HPB: A Model for Handling BN Nodes with High Cardinality Parents

Jorge Jambeiro Filho JORGE.FILHO@JAMBEIRO.COM.BR
Alfândega do Aeroporto de Viracopos
Rodovia Santos Dummont, Km 66
Campinas-SP, Brazil, CEP 13055-900

Jacques Wainer WAINER@IC.UNICAMP.BR
Instituto de Computação
Universidade Estadual de Campinas
Caixa Postal 6176
Campinas - SP, Brazil, CEP 13083-970

Editor: Bianca Zadrozny

Abstract

We replaced the conditional probability tables of Bayesian network nodes whose parents have high cardinality with a multilevel empirical hierarchical Bayesian model called hierarchical pattern Bayes (HPB).[1] The resulting Bayesian networks achieved significant performance improvements over Bayesian networks with the same structure and traditional conditional probability tables, over Bayesian networks with simpler structures like naïve Bayes and tree augmented naïve Bayes, over Bayesian networks where traditional conditional probability tables were substituted by noisy-OR gates, default tables, decision trees and decision graphs and over Bayesian networks constructed after a cardinality reduction preprocessing phase using the agglomerative information bottleneck method. Our main tests took place in important fraud detection domains, which are characterized by the presence of high cardinality attributes and by the existence of relevant interactions among them. Other tests, over UCI data sets, show that HPB may have a quite wide applicability.

Keywords: probabilistic reasoning, Bayesian networks, smoothing, hierarchical Bayes, empirical Bayes

1. Introduction

In most countries, imported goods must be declared by the importer to belong to one of large set of classes (customs codes). It is important that each good is correctly classified, because each of the customs codes implies not only different customs duties but also different administrative, sanitary, and safety requirements. The original goal of this work was to develop a tool that, considering four explanatory attributes: *declared custom code* (DCC), *importer* (IMP), *country of production* (CP) and *entry point in the receiving country* (EPR), will estimate, for each new example, the probability that it involves a misclassification. Such estimates will be used later by a larger system that allocates human resources for different types of anti-fraud operations.

Our main study data set contains 682226 examples of correct classification (which we will call negative examples) and 6460 examples of misclassification (positive examples). In this data set, the

[1] This paper is a an extended version of a conference paper (Jambeiro Filho and Wainer, 2007).

first attribute assumes 7608 distinct values, the second, 18846 values, the third, 161 values, and the fourth 80 values. Thus, the domain is characterized by the presence of high cardinality attributes.

The data set is imbalanced, with only 0.93% of positive examples. This is usually handled with different resampling strategies (Chawla et al., 2002). However, resampling requires retraining the classifiers for each different assignment of costs for *false positives* and *false negatives*. In our context, such costs are not known in advance (priorities change according to other anti-fraud demands) and they vary from example to example (not all false negatives cost the same). These facts make the use of resampling techniques unattractive.

On the other hand, if we can produce reliable probability estimates directly from the original data set, the work of the human resource allocation system becomes much easier. It can at any time, define a selection rate that matches the available human resources for the specific task of detecting wrong customs codes considering all other anti-fraud demands at the moment. If the selection rate is, for example, 10%, the examples to be verified will naturally be the 10% that are most likely to involve a misclassification according to the calculated probability estimates. The allocation system may also combine the probability estimates with costs that may vary from example to example without any retraining. Thus, we decided to concentrate on Bayesian techniques.

Domain specialists claim that there are combinations of attribute values (some involving all of them) that make the probability of an instance being positive significantly higher then it could be expected looking at each value separately. They call such combinations *critical patterns*. To benefit from critical patterns we would like to use the Bayesian network (BN) (Pearl, 1988) presented in Figure 1, where all explanatory attributes are parents of the class attribute. We call a structure of this kind a *direct BN structure*.

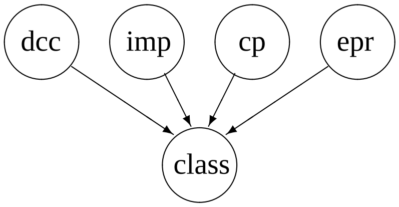

Figure 1: Direct BN structure for misclassification detection

In a BN, considering that x_{ji} is a possible value for node X_j and π_{jk} is a complete combination of values for Π_j, the set of parents of node X_j, the vector, θ_{jk}, such that $\theta_{jki} = P(x_{ji}|\pi_{jk})$ is stored in a table that is called conditional probability table (CPT) of node X_j and is assessed from the frequencies of the values of X_j among the training instances where $\Pi_j = \pi_{jk}$. The distributions of X_j given any two different combinations of values for its parents are assumed to be independent and a Dirichlet prior probability distribution for θ_{jk} is usually adopted. Applying Bayes rule and integrating over all possible values for θ_{jk} it is found that

$$E(\theta_{jki}) = P(x_{ji}|\pi_{jk}) = \frac{N_{jki} + \alpha_{jki}}{N_{jk} + \alpha_{jk}}, \tag{1}$$

where N_{jki} is the number of simultaneous observations of x_{ji} and π_{jk} in the training set, $N_{jk} = \sum_{\forall i} N_{jki}$, α_{jki} is the value of one of the parameters of the Dirichlet prior probability distribution and $\alpha_{jk} = \sum_{\forall i} \alpha_{jki}$, the equivalent sample size of the prior probability distribution.

The Dirichlet prior probability distribution is usually assumed to be noninformative, thus

$$P(x_{ji}|\pi_{jk}) = \frac{N_{jki} + \lambda}{N_{jk} + \lambda M_j}, \tag{2}$$

where all parameters of the Dirichlet distribution are equal to a small smoothing constant λ, and M_j is the number of possible values for node X_j. We call this *direct estimation* (DE). DE is sometimes called Lidstone estimate and if $\lambda = 1$ it is called Laplace estimate.

The conditional probability table of the class node of a BN with the structure in Figure 1 contains more than 1.8×10^{12} parameters. It is clear that for rarely seen combinations of attributes the choice of such structure and Equation (2) tends to produce unreliable probabilities whose calculation is dominated by the noninformative prior probability distribution.

Instead of the structure in Figure 1, we can choose a network structure that does not lead to too large tables. This can be achieved limiting the number of parents for a network node. Naïve Bayes(Duda and Hart, 1973) is an extreme example where the maximum number of parents is limited to one (the class node is the only parent of any other node). Tree augmented naïve Bayes (TAN) (Friedman et al., 1997) adds a tree to the structure of naïve Bayes connecting the explanatory attributes and limits the maximum number of parent nodes to two. However, limiting the maximum number of parents also limits the representational power of the Bayesian network(Boullé, 2005) and, thus, limits our ability to capture interactions among attributes and benefit from critical patterns. Therefore, we would prefer not to do it.

Since the high cardinality of our attributes is creating trouble, it is a reasonable idea to preprocess the data, reducing the cardinality of the attributes. We can use, for example, the agglomerative information bottleneck (AIBN) method (Slonim and Tishby, 1999) for this task. However, the process of reducing the cardinality of one attribute is blind with respect to the others (except for the class attribute) (Slonim and Tishby, 1999; Boullé, 2005; Micci-Barreca, 2001), and thus it is unlikely that cardinality reduction will result in any significant improvement in the ability to capture critical patterns, which always depend on more than one attribute.

When the number of probabilities to be estimated is too large if compared to the size of the training set and we cannot fill the traditional conditional probability tables satisfactorily, Pearl (1988) recommends the adoption of a model that resorts to causal independence assumptions like the noisy-OR gate. Using noisy-OR, the number of parameters required to represent the conditional probability distribution (CPD) of a node given its parents, instead of being proportional to the product of the cardinality of all parents attributes, becomes proportional to the sum of their cardinality. However, causal independence assumptions are incompatible with our goal of capturing critical patterns.

It is possible to use more flexible representations for the conditional probability distributions of a node given its parents, like default tables (DFs) (Friedman and Goldszmidt, 1996b), decision trees (DTs) (Friedman and Goldszmidt, 1996b) and decision graphs (DGs) (Chickering et al., 1997). According to Friedman and Goldszmidt (1996b), using such representations together with adequate learning procedures induces models that better emulate the real complexity of the interactions present in the data and the resulting network structures tend to be more complex (in terms of arcs) but require fewer parameters. Fewer parameters may result in more reliable probability estimates.

