Discrete Bayesian Network Classifiers: A Survey

CONCHA BIELZA and PEDRO LARRAÑAGA, Universidad Politécnica de Madrid

We have had to wait over 30 years since the naive Bayes model was first introduced in 1960 for the so-called Bayesian network classifiers to resurge. Based on Bayesian networks, these classifiers have many strengths, like model interpretability, accommodation to complex data and classification problem settings, existence of efficient algorithms for learning and classification tasks, and successful applicability in real-world problems. In this article, we survey the whole set of discrete Bayesian network classifiers devised to date, organized in increasing order of structure complexity: naive Bayes, selective naive Bayes, seminaive Bayes, one-dependence Bayesian classifiers, *k*-dependence Bayesian classifiers, Bayesian network-augmented naive Bayes, Markov blanket-based Bayesian classifier, unrestricted Bayesian classifiers, and Bayesian multinets. Issues of feature subset selection and generative and discriminative structure and parameter learning are also covered.

Categories and Subject Descriptors: I.5.1 [**Pattern Recognition**]: Models

General Terms: Algorithms, Design, Performance

Additional Key Words and Phrases: Supervised classification, Bayesian network, naive Bayes, Markov blanket, Bayesian multinets, feature subset selection, generative and discriminative classifiers

ACM Reference Format:
Concha Bielza and Pedro Larrañaga. 2014. Discrete Bayesian network classifiers: A survey. ACM Comput. Surv. 47, 1, Article 60 (April 2014), 43 pages.
DOI: http://dx.doi.org/10.1145/2576868

60

1. INTRODUCTION

Bayesian network classifiers are special types of Bayesian networks designed for classification problems. Supervised classification aims at assigning labels or categories to instances described by a set of predictor variables or features. The classification model that assigns labels to instances is automatically induced from a dataset containing labeled instances or sometimes by hand with the aid of an expert. We will focus on learning models from data, favored by the large amount of data collected and accessible nowadays.

Bayesian network classifiers have many advantages over other classification techniques, as follows: (1) They offer an explicit, graphical, and interpretable representation of uncertain knowledge. Their semantics is based on the sound concept of conditional independence since they are an example of a probabilistic graphical model. (2) As they output a probabilistic model, decision theory is naturally applicable for dealing with cost-sensitive problems, thereby providing a confidence measure on the chosen predicted label. (3) Thanks to the model expressiveness of Bayesian network classifiers,

Research partially supported by the Spanish Ministry of Economy and Competitiveness, projects TIN2010-20900-C04-04 and Cajal Blue Brain.
Author's addresses: Concha Bielza and Pedro Larrañaga, Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid.
Q1

DOI: http://dx.doi.org/10.1145/2576868

they can easily accommodate feature selection methods and handle missing data in both learning and inference phases. Also, they fit more complex classification problems in any type of domain (discrete, continuous, and mixed data), with undetermined labels, partial labels, many class variables to be simultaneously predicted, new flows of streaming data, and so forth. (4) There is an active research field developing a plethora of learning from data algorithms, covering different frequentist and Bayesian, expert, and/or data-based viewpoints. Besides, the induced models can be organized hierarchically according to their structure complexity. (5) Bayesian network classifiers can be built with computationally efficient algorithms whose learning time complexity is linear on the number of instances and linear, quadratic, or cubic (depending on model complexity) on the number of variables, and whose classification time is linear on the number of variables. (6) These algorithms are easily implemented, although most of the available software only contains the simplest options (naive Bayes and tree-augmented naive Bayes), focusing instead on learning general-purpose Bayesian networks. (7) Numerous successful real-world applications have been reported in the literature, with competitive performance results against state-of-the-art classifiers.

This article offers a comprehensive survey of the state of the art of the Bayesian network classifier in discrete domains. Unlike other reviews mentioned later, this article covers many model specificities: (1) for naive Bayes, its weighted version, inclusion of hidden variables, metaclassifiers, special situations like homologous sets, multiple instances, cost-sensitive problems, instance ranking, imprecise probabilities, text categorization, and discriminative learning of parameters; (2) for selective naive Bayes, univariate and multivariate filter approaches and wrapper and embedded methods; (3) the not-so-well-known seminaive Bayes classifier; (4) for one-dependence Bayesian classifiers, wrapper approaches, metaclassifiers based on tree-augmented naive Bayes, and discriminative learning; (5) for general Bayesian network classifiers, classifiers based on identifying the class variable Markov blanket, metaclassifiers, and discriminative and generative learning of general Bayesian networks used for classification problems; and (6) Bayesian multinets for encoding probabilistic relationships of asymmetric independence. Besides, we provide a clear unified notation for all models and graphical representations of their corresponding networks.

A recent overview of Bayesian network classifiers is Flores et al. [2012]. However, the authors only cover the basic details of naive Bayes, tree-augmented naive Bayes, k-dependence Bayesian classifiers, averaged one-dependence estimators, Bayesian multinets, dependency networks, and probabilistic decision graphs. Other shorter reviews of Bayesian network classifiers are Goldszmidt [2010], discussing only naive Bayes and tree-augmented naive Bayes, and Al-Aidaroos et al. [2010], focusing on variants of naive Bayes classifiers. This article is a comprehensive, methodical, and detailed survey of Bayesian network classifiers ever conducted, elaborating on a variety of facets and a diversity of models.

The article is organized as follows. Section 2 reviews the fundamentals of Bayesian network classifiers in discrete domains. Then, different models of increasing structure complexity are presented consecutively. Section 3 describes naive Bayes. Section 4 addresses selective naive Bayes. Section 5 introduces seminaive Bayes. Section 6 focuses on one-dependence Bayesian classifiers, like tree-augmented naive Bayes and the super-parent one-dependence estimator. Section 7 discusses k-dependence Bayesian classifiers. Section 8 sets out general Bayesian network classifiers, covering Bayesian network-augmented naive Bayes, classifiers based on identifying the Markov blanket of the class variable, unrestricted Bayesian classifiers, and discriminative learning. Section 9 discusses the broadest models, Bayesian multinets. Section 10 shows an illustrative example highlighting the differences between the most important classifiers. Finally, Section 11 rounds the article off with a discussion and future work.

2. FUNDAMENTALS

Let $\mathbf{X} = (X_1, \ldots, X_n)$ be a vector of discrete predictor random variables or features, with $x_i \in \Omega_{X_i} = \{1, 2, \ldots, r_i\}$, and let C be a label or class variable, with $c \in \Omega_C = \{1, 2, \ldots, r_c\}$. Given a simple random sample $\mathcal{D} = \{(\mathbf{x}^{(1)}, c^{(1)}), \ldots, (\mathbf{x}^{(N)}, c^{(N)})\}$, of size N, with $\mathbf{x}^{(j)} = (x_{j1}, \ldots, x_{jn})$, drawn from the joint probability distribution $p(\mathbf{X}, C)$, the supervised classification problem consists of inducing a classification model from $\mathcal{D}$ able to assign labels to new instances given by the value of their predictor variables. Common performance measures include classification accuracy, sensitivity, specificity, the F-measure, and area under the ROC curve. All these measures must be estimated using honest evaluation methods, like hold-out, k-fold cross-validation, bootstrapping, and so forth [Japkowicz and Mohak 2011].

A *Bayes classifier* assigns the most probable a posteriori (MAP) class to a given instance $\mathbf{x} = (x_1, \ldots, x_n)$, that is,

$$\arg\max_c p(c|\mathbf{x}) = \arg\max_c p(\mathbf{x}, c), \tag{1}$$

which, under a 0/1 loss function, is optimal in terms of minimizing the conditional risk [Duda et al. 2001].

For a general *loss function*, $\lambda(c', c)$, where c' is the class value output by a model and c is the true class value, the Bayesian classifier can be learned by using the Bayes decision rule that minimizes the expected loss or conditional risk $R(c'|\mathbf{x}) = \sum_{c \in \Omega_C} \lambda(c', c) p(c|\mathbf{x})$, for any instance $\mathbf{x}$ [Duda et al. 2001].

Bayesian network classifiers [Friedman et al. 1997] approximate $p(\mathbf{x}, c)$ with a factorization according to a *Bayesian network* [Pearl 1988]. The structure of a Bayesian network on the random variables $X_1, \ldots, X_n, C$ is a directed acyclic graph (DAG) whose vertices correspond to the random variables and whose arcs encode the probabilistic (in)dependences among triplets of variables; that is, each factor is a categorical distribution $p(x_i|\mathbf{pa}(x_i))$ or $p(c|\mathbf{pa}(c))$, where $\mathbf{pa}(x_i)$ is a value of the set of variables $\mathbf{Pa}(X_i)$, which are parents of variable X_i in the graphical structure. The same applies for $\mathbf{pa}(c)$. Thus,

$$p(\mathbf{x}, c) = p(c|\mathbf{pa}(c)) \prod_{i=1}^{n} p(x_i|\mathbf{pa}(x_i)). \tag{2}$$

When the sets $\mathbf{Pa}(X_i)$ are sparse, this factorization prevents having to estimate an exponential number of parameters, which would otherwise be required.

For the special case of $\mathbf{Pa}(C) = \emptyset$, the problem is to maximize on c:

$$p(\mathbf{x}, c) = p(c)p(\mathbf{x}|c).$$

Therefore, the different Bayesian network classifiers explained later correspond with different factorizations of $p(\mathbf{x}|c)$. The simplest model is the naive Bayes, where C is the parent of all predictor variables and there are no dependence relationships among them (Sections 3 and 4). We can progressively increase the level of dependence in these relationships (one-dependence, k-dependence, etc.) giving rise to a family of augmented naive Bayes models, explained in Sections 5 through 8.1; see Figure 1.

Equation (2) states a more general case; see also Figure 1. $p(\mathbf{x}, c)$ is factorized in different ways, C can have parents, and we have to search the Markov blanket of C to solve Equation (1) (Section 8.2). The *Markov blanket* (see Pearl [1988, p. 97]) of C is the set of variables MB_C that make C conditionally independent of the other variables in the network, given MB_C, that is,

$$p(c|\mathbf{x}) = p(c|\mathbf{x}_{MB_C}), \tag{3}$$

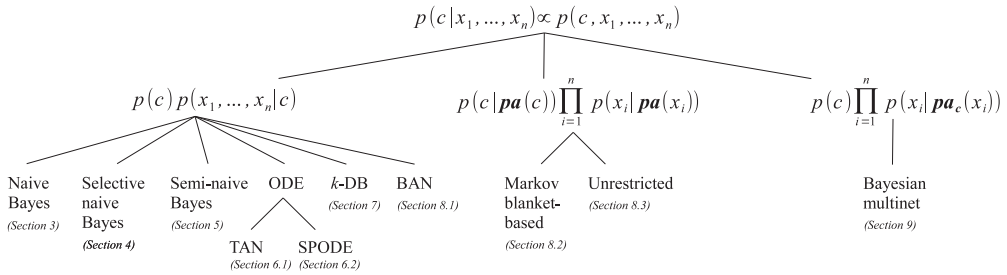


Fig. 1. Categorization of discrete Bayesian network classifiers according to the factorization of $p(\mathbf{x}, c)$.

where $\mathbf{x}_{MB_C}$ denotes the projection of $\mathbf{x}$ onto the variables in MB_C. Therefore, the Markov blanket of C is the only knowledge needed to predict its behavior. A probability distribution p is *faithful* to a DAG representing a Bayesian network if, for all triplets of variables, they are conditionally independent with respect to p iff they are d-separated in the DAG. For such p, MB_C is unique and is composed of C's parents, children, and the children's other parents (spouses) [Pearl 1988].

There are two strategies for learning both the Markov blanket and the structures for augmented naive Bayes: testing conditional independences (constraint-based techniques [Spirtes et al. 1993]) and searching in the space of models guided by a score to be optimized (score + search techniques [Cooper and Herskovits 1992]). They can also be combined in hybrid techniques. Alternatively, we can use these strategies to learn an *unrestricted* Bayesian network, which does not consider C as a distinguished variable, from which only the Markov blanket of C must be extracted for classification purposes (Section 8.3). Finally, specific conditional independence relationships can be modeled for different c values, giving rise to different Bayesian classifiers, which are then joined in the more complex Bayesian multinet (Section 9). The parents of X_i, $\mathbf{Pa}_c(X_i)$, may be different depending on c; see Figure 1.

Apart from learning the network structure, the probabilities $p(x_i|\mathbf{pa}(x_i))$ are estimated from $\mathcal{D}$ by standard methods like maximum likelihood or Bayesian estimation. In Bayesian estimation, assuming a Dirichlet prior distribution over $(p(X_i = 1|\mathbf{Pa}(X_i) = j), \ldots, p(X_i = r_i|\mathbf{Pa}(X_i) = j))$ with all hyperparameters equal to α, then the posterior distribution is Dirichlet with hyperparameters equal to $N_{ijk}+\alpha, k = 1, \ldots, r_i$, where N_{ijk} is the frequency in $\mathcal{D}$ of cases with $X_i = k$ and $\mathbf{Pa}(X_i) = j$. Hence, $p(X_i = k|\mathbf{Pa}(X_i) = j)$ is estimated by

$$\frac{N_{ijk} + \alpha}{N_{\cdot j\cdot} + r_i\alpha}, \tag{4}$$

where $N_{\cdot j\cdot}$ is the frequency in $\mathcal{D}$ of cases with $\mathbf{Pa}(X_i) = j$. This is called the *Lindstone rule*. A special case of the Lindstone rule called *Laplace estimation*, with $\alpha = 1$ in Equation (4), is used in Good [1965]. Also, the *Schurmann-Grassberger rule*, where $\alpha = \frac{1}{r_i}$, is employed in Hilden and Bjerregaard [1976] and Titterington et al. [1981]. Obviously, the maximum likelihood estimate is given by $\frac{N_{ijk}}{N_{\cdot j\cdot}}$.

So far we have proceeded with only one selected Bayesian network classifier, as if that model had generated the data, thus ignoring uncertainty in model selection. *Bayesian model averaging* provides a way of accounting for model uncertainty. It uses the Bayes rule to combine the posterior distributions under each of the models considered with structure S_m in a space $\mathcal{S}$, each weighted by its posterior model probabilities:

$$p(\mathbf{x}, c|\mathcal{D}) = \sum_{S_m \in \mathcal{S}} p(\mathbf{x}, c|S_m, \mathcal{D}) p(S_m|\mathcal{D}). \tag{5}$$

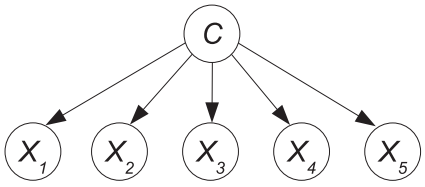


Fig. 2. A naive Bayes structure from which $p(c|\mathbf{x}) \propto p(c)p(x_1|c)p(x_2|c)p(x_3|c)p(x_4|c)p(x_5|c)$.

The posterior probability of model S_m is given by

$$p(S_m|\mathcal{D}) = \frac{p(\mathcal{D}|S_m)p(S_m)}{\sum_{S_l \in \mathcal{S}} p(\mathcal{D}|S_l)p(S_l)} \tag{6}$$

and the (marginal) likelihood of model S_m is

$$p(\mathcal{D}|S_m) = \int p(\mathcal{D}|\boldsymbol{\theta}_m, S_m)p(\boldsymbol{\theta}_m|S_m)d\boldsymbol{\theta}_m, \tag{7}$$

where the vector of parameters of model S_m is $\boldsymbol{\theta}_m = (\boldsymbol{\theta}_C, \boldsymbol{\theta}_{X_1}, \ldots, \boldsymbol{\theta}_{X_n})$, and for the case of $\mathbf{Pa}(C) = \emptyset$, $\boldsymbol{\theta}_C = ((p(c))_{c=1}^{r_c})$ and $\boldsymbol{\theta}_{X_i} = ((((\theta_{ijk}))_{k=1}^{r_i})_{j=1}^{q_i})$. θ_{ijk} denote $p(X_i = k|\mathbf{Pa}(X_i) = j)$ and q_i represents the total number of different configurations of $\mathbf{Pa}(X_i)$.

Since our models are Bayesian network classifiers and, according to Equation (2), $p(\mathbf{x}, c|S_m, \mathcal{D}) = p(c)\prod_{i=1}^{n}\theta_{ijk}$, Equation (5) is then simplified as

$$p(\mathbf{x}, c|\mathcal{D}) \propto \sum_{S_m \in \mathcal{S}} p(c)\left(\prod_{i=1}^{n}\theta_{ijk}\right)p(\mathcal{D}|S_m)p(S_m).$$

3. NAIVE BAYES

Naive Bayes [Maron and Kuhns 1960; Minsky 1961] is the simplest Bayesian network classifier (Figure 2), since the predictive variables are assumed to be conditionally independent given the class, transforming Equation (1) into

$$p(c|\mathbf{x}) \propto p(c)\prod_{i=1}^{n} p(x_i|c). \tag{8}$$

This assumption is useful when n is high and/or N is small, making $p(\mathbf{x}|c)$ difficult to estimate. Even if the assumption does not hold, the model classification performance may still be good in practice (although the probabilities are not well calibrated) because the decision boundaries may be insensitive to the specificities of the class-conditional probabilities $p(x_i|c)$ [Domingos and Pazzani 1997]; that is, variance is reduced because few parameters are required and the biased probability estimates may not matter since the aim is classification rather than accurate posterior class probability estimation [Hand and Yu 2001].

Other approaches transform the data to avoid the effects of violating the conditional independence assumption, thereby improving the probability estimates made by naive Bayes. The class dispersion problem covers distributions $p(\mathbf{x}|c)$, where clusters of cases that belong to the same class are dispersed across the input space. One possible solution is to transform the class distribution by applying a clustering algorithm to each subset of cases with the same label, producing a refinement (extension) on the number of labels. This is proposed in Vilalta and Rish [2003], where a naive Bayes is then learned over this new dataset, and finally the predicted (extended) labels are mapped to the original space of labels.

From a theoretical point of view, if all variables (predictors and class) are binary, the decision boundary has been shown to be a hyperplane [Minsky 1961]. For ordinal nonbinary predictor variables, the decision boundary is a sum of n polynomials, one for each variable X_i, with a degree equal to $r_i - 1$ [Duda et al. 2001]. Naive Bayes has proved to be optimal (i.e., achieving lower zero-one loss than any other classifier) for learning conjunctions and disjunctions of literals [Domingos and Pazzani 1997]. A bound for the degradation of the probability of correct classification when naive Bayes is used as an approximation of the Bayes classifier is given in Ekdahl and Koski [2006].

The inclusion of irrelevant (redundant) variables for the class does not (does) worsen the performance of a naive Bayes classifier [Langley and Sage 1994]. Hence, it is important to remove irrelevant and redundant variables, as the so-called *selective naive Bayes* should ideally do (see Section 4).

