and A_7 will represent the most powerful dependence in the Bayesian network classifier. But, since the MDL-FS score uses an auxiliary structure that includes also this arc, MDL-FS eliminates the influence of A_5 from the classifier, whereas the MDL score wrongly keeps it in the classifier. Similar conclusions we draw for A_6 – when the set of parents includes $\{A_1, A_2, A_4\}$, A_3 – when the set of parents includes A_2, and A_8 - when the set of parents includes A_2.

We conclude that when the MDL score is employed none of the attributes will be removed from $\mathcal{C}$. When the MDL-FS score is employed, the remaining attributes A_1, A_2 and A_4 are irredundant for the class variable at level $|A| - 1$ and are not removed from the classifier. Then, the fully connected classifier has the same conditional auxiliary log-likelihood as the optimal classifier and the same number of attributes. We have that the conditional auxiliary log-likelihood when a complete network is used for the auxiliary network is equal to the conditional entropy $H_{\widehat{P}}(C|A_1, A_2, A_4) = 0$.

6. Learning Bayesian network classifiers with MDL-FS in practice

In the previous section we have investigated general properties of the feature-selection behaviour of the two functions. In this section we use the MDL-FS function in a more practical setting for constructing selective Naive Bayes and TAN classifiers from data using tree structured auxiliary networks of maximum log-likelihood.

Finding an appropriate subset of attributes for inclusion in a classifier amounts to searching the space of all possible selective classifiers, given some predefined measure of quality. Since this search space is infeasible large, often a heuristic algorithm is employed for its traversal. Various different algorithms have been proposed to this end. These algorithms essentially take one of two approaches [6,31,32,42]. Within the *filter approach* [6,33,34,36], feature subset selection is performed in a preprocessing step; within the *wrapper approach* [31,32], feature selection is merged with the learning algorithm. The difference between the two approaches in practice often lies in whether or not the algorithms employ the same measure for the selection of attributes and for measuring performance. In this paper, we will present our fundamental results from both a wrapper and a filter perspective. All algorithms used in this paper are characterised by their starting point(s) in the search space, by the search operator(s) they apply, and by their stopping criterion [5]. Possible starting points in the space of selective classifiers are the *empty classifier* that is built from the empty set of attributes and the *full classifier* that includes all attributes. If the starting point for the search is the empty classifier, then the algorithm typically applies the operator of adding a single attribute; the algorithm is said to perform *forward selection* [32,30]. If the starting point is the full classifier, on the other hand, the algorithm typically applies the operator of removing a single attribute; it then is said to perform *backward elimination* [13,33,36]. The stopping criterion that is commonly employed with the various algorithms, is to stop the traversal of the search space as soon as application of the search operators does no longer result in classifiers of improved quality. We will return to these algorithmic issues in Section 7 where we discuss our experimental results. The MDL or MDL-FS function, for example, are used for comparing the qualities of the classifiers that are supplemented by tree-structured auxiliary networks of maximum log-likelihood. As soon as the algorithm cannot construct a new classifier that improves upon the MDL (or MDL-FS) score of the currently best classifier, the algorithm is halted. In the following, we investigate the feature-selection behaviour of the MDL-FS function in this context.

In this section, we show how to construct selective Naive Bayes and TAN classifiers with the MDL and MDL-FS function and with several other feature selection algorithms implemented in Section 7.

6.1. Learning selective Naive Bayes and TAN classifiers with MDL

Learning a Naive Bayes classifier over a given set of attributes is straightforward as the classifier's graphical structure is uniquely defined. At the beginning of the learning process, we compute the conditional entropies for each attribute given the class variable. Such an algorithm has a time complexity of $O(n \cdot N)$. Learning a selective Naive Bayes classifier, on the other hand, amounts to selecting a graphical structure from among exponentially many alternatives. We recall that the forward-selection algorithm for this purpose starts with the empty Naive Bayes classifier and iteratively adds single attributes that upon removal serve to maximally decrease the MDL score of the classifier. The algorithm stops as soon as adding a single attribute can no longer decrease the classifier's score [32].

As we already have stated in Section 5.1 in Proposition 2, the MDL function tends to eliminate from a Naive Bayes classifier *only* attributes redundant at level 0. Since a Naive Bayes classifier cannot express the information contributed by an attribute at a level higher than level 0, we may look upon an attribute's contribution at level 0 as an *approximation* of its contribution at level $|A| - 1$. Thus, the redundant attributes for the class variable at level 0 are correctly removed from a Naive Bayes classifier.

Learning a TAN classifier over a given set of attributes is more involved than learning a Naive Bayes classifier, because the graphical structure of the TAN classifier is not unique. A well-known search algorithm for learning TAN classifiers [11] starts with a Naive Bayes classifier and iteratively inserts undirected edges between pairs of attributes, under the constraint of acyclicity; the selection of the edges to be inserted is based upon the conditional mutual information of two attributes given the class variable. The algorithm stops adding edges as soon as the undirected graphical structure over the attributes constitutes a tree. After randomly selecting a root for the tree, the edges in the structure are oriented from the root towards the leaves. The resulting TAN classifier is guaranteed to have *maximum log-likelihood* given the data. In the sequel, we assume a

TAN classifier to be of maximum log-likelihood unless explicitly stated otherwise. The time complexity of this algorithm is $O(n^3 \cdot N)$ and is given by the preprocessing step, where the conditional mutual information between each pair of attributes is computed [11].

The forward-selection algorithm for constructing a selective TAN classifier now starts with the empty TAN classifier and iteratively adds single attributes. In each iteration, it computes a TAN classifier over the selected set of attributes by means of the algorithm described above. The MDL function again is used for selecting the attributes to be added as well as for a stopping criterion: the algorithm stops as soon as adding a single attribute cannot result in a TAN classifier of higher score.

In contrast with Naive Bayes classifiers, TAN classifiers can express information at level 1: they can model the relationship of an attribute with the class variable conditional on a single other attribute. Although similar, Proposition 2 from Section 5.1 does not cover this case; when deleting an attribute from a TAN classifier, the resulting classifier is a not necessarily a TAN. Therefore, we need to construct a TAN classifier of maximum log-likelihood over the given (sub)set of attributes. In general, the MDL score wrongly keeps the attributes redundant for the class variable at level 1. Because of the lack of space we refer to [43] for the formal proof.

6.2. Learning selective Naive Bayes classifiers with MDL-FS

Previously, we argued that to be able to exploit the underlying idea of the MDL-FS function, a more complex auxiliary network than the empty structure needs to be used. The auxiliary network should not have a structure too complex, however, because of the number of instances and the computational effort it requires for its construction. The use of *tree-structured auxiliary networks* is motivated by the efficient $(O(n^3 \cdot N)$ learning algorithm from Chow and Liu [44], that is guaranteed to result in a tree-structured network of maximum log-likelihood.

The forward-selection algorithm for learning selective Naive Bayes classifiers starts with the empty Naive Bayes classifier and auxiliary network. The algorithm iteratively adds single attributes, where in each iteration it computes a Naive Bayes classifier and a maximum log-likelihood tree over the selected set of attributes. The MDL-FS function is used for selecting the attributes to be added as well as for a stopping criterion: the algorithm stops as soon as adding a single attribute cannot result in a classifier of smaller score. We observe that this algorithm has also a time complexity of $O(n^3 \cdot N)$ which is given by the preprocessing step, where the conditional mutual information between each pair of attributes is computed [11], and the searching process, where a tree of maximum log-likelihood over the remaining set of attributes is computed each step.