Using traditional CPTs, we assume that the probability distributions for a node given any two combinations of values for the parents are independent. If some of these distributions are actually identical, DTs, DFs and DGs, can reflect it and represent the CPD using a variable number of parameters that is only proportional to the number of actually different distributions.

On the other hand, using DTs, DFs or DGs to represent the conditional probability distributions of a node given its parents, we assume that the probability distribution of the node given two different combinations of values for the parents may be either identical or completely independent. It is possible that neither of the two assumptions hold.

Gelman et al. (2003) assert that modeling hierarchical data nonhierarchically leads to poor results. With few parameters, nonhierarchical models cannot fit the data accurately. With many parameters they fit the existing data well but lead to inferior predictions for new data. In other words they overfit the training set. In contrast, hierarchical models can fit the data well without overfitting. They can reflect similarities among distributions without assuming equality.

The slight modification in Equation (2) used by Friedman et al. (1997) in the definition of a smoothing schema for TAN shows that we can treat the data that is used to estimate a CPT as hierarchical:

$$P(x_{ji}|\pi_{jk}) = \frac{N_{jki} + S \cdot P(x_{ji})}{N_{jk} + S},$$

where S is a constant that defines the equivalent sample size of the prior probability distribution. We call this *almost direct estimation* (ADE). ADE is the consequence of adopting an informative Dirichlet prior probability distribution where $\alpha_{jki} \propto P(x_{ji})$, where $P(x_{ji})$ is the unconditional probability of x_{ji} (for the meaning of α_{jki}, see Equation 1). ADE uses the probability distribution assessed in a wider population (the whole training set) to build an informative prior probability distribution for a narrower population and so it has a hierarchical nature. In the sense of Gelman et al. (2003) ADE is an empirical hierarchical Bayesian model, not a full hierarchical Bayesian model. Probability estimation methods which use such empirical models are popularly known as empirical Bayes (EB) methods. ADE is also considered a m-estimation method (Cestnik, 1990; Zadrozny and Elkan, 2001).

We believe that ADE can get closer to the true probability distribution, but not that its discrimination power can be significantly better than DE's. It is a linear combination of two factors N_{jki}/N_{jk} and $P(x_{ji})$. The second factor is closer to the true probability distribution than its constant counterpart in *direct estimation* but it is still equal for any combination of values of Π_j and thus has no discrimination power.

ADE jumps from a very specific population (the set of training examples where $\Pi_j = \pi_{jk}$) to a very general population (the whole training set). In contrast, we present a model, that we call hierarchical pattern Bayes (HPB), which moves slowly from smaller populations to larger ones benefiting from the discrimination power available at each level.

2. Hierarchical Pattern Bayes

HPB is an empirical Bayes method that generalizes ADE into an aggressive multilevel smoothing strategy. Its name comes from the fact that it explores an hierarchy of patterns intensively, though it is not a full hierarchical Bayesian model.

Given a pattern W and a training set, D, of pairs (U_t, C_t), where U_t is the t^{th} instance in D and C_t is the class label of U_t, HPB calculates $P(C_r|W)$ for any class C_r, where a pattern is as defined below:

Definition 1 *A pattern is a set of pairs of the form $(Attribute = Value)$, where any attribute can appear at most once.*

An attribute that is not in the set is said to be undefined or missing. Before presenting HPB details we need a few more definitions:

Definition 2 *An instance U is a pair $(iid, Pat(U))$ where $Pat(U)$ is a pattern and iid is an identifier that makes each instance unique.*

Definition 3 *A pattern Y is more generic than a pattern W if and only if $Y \subseteq W$*

If Y is more generic than W, we say that W satisfies Y. If an instance U_t is such that $W = Pat(U_t)$ and W satisfies Y, we also say that U_t satisfies Y. It is worth noting that, if $Y \subseteq W$ then $S_Y \supseteq S_W$ where S_Y is the set of instances satisfying Y and S_W is the set of instances satisfying W.

Definition 4 *The level of a pattern W, $level(W)$, is the number of attributes defined in W.*

Definition 5 *$g(W)$ is the set of all patterns more generic than a pattern W whose elements have level equal to $level(W) - 1$.*

For example, if W is $\{A = a, B = b, C = c\}$, $g(W)$ is

$$\{\ \{B = b, C = c\}, \{A = a, C = c\}, \{A = a, B = b\}\ \}.$$

2.1 The Hierarchical Model

HPB calculates the posterior probability $P(C_r|W)$, using a strategy that is similar to almost direct estimation, but the prior probabilities are considered to be given by $P(C_r|g(W))$.

The parameters of the Dirichlet prior probability distribution used by HPB are given by $\alpha_r = S \cdot P(C_r|g(W))$, where S is a smoothing coefficient. Consequently,

$$P(C_r|W) = \frac{N_{wr} + S \cdot P(C_r|g(W))}{N_w + S}, \tag{3}$$

where N_w is the number of instances in the training set satisfying the pattern W and N_{wr} is the number of instances in the training set satisfying the pattern W whose class label is C_r.

Given Equation (3), the problem becomes to calculate $P(C_r|g(W))$. Our basic idea is to write $P(C_r|g(W))$ as a function of the various $P(C_r|W_j)$ where the W_j are patterns belonging to $g(W)$ and calculate each $P(C_r|W_j)$ recursively, using Equation (3).

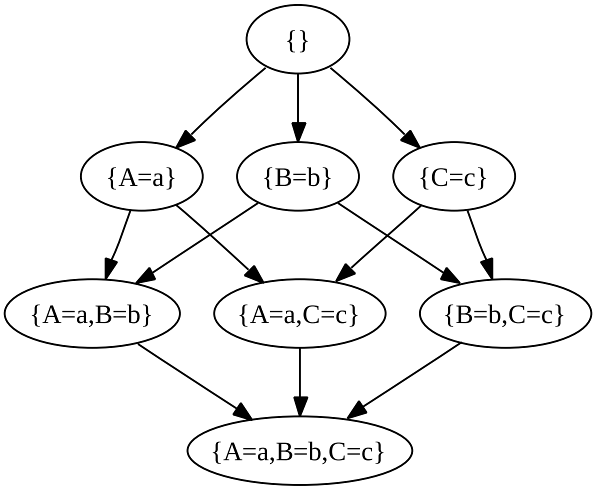

Figure 2: Example of HPB structure

Figure 2 shows a pattern hierarchy,[2] where A, B and C are the attributes. Each pattern is represented by a node and the set of parents of a pattern W in the DAG presented in Figure 2 is $g(W)$. HPB combines the posterior predictive probability distributions, $P(C_r|W_j)$, of the class given each parent, W_j, of a pattern W, to build the prior predictive probability distribution for the class given W, $P(C_r|g(W))$.

The first step to write $P(C_r|g(W))$ as a function of all the $P(C_r|W_j)$ is to apply Bayes theorem:

$$\begin{aligned} P(C_r|g(W)) &= \frac{P(g(W)|C_r)P(C_r)}{P(g(W))} \\ &\propto P(W_1,W_2,\dots,W_L|C_r)P(C_r), \end{aligned}$$

where W_1,W_2,...,W_L are the elements of $g(W)$. Then we approximate the joint probability $P(W_1,W_2,$ $\dots,W_L|C_r)$ by the product of the marginal probabilities:

$$P'(C_r|g(W)) \propto P(C_r)\prod_{j=1}^{L} P(W_j|C_r), \tag{4}$$

Note that we do not assume any kind of independence when using Equation (3) to calculate posterior predictive probabilities, but we do assume independence in a naïve Bayes fashion when calculating the prior probabilities using Equation (4). Naïve Bayes is known to perform well with regard to classification error (Domingos and Pazzani, 1997) and ranking (Zhang and Su, 2004), even when its independence suppositions are violated. Assuming independence among overlapping patterns, as Equation (4) does, is equivalent to assuming independence among attributes which are known to be highly correlated, what may appear to be strange. However, naïve Bayes has been reported to perform better when attributes are highly correlated than when correlation is moderate (Rish et al., 2001).

[2] Note that the DAG in Figure 2 is not a Bayesian network and the dependencies among its nodes do not follow BN conventions.