From a practical point of view, there have been some attempts to visualize the effects of individual predictor values on the classification decision. Most are based on an equivalent expression for a naive Bayes model in terms of the log odds that for a binary class (c vs. $\bar{c}$) results in

$$logit\ p(c|\boldsymbol{x}) = \log \frac{p(c|\boldsymbol{x})}{p(\bar{c}|\boldsymbol{x})} = \log \frac{p(c)}{p(\bar{c})} + \sum_{i=1}^{n} \log \frac{p(x_i|c)}{p(x_i|\bar{c})}.$$

While `Orange` software [Možina et al. 2004] uses nomograms to represent the additive influence of each predictor value, `ExplainD` [Poulin et al. 2006] uses bar-based charts with different levels of explanation capabilities.

3.1. Parameter Estimation

The Bayesian probability estimate called *m-estimate* is successfully used in the naive Bayes classifier [Cestnik 1990]. It has a tunable parameter m whereby it can adapt to domain properties, such as the level of noise in the dataset.

A Bayesian bootstrap method of probability estimation is presented in Norén and Orre [2005]. This results in sampling from the dataset of just the $N' \leq N$ different cases of $\mathcal{D}$ with a Dirichlet distribution with hyperparameters related to the frequency of these N' distinct values in $\mathcal{D}$. The variables in a Dirichlet random vector can never be positively correlated and must have the same normalized variance. These constraints deteriorate the performance of the naive Bayes classifier and motivate the introduction of other prior distributions, like the generalized Dirichlet and the Liouville distributions [Wong 2009].

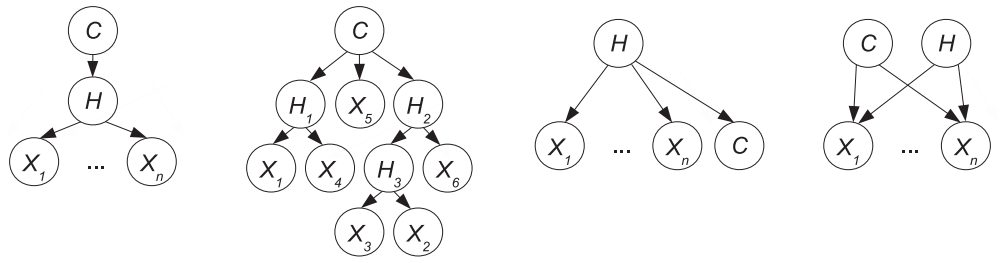
An estimation inspired by an iterative Hebbian rule is proposed in Gama [1999]. In each iteration and for each of the N cases, if the case is well (incorrectly) classified by the current naive Bayes model, then $p(x_i|c)$ for its corresponding values x_i and its true class c should be increased (decreased), adjusting the other conditional probabilities.

3.2. Weighted Naive Bayes

Adjusting the naive Bayesian probabilities during classification may significantly improve predictive accuracy. A general formula is

$$p(c|\mathbf{x}) \propto w_c p(c) \prod_{i=1}^{n} [p(x_i|c)]^{w_i} \tag{9}$$

for some weights $w_c, w_i, i = 1, \ldots, n$. In Hilden and Bjerregaard [1976], $w_c = 1$ and $w_i = w \in (0, 1), \forall i$, attaching more importance to the prior probability of the class variable. w is fixed by looking for a good performance after some trials. Also, in Hall [2007], $w_c = 1$ and w_i is set to $1/\sqrt{d_i}$, where d_i is the minimum depth at which variable



(a) (b) (c) (d)

Fig. 3. (a) Naive Bayes with a hidden variable H [Kwoh and Gillies 1996]; (b) hierarchical naive Bayes [Zhang et al. 2004; Langseth and Nielsen 2006]; (c) finite mixture model, with a hidden variable as a parent of predictor variables and the class [Kontkanen et al. 1996]; (d) finite-mixture-augmented naive Bayes [Monti and Cooper 1999].

X_i is tested in the unpruned decision tree constructed from the data. Fixing the root node to depth 1, d_i weighs X_i according to the degree to which it depends on the values of other variables. Finally, in Webb and Pazzani [1998], the linear adjustment w_c is found by employing a hill-climbing search maximizing the resubstitution accuracy and $w_i = 1, \forall i$.

3.3. Missing Data

When the training set is incomplete (i.e., some variable values are unknown), both classifier efficiency and accuracy can be lost.

Simple solutions for handling missing data are either to ignore the cases including unknown values or to consider unknowns to be a separate value of the respective variables [Kohavi et al. 1997]. These solutions introduce biases in the estimates. Another common solution is imputation, where likely values (mode or class-conditional mode) stand in for the missing data. Other suggestions [Friedman et al. 1997] are to use the *expectation-maximization* (EM) *algorithm* [Dempster et al. 1977] or gradient descent method. However, these methods rely on the assumption that data are *missing at random* (i.e., the probability that an entry will be missing is a function of the observed values in the dataset). This cannot be verified in a particular dataset, and if violated, the methods lead to decreased accuracy.

This is why the *robust Bayesian estimator* is introduced in Ramoni and Sebastiani [2001b] to learn conditional probability distributions from incomplete datasets without any assumption about the missing data mechanism. The estimation is given by an interval including all the estimates induced from all possible completions of the original dataset. A new algorithm to compute posterior probability intervals from interval-valued probabilities is then proposed in Ramoni and Sebastiani [2001a]. In the classification phase, all these intervals are ranked according to a score to decide the class with the highest-ranked interval.

3.4. Including Hidden Variables

The violation of the conditional independence assumption in naive Bayes can be interpreted as an indication of the presence of hidden or latent variables. Introducing one hidden variable in the naive Bayes model as a child of the class variable and parent of all predictor variables is the simplest solution to this problem; see Figure 3(a). This is the approach reported in Kwoh and Gillies [1996], where the conditional probabilities attached to the hidden node are determined using a gradient descent method. The objective function to be minimized is the squared error between the real class values

and the class posterior probabilities. The approach taken in Zhang et al. [2004] is more general, since many hidden variables are arranged in a tree-shaped Bayesian network called *hierarchical naive Bayes*. The root is the class variable, the leaves are the predictor variables, and the internal nodes are the hidden variables. An example is given in Figure 3(b). This structure is learned using a hill-climbing algorithm that compares candidate models with the Bayesian information criterion (BIC), whereas its parameters are estimated using the EM algorithm [Dempster et al. 1977]. A classification accuracy-focused improvement is shown in Langseth and Nielsen [2006]. This strategy is faster since latent variables are proposed by testing for conditional independencies.

There are other options for relaxing the conditional independence assumption. First, the *finite mixture model* introduced in Kontkanen et al. [1996] leaves the class variable as a child node, whereas the common parent for both the discrete or continuous predictors and the class variable is a hidden variable; see Figure 3(c). This unmeasured discrete variable is learned using the EM algorithm and models the interaction between the predictor variables and between the predictor variables and the class variable. Thus, the class and the predictor variables are conditionally independent given the hidden variable. Second, the *finite-mixture-augmented naive Bayes* [Monti and Cooper 1999] is a combination of this model and naive Bayes. The standard naive Bayes is augmented with another naive Bayes with a hidden variable acting as the parent of the predictor variables; see Figure 3(d). The hidden variable models the dependences among the predictor variables that are not captured by the class variable. Therefore, it is expected to have fewer states in its domain (i.e., the mixture will have fewer components) than the finite mixture model.

3.5. Metaclassifiers

We may use many rather than just one naive Bayes. Thus, the *recursive Bayesian classifier* [Langley 1993] observes each predicted label (given by the naive Bayes) separately. Whenever a label is misclassified, a new naive Bayes is induced from those cases having that predicted label. Otherwise, the process stops. The *successive naive Bayes classifier* [Kononenko 1993] repeats for a fixed number of iterations the learning of a naive Bayes from the whole data with redefined labels: a special label c_0 is assigned to cases correctly classified by the current naive Bayes, whereas their original labels are retained in the other instances. When classifying a new instance, the naive Bayes learned last should be applied first. If c_0 is predicted, the next latest naive Bayes must be applied; otherwise, the predicted label will be the answer. Also, any *ensemble method* can be used taking naive Bayes as the base classifier. A specific property of the AdaBoost algorithm based on naive Bayes models is that the final boosted model is shown to be another naive Bayes [Ridgeway et al. 1998]. Finally, two naive Bayes can be used as the base classifier in a *random oracle classifier* [Rodríguez and Kuncheva 2007]. This is formed by two naive Bayes models and a random oracle that chooses one of them in the classification phase. The oracle first divides the predictive variable space into two disjoint subspaces based on some random decisions. A naive Bayes is then learned from those instances belonging to each subspace. A possible reason for the success of (ensembles based on) random oracle classifiers is that the classification may be easier in each subspace than in the original space.

Multiclass problems are often transformed into a set of binary problems via class binarization techniques. Prominent examples are pairwise classification and one-against-all binarization. Training all these binary classifiers, each of which is less complex and has simpler decision boundaries, increases the robustness of the final classifier with probably less computational burden. The classifier resulting from an *ensemble of pairwise naive Bayes* (c_i vs. c_j) that combines the predictions of the individual classifiers

using voting and weighted voting techniques is equivalent to a common naive Bayes. This does not hold for one-against-all binarization [Sulzmann et al. 2007].

Alternatively, naive Bayes can be hybridized with other classification models. The *NBtree* is introduced in Kohavi [1996], combining naive Bayes and decision trees. NBtree partitions the training data using a tree structure and builds a local naive Bayes in each leaf with nontested variables. The particular case of a tree with only one branching variable is reported in Cano et al. [2005], where several methods for choosing this variable are proposed. Optionally, for each new case to be classified, a (local) naive Bayes can be induced only from its k closest cases in the dataset. This hybrid between naive Bayes and the k-nearest neighbor model is called *locally weighted naive Bayes* [Frank et al. 2003], since the instances in the neighborhood are weighted, attaching less weight to instances that are further from the test instance. Finally, the *lazy Bayesian rule* learning algorithm [Zheng and Webb 2000] induces a rule for each example, whose antecedent is a variable-value conjunction while the consequent is a local naive Bayes with features that are not in the antecedent.

3.6. Special Situations

(a) Homologous sets. We sometimes have to classify a set of cases that belong to the same unknown class (i.e., a homologous set), for example, a set of leaves taken from the same unknown plant whose species we intend to identify. The *homologous naive Bayes* [Huang and Hsu 2002] takes this knowledge into account, where Equation (8) is now given by

$$p(c|\mathbf{x}_1,\ldots,\mathbf{x}_H,\mathcal{H}) \propto p(c)\prod_{h=1}^{H}\prod_{i=1}^{n} p(x_{hi}|c),$$

since we wish to classify the homologous set $\{\mathbf{x}_1,\ldots,\mathbf{x}_H\}$, and $\mathcal{H}$ denotes that all cases in this set have the same unknown class label. This way, we ensure that different labels are not assigned to all these cases.

(b) Multiple instances. In this setting, the learner receives a set of bags that are labeled positive or negative. Each bag contains many instances. A bag is labeled positive (negative) if at least one (all) of its instances is (are) positive (negative). We are looking for a standard classification of individual instances from a collection of labeled bags, for example, learning a simple description of a person from a series of images that are positively labeled if they contain the person and negatively labeled otherwise.

The *multiple-instance naive Bayes* [Murray et al. 2005] starts by assigning negative labels to all the instances in a negative bag. In a positive bag, all the instances are assigned a negative label except one, which receives a positive label. Then a naive Bayes is applied to this dataset. For every positive bag that was misclassified (i.e., all its instances were classified as negative), the instance with the maximum a posteriori probability of being positive is relabeled as positive. A second naive Bayes is applied to this new dataset. This succession of naive Bayes models is halted when a stopping condition is met.

(c) Cost sensitivity. For general loss functions, a *cost-sensitive naive Bayes* selects, for each instance $\mathbf{x}$, the class value minimizing the expected loss [Ibáñez et al. 2014] of predictions.

We can consider the associated costs of obtaining the missing values in a new case to be classified (e.g., an X-ray test). In this respect, a *test-cost-sensitive naive Bayes classifier* is proposed in Chai et al. [2004], whose aim is to minimize the expected loss by finding how the unknown test variables should be chosen (sequentially or batch-wise). A different situation arises when we have a fixed budget and we are concerned with costs during the learning phase. Here we wish to decide sequentially which tests

to run on which instance subject to the budget (i.e., *budgeted learning* [Lizotte et al. 2003]). Naive Bayes's conditional independence assumption simplifies the sequential process for test selection.

(d) Instance ranking. In many applications, an accurate ranking of instances is more desirable than their mere classification, for example, a ranking of candidates in terms of several aspects in order to award scholarships. Since naive Bayes produces poor probability estimates [Domingos and Pazzani 1997], an interesting question is to examine this model's ranking behavior in terms of a well-known ranking quality measure, the area under the ROC curve or AUC. When all variables are binary, theoretical results on its optimality for ranking m-of-n concepts are given in Zhang and Su [2008], unlike for classification, where naive Bayes cannot learn all m-of-n concepts [Domingos and Pazzani 1997]. The ideas are extended in Zhang and Sheng [2004] to a weighted naive Bayes given by Equation (9) with $w_c = 1$, where weights w_i are learned using several heuristics.

(e) Imprecise and inaccurate probabilities. Unobserved or rare events, expert estimates, missing data, or small sample sizes can possibly generate imprecise and inaccurate probabilities. Using confidence intervals rather than point estimates for $p(x_i|c)$ and $p(c)$ is an option, as in the *interval estimation naive Bayes* [Robles et al. 2003]. An evolutionary algorithm can search all the possible (precise) models obtained by taking values in those confidence intervals for the most accurate model. A more general way to deal with imprecision in probabilities is by giving a credal set (i.e., the convex hull of a nonempty and finite family of probability distributions). The *naive credal classifier* [Zaffalon 2002] uses the class posterior probability intervals and a dominance criterion to obtain the output of the classification procedure, which, in this case, can be a set of labels instead of singletons. The effects of parameter inaccuracies are investigated in Renooij and van der Gaag [2008] with sensitivity analysis techniques. The effect of varying one parameter on the posterior probability of the class does not significantly influence the performance of the naive Bayes model. However, this article does not investigate the effect of varying more than one parameter at a time.

(f) Text categorization. In this field, documents are represented by a set of random variables $C, X_1, \ldots, X_n$, where C denotes the class of document. X_i has a different meaning depending on the chosen model [Eyheramendy et al. 2002]. Thus, in the *binary independence model*, it represents the presence/absence of a particular term (word) in the document, and $p(x_i|c)$ follows a Bernoulli distribution with parameter p_{ic}. In other models, X_i represents the number of occurrences of particular words in the document. The *multinomial model* assumes that the document length and document class are marginally independent, transforming Equation (8) into

$$p(c|\mathbf{x}) \propto p(c) \left(\sum_{i=1}^{n} x_i\right)! \prod_{i=1}^{n} \frac{p_{ic}^{x_i}}{x_i!}, \tag{10}$$

where, for each c, p_{ic} denotes the probability of occurrence of the ith word and $\sum_{i=1}^{n} p_{ic} = 1$. The *Poisson naive Bayes model* assumes that, in Equation (8), $p(x_i|c)$ follows a Poisson distribution, whereas in the *negative binomial naive Bayes model*, it is a negative binomial distribution.

3.7. Discriminative Learning of Parameters

All previous research models the joint probability distribution $p(\mathbf{x}, c)$ according to what is called a *generative* approach. A *discriminative* approach [Jebara 2004], however, directly models the conditional distribution $p(c|\mathbf{x})$.

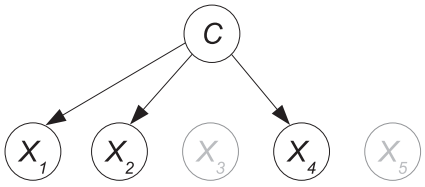


Fig. 4. A selective naive Bayes structure from which $p(c|\mathbf{x}) \propto p(c)p(x_1|c)p(x_2|c)p(x_4|c)$. The variables in the shaded nodes have not been selected.

When computing $p(c|\mathbf{x})$ from the joint probability distribution given by a naive Bayes model, it has been shown [Bishop 1995] to be a linear softmax regression. The parameters of this discriminative model may be estimated by standard techniques (like the Newton-Raphson method). Another more direct way of discriminative learning of the naive Bayes parameters is given in Santafé et al. [2005]: the estimations of parameters maximizing the *conditional likelihood* are approximated using the TM algorithm [Edwards and Lauritzen 2001].

4. SELECTIVE NAIVE BAYES

As mentioned in the previous section, the classification performance of naive Bayes will improve if only relevant, and especially nonredundant, variables are selected to be in the model. Generally, parsimonious models reduce the cost of data acquisition and model learning time, are easier to explain and understand, and increase model applicability, robustness, and performance. Then, a *selective naive Bayes* (Figure 4) is stated as a *feature subset selection* problem, with $\mathbf{X}_F$ denoting the projection of $\mathbf{X}$ onto the selected feature subset $F \subseteq \{1, 2, \ldots, n\}$, where Equation (8) is now

$$p(c|\mathbf{x}) \propto p(c|\mathbf{x}_F) = p(c) \prod_{i \in F} p(x_i|c).$$

The exhaustive search in the space of all possible selective naive Bayes requires the computation of 2^n structures. Although the induction and classification time for a naive Bayes model is short, the enumerative search for the optimal model can be prohibitive. This justifies the use of heuristic approaches for this search.

When a *filter* approach is applied for feature selection, each proposed feature subset is assessed using a scoring measure based on intrinsic characteristics of the data computed from simple statistics on the empirical distribution, totally ignoring the effects on classifier performance. A *wrapper* approach assesses each subset using the classifier performance (accuracy, AUC, F_1 measure, etc.). Finally, an *embedded* approach selects features using the information obtained from training a classifier and is thereby embedded (learning and feature selection tasks cannot be separated) in and specific to a model [Saeys et al. 2007].

4.1. Filter Approaches

When the feature subset is a singleton, we have *univariate filter* methods. This leads to a ranking of features from which the selected feature set is chosen once a threshold on the scoring measure is fixed. The most used scoring measure is the mutual information of each feature and the class variable $I(X_i, C)$ [Pazzani and Billsus 1997]. Other scoring measures for a feature, like odds ratio, weight of evidence, and symmetrical uncertainty coefficient, can be used, some of which are empirically compared in Mladenic and Grobelnik [1999].

The scoring measures in *multivariate filter* methods are defined on a feature subset. The scoring measure introduced in Hall [1999], called *correlation-based feature*

selection (CFS), promotes the inclusion of variables that are relevant for classification and, at the same time, avoids including redundant variables. Any kind of heuristic (forward selection, backward elimination, best first, etc.) can be used to search for this optimal subset. Another possibility is to simply select those features that the `C4.5` algorithm would use in its classification tree, as in Ratanamahatana and Gunopulos [2003]. A Bayesian criterion for feature selection proposed in Kontkanen et al. [1998] is based on approximating the *supervised marginal likelihood* of the class value vector given the rest of the data. This is closely related to the conditional log-likelihood (see Section 8.4), turning the learning of the selective naive Bayes into a discriminative approach.