Even though a Naive Bayes classifier can only express the information contributed by an attribute at level 0 for the class variable, a tree structured Bayesian network over A as auxiliary structure can express the attribute's dependency at level 0 with other attributes from A. We now may look upon the attribute's contribution in the auxiliary structure as an *approximation* of its contribution at level 1; furthermore we may look upon this as an *approximation* of its contribution at level $|A| - 1$. Then, the attribute A_i should not be removed from the Naive Bayes classifier. We note that, for identifying redundancy at a higher level, an auxiliary network of higher complexity is required. To conclude, we illustrate the basic idea by means of an example.

Example 3. We consider again the classification problem from Example 1. We note $A = \{A_1, \ldots, A_8\}$. Let us consider a Naive Bayes classifier $\mathcal{C}$ and a selective Naive Bayes classifier $\mathcal{C}^-$ as before. Associated to these classifiers are a tree-structured Bayesian network $\mathcal{S}$ and a selective tree-structured Bayesian network $\mathcal{S}^-$ as before.

We have that A_5 and A_7, where $A_5 \equiv A_7$, are redundant for the class variable C at level 0 and higher. From Corollary 1, with the MDL score, A_5 is removed from $\mathcal{C}$. With the MDL-FS score, the selective classifier $\mathcal{C}^-$ is preferred over the full classifier $\mathcal{C}$, and A_5 is effectively removed. We note that A_5 is removed regardless of the strengths of its relationships with the other attributes. A similar observation holds for the attribute $A_7 \equiv A_5$.

We further recall from Example 1 that the attributes A_1 and A_4 are redundant for the class variable at level 0 but irredundant at higher levels than 1. A full Naive Bayes classifier over $A \cup \{C\}$ only includes the prior probability distribution $P(C)$ and the conditional probability distributions $P(A_1|C)$ and $P(A_4|C)$. The XOR operator that captures the combined influence of A_1 and A_4 on C cannot be modeled by a Naive Bayes classifier. Although these attributes cannot bias the classification as it does not contribute any information to the class variable C at level 0, it adds to the complexity of the classifier and therefore they are correctly removed by the MDL and MDL-FS functions.

Suppose that $A_i \equiv A_8$, which is redundant for C given A_2 and irredundant at level 0. Then $A_j \equiv A_2$ since $A_8 \equiv A_2$ and then the mutual information of A_8 and A_2 is maximal. The MDL-FS function prefers the selective classifier $\mathcal{C}^-$ over the selective classifier $\mathcal{C}$; the attribute A_i is thus removed from the classifier. Unlike the MDL-FS score, the MDL function prefers $\mathcal{C}$ over $\mathcal{C}^-$ whenever $H_{\widehat{P}}(A_2) - H_{\widehat{P}}(A_2|C) > \xi(A_2;C)$. For $N = 32$, for example, we find that $\xi(A_2;C) < \log 32/(2 \cdot 32) \cdot (|\Omega(A_2)| -1) \cdot (|\Omega(C)| - 1) \approx 0.08$. We further have that $H_{\widehat{P}}(A_2) - H_{\widehat{P}}(A_2|C) \approx 0.16$. We conclude that the full classifier $\mathcal{C}$ therefore is always preferred over the selective classifier $\mathcal{C}^-$ and the attribute A_2 is not removed. Similar observations hold for the attribute A_3 that is strongly connected to A_2.

With MDL-FS, after eliminating the redundant attributes at level 0 and 1, there are only two left: A_2 and A_6. Suppose that $A_i \equiv A_2$, where A_2 is irredundant for C at level 0 and higher, and $A_j \equiv A_6$. The MDL-FS function now prefers the full classifier $\mathcal{C}$ over the selective classifier $\mathcal{C}^-$ because, for $N = 32$, for example, we have that $H_{\widehat{P}}(A_2|A_6) - H_{\widehat{P}}(A_2|C) = 0.87 - 0.69 > \xi(A_2;C) = 0.08$. The attribute A_2 is thus not removed from the classifier. Similar observations hold for the attribute A_6 that is irredundant for C at level 0 up to 3.

The selective Naive Bayes classifier yielded for the example dataset by the MDL-FS function is shown in Fig. 2 on the left. The conditional auxiliary log-likelihood of this Naive Bayes classifier is proportional with $CALL(\mathcal{C}, \mathcal{S}|D)/N \approx 0.47$. This score is higher than the conditional auxiliary log-likelihood of the selective Naive Bayes classifier selected by the MDL score -0.02. With a tree-structured auxiliary network of maximum log-likelihood, therefore, the MDL-FS function serves to remove attributes that are redundant at level 0 and/or at level 1 upon feature selection. We observe that the conditional auxiliary log-likelihood of the selective Naive Bayesian selected by the MDL-FS score is higher than the conditional entropy of the class variable given the attribute set A; the conditional auxiliary log-likelihood of the Naive Bayes classifiers when considering a tree-structured auxiliary network of maximum log-likelihood is positive when the auxiliary network has a higher score than the Naive Bayes classifiers.

6.3. Learning selective TAN classifiers with MDL-FS

Learning a TAN classifier over a given set of variables with the MDL-FS function amounts to constructing both a classifier and an auxiliary network from the available data. Upon learning a selective TAN classifier with the MDL-FS score, moreover, learning the two networks is performed iteratively. In this section, we focus again on the use of tree-structured auxiliary networks with the MDL-FS function. In addition to the use of tree-structured networks of maximum log-likelihood as in the previous section, we also study the use of the attribute tree of the constructed TAN classifier for its associated auxiliary network. Note that using the attribute tree of a TAN classifier with the MDL-FS function serves to substantially reduce the computational effort involved in the learning task. We observe that this algorithm has also a time complexity of $O(n^3 \cdot N)$ as the algorithm for constructing TANs with the MDL score.

In the following, we investigate the ability of the MDL-FS function to identify and remove, from a TAN classifier, redundant attributes at different levels. We study the use of maximum log-likelihood tree-structured auxiliary networks for this purpose and establish the condition under which the MDL-FS function with such a network removes an attribute from a classifier. Although similar, the following property is not a direct consequence of Proposition 4; when deleting an attribute from a TAN classifier, now, the selective classifier is also a TAN classifier of maximum log-likelihood which might be different from the selective classifier obtained from the full TAN by deleting the given attribute and its incident arcs. Similar observations hold also for the selective tree structure auxiliary network. The following observations pertains to an attribute that is either an internal node or a leaf in the attribute tree of the TAN classifier and in the auxiliary tree under consideration. Similar observations also hold for the attribute that constitutes the root of the tree.

We illustrate the basic idea by means of an example.

Example 4. Again, we consider the classification problem from Example 1. We note $A = \{A_1, \ldots, A_8\}$. Let us consider a full TAN classifier $\mathcal{C}$ over $A \cup \{C\}$, its associated tree-structured auxiliary structure $\mathcal{S}$, and a selective TAN classifier $\mathcal{C}^-$ over $(A \setminus \{A_i\}) \cup \{C\}$ and its associated tree structure auxiliary network $\mathcal{S}^-$ as before. We now compare the MDL-FS score of $\mathcal{C}$ and $\mathcal{S}$, with the MDL-FS score of $\mathcal{C}^-$ and $\mathcal{S}^-$.

Since the conditional mutual information of A_5 and A_7 given C is maximal and both variables do not have strong relationships with other attributes, any full TAN classifier of maximum log-likelihood will include an edge between the two attributes. Similar observations hold for the auxiliary network. Since $A_5 \equiv A_7$, A_7 perfectly replaces A_5 in the classifier; we assume, without loss of generality, that A_5 does not have any children. From the equivalence of the two attributes, we now observe that $H_{\widehat{P}}(A_5|A_7, C) = 0$. We find that any TAN classifier $\mathcal{C}$ that contains both A_5 and A_7 has a lower MDL score than the selective classifiers that are constructed from $\mathcal{C}$ by removing A_5 or A_7. Recall that A_5 and A_7 are redundant for C at all levels. However, any TAN classifier $\mathcal{C}$ that contains both A_5 and A_7 has a higher MDL-FS score than the selective TAN classifiers that are constructed from $\mathcal{C}$ by removing A_5 or A_7. These two attributes will therefore be correctly removed. Similar observations hold for the attributes A_3, and A_8 which are redundant for C at level 1 and higher.