On the other hand, naïve Bayes is known to produce extreme probabilities (Domingos and Pazzani, 1997), thus we apply a calibration mechanism (Bennett, 2000; Zadrozny, 2001), which is expressed in Equation (5):

$$P''(C_r|g(W)) = (1-A)\cdot P'(C_r|g(W)) + A\cdot P(C_r), \tag{5}$$

where $A = B/(1+B)$ and B is a calibration coefficient. We discuss this calibration mechanism in Section 2.2. $P''(C_r|g(W))$ is our best estimate for $P(C_r|g(W))$ and it is used in Equation (3) as if it were the true value of $P(C_r|g(W))$.

Given Equations (4) and (5) we need to calculate $P(W_j|C_r)$. Applying Bayes theorem again,

$$P(W_j|C_r) = \frac{P(C_r|W_j)P(W_j)}{P(C_r)}. \tag{6}$$

We can estimate $P(C_r)$ is using the maximum likelihood approach: $P(C_r) = N_r/N$, where N_r is the number of examples in the training set whose class label is C_r, and N is the total number of examples in the training set. If the class variable is binary, this strategy works well, but if the class node has high cardinality it is better to employ a noninformative prior probability distribution:

$$P(C_r) = \frac{N_r + S^{NI}/M_c}{N + S^{NI}},$$

where M_c is the number of classes and S^{NI} is the smoothing constant that defines the equivalent sample size of the noninformative distribution.

When we substitute $P(W_j|C_r)$ by the right side of Equation (6) into Equation (4) we are able to clear out the factor $P(W_j)$ because it is identical for all classes:

$$\begin{aligned} P'(C_r|g(W)) &\propto P(C_r)\prod_{j=1}^{L} P(W_j|C_r) \\ &\propto P(C_r)\prod_{j=1}^{L} \frac{P(C_r|W_j)P(W_j)}{P(C_r)} \\ &\propto P(C_r)\prod_{j=1}^{L} \frac{P(C_r|W_j)}{P(C_r)}, \end{aligned}$$

so we do not need to worry about it.

Since W_j is a pattern, the estimation of $P(C_r|W_j)$ can be done recursively, using Equation (3). The recursion ends when $g(W)$ contains only the empty pattern. In this case $P(C_r|g(W)) = P(C_r|\{\{\}\}) = P(C_r)$.

2.2 Calibration Mechanism

Naïve Bayes is known to perform well in what regards to classification error (Domingos and Pazzani, 1997) and ranking (Zhang and Su, 2004), even when its independence suppositions are violated. However, naïve Bayes is also known to produce unbalanced probability estimates that are typically too "extreme" in the sense that they are too close to zero or too close to one.

The reason why naïve Bayes produces extreme probabilities is that it treats each attribute value in a pattern as if it were new information. Since attributes are not really independent, a new attribute value is not 100% new information, treating it as if it were completely new reinforces the previous beliefs of naïve Bayes towards either zero or one. This reuse of information is explained by Bennett (2000) in the context of text classification.

In order to obtain better posterior probability distributions, calibration mechanisms which try to compensate the overly confident predictions of naïve Bayes have been proposed (Bennett, 2000; Zadrozny, 2001).

Naïve Bayes assumes that attributes are independent given the class. Equation (4) assumes that some aggregations of attributes are independent given the class. Since many of these aggregations have attributes in common, the use of Equation (4) is equivalent to assuming independence among attributes which are known to be highly correlated. Naïve Bayes has been reported to perform better when attributes are highly correlated than when correlation is moderate (Rish et al., 2001), but it is quite obvious that we are reusing a lot of information and that we can expect very extreme probability estimates. Therefore, we need to use a calibration mechanism.

Our mechanism is simpler than the ones presented by Bennett (2000) and by Zadrozny and Elkan (2002) and is unsupervised. This makes it very fast and easy to employ within each step of HPB.

We just made a linear combination of the result of Equation (4) and $P(C_r)$. We did that considering that if the estimates are more extreme than the true probabilities both near zero and near one they must match the true probabilities at some point in the middle. We believe that this point is somewhere near $P(C_r)$.

Extreme probabilities are produced when evidence in favor or against a class is reused. $P(C_r)$ is a point where either there is no evidence or there is evidence in conflicting directions in such way that the effect is null. Thus, such a point cannot be considered extreme. Our calibration mechanism attenuates the probabilities when they are extreme without affecting them in the point $P(C_r)$, where, we believe, they are already correct.

In Figure 3 we show the effect of the calibration mechanism.

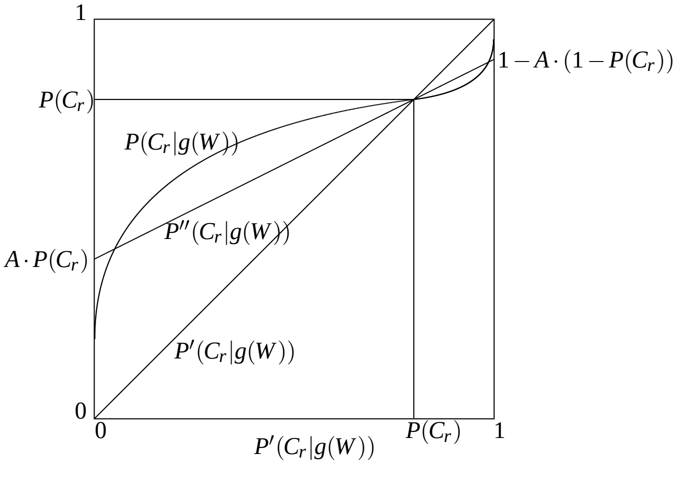

Figure 3: Effect of linear calibration over extreme probabilities

In the horizontal axis the non calibrated estimation $P'(C_r|g(W))$ is represented. The curved line represents the true probability, $P(C_r|g(W))$, as a function of $P'(C_r|g(W))$. Since all information about $P'(C_r|g(W))$ comes from a finite data set such function never hits one or zero. When $P'(C_r|g(W))$ is near zero, $P(C_r|g(W))$ is not as near. The same happens when $P(C_r|g(W))$ is near one.

The 45^o straight line represents what would be our final estimation if we did not do any calibration, that is, $P'(C_r|g(W))$ itself. The other oblique straight line is the result of our calibration mechanism, $P''(C_r|g(W))$. It is still a linear approximation but it is much closer from $P(C_r|g(W))$ than $P'(C_r|g(W))$.

2.3 Analyzing HPB

HPB tries to explore the training set as much as possible. If there are L attributes, HPB starts its work capturing the influence of patterns of level L. At this level, all interactions among attributes may be captured as long as there are enough training instances. However, no training set is so large that we can expect that all level L patterns are well represented. Actually, if there are high cardinality attributes, it is more likely that only a minority of them are represented well. For this minority, level L dominates Equation (3) and prior probabilities are not very important. On the other hand, prior probabilities are critical for the vast majority of cases where level L patterns are not well represented in the training set. Then, HPB moves to level $L-1$. At this level, a greater fraction of patterns are well represented and it is still possible to capture the majority of attribute interactions. Many patterns of level $L-1$, however, are still not well represented and it is necessary to resort to lower level patterns. The lower are the level of the patterns the weaker is HPB's capacity to capture interactions, but less common are problems with small sample sizes.

Equation (3) combines the influence of different level patterns in a way that the most specific patterns always dominate if they are well represented. Equation (4) combines patterns in an naïve Bayes fashion, in spite of the fact that they are highly correlated. This results in extreme probability estimates that are attenuated by the calibration mechanism in Equation (5).

Since the population of instances (both in the training and in the test set) satisfying a pattern W is a subpopulation of the population of instances satisfying $W_j, \forall W_j \in g(W)$, we can say that HPB uses results previously assessed in a wider population to build informative prior probability distributions for narrower populations. Therefore, HPB is a an empirical Bayesian model, not a full hierarchical Bayesian model.

In the work of Gelman et al. (2003); Andreassen et al. (2003); Stewart et al. (2003) full hierarchical Bayesian models are presented, but they have only two levels. HPB deals with a multi level hierarchy recursively and also handles the fact that each subpopulation is contained by several overlapping superpopulations and not only by one superpopulation. These facts make it more difficult to build a full model that allows the calculation of all involved probability distributions at once considering all available evidence.

2.4 HPB as a Replacement for conditional probability tables

HPB's original goal was to be a stand alone classifier well suited a to particular domain, but it is much more relevant as a replacement for conditional probability tables.

HPB's use of space and time is exponential in the number of attributes. Thus, in domains with many attributes, it is not possible to use HPB directly. However, since the number of parents of any node in a Bayesian network is usually small because the size of a CPT is exponential in the number of parent nodes, HPB may be used as a replacement for Bayesian networks conditional probability tables in almost any domain.