4.2. Wrapper Approaches

A wrapper approach outputs the feature subset with a higher computational cost than the filter approach. The key issue is how to search the space of feature subsets of cardinality 2^n. The strategies used range from simple heuristics, like greedy forward [Langley and Sage 1994] and floating search [Pernkopf and O'Leary 2003], to more sophisticated population-based heuristics, like genetic algorithms [Liu et al. 2001] and estimation of distribution algorithms [Inza et al. 2000].

For a large n, a wrapper approach may be impracticable even with the simplest heuristics. This is why many researchers apply a wrapper strategy over a reduced filtered subset, thereby adopting a filter-wrapper option [Inza et al. 2004].

4.3. Embedded Approaches

Regularization techniques are a kind of embedded approach that typically sets out to minimize the negative log-likelihood function of the data given the model plus a penalty term on the size of the model parameters. An L_1 penalty is useful for feature selection because the size of some parameters is driven to zero. An *L_1/L_2-regularized naive Bayes* for continuous and discrete predictor variables is introduced in Vidaurre et al. [2012]. In addition, a *stagewise version of the selective naive Bayes*, which can be considered a regularized version of a naive Bayes, is also presented. Whereas the L_1/L_2-regularized naive Bayes model only discards irrelevant predictors, the stagewise version of the selective naive Bayes can discard both irrelevant and redundant predictors.

4.4. Metaclassifiers

As with naive Bayes (Section 3.5), selective naive Bayes models can be combined in a metaclassifier. The *random naive Bayes* [Prinzie and Van den Poel 2007] is a bagged classifier combining many naive Bayes, each of which has been estimated from a bootstrap sample with $m < n$ randomly selected features. The *naive Bayesian classifier committee* [Zheng 1998] sequentially generates selective naive Bayes models to be members of the committee. The probability that a feature is used for the next model increases if the current model performs better than the naive Bayes (with all features). For each class, the probabilities provided by all committee members are summed up, taking as the predicted class the one with the largest summed probability.

Bayesian model averaging (see Equation (5)) is an ensemble learning technique. Applied to all selective naive Bayes models, this gives rise to a unique naive Bayes model, as shown in Dash and Cooper [2002]. Here Dirichlet priors are assumed for $p(\boldsymbol{\theta}_m|S_m)$ in Equation (7) and uniform priors for $p(S_m)$ in Equation (6).

5. SEMINAIVE BAYES

Seminaive Bayes models (Figure 5) aim to relax the conditional independence assumption of naive Bayes by introducing new features obtained as the Cartesian product of two or more original predictor variables. By doing this, the model is able to represent

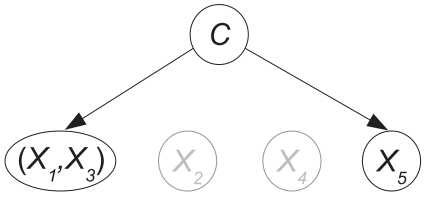


Fig. 5. A seminaive Bayes structure from which $p(c|\mathbf{x}) \propto p(c)p(x_1, x_3|c)p(x_5|c)$.

dependencies between original predictor variables. However, these new predictor variables are still conditionally independent given the class variable. Thus, if $S_j \subseteq \{1, 2, \ldots, n\}$ denotes the indices in the jth feature (original or Cartesian product), $j = 1, \ldots, K$, Equation (8) is now

$$p(c|\mathbf{x}) \propto p(c) \prod_{j=1}^{K} p(\mathbf{x}_{S_j}|c),$$

where $S_j \cap S_l = \emptyset$, for $j \neq l$.

The seminaive Bayes model of Pazzani [1996] starts from an empty structure and considers the best option between (a) adding a variable not used by the current classifier as conditionally independent of the features (original or Cartesian products) used in the classifier, and (b) joining a variable not used by the current classifier with each feature (original or Cartesian products) present in the classifier. This is a greedy search algorithm, called *forward sequential selection and joining*, guided wrapper-wise (the objective function is the classification accuracy), that stops when there is no accuracy improvement. An alternative backward version starting from a naive Bayes, called *backward sequential elimination and joining*, is also proposed by the same author. Evolutionary computation has been used to guide the search for the best semi-naive Bayes model in Robles et al. [2003] wrapper-wise with estimation of distribution algorithms. Using a wrapper approach avoids including redundant variables in the model, since these degrade accuracy, as mentioned in Section 3.

A filter adaptation of the forward sequential selection and joining algorithm is presented in Blanco et al. [2005]. Options (a) and (b) listed previously are evaluated with a χ^2 test of independence based on the mutual information $I(C, X_i)$ of the class and each variable not in the current model (for (a)) and on the mutual information of the class and a joint variable formed by a variable not in the current model and a feature present in the model (for (b)). We always select the variable with the smallest p-value until no more new variables can be added to the model (because they do not reject the null hypothesis of independence). Other filter approaches use alternative scoring metrics like Bayesian Dirichlet equivalence (BDe) [Heckerman et al. 1995], and leave one out and log-likelihood ratio test, as in Abellán et al. [2007]. Every time variables form a new joint variable, this approach [Abellán et al. 2007] tries to merge values of this new variable to reduce its cardinality and computation time. For imprecise probabilities, a filter *seminaive credal classifier* is given in Abellán et al. [2006].

A seminaive Bayes model (or naive Bayes or interval estimation naive Bayes) is the model built in Robles et al. [2004] at the second level of a metaclassifier following a stacked generalization scheme, taking as input data the different labels provided by different classifiers at the first level.

6. ONE-DEPENDENCE BAYESIAN CLASSIFIERS

One-dependence estimators (ODEs) are similar to naive Bayes except that each predictor variable is allowed to depend on at most one other predictor in addition to the class.

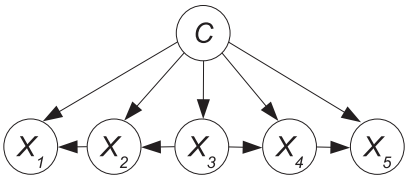


Fig. 6. A TAN structure, whose root node is X_3, from which $p(c|\mathbf{x}) \propto p(c)p(x_1|c, x_2)p(x_2|c, x_3)p(x_3|c)p(x_4|c, x_3)$ $p(x_5|c, x_4)$.

They can improve naive Bayes accuracy when its conditional independence assumption is violated.

6.1. Tree-Augmented Naive Bayes

Unlike in seminaive Bayes, which introduces new features to relax the conditional independence assumption of naive Bayes, the *tree-augmented network* (TAN) [Friedman et al. 1997] maintains the original predictor variables and models relationships of at most order 1 among the variables. Specifically, a tree-shaped graph models the predictor subgraph (Figure 6).

Learning a TAN structure first involves constructing an undirected tree. Kruskal's algorithm [Kruskal 1956] is used to calculate the maximum weighted spanning tree (MWST), containing $n - 1$ edges, where the weight of an edge $X_i - X_j$ is $I(X_i, X_j|C)$, which is the conditional mutual information of X_i and X_j given C. The undirected tree is then converted into a directed tree by selecting at random a variable as the root node and replacing the edges by arcs. This is the tree shaping the predictor subgraph. Finally, a naive Bayes structure is superimposed to form the TAN structure. The posterior distribution in Equation (1) is then

$$p(c|\mathbf{x}) \propto p(c)p(x_r|c) \prod_{i=1, i\neq r}^{n} p(x_i|c, x_{j(i)}), \tag{11}$$

where X_r denotes the root node and $\{X_{j(i)}\} = \mathbf{Pa}(X_i)\backslash C$, for any $i \neq r$.

These ideas are adapted from Chow and Liu [1968], where several trees, one for each value c of the class, were constructed rather than a single tree for the entire domain. This works like TAN, but uses only the cases from $\mathcal{D}$ satisfying $C = c$ to construct each tree. This collection of trees is a special case of a Bayesian multinet, a terminology introduced by Geiger and Heckerman [1996] for the first time (see Section 9).

From a theoretical point of view, the procedures in Chow and Liu [1968] (Figure 7(a)) and Friedman et al. [1997] (Figure 7(b)) construct, respectively, the tree-based Bayesian multinet and the TAN structure that both maximize the likelihood.

Rather than obtaining a spanning tree, the method described in Ruz and Pham [2009] suggests that Kruskal's algorithm be stopped whenever a Bayesian criterion controlling the likelihood of the data and the complexity of the TAN structure holds. The predictor subgraph will then include $e \leq n-1$ arcs. This procedure has been proven to find an augmented naive Bayes classifier that minimizes the Kullback-Leibler (KL) divergence between the real joint probability distribution and the approximation given by the model, across all network structures with e arcs.

Two special situations are when data are incomplete and probabilities are imprecise. The *structural EM algorithm* [Friedman 1997] in the space of trees is used in François and Leray [2006] for the first case. The *tree-based credal classifier* algorithm that is able to induce credal Bayesian networks with a TAN structure is proposed in Zaffalon and Fagiuoli [2003] for the second case.

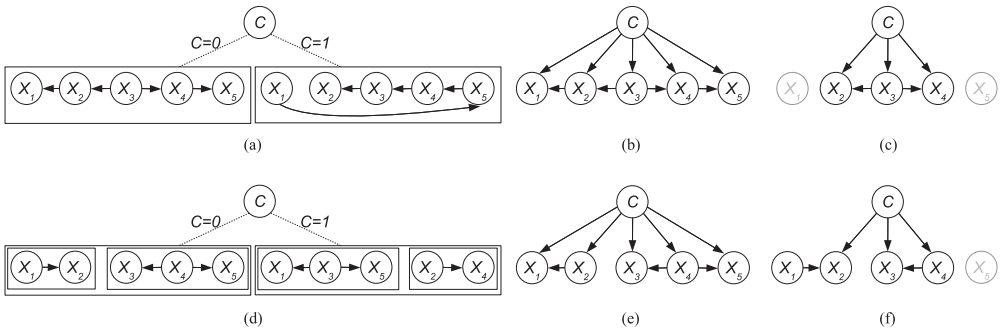


Fig. 7. (a) Bayesian multinet as a collection of trees [Chow and Liu 1968]: $p(C = 0|\mathbf{x}) \propto p(C = 0)p(x_1|C = 0, x_2)p(x_2|C = 0, x_3)p(x_3|C = 0)p(x_4|C = 0, x_3)p(x_5|C = 0, x_4)$ and $p(C = 1|\mathbf{x}) \propto p(C = 1)p(x_1|C = 1)p(x_2|C = 1, x_3)p(x_3|C = 1, x_4)p(x_4|C = 1, x_5)p(x_5|C = 1, x_1)$; (b) TAN [Friedman et al. 1997]: $p(c|\mathbf{x}) \propto p(c)p(x_1|c, x_2)p(x_2|c, x_3)p(x_3|c)p(x_4|c, x_3)p(x_5|c, x_4)$; (c) selective TAN [Blanco et al. 2005]: $p(c|\mathbf{x}) \propto p(c)p(x_2|c, x_3)p(x_3|c)p(x_4|c, x_3)$; (d) Bayesian multinet as a collection of forests [Pham et al. 2002]: $p(C = 0|\mathbf{x}) \propto p(C = 0)p(x_1|C = 0)p(x_2|C = 0, x_1)p(x_3|C = 0, x_4)p(x_4|C = 0)p(x_5|C = 0, x_4)$ and $p(C = 1|\mathbf{x}) \propto p(C = 1)p(x_1|C = 1, x_3)p(x_2|C = 1)p(x_3|C = 1)p(x_4|C = 1, x_2)p(x_5|C = 1, x_3)$; (e) FAN [Lucas 2004]: $p(c|\mathbf{x}) \propto p(c)p(x_1|c, x_2)p(x_2|c)p(x_3|c, x_4)p(x_4|c)p(x_5|c, x_4)$; (f) selective FAN [Ziebart et al. 2007]: $p(c|\mathbf{x}) \propto p(c)p(x_2|c, x_1)p(x_3|c, x_4)p(x_4|c)$.

If the weights of the undirected tree based on conditional mutual information are first filtered with a χ^2 test of independence, the resulting structure is the *selective TAN* [Blanco et al. 2005] (Figure 7(c)). The predictor subgraph could be a forest rather than a tree since it may result in many root nodes.

Other authors propose following a wrapper instead of a filter approach. The next three references, again, lead to forest predictor structures (i.e., a disjoint union of trees). Thus, initializing the network to a naive Bayes, we can consider adding possible arcs from X_i to X_j, for X_j without any predictor variable as parent, and selecting the arc giving the highest accuracy improvement. This hill-climbing search algorithm is described in Keogh and Pazzani [2002]. The authors also propose another less expensive search. Finding the best arc to add is broken down into two steps. First, we consider making each node a superparent in the current classifier (i.e., with arcs directed to all nodes without a predictor parent). The best superparent yields the highest accuracy. Second, we choose one of all the superparent's children (i.e., the favorite child that most improves accuracy) for the final structure. Also starting from a naive Bayes, a sequential floating search heuristic is used in Pernkopf and O'Leary [2003]. In Blanco et al. [2005], by initializing with an empty predictor subgraph, an algorithm greedily decides whether to add a new predictor or to create an arc between two predictors already in the model. Unlike the last two wrapper techniques, it actually performs a feature subset selection.

Forest-augmented naive Bayes. Rather than using a collection of trees as in Chow and Liu [1968], a collection of forests, one for each value c of the class, is built in Pham et al. [2002] (Figure 7(d)). The forests are obtained using a maximum weighted spanning forest algorithm (e.g., [Fredman and Tarjan 1987]). The *forest-augmented naive Bayes* (FAN) was first defined in Lucas [2004], with only one rather than a collection of forests in the predictor subgraph, augmented with a naive Bayes (Figure 7(e)). Therefore, the research reported in Lucas [2004] adapts Pham et al. [2002] for FAN models as Friedman et al. [1997] did with Chow and Liu [1968] for TAN. The *selective FAN* introduced in Ziebart et al. [2007] adds the novelty of allowing the predictor variables to be optionally dependent on the class variable; that is, missing arcs from C to some X_i can be found (Figure 7(f)). Moreover, the learning approach is based on maximizing

the likelihood of the data, which is penalized for avoiding the class variable as a parent.

Metaclassifiers. Bagging-type metaclassifiers use bootstrap samples and thus require an unstable base classifier to generate diverse results from the different classifiers. However, the TAN classifier is stable. A randomization is then needed in the standard TAN algorithm. Thus, the *bagging-randomTAN* in Ma and Shi [2004] takes randomTAN as base classifiers in a bagging scheme. The *randomTAN* randomly selects the edges between predictor variables whose conditional mutual information surpasses a fixed threshold. These selective TAN models vote for the final classification. Using boosting instead means sampling the original data with weights according to the classification results of each data item to form a new dataset for the next classifier. This scheme is employed in the *boosted augmented naive Bayes* (*b*AN) [Jing et al. 2008]. The base classifier is chosen by first running a trial with a naive Bayes, then greedily augmenting the current structure at iteration s with the sth edge having the highest conditional mutual information. We stop when the added edge does not improve the classification accuracy. Note that the final structure of the base classifier can be a FAN.

The *averaged TAN* (ATAN) [Jiang et al. 2012] takes not a random node but each predictor variable as root node and then builds the corresponding MWST conditioned to that selection. Finally, the posterior probabilities $p(c|\mathbf{x})$ of ATAN are given by the average of the n TAN classifier posterior probabilities.

Bayesian model averaging (see Equation (5)) over TAN structures and parameters is carried out in Cerquides and López de Mántaras [2005b]. The authors define decomposable (conjugate) distributions as priors for $p(S_m)$ in Equation (6) and choose Dirichlet priors for $p(\boldsymbol{\theta}_m|S_m)$ in Equation (7). They compute the exact *Bayesian model averaging over TANs*. In addition, they propose an ensemble of the k most probable a posteriori TAN models.

Discriminative learning. A discriminative learning of a TAN model is proposed in Feng et al. [2007]. First, the TAN structure is learned as in Friedman et al. [1997] but replacing the conditional mutual information by the explaining away residual (EAR) criterion [Bilmes 2000], that is, using $I(X_i, X_j|C) - I(X_i, X_j)$. Maximizing EAR over the tree is in fact an approximation to maximizing the conditional likelihood. Second, they define an objective function based mainly on the KL divergence between the empirical distribution and the distribution given by the previous TAN structure for each value c of the class to discriminatively learn the parameters.

A different discriminative score, the maximum margin, is proposed in Pernkopf and Wohlmayr [2013] to search for the structure of TAN with both greedy hill-climbing and simulated annealing strategies. The multiclass margin of an instance $\mathbf{x}^{(i)}$ is $d^{(i)} = \frac{p(c^{(i)}|\mathbf{x}^{(i)})}{\max_{c\neq c^{(i)}} p(c|\mathbf{x}^{(i)})}$. Rather than searching for the structure that maximizes $\min_{i=1,\ldots,N} d^{(i)}$, this is relaxed with a soft margin, finally defining the *maximum margin score* of a structure as $\sum_{i=1}^{N} \min\{1, \lambda \log d^{(i)}\}$, where $\lambda > 0$ is a scaling parameter and is set by cross-validation.

As in Section 3.7 with naive Bayes, the TM algorithm [Edwards and Lauritzen 2001] can be adapted for the discriminatively learning parameters in a TAN classifier [Santafé et al. 2005].

6.2. SuperParent-One-Dependence Estimators

SuperParent-One-Dependence Estimators (SPODEs) are an ODE where all predictors depend on the same predictor (the superparent) in addition to the class [Keogh and Pazzani 2002] (Figure 8). Note that this is a particular case of a TAN model. The

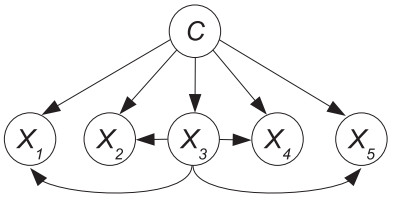


Fig. 8. A SPODE structure, with X_3 as superparent, from which $p(c|\mathbf{x}) \propto p(c)p(x_1|c,x_3)p(x_2|c,x_3)p(x_3|c)$ $p(x_4|c,x_3)p(x_5|c,x_3)$.

posterior distribution in Equation (1) is

$$p(c|\mathbf{x}) \propto p(c)p(x_{sp}|c) \prod_{i=1,i\neq sp}^{n} p(x_i|c,x_{sp}),$$

where X_{sp} denotes the superparent node. This equation is similar to Equation (11), particularized as $X_r = X_{j(i)} = X_{sp}$, for any $i \neq sp$.

Metaclassifiers. The *averaged one-dependence estimator* (AODE) [Webb et al. 2005] averages the predictions of all qualified SPODEs, where "qualified" means that it includes, for each instance $\mathbf{x} = (x_1, \ldots, x_{sp}, \ldots, x_n)$, only the SPODEs for which the probability estimates are accurate, that is, where the training data contain more than m instances verifying $X_{sp} = x_{sp}$. The authors suggest fixing $m = 30$. The average prediction is given by

$$p(c|\mathbf{x}) \propto p(c,\mathbf{x}) = \frac{1}{|\mathcal{SP}^m_{\mathbf{x}}|} \sum_{X_{sp}\in\mathcal{SP}^m_{\mathbf{x}}} p(c)p(x_{sp}|c) \prod_{i=1,i\neq sp}^{n} p(x_i|c,x_{sp}), \tag{12}$$

where $\mathcal{SP}^m_{\mathbf{x}}$ denotes for each $\mathbf{x}$ the set of predictor variables qualified as superparents and $|\cdot|$ is its cardinal. AODE avoids model selection, thereby decreasing the variance component of the classifier.