The attributes $A_1,\ A_2,\ A_4$ and A_6 again are identified as being irredundant at level 1 and therefore are correctly kept in the classifier by both scoring functions.

With a tree-structured auxiliary network of maximum log-likelihood, therefore, the MDL-FS function serves to remove from TAN classifiers attributes that are redundant at level 1 upon feature selection. A selective TAN classifier yielded for the

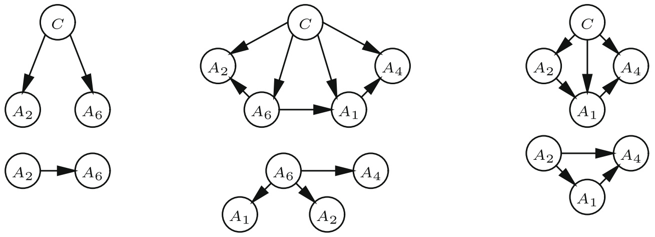

Fig. 2. The selective Naive Bayes classifier and its associated tree structured auxiliary network (left), the selective TAN classifier and its tree-structured auxiliary network (middle) and the selective TAN classifier and its complete auxiliary network (right), constructed with the MDL-FS function.

example dataset by the MDL-FS function is shown in Fig. 2 in the middle. We observe that the conditional auxiliary log-likelihood of this classifier is proportional with $CALL(\mathcal{C}, \mathcal{S}|D)/N = 0.75$. This score is lower than the conditional entropy of the selective Naive Bayesian classifier obtained by the MDL score 0.84 but higher than the conditional entropy of the class variable given the minimum set of irredundant attributes $\{A_1, A_2, A_4\}$ that is 0.

We observe that the MDL-FS function, when using a tree-structured auxiliary structure, tends to eliminate attributes redundant at level 1 within the allowed noise modeled by the penalty term, $\log N/(2 \cdot N)(|\mathcal{C}| - |\mathcal{C}^-|)$, from a TAN classifier. Since we look upon the attribute's contribution at level 1 as an approximation of its contribution at level $|A| - 1$, then A_i is correctly removed from the classifier. To the attributes that are redundant for the class variable at a level higher than level 1, a similar analysis applies as the previous one. We note that, for identifying redundancy at a higher level, an auxiliary network of higher complexity is required.

The attributes that are redundant for the class variable at a level higher than level 1 tend not to be removed from the classifier. To study if such an attribute should indeed be removed, we now consider an attribute A_i that is redundant for the class variable C at level 0 yet irredundant at level 1. If it is redundant at the highest level $|A| - 1$, then the attribute should not contribute to the classification. In fact, it may bias the classification by its contribution to the class variable at level 1 if it is not removed. We note, however, that the MDL-FS function tends not to remove the attribute. If, on the other hand, the attribute A_i is irredundant at level $|A| - 1$, then it should in essence contribute to the classification at this level. We recall, however, that a TAN classifier can only express the information contributed by an attribute at levels 0 and 1. If we may look upon the attribute's contribution at levels 0 and 1 as an approximation of its contribution at level $|A| - 1$, then A_i should not be removed from the classifier. The MDL-FS function indeed tends to not remove it.

We would like to note that, in practical applications, generally good feature-selection results are obtained with the MDL function for Naive Bayes classifiers [32]. Apparently, the function's ability to identify and remove attributes that are redundant for the class variable at level 0 suffices to yield relatively simple classifiers of good accuracy. However, the MDL-FS score is reducing even more redundant attributes – that are attributes redundant at level 1 – since the contribution of these attributes are captured by the tree-structured auxiliary network. Thus, we consider that MDL-FS is more suited for the task of constructing selective Naive classifier from data with the reduction of redundant attributes at chosen levels.

6.4. Related work

After the crisp theoretical definitions of "useful" and "not useful" attributes for the class variable given a set of attributes based on the associated conditional probability, for practical use, Tsamardinos and Aliferis [6] and Koller and Sahami [33] propose heuristics where they consider only the relationships between two attributes given the class variable at the time. Koller and Sahami [33] propose an approximative iterative algorithm that uses the cross-entropy measure of two attributes given the class variable to find an approximative Markov Blanket and therefore the feature selection is independent on the selected family of Bayesian network classifiers. The cross-entropy of A_i and A_j given C, using the conditional entropy is $H_P(C|A_i, A_j) - H_P(C|A_i)$, which is equivalent with the amount of irredundancy of C for A_i given A_j from Section 3. This algorithm iteratively deletes the attribute with the smallest cross-entropy for the class variable given one other attribute from the dataset until some stopping criteria is met (e.g. some predefined number of attributes are deleted or the cross-entropy of the remaining attributes is larger than a threshold γ). Using Example 1, Koller and Sahami's algorithm deletes, in a random order, the attributes A_5 and A_7 – they are redundant for C and for the other attributes; A_8 and A_3 – they are conditionally independent for C given A_2; and A_1 and A_4 – they are redundant for C given A_2. Since this algorithm deletes attributes only by looking at its redundancy for C given another attribute, the algorithm fails to identify that A_1 and A_4 are important for the classification task. Recall that the MDL-FS score for both selective Naive Bayes and TAN classifiers using a tree auxiliary structure of maximum log-likelihood correctly identifies the attributes A_1, A_4, A_2 and A_6 as "useful" for the classification task.

Since Tsamardinos and Aliferis's, Yu and Liu [45]'s and Pena et al. [8] heuristics also consider the interactions between attributes one at a time in an iterative algorithm similar with Koller and Sahami's algorithm, they have similar properties. Thus, they will also wrongly delete the attributes A_1 and A_4 from our example.

Another popular algorithm for feature selection is RELIEF [46,47]. This heuristic and its later developments, Gadat and Younes [48] and Sun [49], attributes a weight (gain) to each feature according with its importance for the classification task. The early developments of RELIEF evaluates the importance of an attribute according to the class variable and therefore the attributes that are conditionally independent given the class variable, like A_8 and A_3 from our examples, will be all deemed important for the classification task and thus wrongly kept in the classifier. An advantage of RELIEF is that it can handle missing instances. The later versions of RELIEF overcomes (some) of these problems by analysing the contribution of an attribute in the context of the class variable and also the other features in the dataset.

Fleuret [50] related the conditional probability definition to the notion of conditional mutual information. In his heuristic, he iteratively adds attributes that have high conditional mutual information scores with the class variable given each of the attributes that are already picked. Note that, in our example, the attributes A_2 and A_6 will be the first to be picked by the algorithm. However, the attributes A_1 and A_4 are again not picked because the conditional mutual information of A_2 and C given A_1, and A_4 respectively, are low.

In fact (conditional) mutual information and its variants are rather popular methods for feature selection used in many recent papers. Huang et al. [51], for example, introduce some parameters to learn from data when attributes are relevant or irredundant for the class variable. Recall that we use the MDL's penalty term to deal with noisy datasets. Since the (conditional) mutual information, however, only considers the relationship between two attributes and the class variable at a time, the more complex relationships between three and more attributes cannot be captured. For example, the interaction between the three relevant attributes for the class variable A_1, A_2 and A_4 is not captured and the attributes A_1 and A_4 are wrongly deleted. Liu et al. [34] extends the work of Huang et al. [51] by expanding the concept of mutual information between a feature and the class variable given the rest of the features in the classifier. However, their proposed heuristics, at first, select attributes relevant only for the class variable, and therefore, again, for our example, the attributes A_1 and A_4 that are irredundant for the classification task but redundant for C at level 0 are eliminated.