Space and time are frequently not the limiting factor for the number of parents of a BN node. More parents usually mean less reliable probabilities (Keogh and Pazzani, 1999) and it is not uncommon to limit their number to two (Friedman and Goldszmidt, 1996a; Keogh and Pazzani, 1999; Hamine and Helman, 2004). So, if HPB produces better probability estimates, it will actually allow for the addition of more parent nodes.

If the BN structure is given, the use of HPB as a replacement of the CPT of any node, X_j, is straightforward. To calculate, $P(x_{jk}|\pi_{ji})$ it is just a matter of acting as if $C_r = x_{jk}$ and $W = \pi_{ji}$, ignoring all other attributes and using HPB to calculate $P(C_r|W)$.

If the BN structure needs to be learned from data, it is necessary to choose a scoring metric that can work together with HPB in the task of choosing among the possible BN structures. We propose the use of the log-likelihood evaluated using leave-one-out cross validation:

$$LLLOO = \sum_t \log P(U_t|S, D - \{U_t\}) = \sum_t \sum_j \log P(x_{jt}|\pi^S_{jt}, D - \{U_t\}),$$

where D is the training set, U_t is the t^{th} instance of D, S is the BN structure being scored, x_{jt} is the value assumed by attribute X_j in the instance U_t, π^S_{jt} is the set of values assumed, in U_t, by the parents of X_j in S and $P(x_{jt}|\pi^S_{jt}, D - \{U_t\})$ is the value calculated by HPB for $P(x_{jt}|\pi^S_{jt})$ using $D - \{U_t\}$ as the training set.

HPB uses the training set only through the frequencies N_{wr} and N_w in Equation (3). For fast computation of $LLLOO$, we can assess these frequencies in D and rely on the relations:

$$\begin{aligned} N_w^{D-\{U_t\}} &= \begin{cases} N_w^D - 1 & \text{if } W \subset \pi^S_{jt}; \\ N_w^D & \text{otherwise}; \end{cases} \\ N_{wr}^{D-\{U_t\}} &= \begin{cases} N_{wr}^D - 1 & \text{if } W \subset \pi^S_{jt} \wedge x_{jr} = x_{jt}; \\ N_{wr}^D & \text{otherwise}. \end{cases} \end{aligned}$$

2.5 Selecting HPB Coefficients

Equations (3) and (5) require respectively the specifications of coefficients S and B. In the classification of a single instance, these equations are applied by HPB in the calculation of $P(C_r|W)$ for several different patterns, W. The optimal values of S and B can be different for each pattern.

In the case of the B coefficients, we use a heuristic motivated by the fact that the level of any pattern in $g(W)$ is $level(W) - 1$. The higher such level is, the more attributes in common the aggregations have, the more extreme probability estimates are and the stronger must be the effect of the calibration mechanism. Thus, we made the coefficient B in Equation (5) equal to $b(level(W) - 1)$, where b is an experimental constant.

In the case of the S coefficients, we can employ a greedy optimization approach, or, for faster training, simply define S to be a constant.

The optimization process we propose uses the area under the hit curve(Zhu, 2004) as a scoring metric. The hit curve of a classifier C over a data set D is a function, $h_{C,D}(r)$, where r is a selection rate (a real number in the interval $[0,1]$). The classifier is used to assign to each example, U_t in D the probability that U_t is a positive instance. The value of $h_{C,D}(r)$ is the number of positive instances among the $r \cdot |D|$ instances that were considered the most likely to be positive by the classifier.

We employed hit curves, instead of the more popular *Receiver Operating Characteristic Curves* (ROC) (Egan, 1975), because they match the interests of the user of a fraud detection system directly. Given a selection rate that reflects the available human resources, he/she wants to maximize the number of detected frauds.

Since the concept of a positive instance only makes sense for binary class variables, the optimization process only works for binary class problems.

When applicable, the process starts from the most general pattern family and moves toward the more specific ones, where a pattern family is the set containing all patterns that define exactly the same attributes (possibly with different values).

Assuming that the S coefficients have already been fixed for all pattern families that are more generic than a family F, there is a single S coefficient that needs to be specified to allow the use of Equation (3) to calculate $P(C_r|W)$, where W is any pattern belonging to F.

This coefficient is selected in order to maximize the area under the hit curve that is induced when, using leave-one-out cross validation, we calculate $P(C_0|W)$ for all training patterns, W, in F, where C_0 is the class that is defined to be the *positive* class.

Calculating $P(C_0|W)$ using leave-one-out cross validation, means, as explained in Section 2.4, simply subtracting one from some frequencies used by Equation (3).

2.6 Computational Complexity

The training phase of the version of HPB where constant smoothing coefficients are employed consists solely in assessing the frequencies used by Equation (3). It is easy to see that each instance, U, such that $W = Pat(U)$, in the training set, D, requires that exactly 2^L frequencies are incremented, where L is the number of parent attributes. Thus, HPB training time is

$$O(N_{tr} \cdot 2^L),$$

where N_{tr} in the number of training instances.

The test (or application) phase of HPB requires that, for each test instance, U, such that $W = Pat(U)$, the probability distribution for the class is computed given 2^L patterns. Since each computation is proportional to the number of classes, HPB test time is

$$O(N_{ts} \cdot M_c \cdot 2^L),$$

where N_{ts} in the number of test instances and M_c is the number of classes.

Note that, in both cases, HPB running time is exponential in the number of parent attributes, linear in the number of instances and independent of the cardinality of the parent attributes.

When the S coefficients are chosen by the optimization process described in Section 2.5, HPB test time does not change, but training requires that, for each pattern family, several S candidates are tested. There are 2^L pattern families and each test requires applying HPB to all training instances. Thus, HPB training time becomes

$$O(N_{tr} \cdot 2^L + N_{cand} \cdot 2^L \cdot N_{tr} \cdot M_c \cdot 2^L) = O(N_{cand} \cdot N_{tr} \cdot M_c \cdot 2^{2L}),$$

where N_{cand} is the number of candidates considered to choose a single S coefficient, which depends on the search algorithm.

HPB needs to save, for each training pattern, less than 2^L frequencies. Thus HPB use of space is

$$O(N_{tr} \cdot 2^L).$$

3. Experimental Results

We evaluated HPB in three different contexts:

- *misclassification detection*: HPB's motivation problem, an important classification problem for Brazil's Federal Revenue, where four high cardinality attributes which are supposed to have relevant interactions are used to predict a binary class attribute;

- *prediction of joint behavior*: another problem originated from Brazil's Federal Revenue where two high cardinality attributes are used to predict a third high cardinality attribute;

- *HPB as a general replacement for CPTs of Bayesian Networks*: tests over several UCI data sets comparing HPB to CPTs and other representations of the conditional probability distribution of a BN node given its parents.

In all cases the classification methods were tested using the Weka Experimenter tool (Witten and Frank, 1999) with five-fold cross validation. The machine used in the tests was an Intel Core 2 Duo 6300 with 2 GB of primary memory.

3.1 Misclassification Detection

This is the motivation problem for HPB. Considering four explanatory attributes: *declared custom code* (DCC), *importer* (IMP), *country of production* (CP) and *entry point in the receiving country* (EPR), we need to estimate, for each new example, the probability that it involves a misclassification, that is, the probability that the DCC is not the correct custom code for the goods being traded.

Our data set has 682226 examples of correct classification (which we will call negative examples) and 6460 examples of misclassification (positive examples). In this data set, the first attribute assumed 7608 distinct values, the second, 18846 values, the third, 161 values, and the fourth 80 values. There are no missing values.

We compared classifiers built using the following methods:

- *HPB-OPT*: BN with the *direct BN structure* (Figure 1), where the CPT of the class node was replaced by HPB with selection of S coefficients by the optimization process described in Section 2.5.

- *HPB*: BN with the *direct BN structure* (Figure 1), where the CPT of the class node was replaced by HPB with fixed S coefficients;

- *NB*: naïve Bayes;

- *Noisy-OR*: BN with the *direct BN structure* (Figure 1) using a noisy-OR gate instead of a CPT;

- *TAN*: Smoothed version of tree augmented naïve Bayes as described by Friedman et al. (1997) ;

- *ADE*: *almost direct estimation*. BN with the *direct BN structure*, traditional CPTs and the smoothing schema described by Friedman et al. (1997);

- *DE*: *direct estimation*. BN with the *direct BN structure* (Figure 1) and traditional CPTs;

- *DG*: Decision Graph constructed following Chickering et al. (1997). In this larger experiment, deviating from what was proposed by Chickering et al. (1997), we did not use DGs within BNs, but as standalone classification methods.