The AODE can be further improved by deleting X_j from the set of predictors whenever $P(x_j|x_i) = 1$ (x_i and x_j are highly dependent predictor values) when classifying a new instance $\mathbf{x}$. Note that this technique introduced in Zheng and Webb [2006] is performed at classification time for each new instance, and this is why it is called *lazy elimination*. It is shown that it significantly reduces classification bias and error without undue computation.

Another improvement is the *lazy AODE* [Jiang and Zhang 2006], which builds an AODE for each test instance. The training data is expanded by adding a number of copies (clones) of each training instance equal to its similarity to the test instance. This similarity is the number of identical predictor variables.

Since AODE requires all the SPODE models to be stored in main memory, *generalized additive Bayesian network classifiers* (GABNs) defined in Li et al. [2007] propose aggregating only some SPODEs (or other simple Bayesian classifiers) within the framework of generalized additive models. SPODEs with the lowest mutual information scores $I(X_{sp}, C)$ are not considered in the aggregation. Thus, this aggregation is given by the linear combination of $n' \leq n$ probabilities $p_{sp}(\mathbf{x}, c)$ obtained in the SPODE models:

$$\sum_{sp=1}^{n'} \lambda_{sp} g_{sp}(p_{sp}(\mathbf{x}, c)),$$

where g_{sp} is the link function and $0 \leq \lambda_{sp} \leq 1$ are parameters to be estimated such that $\sum_{sp=1}^{n'} \lambda_{sp} = 1$. When g_{sp} is the log function, then $p(\mathbf{x}, c) \propto \prod_{sp=1}^{n'} p_{sp}^{\lambda_{sp}}(\mathbf{x}, c)$. It is

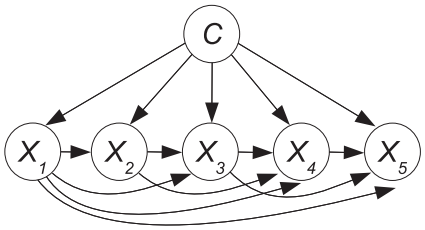


Fig. 9. An example of 3-DB structure from which $p(c|\mathbf{x}) \propto p(c)p(x_1|c)p(x_2|c,x_1)p(x_3|c,x_1,x_2)p(x_4|c,x_1,x_2,x_3)p(x_5|c,x_1,x_3,x_4)$.

easy to design a gradient-based method to optimize its associated quasi-likelihood that outputs the combining parameters λ_{sp}.

Another way to obtain an ensemble of SPODEs in the AODE is proposed in Yang et al. [2005] as a wrapper approach. The aim is to select SPODEs so as to maximize classification accuracy. We need a metric (like minimum description length [MDL], minimum message length [MML], leave-one-out classification accuracy, accuracy from backward sequential elimination, or forward sequential addition processes) to order the n possible SPODEs for selection, and a stopping criterion always based on the accuracy.

The idea of Yang et al. [2007] is to compute the final predictions as a weighted average in Equation (12), rather than as an average. Four different weighting schemes are then proposed. Two of them use the posterior probability of each SPODE given the data as its weight. The first is based on the inversion of Shannon's law and the second is within a Bayesian model averaging, where uniform priors over the n SPODE structures and Dirichlet priors over the corresponding parameters are assumed. The other two schemes use a MAP estimation to find the most probable a posteriori set of weights for a SPODE ensemble, assuming a Dirichlet prior over the weights. These two last schemes differ as to the posterior, generative, or discriminative models (see Cerquides and López de Mántaras [2005a] for further details).

6.3. Other One-Dependence Estimators

The *weighted ODE* can be used to approximate the conditional probabilities $p(x_i|c)$ in the naive Bayes. This was proposed by Jiang et al. [2009], resulting in

$$p(c|\mathbf{x}) \propto p(c,\mathbf{x}) \approx p(c)\prod_{i=1}^{n}\left(\sum_{j=1,j\neq i}^{n} w_{ij}\,p(x_i|c,x_j)\right), \tag{13}$$

where $w_{ij} \propto I(X_i,X_j|C)$. The same authors propose in Jiang et al. [2012] other weighting schemes, based on performance measures of the different ODE models, like AUC or classification accuracy.

The *hidden one-dependence estimator* classifier (HODE) [Flores et al. 2009] avoids using any SPODE. HODE introduces, via the EM algorithm, a new variable (the hidden variable H), with the aim of representing the links existing in the n SPODE models. Node C in the naive Bayes structure is replaced by the Cartesian product of C and H. Then we have to estimate the probability of x_i conditioned by c and h searching for $\arg\max_c \sum_h p(c,h)\prod_{i=1}^{n} p(x_i|c,h)$.

7. *k*-DEPENDENCE BAYESIAN CLASSIFIERS

The *k-dependence Bayesian classifier* (k-DB) [Sahami 1996] allows each predictor variable to have a maximum of k parent variables apart from the class variable (Figure 9). The inclusion order of the predictor variables X_i in the model is given by $I(X_i,C)$,

starting with the highest. Once X_i enters the model, its parents are selected by choosing those k variables X_j in the model with the highest values of $I(X_i, X_j|C)$. The main disadvantages of the standard k-DB are the lack of feature selection (all the original predictor variables are included in the final model) and the need to determine the optimal value for k. Also, once k has been fixed, the number of parents of each predictor variable is inflexible. Obviously, naive Bayes and TAN are particular cases of k-DBs, with $k = 0$ and $k = 1$, respectively.

The posterior distribution in Equation (1) is

$$p(c|\mathbf{x}) \propto p(c) \prod_{i=1}^{n} p(x_i|c, x_{i_1}, \ldots, x_{i_k}),$$

where $X_{i_1}, \ldots, X_{i_k}$ are parents of X_i in the structure. Note that the first k variables entering the model will have fewer than k parents (the first variable entering the model has no parents, the second variable has one parent, and so on) and the remaining $n - k$ variables have exactly k parents.

Feature subset selection is performed in Blanco et al. [2005] within a k-DB using filter and wrapper approaches. In the filter approach, an initial step selects the predictor variables that pass a χ^2 test of independence based on the mutual information $I(C, X_i)$. Then the standard k-DB algorithm is applied on this reduced subset, considering only those arcs that pass an analogous independence test based on conditional mutual information $I(X_i, X_j|C)$. In the wrapper approach, as in the wrapper TAN approach discussed in Section 6.1, the decision on whether to add a new predictor or to create an arc between two predictors already in the model is guided by accuracy, provided that the added arc does not violate the k-DB restrictions. As a consequence, all the predictors in the structures output by this wrapper approach have at most k parents, but there is no need to have $n - k$ variables with exactly k parents. In general, graphs where each node has at most k parents are called *k-graphs*.

A k-graph as the predictor subgraph is also the result of a kind of evolutionary computation method described in Xiao et al. [2009], inspired by the so-called group method of data handling (GMDH) [Ivakhnenko 1970]. The algorithm to build *GMDH-based Bayesian classifiers* starts from a set of $s \propto n + 1$ models with only one arc, corresponding to the pair of variables (C included) with the highest mutual information. Then a new set of $\binom{s}{2}$ models is obtained by pairwise joining the previous structures. The best s models according to BDe or BIC are selected. This process that incrementally increases the model complexity is repeated until the new best does not improve the current best model. The number of parents is always bounded by a fixed k.

The k-graphs obtained in Carvalho et al. [2007] are obliged to be consistent with an order between the predictor variables. This order, σ, is based on a breadth-first search (BFS) over the TAN predictor subgraph obtained in the usual manner [Friedman et al. 1997]. This means that for any arc $X_i \to X_j$ in the k-graph, X_i is visited before X_j in a total order completing σ. The learning algorithm of *BFS-consistent Bayesian network classifiers* can cope with any decomposable score, score expressible as a sum of local scores that depend only on each node and its parents.

k-graphs are also induced in Pernkopf and Bilmes [2010]. They first establish an ordering of the predictor variables by using a greedy algorithm. A variable X is chosen whenever it is the most informative about C given the previous variables in the order, where informativeness is measured by the conditional mutual information, $I(C, X|\mathbf{X}_{\text{prev}})$. This order can alternatively use classification accuracy as a score assuming a fully connected subgraph over C, X, and $\mathbf{X}_{\text{prev}}$. In any case, the best k parents for each variable among $\mathbf{X}_{\text{prev}}$ are selected in a second step by scoring each possibility

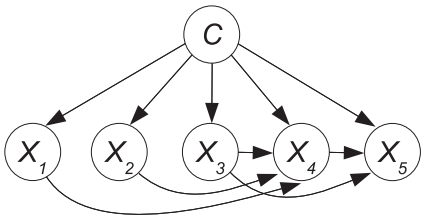


Fig. 10. A Bayesian network-augmented naive Bayes structure from which $p(c|\mathbf{x}) \propto p(c)p(x_1|c)p(x_2|c)p(x_3|c)$ $p(x_4|c, x_1, x_2, x_3)p(x_5|c, x_3, x_4)$.

with the classification accuracy. Here a naive Bayes assumption is used for $\mathbf{X}\backslash\{\mathbf{X}_{\text{prev}}, X\}$, that is, the variables whose parents have not yet been chosen.

Metaclassifiers. A combination of k-DB models in a bagging fashion is proposed in Louzada and Ara [2012].

8. GENERAL BAYESIAN NETWORK CLASSIFIERS

This section discusses more general structures. First, relaxing the structure of the predictor subgraph but maintaining C without any parent defines a Bayesian network-augmented naive Bayes (Section 8.1). Second, if C is allowed to have parents, its Markov blanket is the only knowledge needed to predict its behavior (see Equation (3)), and some classifiers have been designed to search for the Markov blanket (Section 8.2). Finally, a very general unrestricted Bayesian network that does not consider C as a special variable can be induced with any existing Bayesian network structure learning algorithm. The corresponding Markov blanket of C can be used later for classification purposes (Section 8.3). In all three cases, Equation (1) is

$$p(c|\mathbf{x}) \propto p(c|\mathbf{pa}(c)) \prod_{i=1}^{n} p(x_i|\mathbf{pa}(x_i)),$$

where $\mathbf{Pa}(C) = \emptyset$ in Section 8.1.

8.1. Bayesian Network-Augmented Naive Bayes

Relaxing the fixed number of parents, k, in a k-DB, does not place any limitations on links among predictor variables (except that they do not form a cycle); that is, a Bayesian network structure can be the predictor subgraph (Figure 10). This model is called *Bayesian network-augmented naive Bayes* (BAN), a term first coined by Friedman et al. [1997]. The factorization is

$$p(c|\mathbf{x}) \propto p(c) \prod_{i=1}^{n} p(x_i|\mathbf{pa}(x_i)).$$

The first reference to a learning algorithm for this model is Ezawa and Norton [1996]. First, it ranks the n predictor variables based on $I(X_i, C)$, and then it selects the minimum number of predictor variables k verifying $\sum_{j=1}^{k} I(X_j, C) \geq t_{CX} \sum_{j=1}^{n} I(X_j, C)$, where $0 < t_{CX} < 1$ is the threshold. Second, $I(X_i, X_j|C)$ is computed for all pairs of selected variables. The edges corresponding to the highest values are selected until a percentage t_{XX} of the overall conditional mutual information $\sum_{i<j}^{k} I(X_i, X_j|C)$ is surpassed. Edge directionality is based on the variable ranking of the first step: higher-ranked variables point toward lower-ranked variables. Note that this algorithm resembles the initial proposal for learning a k-DB model [Sahami 1996]; see Section 7.

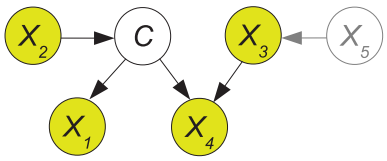


Fig. 11. A Markov blanket structure for C from which $p(c|\mathbf{x}) \propto p(c|x_2)p(x_1|c)p(x_2)p(x_3)p(x_4|c,x_3)$.

As explained in Section 2, a Bayesian network can be learned using conditional independence tests. This is the strategy adopted in Cheng and Greiner [1999] to obtain the predictor subgraph. This algorithm has three phases: drafting, thickening, and thinning. First, it computes $I(X_i, X_j|C)$ as a measure of closeness and creates a draft based on this information. Second, it adds arcs (thickening) when the pairs of nodes cannot be d-separated, resulting in an independence map (I-map) of the underlying dependency model. Third, each arc of the I-map is examined using conditional independence tests and will be removed (thinning) if both nodes of the arc can be d-separated. The final result is the minimal I-map [Pearl 1988].

Also, a Bayesian network can be learned with a score + search technique. In Friedman et al. [1997], the structure is learned by minimizing the MDL score with a greedy forward search. In van Gerven and Lucas [2004], the (conditional) mutual information score and a forward greedy search is used in the *maximum mutual information* (MMI) algorithm. MMI iteratively selects the arc with the highest (conditional) mutual information from two sets of candidate arcs: $C \rightarrow X_i$-type arcs, chosen with $I(X_i, C)$, followed, as soon as C has children, by $X_j \rightarrow X_i$-type arcs where X_i is a child of C, chosen with $I(X_i, X_j|\mathbf{Pa}(X_i))$. Note that $\mathbf{Pa}(X_i)$ can add new variables at each iteration, and the conditional mutual information should be recomputed accordingly. The parameter learning uses nonuniform Dirichlet priors to avoid spurious dependences. Another example of a score + search approach is reported in Pernkopf and O'Leary [2003], where accuracy is used as the score with a sequential floating search heuristic.

8.2. Bayesian Classifiers Based on Identifying the Markov Blanket of the Class Variable

(a) Detecting conditional independences. Finding the Markov blanket of C (Figure 11), MB_C, can be stated as a feature selection problem, where we start from the set of all the predictor variables and eliminate a variable at each step (backward greedy strategy) until we have approximated MB_C. A feature is eliminated if it gives little or no additional information about C beyond what is subsumed by the remaining features. The method in Koller and Sahami [1996] eliminates feature by feature trying to keep $p(C|MB_C^{(t)})$, the conditional probability of C given the current estimation of the Markov blanket at step t, as close to $p(C|\mathbf{X})$ as possible. Closeness is defined by the expected KL divergence. The main idea is to note that eliminating a variable X_i^*, which is conditionally independent of C given $MB_C^{(t)}$, keeps the expected "distance" from $p(C|MB_C^{(t)}, X_i)$ to $p(C|MB_C^{(t)})$ close to zero. The obtained succession of $\{MB_C^{(t)}\}_t$, where $MB_C^{(t)} = MB_C^{(t-1)} \backslash \{X_i^*\}$, should converge to the true MB_C.

At each step t, the algorithm chooses which variable X_i^* to eliminate, as follows. For each X_i, we compute for any X_j not yet eliminated, $D_{KL}(p(C|X_i = x_i, X_j = x_j), p(C|X_j = x_j)), \forall x_i, x_j, j \neq i$, where D_{KL} is the KL divergence. The expected D_{KL} is then computed as $\delta(X_i|X_j) = \sum_{x_i,x_j} p(x_i, x_j) D_{KL}(p(C|X_i = x_i, X_j = x_j), p(C|X_j = x_j))$. We select the K features $(X_{i_1}, \ldots, X_{i_K}) = \mathbf{M}_i$ for which $\delta(X_i|X_j)$ is smallest. $\mathbf{M}_i$ tries to capture the variables X_j for which X_i is conditionally independent of C given X_j. The process is repeated for each X_i, and then we choose the variable X_i^* to be eliminated as the one

with minimum

$$\sum_{\mathbf{m}_i, x_i} p(\mathbf{m}_i, x_i) D_{KL}(p(C|\mathbf{M}_i = \mathbf{m}_i, X_i = x_i), p(C|\mathbf{M}_i = \mathbf{m}_i)).$$

Finally, the next step $t+1$ is started with $MB_C^{(t+1)} = MB_C^{(t)} \backslash \{X_i^*\}$. The number of steps is prespecified and is the number of variables for elimination from the approximate Markov blanket. Note that, as mentioned in Koller and Sahami [1996], the algorithm is suboptimal in many ways, particularly due to the very naive approximations that it uses and the need to specify a good value for K and for the number of variables in the Markov blanket.

This and the following algorithms are based on the observation that if $X_i \notin MB_C$ then $I_p(C, X_i|MB_C)$ holds; that is, C and X_i are conditionally independent under p given MB_C. This holds if we apply the decomposition property of the conditional independence [Pearl 1988]

$$I_p(T, Y \cup W|Z) \Rightarrow I_p(T, Y|Z), I_p(T, W|Z) \tag{14}$$

to Equation (3).

A common assumption in all these algorithms is that $\mathcal{D}$ is a sample from a probability distribution p faithful to a DAG representing a Bayesian network.

The *grow-shrink (GS) Markov blanket algorithm* [Margaritis and Thrun 2000] starts from an empty Markov blanket, current Markov blanket CMB_C, and adds a variable X_i as long as the Markov blanket property of C is violated, that is, $\neg I_p(C, X_i|CMB_C)$, until there are no more such variables (growing phase). Many false positives may have entered the MB_C during the growing phase. Thus, the second phase identifies and removes the variables that are independent of C given the other variables in the MB_C one by one (shrinking phase). In practice, it is possible to reduce the number of tests in the shrinking phase by heuristically ordering the variables by ascending $I(X_i, C)$ or the probability of dependence between X_i and C in the growing step. Orientation rules are then applied to this Markov blanket to get its directed version. GS is the first correct Markov blanket induction algorithm under the faithfulness assumption; that is, it returns the true MB_C. GS is scalable because it outputs the Markov blanket of C without learning a Bayesian network for all variables $\mathbf{X}$ and C. GS has to condition on at least as many variables simultaneously as the Markov blanket size, and it is therefore impractical, because it requires a sample that grows exponentially to this size if the conditional independence tests are to be reliable. This means that GS is not data efficient. A randomized version of the GS algorithm with members of the conditioning set chosen randomly from CMB_C is also proposed as a faster and more reliable variant.

The *incremental association Markov blanket* (IAMB) algorithm [Tsamardinos and Aliferis 2003], a modified version of GS, consists of a forward phase followed by a backward phase. Starting from an empty Markov blanket, it iteratively includes the variable X_i that has the highest association with C conditioned on CMB_C (e.g., conditional mutual information) in the first forward (admission) phase, after checking the same condition as in GS ($\neg I_p(C, X_i|CMB_C)$). We stop when this association is weak. For each $X_i \in CMB_C$, we remove X_i from CMB_C if $I_p(C, X_i|CMB_C \backslash \{X_i\})$ holds to eliminate the false positives in the second backward conditioning phase. IAMB scales to high-dimensional datasets. The authors prove that the Markov blanket corresponds to the strongly relevant features as defined by Kohavi and John [1997]. Likewise to GS, IAMB is correct and scalable but data inefficient.