In the sequel, one can use Pearson's correlation to study the paired correlation between two variables [42]. We now briefly review the concept of redundancy build upon Pearson's correlation coefficient used by Hall [30] and Liang et al. [52]. Informally speaking, he defines a subset of attributes S_c to be important for the classification task if each attribute from S_c is correlated to the class variable and not correlated to any other attribute from S_c. He measures the correlation between two variables using the difference between the entropy of a variable and the conditional entropy between the variable given the other variables. Upon applying Hall's concept of redundancy to our example, we find that the correlations between A_1 and C, between A_4 and C, and between A_1 and A_4 are considered. Since the relationship between the attributes A_1 and A_4 on the one hand and the class variable C on the other hand is not considered, the two attributes A_1 and A_4 would be deemed irrelevant for the classification task.

Our approach differs from previous work in the sense that we work with the conditional probability distribution directly and include the conditional log-likelihoods in an MDL-based function. As a consequence, the impact of these methods as compared with MDL-FS score for feature selection task is very different although they use similar definitions based on conditional entropy of the class variable given a set of attributes to denote "useful" and "not useful" set of attributes. The analysis is in favour for our method. Informally speaking, the previous algorithms will require that an attribute is irredundant for the class variable given *all* other attributes from the selected set. In our example, the previous algorithms will fail to identify the attributes A_4 and A_1 as important for the classification task because these attributes are redundant for the class variable given A_2 and thus their relevancy for the class variable at level 1 given each other is overlooked. In the previous sections we have shown that the MDL-FS score overcomes this drawback by: capturing first the strongest relationships between attributes and the class variable and evaluating a sum of terms that indicates the amount of irredundancy a set of attributes has for the class variable.

Other interesting approaches, but somehow incomparable with our algorithm since they use both labelled and unlabelled records are the algorithms that uses a mixture between supervised and unsupervised learning [53,54].

7. Experimental results

In the following, we study the feature-selection behaviour of the MDL and MDL-FS functions in a practical setting by constructing selective Naive Bayes and TAN classifiers from various different datasets using both functions. Then, we compare them with three other popular methods from feature selection literature: a wrapper method which uses the accuracy measure for training and testing the selective classifiers [32], Koller and Sahami's [33] and Hall's [30] filter methods.

For our experimental study, we use 15 datasets: 13 from the UCI Irvine repository and two artificially generated datasets. The characteristics of some of these UCI datasets are thoroughly analyzed in literature. For example, the *chess* dataset, Hall [30] obtains accurate selective Naive Bayes classifiers over just 3 out of 36 attributes. From the *mushrooms* dataset simple logic rules can be extracted that contain only a small number of attributes (e.g. rules with only 2, 3 or 4 out of 22 attributes might have an accuracy between 98% and 100%). For the *splice* dataset, Domingos and Pazzani [24] point out that the most accurate classifiers were Naive Bayes. The *oesoca dataset* was generated from a hand-crafted Bayesian network in the field of esophageal cancer. The *artificial dataset* was generated by copying the 32 instances of Example 1, 100 times, resulting in a dataset with 3200 instances. We used the method of Fayyad and Irani to discretize any continuous attributes in the various datasets [55] and we eliminate any incomplete instances.

In our study, we used the 15 datasets for learning several Naive Bayes classifiers and TAN classifiers, with different algorithms and different scoring functions. In each experiment, we split each dataset randomly into a *training set* and a *test set* at a 2:1 ratio; the training set was used to construct the classifier and the test set was used to establish the performance of the constructed classifier. We observe that in this case the training and test sets will be at different size. Thus, to fairly measure the conditional auxiliary log-likelihood, we divide it by the size of the training set. Furthermore, we construct auxiliary networks of the same complexities for all methods on the selected attributes in order to compute the conditional auxiliary log-likelihood values. We repeated each experiment 50 times, each time splitting the dataset anew in a training set and a test set to be able to compare the mean scores.

Prior to learning the classifiers, we established from the training set the numbers of redundant attributes at the levels 0 and 1. We report the averages and associated standard deviations obtained over all runs in percentage in Table 2. Recall that in Example 1, four attributes are redundant for the class variable at level 0 – they are A_1, A_4, A_5 and A_7 – and four attributes are redundant for the class variable at level 1 given one other variable – they are A_3, A_5, A_7 and A_8. However, when we split the artificial dataset in training and test set as described before, there is noise inherent to this process. We observe that most attributes are deemed irredundant for C except the copies A_5 and A_7 that are redundant for C at level 1 regardless of the splitting in training sets.

In our first experiment, we constructed from each dataset a full Naive Bayes classifier and a full TAN classifier. The basic idea of this experiment was to establish baseline performances to compare the selective classifiers resulting from the other experiments against. Table 2 summarizes the results from the first experiment; it reports for each dataset the accuracy and the conditional auxiliary log-likelihood divided by the size of the training set of constructed classifiers with the algorithms described in previous sections. We would expect the performance of the TAN classifier constructed from a specific dataset which has interactions between attributes, to be higher than that of the corresponding Naive Bayes classifier, since a TAN classifier takes them into consideration while a Naive Bayes classifier does not. For *mushrooms* and *pima* the difference in accuracy of NB and TAN is not significant. Domingos and Pazzani [24] show that the improvement in accuracy classification by using a more complex Bayesian network classifier than the Naive Bayes classifier is not necessary to be significant. For the *splice*, *credit* and *german* datasets we find that the accuracy of the Naive Bayes classifier exceeds that of the TAN classifier. The lower accuracy of these TAN classifiers can be explained by the negative effect of the relationships between redundant attributes over the classification task. The accuracies of the selective TANs generated from the *credit* database in Table 3 are about the accuracy of the full NB. For the other 10 datasets, we found statistical significantly higher accuracies for TAN than for Naive Bayes classifiers.

Furthermore, for all databases, we found significantly higher conditional auxiliary log-likelihoods for TAN than for Naive Bayes classifiers. Recall that a TAN classifier has equal or higher log-likelihood than a Naive Bayesian classifier over the same attributes set; thus, it has equal or higher conditional auxiliary log-likelihood with the same auxiliary network. We observe that, for all but three datasets, *splice*, *car* and *nursery*, the conditional auxiliary log-likelihoods of the full TAN classifiers are positive but the ones of the Naive Bayes classifiers are negative. Then, the attributes are strongly connected to each other given the classifier since the conditional auxiliary log-likelihoods of TANs are higher than of NBs. But the attributes are weakly connected between them in the absence of the class variable – since a low log-likelihood for the auxiliary structure dominates the log-likelihood of the classifier.

For the second experiment, Table 3 reports the percentages of selected attributes and the classification accuracies of the selective Naive Bayes and TAN classifiers constructed with five feature selection methods. We use an forward algorithm to optimize the MDL and MDL-FS functions, two popular feature-selection specific scores, the accuracy method for evaluating a specific classifier [56,32] and Hall [30]'s algorithms described in the previous section. The fifth method is the backward elimi method of Koller and Sahami [33]'s. The disadvantage of the forward elimination and backward elimination is that they can stuck in the local optimum. We have also run experiments with a standard genetic algorithm [57] that overcomes this disadvantage. However, due to the lack of space and because of the similar results with forward elimination for most of the datasets, we do not show these results except for some interesting cases.

According with the definition of the wrapper and filter approach [32,6], the MDL and MDL-FS algorithms are wrappers when the performance measure is the conditional auxiliary log-likelihood, and they becomes filters when the performance measure is the classification accuracy. Whereas, if the accuracy measure is used instead of the MDL score, the resulting algorithm is a wrapper when the algorithm evaluates selective NB and TAN classifiers over the test set using the accuracy measure, and a filter when the algorithm evaluates the performance of the specific classifiers with the conditional auxiliary log-likelihood.