- *BN-HC-DT*: BN with decision trees learned using hill climbing (HC) and MDL as the scoring metric as described by Friedman and Goldszmidt (1996b);
- *BN-HC-DF*: BN with default tables learned using HC and MDL as described by Friedman and Goldszmidt (1996b);
- *PRIOR*: Trivial classifier that assigns the prior probability to every instance.

We were unable to build BNs with DGs replacing CPTs following Chickering et al. (1997) because it took too long (more than one day without completing a single fold). We found that the construction of a DG becomes very slow when the BN node in question has high cardinality and its parents also have high cardinality. High cardinality parents imply many possible split/merge operations to compare in each step of the learning algorithm and a high cardinality child implies that each comparison requires a lot of calculation.

In some experiments in the same domain, with BNs with DGs applied over smaller data sets, we found that in the global BN structures chosen by the search algorithm described by Chickering et al. (1997), all four explanatory attributes were parents of the class attribute. This means that if we had used a decision graph as a standalone classification method we would have had exactly the same results. Thus we concluded that it was worth to test a DG as a standalone classification method over our large data set. Since our class variable is binary the running time becomes acceptable.

We tried different parameterizations for each method and chose the parameter set that provided the best results in the five-fold cross-validation process, where best results mean best area under the hit curve up to 20% of selection rate. We ignored the area under the curve for selection rates above 20%, because all selection rates of interest are below this threshold.

Besides using the hit curve, we compared the probability distributions estimated by the models with the distribution actually found in the test set using two measures: root mean squared error (RMSE) and mean cross entropy (MCE):

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N}\sum_{r=1}^{M}(P'(C_{rt}) - P(C_{rt}))^2}{MN}}, \quad MCE = \frac{\sum_{i=1}^{N}\sum_{t=1}^{M} -P(C_{rt})\log_2 P'(C_{rt})}{MN},$$

where N is the number of instances in the test set, M is the number of classes, $P'(C_{jt})$ is the estimated probability that the t^{th} instance belongs to class C_r and $P(C_{tr})$ is the true probability that t^{th} instance belongs to class C_r. $P(C_r)$ is always either 0 or 1.

Many of the methods tested require the specification of parameters and many of them are real constants. We used a common strategy to chose such constants:

1. Based on experience, decide on a search interval, $SI = [beg, end]$, within which we believe the ideal constant is;
2. Build a search enumeration SE containing all powers of 10, all halves of powers of 10 and quarters of powers of 10 within SI;
3. Try all constants in SE. If the method requires more than one constant try all possible combinations exhaustively;
4. If the optimal constant, C is in the middle of SE take C as the final constant;

5. If the optimal constant, C is one of the extreme values of SE expand SE adding one more value to it and try again. The value to be added is the real number that is the nearest to the current optimal value that was not in SE and is a power of 10, a half of a power 10 or a quarter of a power of 10.

By restricting ourselves to powers of 10, halves of powers of 10 and quarters of powers of 10 we try different orders of magnitude for the constants and avoid fine tuning them.

The smoothing coefficients employed by HPB-OPT are all automatically selected. The selection involves a leave-one-out cross validation that takes place within the current training set (the five-fold cross validation varies the current training set). The B coefficients are defined by the heuristic described in Section 2.5 and by the constant b. The choice of b was done starting with $SI = [0.5, 2.5]$. The value of S^{NI} was set to zero.

HPB requires the specification of the S constant, which is used directly and the b constant which defines the B coefficients through the heuristic in Section 2.5. The choice of b was done starting with $SI = [0.5, 2.5]$. To choose S we defined $s = S/NumClasses = S/2$ and chose s starting from $SI = [1.0, 10.0]$. The reason to introduce the constant s is just to follow the way Weka usually handles smoothing constants. Again, the value of S^{NI} was set to zero.

DGs have four parameters: the smoothing constant and three boolean parameters defining the activation state of each of the possible operations, which are complete splits, binary splits and merges. The smoothing constant was chosen starting from $SI = [0.01, 1.0]$. We always kept complete splits enabled and tried the variations resulted from enabling/disabling binary splits and merges exhaustively for each smoothing constant.

Noisy-OR and PRIOR have no parameters. The optimization of all other methods involves only the smoothing constant, which, in all cases, was chosen starting from $SI = [0.01, 2.5]$.

Below we report the optimal parameters for each method:

- *HPB-OPT*: $b = 1.0$;
- *HPB*: $s = 5.0$ and $b = 1.0$;
- *NB*: $s = 0.1$;
- *TAN*: $s = 0.25$;
- *ADE*: $s = 0.01$;
- *DE*: $s = 2.5$;
- *DG CBM*: $s = 0.05$, complete splits, binary splits and merges enabled;
- *BN-HC-DT*: $s = 0.01$;
- *BN-HC-DF*: $s = 0.025$;

In Figure 4, we show the hit curves produced by each classification method. We chose to represent the $Recall = N_{TruePositives}/N_{Positives}$, in the vertical axis, instead of the absolute number of hits, because this does not change the form of the curve and makes interpretation easier. We represented the selection rate in log scale to emphasize the beginning of the curves. In Table 2 we show the recall values for different selection rates.

In Table 1, we show the area under the hit curve (AUC), the area under the hit curve up to 20% of selection rate (AUC20), the root mean squared error (RMSE), the mean cross entropy (MCE),[3] the training time (TR) and the test time (TS) of each method. The presence of the symbol *!* before any result means that it is significantly worse than its counterpart in the first row of the table using a 95% confidence t-test. Since HPB is in the first row, we can see that HPB is significantly better than all other classifiers with regard to AUC, AUC20 and MCE. With regard to RMSE, HPB was not better than BN-HC-DT, BN-HC-DF and PRIOR.

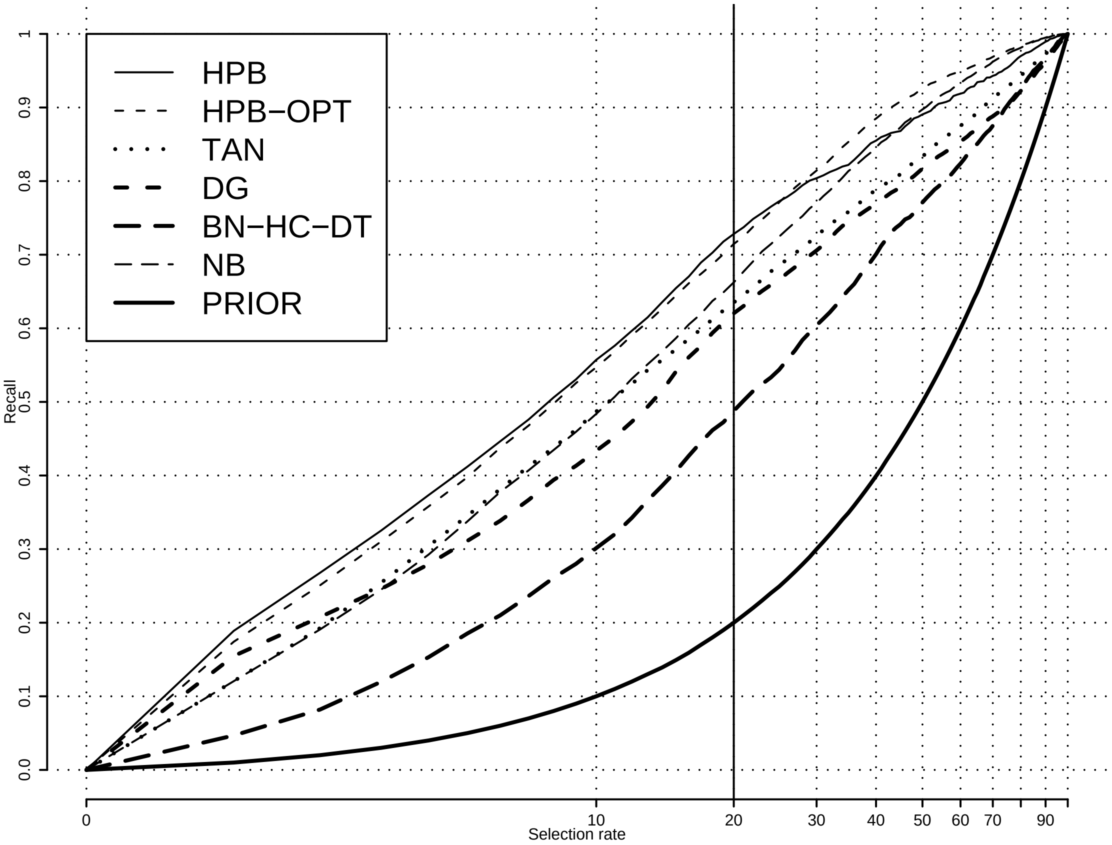

Figure 4: Misclassification detection - hit curves (to avoid pollution we only present curves related to a subset of the tested methods)

[3] For better visualization of the RMSE values, MCE values and their deviations, all RMSE and MCE values presented in this paper were multiplied by 10^4.