There have been many variants of the IAMB algorithm. The InterIAMBnPC algorithm [Tsamardinos et al. 2003a] interleaves the admission phase with backward conditioning attempting to keep the size of CMB_C as small as possible during all

the steps. It also substitutes the backward conditioning phase with the PC algorithm [Spirtes et al. 1993]. Fast-IAMB [Yaramakala and Margaritis 2005] speeds up IAMB, reducing the number of tests in the admission phase by adding not one but a number of variables at a time.

The *HITON* algorithm [Aliferis et al. 2003] consists of three steps. First, HITON-PC identifies the parents and children of C, the set PC. This is started from an empty set and includes the variable X_i that has the maximum association with C in the current PC, CPC. Then, a variable $X_j \in CPC$ that meets $\neg I_p(C, X_j|S)$ for some subset S from CPC is removed from CPC and not considered again for admission. The process is repeated until no more variables are left. After outputting PC, in the second step, HITON-PC is again applied to each variable in PC to obtain $PCPC$, the parents and children of PC. Thus, the current MB_C is $CMB_C = PC \cup PCPC$. False positives, which retain just the spouses of C, are removed from CMB_C: $X_j \in CMB_C$ is only retained if $\nexists S \in CMB_C \backslash PC$ such that $\neg I_p(C, X_j|S)$. Unlike the GS and IAMB algorithms, HITON works with conditional (in)dependence statements involving any subset S in CMB_C, rather than just with CMB_C. Finally, in a third step, a greedy backward elimination approach is applied wrapper-like to the previously obtained Markov blanket. HITON is scalable and data efficient because the number of instances required to identify the Markov blanket does not depend on its size but on its topology. However, HITON is incorrect, as proved by Peña et al. [2007].

The *max-min Markov blanket* (MMMB) algorithm [Tsamardinos et al. 2003b] is similar to HITON. However, it chooses the variable X_i in CPC that exhibits the maximum association with C conditioned on the subset S^* of CPC that achieves the minimum association possible for this variable; that is, S^* is the subset S of CPC that minimizes the association of X_i and C given S. This selection method typically admits very few false positives, whereby all subsets on which we condition in the next steps have a manageable size. Also, the second step of MMMB introduces a more sophisticated criterion to identify the spouses of C than HITON. MMMB has the same properties as HITON.

The *parents- and children-based Markov boundary* (PCMB) algorithm [Peña et al. 2007] is a variant of MMMB that incorporates so-called "symmetry correction." The parents–children relationship is symmetric in the sense that X_i belongs to the set of parents and children of C, and C should also belong to the set of parents and children of X_i. A breach of this symmetry is a sign of a false-positive member in the Markov blanket. This leads to the first algorithm that is correct, scalable, and data efficient. This symmetry correction, based on an AND operator, makes it harder for a true positive to enter the Markov blanket. This is relaxed in the MBOR algorithm [Rodrigues de Morais and Aussem 2010], which uses an OR operator and is correct and scalable but data inefficient. A faster PCMB called *breadth-first search of Markov blanket* (BFMB) [Fu and Desmarais 2007] relies on fewer data passes and conditioning on the minimum set.

The *generalized local learning framework for Markov blanket induction* algorithms is proposed in Aliferis et al. [2010]. It can be instantiated in many ways, giving rise to existing state-of-the-art (HITON and MMPC) algorithms. Both the PC set and the Markov blanket are seen as the results of searching for direct causes, direct effects, and direct causes of the direct effects of a variable C.

Table I shows a summary of the main algorithms assuming faithfulness and their properties.

Few algorithms have tried to relax the faithfulness assumption. A weaker condition is the *composition property*, which is the converse of Equation (14), which does not have the guarantee of the Markov blanket being unique. IAMB is still correct under this composition property, but because it is a deterministic algorithm, it cannot discover

Table I. Properties of the Main Algorithms for Markov Blanket Discovery under the Faithfulness Assumption

	Correct	Scalable	Data efficient
GS [Margaritis and Thrun 2000]	✓	✓	
IAMB [Tsamardinos and Aliferis 2003]	✓	✓	
HITON [Aliferis et al. 2003]		✓	✓
MMMB [Tsamardinos et al. 2003b]		✓	✓
PCMB [Peña et al. 2007]	✓	✓	✓
MBOR [Rodrigues de Morais and Aussem 2010]	✓	✓	

different Markov blankets. This drawback is overcome by KIAMB [Peña et al. 2007], a stochastic version of IAMB, which is not only correct and scalable like IAMB but also data efficient unlike IAMB. Rather than conditioning on CMB_C when searching for the highest association in the IAMB admission phase, KIAMB conditions on a random subset of CMB_C, whose size is proportional to $K \in [0, 1]$. IAMB corresponds to KIAMB with $K = 1$.

Note that none of these algorithms takes into account arcs between either the children of C or $\mathbf{Pa}(C)$ and the children of C.

(b) Score + search techniques. The *partial Bayesian network* (PBN) for the Markov blanket around C [Madden 2002] involves three steps. In the first step, each predictor variable is classified as either parent of C, child of C, or unconnected to C. During the second step, the spouses of C are added from the set of parents and unconnected nodes. The third step determines the dependences between the nodes that are children of C. The three steps are guided by the K2 score [Cooper and Herskovits 1992], thereby requiring a node ordering. The inclusion of an arc is decided with the score in a forward greedy way. A similar idea is presented in dos Santos et al. [2011], where the K2 algorithm [Cooper and Herskovits 1992] is applied on an ordering starting with C. This ordering prevents C from having parents resulting in an approximated Markov blanket of C.

For small sample situations, a bootstrap procedure for determining membership in the Markov blanket is proposed in Friedman et al. [1999]. They answer the question of how confident we can be that X_i is in X_j's Markov blanket (in our case we would be interested in $X_j = C$). From each bootstrap sample, a Bayesian network is learned using the BDe score with a uniform prior distribution and using a greedy hill-climbing search. Using the procedure described in Chickering [1995], each Bayesian network is converted into a partially directed acyclic graph (PDAG). From these PDAGs, the final PDAG is composed of the arcs and edges whose confidence (measured by their occurrence frequency in these networks) surpasses a given threshold. A PDAG represents an *equivalence class* of Bayesian network structures, where equivalence means that all networks in the class imply the same set of independence statements. Thus, an equivalence class includes equivalent networks, with the same skeleton (the undirected version of the DAG) and the same set of immoralities or v-structures (arcs $X \rightarrow Z$ and $Y \rightarrow Z$ but with nonadjacent X and Y) [Verma and Pearl 1990]. An arc in a PDAG denotes that all members in the equivalence class contain that arc; an edge $X_i - X_j$ in a PDAG indicates that some members contain the arc $X_i \rightarrow X_j$ and some contain $X_j \rightarrow X_i$.

Rather than using a filter score, the search can be guided in a wrapper-wise using classification accuracy as the score. An example is given in Sierra and Larrañaga [1998], where the search is performed by means of a genetic algorithm. Each individual in the population represents a Markov blanket structure for C.

(c) Hybrid techniques. A two-stage algorithm called *tabu search-enhanced Markov blanket* is presented in Bai et al. [2008]. In the first stage, an initial Markov blanket is

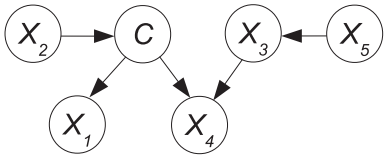


Fig. 12. An unrestricted Bayesian network classifier structure from which $p(c|\mathbf{x}) \propto p(c|x_2)p(x_1|c)p(x_2)p(x_3)p(x_4|c,x_3)$.

obtained based on conditional independence tests carried out according to a breadth-first search heuristic. In the second stage, tabu search enhancement, allowing four kinds of move (arc addition, arc deletion, arc switch, and arc switch with node pruning) is introduced. Each possible move is evaluated taking into account classification accuracy.

8.3. Unrestricted Bayesian Classifiers

This section includes the general unrestricted Bayesian classifiers where C is not considered as a special variable in the induction process (Figure 12).

The complexity of algorithms that learn Bayesian networks from data identifying high-scoring structures in which each node has at most k parents, for all $k \geq 3$, has been shown to be NP hard [Chickering et al. 2004]. It holds whenever the learning algorithm uses a consistent scoring criterion and is applied to a sufficiently large dataset. This justifies the use of search heuristics.

The *K2-attribute selection* (K2-AS) algorithm [Provan and Singh 1995] consists of two main steps. The node selection phase chooses the set of nodes from which the final network is built. In the network construction phase, the network is built with those nodes. Nodes are selected incrementally by adding the variable whose inclusion results in the maximum increase in accuracy (of the resulting network). Using these selected variables, the final network is built using the *CB algorithm* [Singh and Valtorta 1995]. This algorithm uses conditional independence tests to generate a "good" node ordering and then uses the K2 algorithm on that ordering to induce the Bayesian network. A variant of K2-AS is Info-AS [Singh and Provan 1996]. They differ only as to node selection being guided by a conditional information-theoretic metric (conditional information gain, conditional gain ratio, or complement of conditional distance). A simpler approach is to use a node ordering for the K2 algorithm given by the ranking of variables yielded with a score (like information gain or chi-squared score) as in Hruschka and Ebecken [2007].

Instead of searching the Bayesian classifier in the space of DAGs, we can use a reduced search space that consists of a type of PDAGs, called *class-focused restricted PDAGs* (C-RPDAGs) [Acid et al. 2005]. C-RPDAGs combine two concepts of DAG equivalence: independence equivalence and a new concept, classification equivalence. This classification equivalence means producing the same posterior probabilities for the class. Local search is performed by means of specific operators to move from one C-RPDAG to another neighboring C-RPDAG. Standard decomposable and score-equivalent (where equivalent networks have the same score) functions guide the search.

As mentioned at the beginning of this section, from the general Bayesian network obtained with all these methods, the Markov blanket of C is used for classification.

Metaclassifiers. Following the stacked generalization method, a general Bayesian network classifier is built in Sierra et al. [2001] from the response given by a set of classifiers. The algorithm for building this network searches for the structure that maximizes classification accuracy, guided by a genetic algorithm.

Exact Bayesian model averaging of a particular class of structures, consistent with a fixed partial ordering of the nodes and with bounded in-degree k, is considered in

Table II. Generative and Discriminative Approaches for Structure and Parameter Learning of General Bayesian Network Classifiers

		Structure learning	
		Generative	Discriminative
Parameter learning	Generative	Sections 8.1, 8.2, 8.3	CMDL [Grossman and Domingos 2004], CBIC [Guo and Greiner 2005], $\hat{f}$CLL [Carvalho et al. 2011], *ACL-MLE* [Burge and Lane 2005], EAR [Narasimhan and Bilmes 2005], MDL-FS [Drugan and Wiering 2010], *Hist-dist* [Sierra et al. 2009]
	Discriminative	LR-Roos [Roos et al. 2005], LR-Feelders [Feelders and Ivanovs 2006], ELR [Greiner and Zhou 2002; Greiner et al. 2005], DFE [Su et al. 2008], ECL, ACL, and EBW [Pernkopf and Wohlmayr 2009], MCLR [Guo et al. 2005; Pernkopf et al. 2012]	CMDL-ELR [Grossman and Domingos 2004], CBIC-ELR [Guo and Greiner 2005], *ACL-Max* [Burge and Lane 2005]
	Generative-Discriminative	Normalized hybrid [Raina et al. 2004; Fujino et al. 2007], JoDiG [Xue and Titterington 2010], HBayes [Kang and Tian 2006], Bayesian blending [Bishop and Lasserre 2007]	

Dash and Cooper [2004]. The authors prove that there is a single Bayesian network whose prediction is equivalent to the one obtained by averaging the structures of this particular class. Since constructing this network is computationally prohibitive, they provide a tractable approximation whereby approximate model-averaging probability calculations can be performed in linear time. Rather than starting from a fixed node order, which is hard to obtain and may affect classification performance, the idea of Hwang and Zhang [2005] is to extend Bayesian model averaging of general Bayesian network classifiers by averaging over several distinct node orders. The average is approximated using the Markov chain Monte Carlo sampling technique. This method performs well when the dataset is sparse and noisy.

8.4. Discriminative Learning of General Bayesian Network Classifiers

As mentioned in Section 3.7, generative classifiers learn a model of the joint probability distribution $p(\mathbf{x}, c)$ and perform classification using Bayes's rule to compute the posterior probability of the class variable. The standard approach for learning generative classifiers is maximum likelihood estimation, possibly augmented with a (Bayesian) smoothing prior. Discriminative classifiers directly model the posterior probability of the class variable, which is the distribution used for classification. Therefore, generative models maximize the log-likelihood or a related function, whereas discriminative models maximize the conditional log-likelihood. Table II summarizes the content of this section.

(a) Discriminative learning of structures. The log-likelihood of the data $\mathcal{D}$ given a Bayesian network classifier B, $LL(\mathcal{D}|B)$, and the conditional log-likelihood, $CLL(\mathcal{D}|B)$,

are both related as follows:

$$\begin{aligned} LL(\mathcal{D}|B) &= \sum_{i=1}^{N} \log p_B\big(c^{(i)}, x_1^{(i)}, \ldots, x_n^{(i)}\big) \\ &= \sum_{i=1}^{N} \log p_B\big(c^{(i)}|x_1^{(i)}, \ldots, x_n^{(i)}\big) + \sum_{i=1}^{N} \log p_B\big(x_1^{(i)}, \ldots, x_n^{(i)}\big) \\ &= CLL(\mathcal{D}|B) + \sum_{i=1}^{N} \log p_B\big(x_1^{(i)}, \ldots, x_n^{(i)}\big). \end{aligned} \tag{15}$$

It is the first addend that matters in classification, and a better approach would be to use $CLL(\mathcal{D}|B)$ alone as the objective function. Unfortunately, the CLL function does not decompose into a separate term for each variable, and there is no known closed-form solution for the optimal parameter estimates.

The CLL function is used in Grossman and Domingos [2004] to learn the structure of the network, where the maximum number of parents per variable is bounded, while parameters are approximated by their maximum likelihood estimates (MLEs), which is extremely fast. Also, they propose using a modified CLL, which penalizes complex structures via the number of parameters in the network, that is, a *conditional MDL* score (CMDL). A hill-climbing algorithm is used to maximize CLL and CMDL, starting from an empty network and at each step considering the addition, deletion, or reversion of an arc. Additionally, this discriminative learning of structures is extended to a discriminative learning of parameters by computing their estimates via the extended logistic regression (ELR) algorithm [Greiner and Zhou 2002], although the results were not much better.

Another way of modifying CLL is to penalize by the number of parameters in C's Markov blanket. This results in the *conditional BIC* score (CBIC) defined in Guo and Greiner [2005] as an analog of the generative BIC criterion. This CBIC criterion can be accompanied by generative (MLE) or discriminative (ELR) parameter learning.

Rather than working with CLL, other authors propose criteria similar to CLL but with better computational properties. The *factorized conditional log-likelihood* ($\hat{f}$CLL) is introduced in Carvalho et al. [2011] with the properties of being decomposable and score equivalent for BAN classifiers. Note that the addends in CLL (see Equation (15)) can be expressed, for a binary C (c vs. $\neg c$), as a difference of logarithms:

$$\begin{aligned} \log p_B\big(c^{(i)}|x_1^{(i)}, \ldots, x_n^{(i)}\big)\big) &= \log p\big(c^{(i)}, x_1^{(i)}, \ldots, x_n^{(i)}\big) \\ &\quad - \log\big(p\big(c^{(i)}, x_1^{(i)}, \ldots, x_n^{(i)}\big) + p\big(\neg c^{(i)}, x_1^{(i)}, \ldots, x_n^{(i)}\big)\big), \end{aligned}$$

the second one being the log of a sum of terms, whereby it is nondecomposable. Then these addends are approximated by a linear function of the log of these terms. When substituted in the $\hat{f}$CLL score, this can be rewritten in terms of conditional mutual information and *interaction information* [McGill 1954]. For parameter learning, the authors use MLEs.

Another simpler approximation to CLL is the *approximate conditional likelihood* (ACL) [Burge and Lane 2005], where the sum mentioned earlier is replaced by a single term, that is, by $\log p(\neg c^{(i)}, x_1^{(i)}, \ldots, x_n^{(i)})$, to avoid the nondecomposability drawback. This formulation can be applied even for complex classifiers like Bayesian multinets (see Section 9). This results in a decomposable (although unbounded) score. The (discriminatively learned) parameters maximizing this score (*ACL-Max*) have a closed form. Alternatively, MLEs can be used for parameter learning (*ACL-MLE*).

The EAR measure is the criterion maximized in Narasimhan and Bilmes [2005] using a greedy forward algorithm with an MLE of parameters.

The idea of Drugan and Wiering [2010] is to use both the Bayesian network classifier that factorizes the joint distribution $p(c, \mathbf{x})$ and an auxiliary Bayesian network that factorizes $p(\mathbf{x})$. Since the quotient between these two distributions is $p(c|\mathbf{x})$, the conditional log-likelihood, CLL, of the classifier is then approximated by the difference between the unconditional log-likelihood of the classifier and the log-likelihood of the auxiliary network; see the first three sums in Equation (15). Both structures are learned using a generative method. A new score, called *minimum description length for feature selection* (MDL-FS), is introduced to guide the search for good structures, also allowing feature selection. MDL-FS, like MDL, penalizes the complexity of the classifier and, rather than including the log-likelihood, it includes the so-called *conditional auxiliary log-likelihood*, the difference between the log-likelihood of the data given the Bayesian network classifier and that given the auxiliary Bayesian network over $\mathbf{X}$. In practical applications, they propose to set a specific family of auxiliary networks beforehand. Depending on their complexity, the MDL-FS can serve to identify and remove redundant variables at various levels. Thus, with trees as auxiliary networks, learning a selective TAN classifier starts with all predictor variables in both types of structures. The corresponding MDL-FS is computed and guides the next variable to be deleted following a backward elimination strategy. New structures are learned from the new set of variables. MLE is used for parameter learning.

A score that takes into account the posterior distribution of the class variable during the structure learning process should in principle lead to models with higher classification capabilities. The score introduced in Sierra et al. [2009] (*Hist-dist*) uses, for each case, the distance between the predicted posterior distribution of the class and an approximation of the real (degenerated) posterior distribution. This is defined by giving an α value (close to 1) to the real class of the case and dividing the remaining $1 - \alpha$ evenly across the other class values. The final score to be minimized is the mean of those distances for all cases. Different distance measures are proposed (Euclidean, Kolmogorov-Smirnov, chi-square, etc.). The wrapper approach is based on the greedy *Algorithm B* [Buntine 1991], which searches for the best unrestricted Bayesian classifier.

(b) Discriminative learning of parameters. Logistic regression can be seen as discriminatively trained naive Bayes classifiers [Agresti 1990]. See also Ng and Jordan [2001] for an empirical and theoretical comparison of both models, where for small sample sizes the generative naive Bayes can outperform the discriminatively trained naive Bayes. In general, discriminatively trained classifiers are usually more accurate when N is high.