We observe that, upon constructing a selective Naive Bayes classifier, with the MDL-FS function, more selective classifiers were obtained than with the MDL function. The more selective behaviour was expected of the MDL-FS function as it removes not just attributes that are redundant at level 0, but also many attributes that are redundant for the class variable at level 1. We observe that indeed, for the artificial dataset, in addition to the four attributes redundant at level 0, attributes redundant at level 1 for the class variable – these are A_3 and A_8 – were removed. In the sequel, upon constructing a TAN classifier with

Table 2
The characteristics of the five datasets.

data	inst	nr attrib	% red attr		NB		TAN	
			$l = 0$	$l = 1$	% acc	*CALL*	% acc	*CALL*
oesoca	10,000	25	0 ± 0	0 ± 0	71 ± 1	-1.63 ± 0.02	74 ± 0	1.46 ± 0.01
artif	3200	8	0 ± 0	75 ± 0	81 ± 4	-1.64 ± 0.02	100 ± 0	0.84 ± 0.01
chess	3196	36	2 ± 1	1 ± 1	88 ± 1	-3.11 ± 0.04	92 ± 1	0.89 ± 0.01
mush	5644	22	5 ± 0	14 ± 0	100 ± 0	-6.23 ± 0.03	100 ± 0	2.77 ± 0.02
splice	3000	60	0 ± 0	0 ± 0	96 ± 1	-0.20 ± 0.05	95 ± 1	3.43 ± 0.06
spam	4601	57	4 ± 0	4 ± 0	91 ± 1	-3.78 ± 0.04	92 ± 0	2.33 ± 0.03
adult	32,561	14	7 ± 0	14 ± 0	83 ± 0	-1.70 ± 1.88	84 ± 0	0.25 ± 0.26
car	1728	6	0 ± 0	0 ± 0	85 ± 2	0.68 ± 0.02	94 ± 1	0.91 ± 0.02
nursery	12,960	8	0 ± 0	0 ± 0	91 ± 0	1.29 ± 0	94 ± 0	1.46 ± 0.01
conn	67,557	42	0 ± 0	0 ± 0	72 ± 0	-6.63 ± 0	77 ± 0	0.39 ± 0
german	1000	20	30 ± 0	30 ± 0	74 ± 2	-0.96 ± 0.05	68 ± 3	0.51 ± 0.03
votes	435	16	0 ± 0	0 ± 0	97 ± 0	-0.94 ± 0.0	98 ± 0	1.34 ± 0.0
spect	267	23	0 ± 0	0 ± 0	79 ± 4	-3.42 ± 0.17	81 ± 4	0.59 ± 0.06
pima	768	8	37.5 ± 0	37.5 ± 0	78 ± 2	-0.03 ± 0.03	78 ± 2	0.34 ± 0.02
credit	690	15	6 ± 0	13 ± 0	86 ± 2	-1.53 ± 0.07	78 ± 3	1.04 ± 0.05

Table 3
The feature-selection results obtained for Naive Bayes and TAN classifiers.

data	MDL-FS			MDL			Acc			K&S			Hall		
	% sel	% acc	*CALL*	% sel	% acc	*CALL*	% sel	% acc	*CALL*	% sel	% acc	*CALL*	% sel	% acc	*CALL*
NB															
oesoca	26 ± 2	69 ± 1	0.97 ± 0.01	72 ± 2	72 ± 1	−0.31 ± 0.14	43 ± 9	72 ± 1	0.37 ± 0.25	24 ± 1	70 ± 1	0.83 ± 0.03	24 ± 1	68 ± 1	0.90 ± 0.01
artif	25 ± 0	87 ± 1	0.47 ± 0.01	50 ± 0	81 ± 1	−0.48 ± 0.01	12.5 ± 0	88 ± 1	0.29 ± 0.01	26 ± 0	87 ± 2	0.43 ± 0.03	25 ± 0	87 ± 1	0.46 ± 0.01
chess	13 ± 2	94 ± 0	0.41 ± 0.01	58 ± 3	88 ± 1	−1.50 ± 0.22	14 ± 1	95 ± 1	0.40 ± 0.01	11 ± 0	90 ± 1	0.36 ± 0.01	8 ± 0	91 ± 1	0.38 ± 0.08
mush	5 ± 0	98 ± 0	0.86 ± 0.01	92 ± 3	100 ± 0	−6.08 ± 0.14	14 ± 0	100 ± 0	0.23 ± 0.01	37 ± 1	100 ± 0	−2.41 ± 0.07	9 ± 1	100 ± 0	0.52 ± 0.01
splice	22 ± 1	96 ± 1	1.59 ± 0.03	47 ± 2	96 ± 0	1.25 ± 0.05	19 ± 4	95 ± 1	1.40 ± 0.06	29 ± 1	96 ± 1	1.46 ± 0.05	10 ± 1	94 ± 1	1.25 ± 0.02
spam	22 ± 1	93 ± 0	0.88 ± 0.02	96 ± 0	90 ± 1	−3.74 ± 0.06	25 ± 5	93 ± 1	0.30 ± 0.20	18 ± 1	91 ± 1	−0.11 ± 0.07	25 ± 2	80 ± 1	−1.13 ± 0.16
adult	32 ± 4	85 ± 0	0.36 ± 0	83 ± 0	83 ± 0	−3.23 ± 0.01	40 ± 6	85 ± 1	0.07 ± 0.17	36 ± 1	83 ± 0	0.11 ± 0.05	36 ± 1	83 ± 0	0.11 ± 0.03
car	81 ± 6	85 ± 2	0.65 ± 0.04	82 ± 5	85 ± 2	0.68 ± 0.02	68 ± 34	80 ± 7	0.50 ± 0.28	83 ± 0	86 ± 2	0.68 ± 0.02	83 ± 0	86 ± 1	0.68 ± 0.02
nursery	87 ± 2	90 ± 1	1.28 ± 0	87.5 ± 0	90 ± 0	1.28 ± 0	99 ± 4	90 ± 1	1.29 ± 0.01	37.5 ± 0	88 ± 0	1.23 ± 0	37.5 ± 0	88 ± 0	1.23 ± 0
german	10 ± 1	72 ± 2	0.12 ± 0.01	44 ± 9	75 ± 0	−0.02 ± 0.15	19 ± 10	74 ± 3	0.01 ± 0.07	35 ± 0	67 ± 0	0.05 ± 0	11 ± 2	70 ± 2	0.10 ± 0.01
conn	9.5 ± 0	70 ± 2	0.16 ± 0	86 ± 2	72 ± 0	−6.26 ± 0	65 ± 4	73 ± 0	0.23 ± 0	2 ± 0	66 ± 0	0.01 ± 0	2 ± 0	66 ± 0	0.01 ± 0
votes	14 ± 0	97 ± 1	0.87 ± 0.03	87.5 ± 0	93 ± 3	−0.59 ± 0.19	12 ± 5	97 ± 2	0.95 ± 0.16	56 ± 6	92 ± 3	0.13 ± 0.07	57 ± 5	92 ± 3	0.12 ± 0.24
spect	14.5 ± 3	77 ± 4	0.20 ± 0.03	85 ± 9	79 ± 4	−2.48 ± 0.55	7 ± 4	78 ± 3	0.07 ± 0.04	13 ± 4	77 ± 5	0.03 ± 0.07	13 ± 4	77 ± 5	0.03 ± 0.08
pima	37 ± 3	76 ± 2	0.26 ± 0.02	62.5 ± 0	78 ± 2	−0.03 ± 0.03	45 ± 14	78 ± 3	0.34 ± 0.06	41 ± 6	78 ± 2	0.22 ± 0.06	41 ± 6	78 ± 2	0.22 ± 0.06
credit	25 ± 5	86 ± 2	0.53 ± 0.03	58 ± 6	87 ± 2	−1.15 ± 0.33	18 ± 8	86 ± 2	0.44 ± 0.13	26 ± 3	86 ± 2	0.49 ± 0.07	26 ± 2	86 ± 2	0.49 ± 0.09
TAN															
oesoca	47 ± 5	73 ± 1	1.24 ± 0.05	98 ± 2	74 ± 1	1.45 ± 0.02	64 ± 6	75 ± 1	1.32 ± 0.03	24 ± 1	71 ± 1	1.08 ± 0.02	24 ± 1	68 ± 1	1.00 ± 0.01
artif	50 ± 0	100 ± 0	0.76 ± 0.01	75 ± 0	100 ± 1	0.84 ± 0.01	12.5 ± 0	100 ± 1	0.30 ± 0.01	26 ± 0	88 ± 1	0.49 ± 0.03	25 ± 0	88 ± 1	0.46 ± 0.01
chess	42 ± 4	93 ± 1	0.86 ± 0.02	97 ± 1	92 ± 1	0.90 ± 0.02	14 ± 0	95 ± 0	0.68 ± 0.01	11 ± 0	90 ± 1	0.63 ± 0.01	8 ± 0	91 ± 1	0.61 ± 0.01
mush	51 ± 4	100 ± 0	2.56 ± 0.05	94 ± 2	100 ± 2	2.77 ± 0.03	14 ± 0	100 ± 0	0.98 ± 0.01	37 ± 1	100 ± 0	1.86 ± 0.05	9 ± 1	100 ± 0	0.95 ± 0
splice	14 ± 1	94 ± 1	1.75 ± 0.05	34 ± 1	96 ± 0	2.47 ± 0.04	19 ± 4	95 ± 1	1.76 ± 0.15	29 ± 1	96 ± 1	2.30 ± 0.04	10 ± 1	94 ± 1	1.40 ± 0.02
spam	59 ± 4	93 ± 0	2.01 ± 0.08	95 ± 1	92 ± 1	2.34 ± 0.04	27 ± 4	93 ± 1	1.14 ± 0.10	18 ± 1	91 ± 1	1.30 ± 0.05	25 ± 2	83 ± 1	0.44 ± 0.03
adult	38 ± 4	84 ± 1	0.39 ± 0.02	85 ± 2	85 ± 0	0.51 ± 0	54 ± 7	86 ± 0	0.45 ± 0.02	36 ± 1	83 ± 0	0.37 ± 0.01	36 ± 1	83 ± 0	0.37 ± 0.01
car	33 ± 0	77 ± 1	0.53 ± 0.01	33 ± 0	77 ± 1	0.53 ± 0.01	77 ± 31	88 ± 10	0.72 ± 0.32	83 ± 0	93 ± 1	0.89 ± 0.02	83 ± 0	93 ± 1	0.90 ± 0.02
nursery	39 ± 4	89 ± 0	1.35 ± 0.01	40 ± 6	89 ± 1	1.35 ± 0.02	100 ± 2	93 ± 1	1.45 ± 0.01	37.5 ± 0	88 ± 0	1.34 ± 0	37.5 ± 0	89 ± 0	1.34 ± 0
german	5 ± 0	69 ± 2	0.10 ± 0.01	5 ± 1	69 ± 2	0.10 ± 0.01	18 ± 10	74 ± 4	0.10 ± 0.07	35 ± 0	67 ± 0	0.05 ± 0	11 ± 2	70 ± 2	0.12 ± 0.02
conn	53 ± 0	76 ± 0	0.35 ± 0	100 ± 0	76 ± 0	0.38 ± 0	90 ± 4	76 ± 0	0.38 ± 0.01	2 ± 0	66 ± 0	0.01 ± 0	2 ± 0	66 ± 0	0.01 ± 0
votes	9 ± 0	97 ± 2	0.85 ± 0.03	93 ± 4	93 ± 3	1.34 ± 0.07	12 ± 3	97 ± 1	0.83 ± 0.04	56 ± 6	94 ± 3	1.09 ± 0.07	57 ± 5	94 ± 3	1.10 ± 0.06
spect	6 ± 3	79 ± 4	0.14 ± 0.05	100 ± 0	83 ± 4	0.57 ± 0.04	7 ± 3	79 ± 4	0.06 ± 0.04	13 ± 4	78 ± 4	0.16 ± 0.05	13 ± 4	79 ± 4	0.16 ± 0.05
pima	15 ± 7	74 ± 2	0.20 ± 0.03	62 ± 0	78 ± 2	0.34 ± 0.02	44 ± 11	77 ± 3	0.28 ± 0.03	41 ± 6	77 ± 2	0.30 ± 0.03	41 ± 6	77 ± 2	0.30 ± 0.03
credit	23 ± 4	85 ± 2	0.58 ± 0.03	51 ± 5	86 ± 2	0.64 ± 0.03	17 ± 7	86 ± 2	0.50 ± 0.05	26 ± 3	84 ± 3	0.61 ± 0.05	26 ± 2	84 ± 2	0.58 ± 0.04