	1%	2%	5%	10%	20%
HPB	18.89±0.77	26.77±0.57	41.20±1.10	55.72±1.82	72.81±1.69
HPB-OPT	17.41±1.55	25.08±1.10	39.76±0.61	54.70±1.44	71.45±1.74
TAN	12.06±0.59	19.26±0.70	34.52±1.32	48.70±1.82	63.52±1.06
ADE	13.32±1.37	15.06±1.46	20.70±1.65	30.61±1.18	49.39±1.06
DE	8.32±0.69	10.42±0.73	16.49±0.73	26.58±0.56	45.54±0.58
DG	15.47±1.29	20.76±0.61	31.12±1.61	43.36±2.19	62.03±1.41
BN-HC-DT	4.68±0.23	8.20±0.62	18.54±0.51	30.14±1.13	48.78±1.32
BN-HC-DF	4.44±0.39	8.22±0.49	18.45±0.44	30.06±0.30	47.45±0.98
NB	12.06±0.35	19.07±0.87	33.76±0.68	48.37±1.70	66.24±1.56
Noisy-Or	12.86±0.46	20.36±1.13	33.45±0.73	47.36±1.69	63.26±1.52
PRIOR	1.00±0.00	2.00±0.00	5.00±0.00	10.00±0.00	20.00±0.00

Table 1: Misclassification detection - other measures

	AUC	AUC20	RMSE($\times 10^4$)	MCE($\times 10^4$)	TR(s)	TS(s)
HPB	83.17±0.73	53.34±1.37	986.05±3.82	347.54±4.01	9.84±0.55	7.79±1.03
HPB-OPT	84.47±0.70	52.21±1.21	!1006.26±5.24	!367.20±5.10	!517.66±4.76	!11.43±1.50
TAN	!78.10±0.72	!45.78±1.17	!1155.36±5.26	!484.05±7.94	!43.67±0.12	1.34±0.01
ADE	!74.96±0.19	!31.43±1.25	!1005.14±6.38	!459.39±4.46	4.04±0.12	0.34±0.09
DE	!72.33±0.57	!27.37±0.40	!3462.81±2.93	!2825.06±3.79	4.35±0.11	0.28±0.00
DG	!76.12±0.90	!42.89±1.55	!1007.47±6.90	!519.49±30.82	!577.78±29.29	4.47±0.48
BN-HC-DT	!70.47±0.76	!29.95±0.85	960.89±0.25	!364.68±1.59	!125.01±1.21	!2446.17±113.19
BN-HC-DF	!69.79±0.76	!29.63±0.43	960.78±0.26	!365.03±1.25	!2433.02±20.20	!265.02±3.41
NB	!81.73±0.79	!46.33±1.08	!1120.25±6.84	!419.47±6.68	4.79±0.06	0.28±0.00
Noisy-Or	!79.13±0.64	!45.07±1.09	!1016.06±5.08	!inf±0.00	4.73±0.07	0.28±0.00
PRIOR	!50.48±0.01	!10.48±0.01	963.96±0.00	!383.27±0.00	4.87±0.46	0.28±0.00

Table 2: Misclassification detection - recall at different selection rates

The PRIOR method is very conservative, assigning the prior probability to every instance. In this data set, such strategy results in a good MCE and a good RMSE. On the other hand, the PRIOR method has absolutely no discrimination power, considering all instances to be equally likely to be positive. In Figure 4 and Table 2, we can see that this results in random selection, just checking that recall is always approximately equal to the selection rate.

BN-HC-DT and BN-HC-DF produced similar hit curves as can be seen in Table 2. In Figure 4 BN-HC-DT is the second worse method. The reason is that the construction of DTs and DFs presented by Friedman and Goldszmidt (1996b) turned out to be very conservative, tending to prefer simple structures: DFs with few rows and DTs with few splits. Observing the PRIOR method results, it is not surprising that this conservative behavior results in a good MCE, a good RMSE and an unsatisfactory hit curve in comparison to other methods.

At a selection rate of 1%, ADE performs better than NB, noisy-OR and TAN, but for higher selection rates it is worse by a very significant margin. The reason is that critical patterns involving all attributes are decisive in the very beginning of the curves. ADE treats all attributes at once and thus can benefit from their presence, but soon ADE is forced to choose among test patterns for which there are no identical training patterns. At this point it starts to choose at random (in the average, 17% of the positive test instances are identical to at least one training instance).

Using Decision Graphs (with binary splits enabled), the most critical patterns were separated from the others and that resulted in a significant improvement in the beginning of the hit curve in comparison to methods like NB, noisy-OR or TAN, which cannot capture the influence of many attributes at once. However, the other patterns were clustered into few leaves in the graph. Within a leaf all patterns are considered equally likely to be positive. This resulted in loss of discrimination power for selection rates above 5%.

HPB (in both versions) benefits from critical patterns involving many or even all attributes, but also considers the influence of less specific patterns. As a consequence, it performs well for any selection rate. The version of HPB that uses a fixed value for the S coefficients is worse than NB for selection rates above 45%, but at this point, recall is already of 87% for both methods and the differences between them are never significant. Except for its non-optimized version, HPB-OPT is better than any other method for all selection rates, but the optimization process makes it fifty times slower than the simpler HPB.

It is worth noting that even the slower version of HPB is faster than the methods involving decision graphs, decision trees and default tables.

Since the cardinality of the attributes is a problem in this domain, we decided to also test all classification methods on a transformed data set where the cardinality of all attributes were reduced by the agglomerative information bottleneck method (AIBN). To prevent AIBN from using information from the test sets, we implemented a Weka meta classifier that applies AIBN immediately before training the real classifier and after each training set was separated from its associated test set in the five-fold cross validation process.

AIBN reduces the cardinality of an attribute by successively executing the merge of two values that results in minimum mutual information lost. The process can continue till a single value lasts, but can be stopped at any convenient point. We chose to limit the loss of mutual information to $1e-4$, a very low value. In spite of this, the cardinality reduction was accentuated. Table 3 shows the cardinality of the attributes before and after reduction.

Attribute	Original Cardinality	Final Cardinality
DCC	7608	101
IMP	18846	84
CP	161	50
EPR	80	28

Table 3: Cardinality reduction using AIBN

Because of the lower cardinality of the resulting attributes, it was possible to test BNs with DGs instead of standalone DGs. Results are in Table 4, Figure 5 and Table 5.

	1%	2%	5%	10%	20%
HPB	14.28±0.40	20.72±0.47	35.05±0.92	51.14±2.00	67.70±2.04
HPB-OPT	10.86±0.51	17.74±0.73	34.00±1.06	50.08±2.09	67.76±2.12
TAN	10.11±0.67	17.66±0.90	32.15±1.54	46.78±1.70	63.78±0.76
ADE	13.10±0.53	16.36±1.20	20.42±1.34	33.82±1.44	55.72±1.06
DE	8.28±0.59	11.17±0.64	19.40±0.64	32.82±0.47	56.66±0.63
BN-DG	8.14±0.46	17.40±0.66	32.12±1.38	45.44±1.12	60.66±1.48
BN-HC-DT	6.10±0.53	15.18±0.19	27.12±1.66	38.68±2.26	57.21±1.99
BN-HC-DF	6.22±0.45	14.94±0.15	26.33±0.56	37.92±1.53	55.05±1.21
NB	10.22±0.55	17.09±0.83	31.50±0.84	46.28±1.73	64.14±1.85
Noisy-Or	4.84±0.26	14.80±0.52	29.79±0.87	44.70±1.72	62.78±1.97
PRIOR	1.00±0.00	2.00±0.00	5.00±0.00	10.00±0.00	20.00±0.00

Table 4: Misclassification detection with cardinality reduction - recall at different selection rates

HPB and HPB-OPT are still the best methods but they lose much of their ability to explore critical patterns, and, at a selection rate of 1%, they do not perform nearly as well as they did over the original data set. The reason is that AIBN joins attribute values looking at each attribute separately and thus ignoring any interaction among them. In this case, relevant interactions were lost.