For a fixed Bayesian network structure, finding the values θ_{ijk} for the conditional probability tables that maximize the CLL is NP hard for a given incomplete dataset [Greiner et al. 2005], something more readily solved in generative models maximizing the likelihood, which have straightforward EM methods for handling missing data.

Given complete data, the complexity of maximizing the CLL for arbitrary structures is unknown. However, the CLL does not have local maxima for structures satisfying a certain graph-theoretic property, and the global maximum can be found by mapping the corresponding optimization problem to an equivalent logistic regression model [Roos et al. 2005]. This model has fewer parameters than its Bayesian network classifier counterpart and is known to have a strictly concave log-likelihood function. The graph-theoretic property is that the structure of the Bayesian network is such that its canonical version is perfect; that is, all nodes having a common child are connected. The canonical version is constructed by first restricting the original structure to C's Markov blanket and then adding as many arcs as needed to make the parents of C

fully connected. All Bayesian networks with the same canonical version are equivalent in terms of $p(c|x_1, \ldots, x_n)$. Naive Bayes and TAN models comply with this property. The conditional distributions $p(c|x_1, \ldots, x_n)$ in the CLL expression are reparameterized using a logistic regression model where the covariates are derived from the original variables. There are two types of covariates: (a) indicator variables for each configuration $\mathbf{pa}(c)$ and (b) indicator variables for each configuration $(x_i, \mathbf{pa}^{\setminus C}(x_i))$, where X_i denotes any children of C, and $\mathbf{Pa}^{\setminus C}(X_i) = \mathbf{Pa}(X_i)\setminus\{C\}$. The original parameters, θ_{ijk}, are recovered via the exponential function of the logistic regression parameters. We call this approach LR-Roos, an acronym of logistic regression for perfect structures.

A different mapping for perfect graphs to an equivalent logistic regression model with fewer parameters than LR-Roos is proposed in Feelders and Ivanovs [2006]. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) method (the best for simple structures) and conjugate gradient are used to optimize the CLL. We call this approach LR-Feelders.

The aforementioned ELR algorithm [Greiner et al. 2005] is the most popular approximation procedure for maximizing the CLL for a given Bayesian network structure. ELR applies to arbitrary Bayesian network structures and works effectively even with an incomplete dataset. It is often superior to classifiers produced by standard generative algorithms, especially in common situations where the given Bayesian network structure is incorrect; that is, it is not an I-map of the underlying distribution. This occurs when the learning algorithm is conservative about adding new arcs to avoid overfitting the data or because the algorithm only considers a restricted class of structures that is not guaranteed to contain the correct structure. For each conditional probability table entry, ELR is a conjugate gradient-ascent algorithm that tries to maximize CLL with respect to a softmax function of θ_{ijk}, that is, $\theta_{ijk} = \frac{e^{\beta_{ijk}}}{\sum_{k'} e^{\beta_{ijk'}}}$.

A different idea is to take the effect of estimating θ_{ijk} on classification into account by adapting the appropriate frequencies from data. θ_{ijk} is initialized as the MLE in iteration $t = 0$. Going through all the training data, the update at iteration $t+1$ consists of summing, for each instance $\mathbf{x}$, the difference between the true posterior probability $p(c|\mathbf{x})$ (assumed to be 1 when $\mathbf{x}$ has label c in the dataset) and the predicted probability generated by the current parameters $p_t(c|\mathbf{x})$, that is, $\theta_{ijk}^{(t+1)} = \theta_{ijk}^{(t)} + p(c|\mathbf{x}) - p_t(c|\mathbf{x})$. This approach was proposed in Su et al. [2008] and named *discriminative frequency estimate* (DFE). DFE can be seen as a more sophisticated approach than the one proposed in Gama [1999].

Three discriminative parameter learning algorithms are introduced in Pernkopf and Wohlmayr [2009] for naive Bayes, TAN, or 2-DB structures. First, the *exact CLL decomposition* (ECL) algorithm tries to optimize the CLL function. Second, the *approximate CLL decomposition* (ACL) algorithm aims at optimizing a lower-bound surrogate of the CLL function. Third, the *extended Baum-Welch* (EBW) algorithm is used for these three structures. All the algorithms initialize the parameters to the MLEs.

A different criterion is optimized in Guo et al. [2005]. The discriminative objective is to maximize the *minimum conditional likelihood ratio* (MCLR):

$$MCLR(\boldsymbol{\theta}) = \min_{i=1,\ldots,N} \min_{c \neq c^{(i)}} \frac{p(c^{(i)}|\mathbf{x}^{(i)}, \boldsymbol{\theta})}{p(c|\mathbf{x}^{(i)}, \boldsymbol{\theta})}.$$

When Bayesian networks are formulated as a form of exponential model, $\log MCLR(\boldsymbol{\theta})$ resembles a large margin criterion of support vector machines, but subject to normalization constraints over each variable (probabilities summing 1). These restrictions are nonlinear, and this yields a difficult optimization problem. The authors solve the problem with convex relaxation for a wide range of graph topologies.

A conjugate gradient algorithm is instead proposed in Pernkopf et al. [2012] and is advantageous in terms of computational requirements.

(c) Generative-discriminative learning. Some researchers try to take advantage of the best of both approaches through hybrid parameter learning (partly generative and partly discriminative) and generative modeling.

Thus, in the context of text classification, the multinomial naive Bayes model of Raina et al. [2004] divides the set of predictors into R regions. For the sake of clarity, we will focus on $R = 2$, and therefore $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)$. Equation (8) is modified as

$$p(c|\mathbf{x}) \propto p(c)p(\mathbf{x}_1|c)^{\frac{w_1}{n_1}} p(\mathbf{x}_2|c)^{\frac{w_2}{n_2}},$$

where (w_1, w_2) controls the relative weighting between the regions, and n_1, n_2 are their lengths. For instance, in emails consisting of two regions, subject and body, $n_2 \gg n_1$ since bodies are usually much longer than subjects, and the usual naive Bayes equation will be mostly dominated by the message body (with many more factors). This model tries instead to convey that different predictors are of different importance (words in the subject might be more important) and counteracts the independence assumption of naive Bayes with normalization factors n_1, n_2. The expression of $p(c|\mathbf{x})$ is then rewritten in a logistic regression form, where its linear combination contains parameters, generatively learned functions of $p(\mathbf{x}_i|c)$. Parameters w_i are discriminatively learned (by maximizing the CLL), $i = 1, 2$. They call this model the *normalized hybrid* algorithm, designed for a binary class. A multiclass extension is reported in Fujino et al. [2007].

The *joint discriminative-generative* (JoDiG) approach of Xue and Titterington [2010] partitions $\mathbf{X}$ into two subvectors: $\mathbf{X} = (\mathbf{X}_D, \mathbf{X}_G)$. A generative approach is applied to $\mathbf{X}_G$ to estimate $p(\mathbf{x}_G|c)$ and a discriminative approach is applied to $\mathbf{X}_D$ to estimate $p(c|\mathbf{x}_D)$. A data-generating process is always assumed in generative but never in discriminative approaches. In general, when this process is well specified, the generative approach performs better than the discriminative approach. This is the idea for finding the partition of $\mathbf{X}$: $\mathbf{X}_D$ will contain the variables that violate the assumption underlying the data-generating process (as given by a statistical test). Finally, since $\mathbf{X}_G$ and $\mathbf{X}_D$ are assumed to be (block-wise) conditionally independent given C, then $p(\mathbf{x}_D, \mathbf{x}_G, c) = p(\mathbf{x}_D)p(c|\mathbf{x}_D)p(\mathbf{x}_G|c)$, and both approaches are probabilistically combined to classify a new instance via the MAP criterion

$$\arg\max_c p(c|\mathbf{x}_D)p(\mathbf{x}_G|c).$$

The *hybrid generative/discriminative Bayesian* (HBayes) classifier [Kang and Tian 2006] uses a similar idea. The difference lies in how the partition is chosen, for which purpose a wrapper strategy is adopted in this case: starting from $\mathbf{X}_G = \mathbf{X}$, the variable producing the greatest improvement in classification performance is greedily moved from $\mathbf{X}_G$ to $\mathbf{X}_D$. Ridge logistic regression is used to estimate $p(c|\mathbf{x}_D)$, whereas naive Bayes or TAN is used to estimate $p(\mathbf{x}_G|c)$. The Bayesian network structure is thereby restricted (Figure 13) to reduce the computational effort.

A Bayesian approach for the combination of generative and discriminative learning of classifiers is found in Bishop and Lasserre [2007]. This is intended to find the appropriate tradeoff between generative and discriminative extremes. Generative and discriminative models correspond to specific choices for the priors over parameters. Since generative approaches can model unlabelled instances while discriminative approaches do not, this *Bayesian blending* can also be applied to semisupervised classification.

9. BAYESIAN MULTINETS

Bayesian networks are unable to encode *asymmetric* independence assertions in their topology. This refers to conditional independence relationships only held for some but

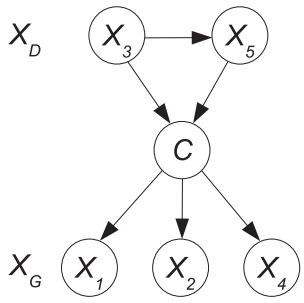


Fig. 13. A HBayes classifier structure from which $p(c|\mathbf{x}) \propto p(c|\mathbf{x}_D)p(\mathbf{x}_G|c)$.

not all the values of the variables involved. *Bayesian multinets* [Geiger and Heckerman 1996] offer a solution. They consist of several (local) Bayesian networks associated with a subset of a partition of the domain of a variable H, called the hypothesis or distinguished variable; that is, each local network represents a joint probability of all (but H) variables conditioned on a subset of H values. As a result of this conditioning, asymmetric independence assertions are represented in each local network topology. Consequently, structures are expected to be simpler, with computational and memory requirement savings. Whereas the typical setting is when H is a root node, other situations are addressed in Geiger and Heckerman [1996]: H is a nonroot node, and there is more than one variable representing hypotheses.

For classification problems, the distinguished variable is naturally the class variable C. All subsets of the C domain partition are commonly singletons. Thus, conditioned on each c, the predictors can form different local networks with different structures. Therefore, the relations among variables do not have to be the same for all c. Equation (1) is, for Bayesian multinets, given by

$$p(c|\mathbf{x}) \propto p(c) \prod_{i=1}^{n} p(x_i|\mathbf{pa}_c(x_i)),$$

where $\mathbf{Pa}_c(X_i)$ is the parent set of X_i in the local Bayesian network associated with $C = c$; see Figure 1. Therefore, a Bayesian multinet is defined via its local Bayesian networks and the prior distribution on C.

Particular cases of multinets were explained in Section 6.1: networks reported in Chow and Liu [1968] and Pham et al. [2002] with trees and forests, respectively, as local Bayesian networks (illustrated in Figure 7(a) and (d)). Trees are also used in Kłopotek [2005], although the learning is based on a new algorithm designed for very large datasets rather than Kruskal's algorithm. The trees in Huang et al. [2003] are learned by optimizing a function that includes a penalty term representing the divergence between the different joint distributions defined at each local network. Finally, the trees in Gurwicz and Lerner [2006] are learned from all instances, instead of learning the local structures from only those instances with $C = c$. The process is guided by a score that simultaneously detects class patterns and rejects patterns of the other classes. Thus, for the local network for $C = c$, the score of $\mathbf{x}$ with true class value c is higher when $p(C = c|\mathbf{x}) \geq p(C = c'|\mathbf{x}), \forall c' \neq c$ and the score of $\mathbf{x}$ with true class value $c' \neq c$ is higher when $p(C = c'|\mathbf{x}) \geq p(C = c|\mathbf{x})$. The search is based on the hill-climbing algorithm described in Keogh and Pazzani [2002] (see Section 6.1).

The local structures are general unrestricted Bayesian networks in Friedman et al. [1997] and Hussein and Santos [2004]. However, the approach taken in Hussein and Santos [2004] is different. The data are not partitioned according to $C = c$. The training

Table III. Mean Accuracies (%) ± Standard Deviations of the 12 Bayesian Network Classifiers
"#" means the number of variables included in the model.

	All variables	#	Filter	#	Wrapper	#
Naive Bayes	71.64 ± 9.78	9	71.98 ± 11.59	5	77.20 ± 8.01	3
Tree-augmented naive Bayes	77.57 ± 8.08	9	76.50 ± 9.10	5	77.55 ± 9.35	5
Bayesian network-augmented naive Bayes	74.78 ± 8.62	9	76.83 ± 10.54	5	77.22 ± 10.14	6
Markov blanket-based Bayesian classifiers	75.16 ± 7.62	9	73.74 ± 7.67	5	76.52 ± 9.00	6

data are first partitioned into clusters from which a set of rules characterizing their cases are derived. Then a local Bayesian network is learned from the cases satisfying the rules. This is why the resulting models are called *case-based Bayesian network classifiers*, capturing case-dependent relationships, a generalization of hypothesis-specific relationships.

10. ILLUSTRATIVE EXAMPLE

This section reports the classification accuracy results of 12 different Bayesian network classifiers, according to four increasing model complexities (naive Bayes, tree-augmented naive Bayes, Bayesian network-augmented naive Bayes, and Markov blanket-based Bayesian classifiers) including all predictor variables and using two feature subset selection methods (a filter and a wrapper approach). The filter approach is univariate and based on information gain, whereas the wrapper search uses a greedy forward strategy in all models but the Markov blanked-based classifier, which employs a genetic algorithm.

The classifiers were learned from the Ljubljana breast cancer dataset [Michalski et al. 1986] with 286 labeled instances of real patients. The classification problem was to predict breast cancer recurrence (yes or no) in the 5 years after surgery. Recurrence was observed in 85 out of the 286 patients. The nine predictor variables, measured at diagnosis, are:

—`age`: patient age in years, discretized into three equal-width intervals
—`menopause`: non-, pre-, or postmenopausal patient
—`deg-malig`: degree of tumor malignancy (histological grade scored 1–3)
—`node-caps`: whether or not the tumor has perforated through the lymph node capsule
—`inv-nodes`: the number (range 0–26) of involved axillary lymph nodes that contain metastatic breast cancer visible on histological examination, discretized into three intervals
—`irradiation`: whether or not the patient has been irradiated
—`breast`: left- or right-sided breast cancer
—`breast-quad`: location of the tumor according to the four breast quadrants (upper-outer, lower-outer, upper-inner, and lower-inner) plus the nipple as a central point
—`size`: maximum excised tumor diameter (in mm), discretized into three equal-width intervals

Table III shows the classification accuracy (%) and standard deviations of all model combinations. They have been estimated with 10-fold stratified cross-validation using WEKA [Hall et al. 2009] software.

Naive Bayes and the filter-based selective naive Bayes (Figure14(a)) are the worst-performing algorithms ($\approx$71% accuracy). However, the accuracy of selective naive Bayes increases considerably (up to 77%) using a wrapper-wise-guided search, with only three predictor variables. WEKA was parameterized to run similar algorithms to those proposed in the literature and reviewed within this article: Maron and Kuhns [1960] for naive Bayes, Pazzani and Billsus [1997] for filter-based selective naive Bayes, and Langley and Sage [1994] for wrapper-based selective naive Bayes.

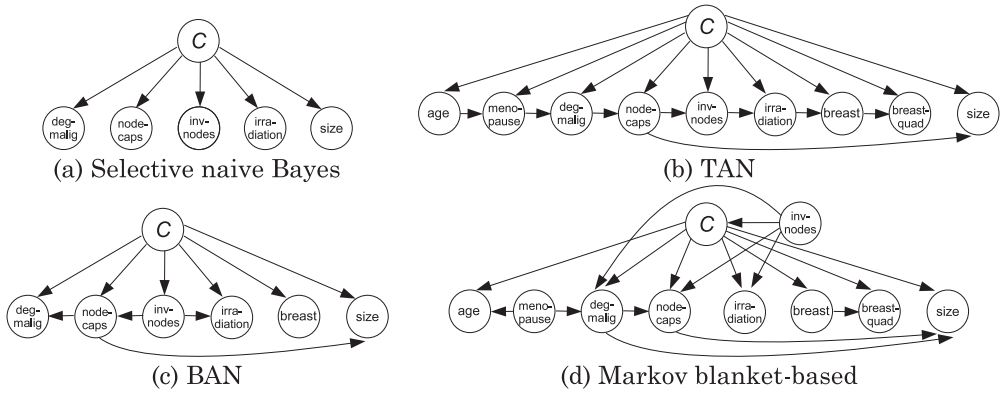


Fig. 14. Structures of (a) selective naive Bayes output using a filter approach, (b) TAN, (c) wrapper BAN, and (d) Markov blanket-based Bayesian classifier.

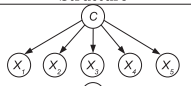
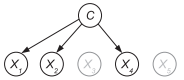
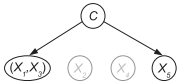
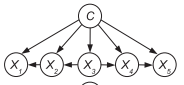
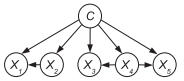
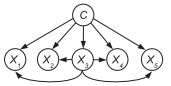
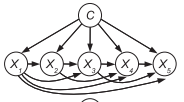
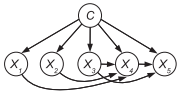
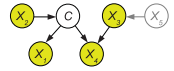
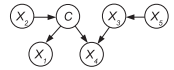
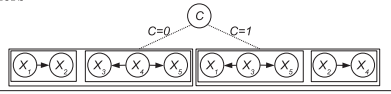
TAN and its selective versions (filter and wrapper) are the best-performing models on average. The TAN spanning tree (Figure 14(b)) is rooted at node `age`. It captures expected relationships, as specified by the arcs `age`→`menopause`, `deg-malig`→`node-caps`, and `node-caps`→`size`. Age and menopause are obviously related. There is a greater likelihood of the tumor penetrating through the lymph node capsule and invading the surrounding tissues at worse tumor grades. Tumor grade also conditions tumor size. The WEKA algorithms for these TAN models were similar to the learning algorithms described in Friedman et al. [1997] for TAN and in Blanco et al. [2005] for both selective TAN models.

BAN models (Table III, row 3) were learned by setting the maximum number of parents to 3. Selective BAN models behave similarly to their TAN counterparts. Without feature selection, BAN accuracy decreases. The best BAN, which is in fact a FAN (Figure14(c)), is the wrapper version. This model did not select `age`, `menopause`, and `breast-quad`. Its structure shares two arcs with the TAN classifier (Figure 14(b)), `node-caps`→`size` and `inv-nodes`→`irradiation`. TAN also identified arcs `inv-nodes`→`node-caps` and `node-caps`→`deg-malig`, albeit reversed. The most similar algorithms to those run in WEKA are Friedman et al. [1997] for BAN, Ezawa and Norton [1996] for the filter-based BAN, and Pernkopf and O'Leary [2003] for the wrapper-based BAN.