the MDL function, hardly any attributes were removed. We recall that the MDL function tends to remove redundant attributes only if they are very weakly related with the other attributes. From the feature-selection results obtained, we thus have that, apparently, most redundant attributes show a relatively strong relationship with one or more other attributes. It is interesting to note that, for the artificial dataset, the redundant attributes at level 0, A_5 and A_7 are not included in the TAN classifier scored with MDL even though they are copies of each other because the use of the forward selection method. However, when backward elimination and genetic algorithms are used, the (selective) TAN classifier that includes these two irrelevant features are preferred over the selective TAN classifiers that do not include them. When selective TANs are generated using the MDL-FS score redundant attributes at level 0 and 1 were indeed eliminated; the resulting TAN classifier is presented in Fig. 2 in the middle. We note that, although with the MDL-FS function far more selective classifiers were yielded, the accuracies obtained are approximately the same as with the MDL function. In the sequel, the selective TAN and NB classifiers have similar accuracy as the full TAN and NB, respectively, for all datasets except the *car* dataset. When running the genetic algorithms with MDL and MDL-FS score, we notice that the accuracy improves when the number of eliminated attributes decreases suggesting that the feature selection is not useful for this dataset. There are also datasets (i.e., *artif*, *chess* and *spam* for NB) where the selective classifiers have better performance than the full classifiers.

The conditional auxiliary log-likelihood values for the selective Naive Bayes are positive for the MDL-FS function and negative for the MDL function (except again for *car* and *nursery* datasets) indicating that the attributes that are weakly correlated with the class variable and strongly correlated with the other attributes in the classifier are eliminated. In opposition, the conditional auxiliary log-likelihoods for the selective TAN and the MDL-FS function are lower than for the MDL function. This is the effect of balancing between the model and the representation of the MDL(-FS) score. From Section 4 we recall that the conditional auxiliary log-likelihood score increases when the complexity of the classifier increases but also the cost to represent (compress) this TAN increases. We note that the feature-selection behaviour of the two functions is relatively robust as the standard deviation of the average accuracies is quite small. Higher standard deviations of the conditional auxiliary log-likelihoods were obtained for the MDL scoring than for the MDL-FS function, showing that the latest function models the conditional auxiliary log-likelihood better than the earlier one.