BNs with DGs lost much of their ability to explore critical patterns too, which also resulted in a much worse performance at a selection rate of 1%.

3.2 Prediction of Joint Behavior

In some problems of interest for Brazil's Federal Revenue it is important to answer the following question: what do two or more actors tend to do when they act together? When BNs are used to model such problems, their structure tend to follow the sketch in Figure 6.

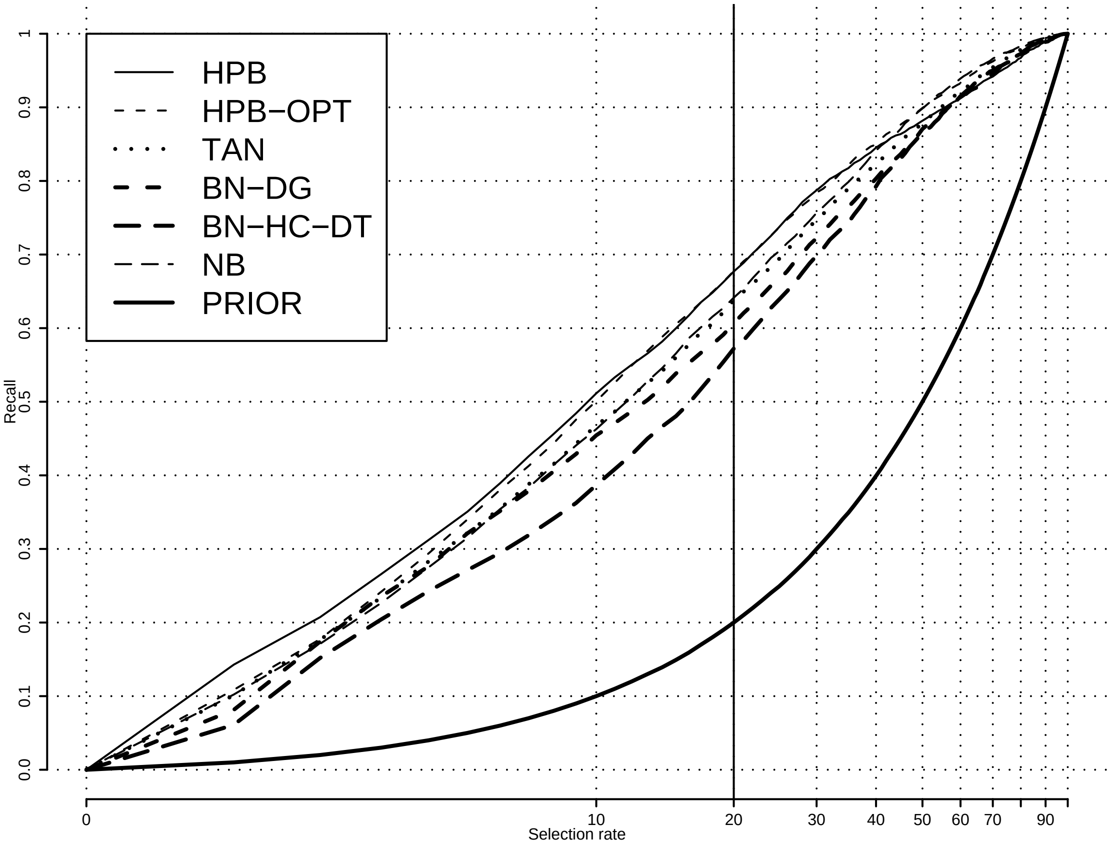

Figure 5: Misclassification detection with cardinality reduction - hit curves (to avoid pollution we only present curves related to a subset of the tested methods)

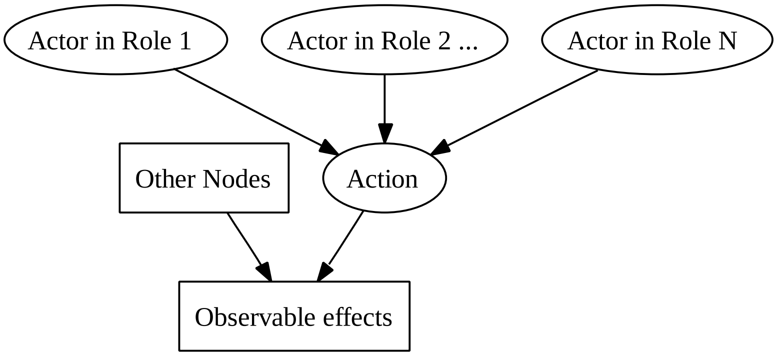

Figure 6: actors Bayesian network

	AUC	AUC20	RMSE($\times 10^4$)	MCE($\times 10^4$)	TR(s)	TS(s)
HPB	81.51±0.72	48.07±1.43	1037.82±3.51	385.10±4.59	8.30±0.07	5.74±0.03
HPB-OPT	82.16±0.85	47.28±1.44	956.09±1.06	350.67±1.98	!148.66±2.73	!6.32±0.02
TAN	!80.27±0.61	!44.21±1.15	!1103.69±5.84	!419.30±5.32	!18.40±0.53	1.40±0.02
ADE	!75.90±0.52	!35.05±1.20	953.39±1.38	354.56±2.25	2.96±0.02	0.69±0.01
DE	!75.85±0.48	!34.45±0.47	!1967.97±7.53	!914.90±6.01	!17.90±0.14	0.70±0.03
BN-DG	!78.98±0.84	!42.59±1.18	!1064.32±7.84	!393.07±8.34	!33.35±1.11	!7.84±0.09
BN-HC-DT	!77.56±0.87	!37.72±1.73	!1065.34±6.17	!393.08±1.54	!154.79±9.83	!21.37±0.80
BN-HC-DF	!77.09±0.68	!36.75±0.99	!1058.77±3.53	389.11±4.16	!234.64±56.99	!7.80±0.11
NB	81.11±0.84	!44.24±1.53	!1142.89±6.04	!429.37±6.61	!16.78±0.11	0.45±0.01
Noisy-Or	!80.11±0.84	!42.15±1.48	!1122.98±5.38	!inf±0.00	!16.65±0.11	0.44±0.02
PRIOR	!50.48±0.01	!10.48±0.01	963.96±0.00	383.27±0.00	!17.42±1.75	0.44±0.01

Table 5: Misclassification detection with cardinality reduction - other measures

Since the number of possible actors can be very big, but the number of roles is usually small, it seems reasonable to replace the CPT of the *Action* node in Figure 6 with HPB. However, in Section 3.1, HPB was used to predict a binary class. The number of possible actions can be high, so we have a different challenge for HPB.

In this section, we present the performance of HPB in a standalone classification problem which was built to resemble the problem of calculating the prior probability distribution of the *Action* node in Figure 6. We used two high cardinality explanatory attributes: the *importer* (IMP) and the *exporter* (EXP)[4] to predict another high cardinality variable, the *declared custom code* (DCC). Note that we are not predicting if there is a misclassification or not, but the DCC itself.

The importer attribute can assume 18846 values, the exporter attribute can assume 43880 values and the declared custom code can assume 7608 values. There are no missing values.

The tested methods were:

- *HPB*: BN with a *direct BN structure* and HPB with fixed *S* coefficients;
- *NB*: naïve Bayes;
- *ADE*: *almost direct estimation*. BN with a *direct BN structure* and the smoothing schema described by Friedman et al. (1997);
- *DE*: *direct estimation*. BN with a *direct BN structure* and traditional CPTs;
- *DE Imp*: *direct estimation*, ignoring the exporter attribute;
- *DE Exp*: *direct estimation*, ignoring the importer attribute.

HPB-OPT was not tested because its optimization process requires a binary class variable. We did not test DGs, DFs and DTs because the combination of high cardinality parents and a high cardinality child makes them too slow.