Finally, despite the flexibility of the Markov blanket-based classifier structures, they do not exhibit very high accuracies. Without variable selection (Figure14(d)), C has only one parent, `inv-nodes`. This model has many relationships in common with TAN (Figure14(b)). However, three nodes (`deg-malig`, `node-caps`, and `size`) have three parents, requiring bigger conditional probability tables. Also, there is a new arc, `deg-malig`→`size` (justified by following the aforementioned reasoning), and a missing arc, C→`menopause`. The algorithm reported in Madden [2002] is close to the WEKA implementations of Markov blanket-based classifiers (all variables and filter), whereas we used WEKA's genetic algorithm-guided search for the wrapper version as reported in Sierra and Larrañaga [1998].

In summary, the wrapper versions are the models that work best here. All of them include at least the `inv-nodes`, `deg-malig`, and `breast` variables. Filter approaches seem to improve the all-variables strategy. With only nine variables, carefully chosen by physicians to be relevant for the problem, the advantages of feature selection are limited. The best model is the wrapper-based TAN. Thus, increasing model complexity does not necessarily imply a better model. This is why it is always worthwhile to explore the whole hierarchy of Bayesian classifiers.

Table IV. Summary of Bayesian Network Classifiers and Their Most Relevant References

Name	Structure	Seminal paper	Feature subset selection: Filter	Feature subset selection: Wrapper	Metaclassifiers
Naive Bayes		[Maron and Kuhns 1960]	NA	NA	[Langley 1993]
Selective naive Bayes		[Langley and Sage 1994]	[Pazzani and Billsus 1997]	[Langley and Sage 1994]	[Zheng 1998]
Semi-naive Bayes		[Pazzani 1996]	[Blanco et al. 2005]	[Robles et al. 2003]	[Robles et al. 2004]
Tree-augmented naive Bayes		[Friedman et al. 1997]	[Blanco et al. 2005]	[Keogh and Pazzani 2002]	[Ma and Shi 2004]
Forest-augmented naive Bayes		[Lucas 2004]	[Ziebart et al. 2007]		
Superparent-one-dependence estimator		[Keogh and Pazzani 2002]	NA	NA	[Webb et al. 2005]
k-dependence Bayesian classifier		[Sahami 1996]	[Blanco et al. 2005]	[Blanco et al. 2005]	[Louzada and Ara 2012]
Bayesian network-augmented naive Bayes		[Friedman et al. 1997]	[Ezawa and Norton 1996]	[Pernkopf and O'Leary 2003]	
Markov blanket-based Bayesian classifiers		[Koller and Sahami 1996]	[Koller and Sahami 1996]	[Sierra and Larrañaga 1998]	
Unrestricted Bayesian classifiers		[Provan and Singh 1995]	[Singh and Provan 1996]	[Provan and Singh 1995]	[Dash and Cooper 2004]
Bayesian multinet		[Geiger and Heckerman 1996]			

11. DISCUSSION

This survey has shown the power of Bayesian network classifiers in terms of model expressiveness and algorithm efficiency/effectiveness for learning models from data and for use in classification. Unlike other pattern recognition classifiers, Bayesian network classifiers can be clearly organized hierarchically from the simplest naive Bayes to the most complex Bayesian multinet.

The Bayesian network classifiers are hierarchized in the rows of Table IV, whereas the columns give an example of their graphical structure, the associated seminal paper, and the first references proposing filter/wrapper approaches for feature subset selection and metaclassifiers.

We did not set out to survey the behavior of these classifiers in big real-world problems. As the no-free-lunch theorem states, this depends on the dataset. However, some relevant papers, already cited within this survey [Friedman et al. 1997; Cheng and Greiner 1999, 2001; Pernkopf 2005; Madden 2009], do include empirical comparisons of the algorithms for learning naive Bayes, TAN, BAN, unrestricted Bayesian classifiers, and Bayesian multinets. They all use datasets from the UCI repository [Bache and Lichman 2013]. Also, both discriminative and generative parameter learning on both discriminatively and generatively structured models are compared in Pernkopf and Bilmes [2005]. The general findings are that more complex structures perform better whenever the sample size is big enough to guarantee reliable probability estimates. Also, smoothing parameter estimation can significantly improve the classification rate. Discriminative parameter learning produces on average a better classifier than maximum likelihood parameter learning. In most datasets, structures learned with wrapper approaches yield the most accurate classifiers.

Since the focus of this article is on Bayesian network classifiers based on Bayesian networks, other models—models with cycles, like dependency networks, and undirected models, like Markov networks—are beyond its scope. We have not considered data-streaming situations or specific problems like multilabel or semisupervised classification or classification with probabilistic labels either. Although the survey has focused on discrete data, research on continuous and mixed data is on-going.

Research on discrete Bayesian network classifiers may in the future target more theoretical studies on determining the decision boundary for classifier types apart from the naive Bayes reviewed here. Also, the gaps in Table IV suggest that there is still room for research on metaclassifiers and feature subset selection. Metaclassifiers might also be formed by hybridizing Bayesian classifiers with different types of classifiers other than the decision trees and k-nearest neighbors mentioned in this article. Finally, we have seen how naive Bayes can tackle complex classification situations (e.g., with homologous sets, multiple instances, cost-sensitive learning, instance ranking, and imprecise probabilities). We expect to see other models dealing with these and more challenging settings soon.

REFERENCES

J. Abellán. 2006. Application of uncertainty measures on credal sets on the naive Bayes classifier. *International Journal of General Systems* 35 (2006), 675–686.

J. Abellán, A. Cano, A. R. Masegosa, and S. Moral. 2007. A semi-naive Bayes classifier with grouping of cases. In *Proceedings of the 9th European Conference in Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU-2007). Lecture Notes in Artificial Intelligence*, Vol. 4724. Springer, 477–488.

S. Acid, L. M. de Campos, and J. G. Castellano. 2005. Learning Bayesian network classifiers: Searching in a space of partially directed acyclic graphs. *Machine Learning* 59, 3 (2005), 213–235.

A. Agresti. 1990. *Categorical Data Analysis*. Wiley.

K. M. Al-Aidaroos, A. A. Bakar, and Z. Othman. 2010. Naive Bayes variants in classification learning. In *Proceedings of the International Conference on Information Retrieval Knowledge Management (CAMP-2010)*. 276–281.

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

C. F. Aliferis, I. Tsamardinos, and M. S. Statnikov. 2003. HITON: A novel Markov blanket algorithm for optimal variable selection. In *AMIA Annual Symposium Proceedings*. 21–25.

K. Bache and M. Lichman. 2013. UCI Machine Learning Repository. (2013). Retrieved from http://archive.ics.uci.edu/ml.

X. Bai, R. Padman, J. Ramsey, and P. Spirtes. 2008. Tabu search-enhanced graphical models for classification in high dimensions. *INFORMS Journal on Computing* 20, 3 (2008), 423–437.

J. Bilmes. 2000. Dynamic Bayesian multinets. In *Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence (UAI-2000)*. Morgan Kaufmann, 38–45.

C. Bishop. 1995. *Neural Networks for Pattern Recognition*. Oxford University Press.

C. M. Bishop and J. Lasserre. 2007. Generative or discriminative? Getting the best of both worlds. In *Bayesian Statistics*, Vol. 8. Oxford University Press, 3–23.

R. Blanco, I. Inza, M. Merino, J. Quiroga, and P. Larrañaga. 2005. Feature selection in Bayesian classifiers for the prognosis of survival of cirrhotic patients treated with TIPS. *Journal of Biomedical Informatics* 38, 5 (2005), 376–388.

W. L. Buntine. 1991. Theory refinement on Bayesian networks. In *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence (UAI-1991)*. Morgan Kaufmann, 52–60.

J. Burge and T. Lane. 2005. Learning class-discriminative dynamic Bayesian networks. In *Proceedings of the 22nd International Conference on Machine Learning (ICML-2005)*. ACM, 97–104.

A. Cano, J. G. Castellano, A. R. Masegosa, and S. Moral. 2005. Methods to determine the branching attribute in Bayesian multinets classifiers. In *Proceedings of the 8th European Conference in Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU-2005). Lecture Notes in Artificial Intelligence*, Vol. 3571. Springer, 932–943.

A. M. Carvalho, A. L. Oliveira, and M.-F. Sagot. 2007. Efficient learning of Bayesian network classifiers. In *Proceedings of the 20th Australian Joint Conference on Artificial Intelligence (AI-2007). Lecture Notes in Computer Science*, Vol. 4830. Springer, 16–25.

A. M. Carvalho, T. Roos, A. L. Oliveira, and P. Myllymäki. 2011. Discriminative learning of Bayesian networks via factorized conditional log-likelihood. *Journal of Machine Learning Research* 12 (2011), 2181–2210.

J. Cerquides and R. López de Mántaras. 2005a. Robust Bayesian linear classifier ensembles. In *Proceedings of the 16th European Conference on Machine Learning (ECML-2005). Lecture Notes in Computer Science*, Vol. 3720. Springer, 72–83.

J. Cerquides and R. López de Mántaras. 2005b. TAN classifiers based on decomposable distributions. *Machine Learning* 59, 3 (2005), 323–354.

B. Cestnik. 1990. Estimating probabilities: A crucial task in machine learning. In *Proceedings of the European Conference in Artificial Intelligence*. 147–149.

X. Chai, L. Deng, Q. Yang, and C. X. Ling. 2004. Test-cost sensitive naive Bayes classification. In *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM-2004)*. IEEE Computer Society, 51–58.

J. Cheng and R. Greiner. 1999. Comparing Bayesian network classifiers. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-1999)*. Morgan Kaufmann Publishers, 101–108.

J. Cheng and R. Greiner. 2001. Learning Bayesian belief networks classifiers: Algorithms and system. In *Proceedings of the 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence (CSCSI-2001)*, Vol. 2056. Springer, 141–151.

D. M. Chickering. 1995. A transformational characterization of equivalent Bayesian network structures. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI-1995)*. Morgan Kaufmann, 87–98.

D. M. Chickering, D. Heckerman, and C. Meek. 2004. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research* 5 (2004), 1287–1330.

C. Chow and C. Liu. 1968. Approximating discrete probability distributions with dependency trees. *IEEE Transactions on Information Theory* 14 (1968), 462–467.

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

D. Dash and G. F. Cooper. 2004. Model averaging for prediction with discrete Bayesian networks. *Journal of Machine Learning Research* 5 (2004), 1177–1203.

D. Dash and G. F. Cooper. 2002. Exact model averaging with naïve Bayesian classifiers. In *Proceedings of the 19th International Conference on Machine Learning (ICML-2002)*. 91–98.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B* 39, 1 (1977), 1–38.

P. Domingos and M. Pazzani. 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* 29 (1997), 103–130.

E. B. dos Santos, E. R. Hruschka Jr., E. R. Hruschka, and N. F. F. Ebecken. 2011. Bayesian network classifiers: Beyond classification accuracy. *Intelligent Data Analysis* 15, 3 (2011), 279–298.

M. M. Drugan and M. A. Wiering. 2010. Feature selection for Bayesian network classifiers using the MDL-FS score. *International Journal of Approximate Reasoning* 51 (2010), 695–717.

R. Duda, P. Hart, and D. G. Stork. 2001. *Pattern Classification*. John Wiley and Sons.

D. Edwards and S. L. Lauritzen. 2001. The TM algorithm for maximising a conditional likelihood function. *Biometrika* 88 (2001), 961–972.

M. Ekdahl and T. Koski. 2006. Bounds for the loss in probability of correct classification under model based approximation. *Journal of Machine Learning Research* 7 (2006), 2449–2480.

S. Eyheramendy, D. D. Lewis, and D. Madigan. 2002. On the naive Bayes model for text categorization. In *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics (AISTATS-2002)*.

K. J. Ezawa and S. W. Norton. 1996. Constructing Bayesian networks to predict uncollectible telecommunications accounts. *IEEE Expert* 11, 5 (1996), 45–51.

A. J. Feelders and J. Ivanovs. 2006. Discriminative scoring of Bayesian network classifiers: A comparative study. In *Proceedings of the 3rd European Workshop on Probabilistic Graphical Models (PGM-2006)*. 75–82.

Q. Feng, F. Tian, and H. Huang. 2007. A discriminative learning method of TAN classifier. In *Proceedings of the 9th European Conference in Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU-2007). Lecture Notes in Artificial Intelligence*, Vol. 4724. Springer, 443–452.

J. Flores, J. A. Gámez, and A. M. Martínez. 2012. Supervised classification with Bayesian networks: A review on models and applications. In *Intelligent Data Analysis for Real World Applications. Theory and Practice*. IGI Global, 72–102.

M. J. Flores, J. A. Gámez, A. M. Martínez, and J. M. Puerta. 2009. HODE: Hidden one-dependence estimator. In *Proceedings of the 10th European Conference in Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU-2009). Lecture Notes in Artificial Intelligence*, Vol. 5590. Springer, 481–492.

O. François and P. Leray. 2006. Learning the tree augmented naive Bayes classifier from incomplete datasets. In *Proceedings of the 3rd European Workshop on Probabilistic Graphical Models (PGM-2006)*. 91–98.

E. Frank, M. Hall, and B. Pfahringer. 2003. Locally weighted naive Bayes. In *Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence (UAI-2003)*. Morgan Kaufmann, 249–256.

M. L. Fredman and R. E. Tarjan. 1987. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal ACM* 34, 3 (1987), 596–615.

N. Friedman. 1997. Learning belief networks in the presence of missing values and hidden variables. In *Proceedings of the 14th International Conference on Machine Learning (ICML-1997)*. Morgan Kaufmann, 125–133.

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

N. Friedman, M. Goldszmidt, and A. Wyner. 1999. Data analysis with Bayesian networks: A bootstrap approach. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-1999)*. Morgan Kaufmann, 196–205.

S. Fu and M. Desmarais. 2007. Local learning algorithm for Markov blanket discovery. In *Proceedings of the 20th Australian Joint Conference on Artificial Intelligence (AI-2007). Lecture Notes in Computer Science*, Vol. 4830. Springer, 68–79.

A. Fujino, N. Ueda, and K. Saito. 2007. A hybrid generative/discriminative approach to text classification with additional information. *Information Processing and Management* 43, 2 (2007), 379–392.

J. Gama. 1999. Iterative naïve Bayes. *Theoretical Computer Science* 292, 2 (1999), 417–430.

D. Geiger and D. Heckerman. 1996. Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence* 82 (1996), 45–74.

M. Goldszmidt. 2010. Bayesian network classifiers. In *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, 1–10.

I. J. Good. 1965. *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*. The MIT Press.

R. Greiner, X. Su, B. Shen, and W. Zhou. 2005. Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. *Machine Learning* 59, 3 (2005), 297–322.

R. Greiner and W. Zhou. 2002. Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-2002)*. AAAI Press/MIT Press, 167–173.

D. Grossman and P. Domingos. 2004. Learning Bayesian network classifiers by maximizing conditional likelihood. In *Proceedings of the 21st International Conference on Machine Learning (ICML-2004)*. 361–368.

Y. Guo and R. Greiner. 2005. Discriminative model selection for belief net structures. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-2005)*. AAAI Press / The MIT Press, 770–776.

Y. Guo, D. F. Wilkinson, and D. Schuurmans. 2005. Maximum margin Bayesian networks. In *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence (UAI-2005)*. AUAI Press, 233–242.

Y. Gurwicz and B. Lerner. 2006. Bayesian class-matched multinet classifier. In *Proceedings of the 2006 Joint IAPR international Conference on Structural, Syntactic, and Statistical Pattern Recognition (SSPR-2006/SPR-2006). Lecture Notes in Computer Science*, Vol. 4109. Springer, 145–153.

M. A. Hall. 1999. *Correlation-Based Feature Selection for Machine Learning*. Ph.D. Dissertation. Department of Computer Science, University of Waikato.

M. Hall. 2007. A decision tree-based attribute weighting filter for naive Bayes. *Knowledge-Based Systems* 20, 2 (2007), 120–126.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Explorations* 11, 1 (2009), 10–18.

D. J. Hand and K. Yu. 2001. Idiot's Bayes - not so stupid after all? *International Statistical Review* 69, 3 (2001), 385–398.

D. Heckerman, D. Geiger, and D. Chickering. 1995. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning* 20 (1995), 197–243.

J. Hilden and B. Bjerregaard. 1976. Computer-aided diagnosis and the atypical case. In *Decision Making and Medical Care. Can Information Science Help?* 365–378.

E. R. Hruschka and N. F. F. Ebecken. 2007. Towards efficient variables ordering for Bayesian network classifiers. *Data and Knowledge Engineering* 63 (2007), 258–269.

H. Huang and C. Hsu. 2002. Bayesian classification for data from the same unknown class. *IEEE Transactions on Systems, Man, and Cybernetics Part B* 32, 2 (2002), 137–145.

K. Huang, I. King, and M. R. Lyu. 2003. Discriminative training of Bayesian Chow-Liu multinet classifiers. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-2003)*, Vol. 1. 484–488.

A. Hussein and E. Santos. 2004. Exploring case-based Bayesian networks and Bayesian multi-nets for classification. In *Proceedings of the 17th Conference of the Canadian Society for Computational Studies of Intelligence (CSCSI-2004). Lecture Notes in Computer Science*, Vol. 3060. Springer, 485–492.

K.-B. Hwang and B. T. Zhang. 2005. Bayesian model averaging of Bayesian network classifiers over multiple node-orders: Application to sparse datasets. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics* 35, 6 (2005), 1302–1310.

A. Ibáñez, P. Larrañaga, and C. Bielza. 2014. Cost-sensitive selective naive Bayes classifiers for predicting the increase of the h-index for scientific journals. *Neurocomputing* in press (2014).

I. Inza, P. Larrañaga, R. Blanco, and A. J. Cerrolaza. 2004. Filter versus wrapper gene selection approaches in DNA microarray domains. *Artificial Intelligence in Medicine* 31, 2 (2004), 91–103.

I. Inza, P. Larrañaga, R. Etxeberria, and B. Sierra. 2000. Feature subset selection by Bayesian network-based optimization. *Artificial Intelligence* 123, 1–2 (2000), 157–184.

A. G. Ivakhnenko. 1970. Heuristic self-organization in problems of engineering cybernetics. *Automatica* 6, 2 (1970), 207–219.

N. Japkowicz and S. Mohak. 2011. *Evaluating Learning Algorithms. A Classification Perspective*. Cambridge University Press.

T. Jebara. 2004. *Machine Learning: Discriminative and Generative*. Kluwer Academic Publishers.

L. Jiang, Z. Cai, D. Wang, and H. Zhang. 2012. Improving tree augmented Naive Bayes for class probability estimation. *Knowledge-Based Systems* 26 (2012), 239–245.

L. Jiang and H. Zhang. 2006. Lazy averaged one-dependence estimators. In *Proceedings of the 19th Canadian Conference on AI (Canadian AI-2006). Lecture Notes in Computer Science*, Vol. 4013. Springer, 515–525.

L. Jiang, H. Zhang, and Z. Cai. 2009. A novel Bayes model: Hidden naive Bayes. *IEEE Transactions on Knowledge and Data Engineering* 21, 10 (2009), 1361–1371.

L. Jiang, H. Zhang, Z. Cai, and D. Wang. 2012. Weighted average of one-dependence estimators. *Journal of Experimental and Theoretical Artificial Intelligence* 24, 2 (2012), 219–230.