The accuracy method is a wrapper method since it evaluates the accuracy itself in constructing selective classifiers. Therefore, it is also by far the slowest method. The other feature selection algorithms tested here are filters and do not win from the accuracy method on the accuracy score, but are much faster. The exception is the *car* dataset where the forward-selection algorithm gets stuck in a local optimum. Alternatively, the genetic algorithm generates less selective TANs which keep $96\% \pm 7$ of the attributes with the high accuracy $94\% \pm 1$ and less selective NB keeping $90\% \pm 10$ of attributes but with the accuracy of $85\% \pm 2$. For most of the datasets the accuracy method is slightly better than those of the MDL-FS and MDL scores. The exception is again the *car* dataset where the MDL-FS score is too selective whereas the MDL score is stuck in a local optima. However, the accuracy algorithm is computationally very expensive since we have to compute each step the accuracy over the training set. Recall that the other quality measures used in this section calculates the required probabilities from the training set only once and store them for later use. The amount of evaluation also increases with the complexity of the classifiers we construct. The conditional auxiliary log-likelihoods of this function are, for most datasets, slightly worse than the conditional auxiliary log-likelihoods for our MDL-FS function. However, we observe that the standard deviations are considerable higher for this function than for the MDL-FS score because various classifiers can have good accuracies but their conditional auxiliary log-likelihood can be rather large. For example, for the artificial dataset, all TAN classifiers which contain the attributes $A_1,\ A_2$ and A_4 and an arc between A_1 and A_4 have the accuracy 1 (e.g. full TAN classifiers and the selective TAN obtained with the MDL-FS function) whereas their conditional auxiliary log-likelihood can vary. For the artificial dataset, the greedy algorithm might fail in learning the complex relationship between attributes (e.g. the XOR relation between A_1 and A_4). However, for the same dataset, the TAN classifier obtained using backward elimination is the one illustrated in Fig. 2 on the right; in Section 5, we showed that we need a more complex auxiliary network than the tree structured one to learn this classifier with the MDL-FS score.

Koller and Sahami's approximative method depends heavily on the threshold γ over which an attribute is considered useful for the classification task. The higher the threshold γ the more selective is this method, but the performance of the selective Naive Bayesian classifier can be diminished. In this paper, we present results with $\gamma = 0.05$; for all datasets the selection is very strong and the accuracies are comparable with the full Naive Bayesian classifiers. Hall's and Koller and Sahami's algorithms, on average, for Naive Bayes classifiers, have a comparable performance regarding the number of selected attributes and the classification accuracy with the algorithm using the MDL-FS function. However, the conditional auxiliary log-likelihood, on average, even though it is positive, is considerable smaller than the conditional auxiliary log-likelihood obtained using the MDL-FS score. When we compare these algorithms with the other algorithms for constructing TAN classifiers, we find they are very selective, with slightly worse accuracies, but with much worse conditional auxiliary log-likelihoods. The exception is, again, the *car* dataset for which the MDL-FS method is too selective and considerably diminishes accuracy. It is interesting to note that the TAN classifier constructed over the attributes selected with the MDL-FS score on NBs from the *car* dataset has the same performance as Hall's and Koller and Sahami's algorithms. For the artificial dataset, both algorithms select on average only two attributes A_2 and A_6, as when the MDL-FS score for Naive Bayesian classifiers is used. As we have showed in the previous section, neither of these two methods, however, are able to identify the XOR relationship between A_1 and A_4.

To conclude our experimental section, in Table 4, we compare the performance of the five discussed algorithms and the non-selective Bayesian network classifier (full BNC) for Naive Bayes and TAN classifiers using three methods. We first mea-

Table 4
Three measures to compare the performance of the five scoring methods and the non-selective (full) Bayesian network classifier (BNC) for Naive Bayes and TAN classifiers.

	alg	Mean			Nr. wins MDL-FS			Rank		
		% sel	% acc	*CALL*	% sel	% acc	*CALL*	% sel	% acc	*CALL*
NB	Full BNC	100	85	−1.99	15–0	4–8	13–1	6.0	2.13	5.2
	MDL-FS	28	85	0.63				2.2	3	1.33
	MDL	72	84	−1.45	15–0	5–7	13–0	4.73	2.53	4.53
	Acc	33	86	0.46	9–6	2–8	11–4	2.73	1.93	2.8
	K&S	32	84	0.23	9–6	7–4	14–0	2.6	3.2	2.93
	Hall	27	83	0.36	8–6	9–3	14–0	2.07	3.73	2.8
TAN	Full BNC	100	87	1.24	15–0	4–8	1–14	5.73	2.27	1.47
	MDL-FS	29	85	0.89				2.33	3.07	3.93
	MDL	72	86	1.1	13–0	3–7	0–12	4.4	2.33	1.87
	Acc	38	87	0.73	10–5	1–10	7–7	3	2.13	4.2
	K&S	32	84	0.83	7–8	10–3	9–6	2.47	4.13	4.07
	Hall	27	84	0.65	6–8	7–3	9–5	2	3.8	4.4

sure the mean and standard deviation of all datasets for: (i) the number of selected attributes, (ii) the accuracy and (iii) the conditional auxiliary log-likelihood (CALL). For the second comparison method, we count the number of significant wins–losses of the MDL-FS function when compared with the other five methods using the Wilcox tests with a *p-value* of 0.05. The last comparison uses a ranking scheme of the discussed scoring methods for the 15 tested datasets. We observe that the MDL-FS has the best conditional auxiliary log-likelihood score for Naive Bayes and it is second best for TANs. It is the second most selective methods (on average only Hall's method reduces more attributes). The accuracies of the MDL-FS method are comparable with the accuracies of the other algorithms.

8. Conclusions

In this paper, we have studied the feature-selection behaviour of the MDL-FS function, an MDL kind of function, for learning Bayesian network classifiers from data. We define the concept of redundant and irredundant attributes for the class variable given sets of attributes; based on the cardinality of these attributes, we have different levels of redundancy and irredundancy. Based on the observation that the poor feature-selection behaviour of the MDL function is due to the use of the joint probability distribution over a classifier's variables, we have analysed an MDL-based function that captures the conditional distribution instead of the joint probability from the standard MDL. Since computing conditional log-likelihood is generally acknowledged to be hard, we associate to each Bayesian network classifier an auxiliary network to model also the distribution over the attributes set. We have argued, both theoretically and experimentally, that the MDL-FS function is better tailored to the task of feature selection for more complicated Bayesian network classifiers than the Naive Bayes classifier than the MDL score: with the MDL-FS function, classifiers are yielded that have a performance comparable to the ones found with the MDL function, yet include fewer attributes. We performed many experiments that compare our method with popular methods from feature selection literature; in many cases, with the MDL-FS score, we have obtained better and/or more selective classifiers. Explanations for the empirical performance of the generated selective Bayesian network classifiers are consistent with the theoretical findings of the paper. Our selective MDL-FS method can be used to provide improved insight into the domains being modeled, since less features are used. Furthermore, for classifying a new instance, fewer feature values need to be obtained which is very useful as well. Finally, the paper has contributed to an additional understanding of feature selection in terms of explaining different methods for discriminating the relevant features for the classification task using various scores and machine learning algorithms.

Acknowledgments

We want to thank Prof. Linda van der Gaag for her help with writing the paper.

References

[1] P. Langley, W. Iba, K. Thompson, An analysis of Bayesian classifiers, in: Conference on Artificial Intelligence, 1992, pp. 223–228.

[2] S. Kotsiantis, Supervised machine learning: a review of classification techniques, Informatica 31 (2007) 249–268.

[3] H. Liu, H. Motoda, Computational Methods of Feature Selection, Chapman & Hall/CRC, 2007.

[4] I. Inza, P. Larranaga, B. Sierra, Feature subset selection by Bayesian networks: a comparison with genetic and sequential algorithms, International Journal of Approximate Reasoning 27 (2) (2001) 143–164.

[5] A. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artificial Intelligence 97 (1–2) (1997) 245–271.

[6] I. Tsamardinos, C. Aliferis, Towards principled feature selection: relevance, filters, and wrappers, in: Workshop AI&Stat, 2003.

[7] R. Nilsson, J. Pena, J. Bjorkegren, J. Tegner, Consistent feature selection for pattern recognition in polynomial time, Journal of Machine Learning Research 8 (2007) 589–612.

[8] J. Pena, R. Nilsson, J. Björkegren, J. Tegnér, Towards scalable and data efficient learning of Markov boundaries, International Journal of Approximate Reasoning 45 (2) (2007) 211–232.