Y. Jing, V. Pavlovic, and J. M. Rehg. 2008. Boosted Bayesian network classifiers. *Machine Learning* 73 (2008), 155–184.

C. Kang and J. Tian. 2006. A Hybrid generative/discriminative Bayesian classifier. In *Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference (FLAIRS-2006)*. AAAI Press, 562–567.

E. J. Keogh and M. J. Pazzani. 2002. Learning the structure of augmented Bayesian classifiers. *International Journal on Artificial Intelligence Tools* 11, 4 (2002), 587–601.

M. A. Kłopotek. 2005. Very large Bayesian multinets for text classification. *Future Generation Computer Systems* 21, 7 (2005), 1068–1082.

R. Kohavi. 1996. Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-1996)*. 202–207.

R. Kohavi, B. Becker, and D. Sommerfield. 1997. *Improving Simple Bayes*. Technical Report. Data Mining and Visualization Group, Silicon Graphics.

R. Kohavi and G. H. John. 1997. Wrappers for feature subset selection. *Artificial Intelligence* 97, 1 (1997), 273–324.

D. Koller and M. Sahami. 1996. Toward optimal feature selection. In *Proceedings of the 13th International Conference on Machine Learning (ICML-1996)*. 284–292.

I. Kononenko. 1993. Successive naive Bayesian classifier. *Informatica (Slovenia)* 17, 2 (1993), 167–174.

P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. 1998. BAYDA: Software for Bayesian classification and feature selection. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-1998)*. AAAI Press, 254–258.

P. Kontkanen, P. Myllymäki, and H. Tirri. 1996. *Constructing Bayesian Finite Mixture Models by the EM Algorithm*. Technical Report C-1996-9. Department of Computer Science, University of Helsinki.

J. B. Kruskal. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society* 7 (1956), 48–50.

C. K. Kwoh and D. Gillies. 1996. Using hidden nodes in Bayesian networks. *Artificial Intelligence* 88 (1996), 1–38.

P. Langley. 1993. Induction of recursive Bayesian classifiers. In *Proceedings of the 8th European Conference on Machine Learning (ECML-1993)*. 153–164.

P. Langley and S. Sage. 1994. Induction of selective Bayesian classifiers. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence (UAI-1994)*. Morgan Kaufmann, 399–406.

H. Langseth and T. D. Nielsen. 2006. Classification using hierarchical naïve Bayes models. *Machine Learning* 63, 2 (2006), 135–159.

J. Li, C. Zhang, T. Wang, and Y. Zhang. 2007. Generalized additive Bayesian network classifiers. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*. 913–918.

J. N. K. Liu, N. L. Li, and T. S. Dillon. 2001. An improved naïve Bayes classifier technique coupled with a novel input solution method. *IEEE Transactions on Systems, Man, and Cybernetics* 31 (2001), 249–256.

D. J. Lizotte, O. Madani, and R. Greiner. 2003. Budgeted learning of naive-Bayes classifiers. In *Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence (UAI-2003)*. Morgan Kaufmann, 378–385.

F. Louzada and A. Ara. 2012. Bagging k-dependence probabilistic networks: An alternative powerful fraud detection tool. *Expert Systems with Applications* 39, 14 (2012), 11583–11592.

P. Lucas. 2004. Restricted Bayesian network structure learning. In *Advances in Bayesian Networks*. Springer, 217–232.

S.-C. Ma and H.-B. Shi. 2004. Tree-augmented naive Bayes ensembles. In *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics*. IEEE, 1497–1502.

M. G. Madden. 2009. On the classification performance of TAN and general Bayesian networks. *Knowledge-Based Systems* 22, 7 (2009), 489–495.

M. G. Madden. 2002. A new Bayesian network structure for classification tasks. In *Proceedings of the 13th Irish Conference on Artificial Intelligence and Cognitive Science*. 203–208.

D. Margaritis and S. Thrun. 2000. Bayesian network induction via local neighborhoods. In *Advances in Neural Information Processing Systems 12 (NIPS-1999)*. MIT Press, 505–511.

M. Maron and J. Kuhns. 1960. On relevance, probabilistic indexing, and information retrieval. *Journal of the Association for Computing Machinery* 7 (1960), 216–244.

W. J. McGill. 1954. Multivariate information transmission. *Psychometrika* 19 (1954), 97–116.

R. S. Michalski, I. Mozetic, J. Hong, and N Lavrac. 1986. The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. In *Proceedings of the 5th National Conference on Artificial Intelligence*. Morgan Kaufman, 1041–1045.

M. Minsky. 1961. Steps toward artificial intelligence. *Transactions on Institute of Radio Engineers* 49 (1961), 8–30.

D. Mladenic and M. Grobelnik. 1999. Feature selection for unbalanced class distribution and naive Bayes. In *Proceedings of the 16th International Conference on Machine Learning (ICML-1999)*. Morgan Kaufmann, 258–267.

S. Monti and G. F. Cooper. 1999. A Bayesian network classifier that combines a finite mixture model and a naïve Bayes model. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-1999)*. 447–456.

M. Možina, J. Demšar, M. Kattan, and B. Zupan. 2004. Nomograms for visualization of naive Bayesian classifier. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2004)*. 337–348.

J. F. Murray, G. F. Hughes, and K. Kreutz-Delgado. 2005. Machine learning methods for predicting failures in hard drives: A multiple-instance application. *Journal of Machine Learning Research* 6 (2005), 783–816.

M. Narasimhan and J. A. Bilmes. 2005. A submodular-supermodular procedure with applications to discriminative structure learning. In *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence (UAI-2005)*. AUAI Press, 404–412.

A. Ng and M. Jordan. 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naïve Bayes. In *Advances in Neural Information Processing Systems 14 (NIPS-2001)*. MIT Press, 841–848.

G. N. Norén and R. Orre. 2005. Case based imprecision estimates for Bayes classifiers with the Bayesian bootstrap. *Machine Learning* 58, 1 (2005), 79–94.

M. Pazzani. 1996. Constructive induction of Cartesian product attributes. In *Proceedings of the Information, Statistics and Induction in Science Conference (ISIS-1996)*. 66–77.

M. Pazzani and D. Billsus. 1997. Learning and revising user profiles: the identification of interesting web sites. *Machine Learning* 27 (1997), 313–331.

J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, Palo Alto, CA.

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

F. Pernkopf. 2005. Bayesian network classifiers versus selective k-NN classifier. *Pattern Recognition* 38 (2005), 1–10.

F. Pernkopf and J. A. Bilmes. 2005. Discriminative versus generative parameter and structure learning of Bayesian network classifiers. In *Proceedings of the 22nd International Conference on Machine Learning (ICML-2005)*. ACM, 657–664.

F. Pernkopf and J. A. Bilmes. 2010. Efficient heuristics for discriminative structure learning of Bayesian network classifiers. *Journal of Machine Learning Research* 11 (2010), 2323–2360.

F. Pernkopf and P. O'Leary. 2003. Floating search algorithm for structure learning of Bayesian network classifiers. *Pattern Recognition Letters* 24 (2003), 2839–2848.

F. Pernkopf and M. Wohlmayr. 2009. On discriminative parameter learning of Bayesian network classifiers. In *Proceedings of the 20th European Conference on Machine Learning (ECML-2009). Lecture Notes in Computer Science*, Vol. 5782. Springer, 221–237.

F. Pernkopf and M. Wohlmayr. 2013. Stochastic margin-based structure learning of Bayesian network classifiers. *Pattern Recognition* 46, 2 (2013), 464–471.

F. Pernkopf, M. Wohlmayr, and S. Tschiatschek. 2012. Maximum margin Bayesian network classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 3 (2012), 521–532.

T. V. Pham, M. Worring, and A. W. M. Smeulders. 2002. Face detection by aggregated Bayesian network classifiers. *Pattern Recognition Letters* 23, 4 (2002), 451–461.

B. Poulin, R. Eisner, D. Szafron, Paul Lu, R. Greiner, D. S. Wishart, A. Fyshe, B. Pearcy, C. MacDonell, and J. Anvik. 2006. Visual explanation of evidence with additive classifiers. In *Proceedings of the 21th National Conference on Artificial Intelligence (AAAI-2006)*. AAAI Press/MIT Press, 1822–1829.

A. Prinzie and D. Van den Poel. 2007. Random multiclass classification: Generalizing random forests to random MNL and random NB. In *Proceedings of the Database and Expert Systems Applications. Lecture Notes in Computer Science*. Vol. 4653. Springer, 349–358.

G. M. Provan and M. Singh. 1995. Learning Bayesian networks using feature selection. In *Proceedings of the 5th International Workshop on Artificial Intelligence and Statistics (AISTATS-1995)*. 450–456.

R. Raina, Y. Shen, A. Y. Ng, and A. McCallum. 2004. Classification with hybrid generative/discriminative models. In *Advances in Neural Information Processing Systems 16 (NIPS-2003)*. The MIT Press.

M. Ramoni and P. Sebastiani. 2001a. Robust Bayes classifiers. *Artificial Intelligence* 125 (2001), 209–226.

M. Ramoni and P. Sebastiani. 2001b. Robust learning with missing data. *Machine Learning* 45, 2 (2001), 147–170.

C. A. Ratanamahatana and D. Gunopulos. 2003. Feature selection for the naive Bayesian classifier using decision trees. *Applied Artificial Intelligence* 17, 5–6 (2003), 475–487.

S. Renooij and L. C. van der Gaag. 2008. Evidence and scenario sensitivities in naive Bayesian classifiers. *International Journal of Approximate Reasoning* 49, 2 (2008), 398–416.

G. Ridgeway, D. Madigan, and T. Richardson. 1998. Interpretable boosted naïve Bayes classification. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-1998)*. 101–104.

V. Robles, P. Larrañaga, J. M. Peña, E. Menasalvas, and M. S. Pérez. 2003. Interval estimation naive Bayes. In *Proceedings of the 5th International Symposium on Intelligent Data Analysis (IDA-2003). Lecture Notes in Computer Science*, Vol. 2810. Springer, 143–154.

V. Robles, P. Larrañaga, J. M. Peña, E. Menasalvas, M. S. Pérez, and V. Herves. 2004. Bayesian networks as consensed voting system in the construction of a multi-classiffier for protein secondary structure prediction. *Artificial Intelligence in Medicine* 31 (2004), 117–136.

V. Robles, P. Larrañaga, J. M. Peña, M. S. Pérez, E. Menasalvas, and V. Herves. 2003. Learning semi naive Bayes structures by estimation of distribution algorithms. In *Proceedings of the 11th Portuguese Conference on Artificial Intelligence (EPIA-2003). Lecture Notes in Computer Science*. 244–258.

S. Rodrigues de Morais and A. Aussem. 2010. A novel Markov boundary based feature subset selection algorithm. *Neurocomputing* 73, 4–6 (2010), 578–584.

J. J. Rodríguez and L. I. Kuncheva. 2007. Naïve Bayes ensembles with a random oracle. In *Proceedings of the 7th International Workshop on Multiple Classifier Systems (MCS-2007). Lecture Notes in Computer Science*, Vol. 4472. Springer, 450–458.

T. Roos, H. Wettig, P. Grünwald, P. Myllymäki, and H. Tirri. 2005. On discriminative Bayesian network classifiers and logistic regression. *Machine Learning* 59, 3 (2005), 267–296.

G. A. Ruz and D. T. Pham. 2009. Building Bayesian networks classifiers thorugh a Bayesian monitoring system. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 223 (2009), 743–755.

Y. Saeys, I. Inza, and P. Larrañaga. 2007. A review of feature selection techniques in bioinformatics. *Bioinformatics* 23, 19 (2007), 2507–2517.

M. Sahami. 1996. Learning limited dependence Bayesian classifiers. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-1996)*. 335–338.

G. Santafé, J. A. Lozano, and P. Larrañaga. 2005. Discriminative learning of Bayesian network classifiers via the TM algorithm. In *Proceedings of the 8th European Conference in Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU-2005). Lecture Notes in Artificial Intelligence*, Vol. 3571. Springer, 148–160.

B. Sierra and P. Larrañaga. 1998. Predicting the survival in malignant skin melanoma using Bayesian networks automatically induced by genetic algorithms. An empirical comparison between different approaches. *Artificial Intelligence in Medicine* 14 (1998), 215–230.

B. Sierra, E. Lazkano, E. Jauregi, and I. Irigoien. 2009. Histogram distance-based Bayesian network structure learning: A supervised classification specific approach. *Decision Support Systems* 48, 1 (2009), 180–190.

B. Sierra, N. Serrano, P. Larrañaga, E. J. Plasencia, I. Inza, J. J. Jiménez, P. Revuelta, and M. L. Mora. 2001. Using Bayesian networks in the construction of a bi-level multi-classifier. A case study using intensive care unit patient data. *Artificial Intelligence in Medicine* 22 (2001), 233–248.

M. Singh and G. Provan. 1996. Efficient learning of selective Bayesian network classifiers. In *Proceedings of the 13th International Conference on Machine Learning (ICML-1996)*. 453–461.

M. Singh and M. Valtorta. 1995. Construction of Bayesian network structures from data: A brief survey and an efficient algorithm. *International Journal of Approximate Reasoning* 12, 2 (1995), 111–131.

P. Spirtes, C. Glymour, and R. Scheines. 1993. *Causation, Prediction, and Search*.

J. Su, H. Zhang, C. X. Ling, and S. Matwin. 2008. Discriminative parameter learning for Bayesian networks. In *Proceedings of the 25th International Conference on Machine Learning (ICML-2008)*, Vol. 307. ACM, 1016–1023.

J.-N. Sulzmann, J. Fürnkranz, and E. Hüllermeier. 2007. On pairwise naive Bayes classifiers. In *Proceedings of the 18th European Conference on Machine Learning (ECML-2007). Lecture Notes in Computer Science*, Vol. 4701. Springer, 371–381.

D. M. Titterington, G. D. Murray, L. S. Spiegelhalter, A. M. Skene, J. D. F. Habbema, and G. J. Gelpke. 1981. Comparison of discrimination techniques applied to a complex data set of head injured patients (with discussion). *Journal of the Royal Statistical Society Series A* 144, 2 (1981), 145–175.

I. Tsamardinos and C. F. Aliferis. 2003. Towards principled feature selection: Relevancy, filters and wrappers. In *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics (AISTATS-2003)*.

I. Tsamardinos, C. F. Aliferis, and A. R. Statnikov. 2003a. Algorithms for large scale Markov blanket discovery. In *Proceedings of the 16th International Florida Artificial Intelligence Research Society Conference (FLAIRS-2003)*. AAAI Press, 376–381.

I. Tsamardinos, C. F. Aliferis, and A. R. Statnikov. 2003b. Time and sample efficient discovery of Markov blankets and direct causal relations. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*. 673–678.

M. van Gerven and P. J. F. Lucas. 2004. Employing maximum mutual information for Bayesian classification. In *Proceedings of the 5th International Symposium on Biological and Medical Data Analysis (ISBMDA-2004). Lecture Notes in Computer Science*, Vol. 3337. Springer, 188–199.

T. Verma and J. Pearl. 1990. Equivalence and synthesis of causal models. In *Proceedings of the 6th Conference on Uncertainty in Artificial Intelligence (UAI-1990)*. Elsevier, 255–270.

D. Vidaurre, C. Bielza, and P. Larrañaga. 2012. Forward stagewise naive Bayes. *Progress in Artificial Intelligence* 1 (2012), 57–69.

R. Vilalta and I. Rish. 2003. A decomposition of classes via clustering to explain and improve naive Bayes. In *Proceedings of the 14th European Conference on Machine Learning (ECML-2003). Lecture Notes in Computer Science*, Vol. 2837. Springer, 444–455.

G. I. Webb, J. Boughton, and Z. Wang. 2005. Not so naive Bayes: Aggregating one-dependence estimators. *Machine Learning* 58 (2005), 5–24.

G. I. Webb and M. J. Pazzani. 1998. Adjusted probability naïve Bayesian induction. In *Proceedings of the 11th Australian Joint Conference on Artificial Intelligence (AI-1998). Lecture Notes in Computer Science*, Vol. 1502. Springer.

T.-T. Wong. 2009. Alternative prior assumptions for improving the performance of naïve Bayesian classifiers. *Data Mining and Knowledge Discovery* 18, 2 (2009), 183–213.

J. Xiao, C. He, and X. Jiang. 2009. Structure identification of Bayesian classifiers based on GMDH. *Knowledge-Based Systems* 22 (2009), 461–470.

J.-H. Xue and D. M. Titterington. 2010. Joint discriminative-generative modelling based on statistical tests for classification. *Pattern Recognition Letters* 31, 9 (2010), 1048–1055.

Y. Yang, K. B. Korb, K. M. Ting, and G. I. Webb. 2005. Ensemble selection for superparent-one-dependence estimators. In *Proceedings of the 18th Australian Conference on Artificial Intelligence*. 102–112.

Y. Yang, G. I. Webb, J. Cerquides, K. B. Korb, J. Boughton, and K. M. Ting. 2007. To select or to weigh: A comparative study of linear combination schemes for superparent-one-dependence estimators. *IEEE Transactions on Knowledge and Data Engineering* 19 (2007), 1652–1665.

S. Yaramakala and D. Margaritis. 2005. Speculative Markov blanket discovery for optimal feature selection. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM-2005)*. IEEE Computer Society, 809–812.

M. Zaffalon. 2002. The naïve credal classifier. *Journal of Statistical Planning and Inference* 105, 1 (2002), 5–21.

M. Zaffalon and E. Fagiuoli. 2003. Tree-based credal networks for classification. *Reliable Computing* 9, 6 (2003), 487–509.

H. Zhang and S. Sheng. 2004. Learning weighted naive Bayes with accurate ranking. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM-2005)*. IEEE Computer Society, 567–570.

H. Zhang and J. Su. 2008. Naive Bayes for optimal ranking. *Journal of Experimental & Theoretical Artificial Intelligence* 20, 2 (2008), 79–93.

N. L. Zhang, T. D. Nielsen, and F. V. Jensen. 2004. Latent variable discovery in classification models. *Artificial Intelligence in Medicine* 30, 3 (2004), 283–299.

F. Zheng and G. I. Webb. 2006. Efficient lazy elimination for averaged one-dependence estimators. In *Proceedings of the 23rd International Conference on Machine Learning (ICML-2006)*, Vol. 148. ACM, 1113–1120.

Z. Zheng. 1998. Naïve Bayesian classifier committees. In *Proceedings of the 10th European Conference on Machine Learning (ECML-1998). Lecture Notes in Computer Science*, Vol. 1398. Springer, 196–207.

Z. Zheng and G. I. Webb. 2000. Lazy learning of Bayesian rules. *Machine Learning* 41 (2000), 53–84.

B. Ziebart, A. K. Dey, and J. A. Bagnell. 2007. Learning selectively conditioned forest structures with applications to DBNs and classification. In *Proceedings of the 23rd Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-2007)*. AUAI Press, 458–465.

Q2 Received January 2013; revised October 2013; accepted xxx

QUERIES

Q1: AU: Please provide full author address for correspondence.
Q2: AU: Please provide accepted date.