[9] M. Minsky, Steps toward artificial intelligence, Transactions of the Institute and Radio Engineers 46 (1961) 8–30.

[10] J. Rissanen, Modeling by shortest data description, Automatica 14 (1978) 465–471.

[11] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Machine Learning 29 (2–3) (1997) 131–163.

[12] Y. Guo, R. Greiner, Discriminative model selection for belief net structures, in: Proceedings of the American Association for Artificial Intelligence (AAAI), 2005, pp. 770–776.

[13] E. Hruschka, M. do Carmo Nicoletti, V. de Oliveira, G. Bressan, Bayesrule: a Markov-blanket based procedure for extracting a set of probabilistic rules from Bayesian classifiers, Hybrid Intelligent Systems 5 (2008) 83–96.

[14] Y. Jing, V. Pavlovic, J. Rehg, Boosted Bayesian network classifiers, Machine Learning 73 (2) (2008) 155–184.

[15] A. Antonucci, M. Zaffalon, Fast algorithms for robust classification with Bayesian nets, International Journal of Approximate Reasoning 44 (3) (2007) 200–223.

[16] A. Pérez, P. Larranaga, I. Inaki, Bayesian classifiers based on kernel density estimation: flexible classifiers, International Journal of Approximate Reasoning 50 (2) (2009) 341–362.

[17] L.M. de Campos, A.E. Romero, Bayesian network models for hierarchical text classification from a thesaurus, International Journal of Approximate Reasoning 50 (7) (2009) 932–944.

[18] M.M. Drugan, L.C. van der Gaag, A new MDL-based feature selection for Bayesian network classifiers, in: Proceedings of the European Conference on Artificial Intelligence, 2004, pp. 999–1000.

[19] D. Grossman, P. Domingos, Learning Bayesian networks classifiers by maximising conditional likelihood, in: Proceedings of the International Conference on Machine Learning, 2004, pp. 361–368.

[20] G. Santafè, J. Lozano, P. Larranaga, Discriminative vs. generative learning of Bayesian network classifiers, in: Proceedings of Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 2007, pp. 453–464.

[21] Y. Rubinstein, T. Hastie, Discriminative vs informative learning, in: Knowledge Discovery and Data Mining, 1997, pp. 49–53.

[22] X. Zhu, X. Wu, Class noise vs. attribute noise: a quantitative study, Artificial Intelligence Review 22 (2004) 177–210.

[23] C.E. Shannon, A mathematical theory of communication, Bell System Technical Journal 27 (1948) 379–423. 623–656.

[24] P. Domingos, M. Pazzani, On the optimality of the simple Bayesian classifier under zero-one loss, Machine Leaning 29 (1997) 103–130.

[25] M.H. Hansen, B. Yu, Model Selection and the Principle of Minimum Description Length, Journal of the American Statistical Association 96 (454) (2001) 746–774.

[26] J. Rissanen, Strong optimality of the normalized ML models as universal codes and information in data, IEEE Transactions on Information Theory 47 (2001) 1712–1717.

[27] P. Kontkanen, W. Buntine, P. Myllymaki, J. Rissanen, H. Tirri, Efficient computation of stochastic complexity, in: Workshop AI&Stat, 2003, pp. 181–188.

[28] T. Silander, T. Roos, P. Myllymäki, Learning locally minimax optimal Bayesian networks, International Journal of Approximate Reasoning 51 (5) (2010) 544–557.

[29] P. Langley, S. Sage, Induction of selective Bayesian classifiers, in: Proceedings of Uncertainty in Artificial Intelligence (UAI), 1994, pp. 399–406.

[30] M.A. Hall, Correlation-based feature selection for discrete and numeric class machine learning, in: Proceedings of the International Conference on Machine Learning (ICML), 2000, pp. 359–366.

[31] G.H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: Proceedings of the International Conference on Machine Learning (ICML), 1994, pp. 121–129.

[32] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence (1997) 273–324.

[33] D. Koller, M. Sahami, Toward optimal features selection., in: Proceedings of the International Conference on Machine Learning (ICML), 1996, pp. 284–292.

[34] H. Liu, J. Sun, L. Liu, H. Zhang, Feature selection with dynamic mutual information, Pattern Recognition 42 (2009) 1330–1339.

[35] R. Blanco, I. Inza, M. Merino, J. Quiroga, P. Larranaga, Feature selection in Bayesian classifiers for the prognosis of survival of cirrhotic patients treated with tips, Journal of Biomedical Informatics 38 (2005) 376–388.

[36] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, Machine Learning 5 (2004) 1205–1224.

[37] P. Kontkanen, P. MyllymSki, T. Silander, H. Tirri, BAYDA: software for Bayesian classification and feature selection, in: Proceedings of Knowledge discovery and Data-Mining (KDD), 1998, pp. 254–258.

[38] T. Jebara, T. Jaakola, Feature selection and dualities in maximum entropy discrimination, in: Proceedings of Uncertainty in Artificial Intelligence (UAI), 2000, pp. 291–300.

[39] J. Bilmes, Dynamic Bayesian multinets, in: Proceedings of Uncertainty in Artificial Intelligence (UAI), 2000, pp. 38–45.

[40] F. Pernkopf, J. Bilmes, Discriminative versus generative parameter and structure learning of Bayesian network classifiers, in: Proceedings of the International Conference on Machine Learning (ICML), 2005, pp. 657–664.

[41] J. Burge, T. Lane, Learning class-discriminative dynamic Bayesian networks, in: Proceedings of the International Conference on Machine Learning, 2005, pp. 97–104.

[42] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Machine Learning 3 (2003) 1157–1182.

[43] M. Drugan, Conditional log-likelihood MDL and Evolutionary MCMCs, Ph.D. Thesis, Utrecht University, The Netherlands, 2006.

[44] C. Chow, C. Liu, Approximating discrete probability distributions with dependence trees, IEEE Transaction on Information Theory (14) (1968) 462–467.

[45] L. Yu, H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, in: Proceedings of the International Conference on Machine Learning (ICML), 2003, pp. 856–863.

[46] K. Kira, L. Rendell, A practical approach to feature selection, in: Proceedings of the International Conference on Machine Learning (ICML), 1992, pp. 249–256.

[47] I. Kononenko, Estimating attributes: Analysis and extensions of relief, in: Proceedings of the European Conference on Machine Learning (ECML), 1994, pp. 171–182.

[48] S. Gadat, L. Younes, A stochastic algorithm for feature selection in pattern recognition, Journal of Machine Learning Research 8 (2007) 509–547.

[49] Y. Sun, Iterative relief for feature weighting: algorithms, theories and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (6) (2007) 1035–1051.

[50] F. Fleuret, Fast binary feature selection with conditional mutual information, Machine Learning 5 (2004) 1531–1555.

[51] J. Huang, N. Lv, S. Li, Y. Cai, Feature selection for classificatory analysis based on information-theoretic criteria, Acta Automatica Sinica 34 (3) (2008) 383–392.

[52] J. Liang, S. Yang, A. Winstanley, Invariant optimal feature selection: a distance discriminant and feature ranking based solution, Pattern Recognition 41 (5) (2008) 1429–1439.

[53] Y. Song, F. Nie, C. Zhang, S. Xiang, A unified framework for semi-supervised dimensionality reduction, Pattern Recognition 41 (9) (2008) 2789–2799.

[54] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning, in: Proceedings of the International Conference on Machine Learning (ICML), 2007, pp. 1151–1157.

[55] U.M. Fayyad, K.B. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, in: Joint Conference on Artificial Intelligence, 1993, pp. 1022–1027.

[56] P. Langley, Selection of relevant features in machine learning, in: Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI), 1994, pp. 140–144.

[57] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, 1989.