[4] The *exporter* attribute was not available when we ran the tests in Section 3.1, so we did not use it there.

The parameters for each method were chosen as in Section 3.1, but MCE was used as the selection criterion. Below we present the initial search intervals and the optimal constants ($s = S/NumClasses = S/7608$):

- *HPB*: The *SI* for the s constant was $[1e-4, 1e-03]$ and the optimal value for s was equal to $1e-3$. The SI for the b constant was $[0.5, 2.5]$ and the optimal value for b was equal to 1.0. S^{NI} was always set to be equal to S;

- *NB*: $SI = [1e-3, 2.5]$, $s = 0.05$;

- *ADE*: $SI = [1e-3, 2.5]$, $s = 1e-3$;

- *DE*: $SI = [1e-3, 2.5]$, $s = 1e-3$;

- *DE Imp*: $SI = [1e-3, 2.5]$, $s = 1e-3$;

- *DE Exp*: $SI = [1e-3, 2.5]$, $s = 2.5e-3$;

Table 6 shows that HPB is the best method with regard to RMSE, MCE and number of correct predictions (NC). The difference is significant with the exception that HPB was not significantly better than NB with respect to number of correct assignments.

	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
HPB	108.37 ± 0.02	8.31 ± 0.00	26882.40 ± 89.76	19.51 ± 0.06	35.48 ± 0.17	1800.73 ± 2.99
DE	!108.88 ± 0.02	!9.37 ± 0.01	!25796.20 ± 63.73	!18.72 ± 0.04	1.87 ± 0.05	39.82 ± 0.03
ADE	!108.87 ± 0.01	!8.78 ± 0.01	!26039.20 ± 58.11	!18.90 ± 0.04	2.13 ± 0.01	46.17 ± 0.06
DE Exp	!108.91 ± 0.01	!9.07 ± 0.01	!25257.60 ± 64.14	!18.33 ± 0.04	2.95 ± 0.15	77.03 ± 7.94
DE Imp	!110.42 ± 0.01	!8.95 ± 0.01	!22077.60 ± 90.97	!16.02 ± 0.06	3.24 ± 0.20	73.50 ± 0.51
NB	!111.89 ± 0.05	!9.23 ± 0.01	26803.00 ± 118.12	19.45 ± 0.08	4.01 ± 0.19	357.24 ± 1.18

Table 6: Prediction of joint behavior

3.3 HPB as a General Replacement for CPTs of Bayesian Networks

In this section we test HPB over UCI data sets. Our goal is to observe its performance in domains whose characteristics are different from the ones which inspired its design. We evaluated the performance of Bayesian networks where the usual CPTs were replaced with HPB models. For comparison we also evaluated BNs using other representations for the conditional probability distribution (CPD) of a node given its parents . Below we list all tested CPD representations:

- *HPB*: HPB as described in Section 2.4;

- *DE*: direct estimation, that is, traditional CPTs;

- *ADE*: almost direct estimation. Also CPTs but using the smoothing strategy presented by Friedman et al. (1997);

- *DG*: decision graphs as presented by Chickering et al. (1997);

- *DT*: decision trees as presented by Friedman et al. (1997);

- *DF*: default tables as presented by Friedman et al. (1997).

In all cases, we learned the global BN structure using the hill climbing search algorithm implemented in Weka 3.4.2 and used NB as the starting point. To guarantee that we would not have excessively long execution times we limited the maximum number of parents to 10 and because HPB does not handle continuous attributes we removed them all. We also removed all instances with missing attributes.

Depending on the chosen representation for the CPDs, we employed different scoring metrics in the BN structure search. Below we list our choices:

- HPB: log-likelihood evaluated using leave-one-out cross validation;
- DE: MDL;
- ADE: MDL;
- DGs: Bayesian Dirichlet scoring metric as presented by Chickering et al. (1997);
- DTs: MDL as presented by Friedman and Goldszmidt (1996b);
- DFs: MDL as presented by Friedman and Goldszmidt (1996b).

The tested data sets were: *anneal*, *audiology*, *autos*, *breast-cancer*, *horse-colic*, *credit-rating*, *german-credit*, *cleveland-14-heart-disease*, *hungarian-14-heart-disease*, *hepatitis*, *hypothyroid*, *kr-vs-kp*, *labor*, *lymphography*, *mushroom*, *primary-tumor*, *sick*, *soybean*, *vote* and *zoo*.

Before building a BN we decided on fixed equivalent sample size for the prior probability distributions (this means a fixed S constant) and used it for all HPB's instances inside the BN. Fortunately, the optimal values for the equivalent sample sizes tend to be similar.

We chose S starting from $SI = [1.0, 25.0]$ and, forced the S^{NI} be identical to the S. The b constant was chosen starting from $SI = [0.5, 2.5]$.

We chose the s constant ($s = S/NumClasses$) for DGs starting from $SI = [0.01, 2.5]$. We always kept complete splits enabled and exhaustively varied the activation state of binary splits and merges. We chose the s constant for the other methods starting from $SI = [0.01, 2.5]$.

In contrast to Sections 3.1 and 3.2 we did not expand the initial search intervals if the optimal value for a constant turned out to be in one of its extreme points.

We compared the results using three criteria: number of correct classifications (NC), mean cross entropy (MCE) and root mean squared error (RMSE). To save space we present only the numbers of times where each method resulted in the best average performance. Since selecting the best parameterization for each method using a criterion and comparing the methods using only the same criterion would possibly not provide the reader enough information, we selected parameterizations using all three criteria and compared the methods using also all the three criteria in exhaustive combinations. In some cases, two or more classifiers resulted in exactly the same value for NC. In these cases, if NC was the comparison criterion, we used MCE to decide who was the winner. Results are in Table 7. Details are available in appendix A.

Sel.Crit.	Comp.Crit.	HPB	DG	DF	DT	ADE	DE
NC	NC	9	6	2	1	1	1
NC	MCE	9	5	2	1	2	1
NC	RMSE	7	6	4	1	1	1
MCE	NC	9	5	2	0	3	1
MCE	MCE	10	5	2	1	0	2
MCE	RMSE	8	6	4	0	1	1
RMSE	NC	7	7	4	0	1	1
RMSE	MCE	9	5	2	1	1	2
RMSE	RMSE	8	6	4	0	1	1

Table 7: Number of winning results in UCI data sets

HPB	DG	DF	DT	ADE	DE
3.18	3.79	1.49	1.56	1.0	1.0

Table 8: Proportions among the number of arcs of BN structures

In Table 8 we show the average proportions between the number of arcs in the BN structures learned using each CPD representation and the BN structures learned using direct estimation (traditional CPTs). We can see that, as predicted by Friedman and Goldszmidt (1996b), the use of structures like DFs, DTs and DGs does result in BNs with more arcs. The use of HPB has a similar effect.

As shown in Section 3.1, HPB is much faster than DGs, DTs and DFs in the task of handling a small set of high cardinality explanatory attributes. However, in UCI tests, many BN structures involved sets of low cardinality parents. This makes HPB comparatively slow. HPB was, in all cases, the slowest method and in some of them more than 10 times slower than the second slowest method.

Moreover, the advantage of HPB in all three criteria (MCE, RMSE and NC) was rarely statistically significant. Thus, we cannot recommend HPB as a general replacement for CPTs.

However, the vast majority of variables in the tested data sets have low cardinality (the highest cardinality variable among all data sets is the audiology class variable with 24 possible values) and many of them are binary. In spite of this, HPB's are clearly the best results in Table 7 showing that good prior probability distributions, can, many times, improve the quality of predictions.

We can say that a BN where CPDs are represented using HPB has a quite high probability of producing better classification predictions than BNs employing other CPD representations. The only explanation we found for this fact is that HPB represents some CPDs better than its alternatives and that such better representations result in BNs with better classification predictions, even when the characteristics of the attributes are opposite to the ones that inspired HPB.

This suggests that it should not be difficult to find problems where, if a BN is employed, there will be one or more BN nodes where it will be worth using HPB.

4. Conclusions

We presented HPB a novel multilevel empirical hierarchical Bayesian model, which is intended to replace conditional probability tables of Bayesian network nodes whose parents have high cardinality.

We presented HPB in two versions. The first version involves an optimization process to choose the best smoothing coefficients for each family of patterns, while the second and simpler version employs a fixed smoothing coefficient. We prefer the simpler version because it is much faster and can handle non binary child nodes.

We evaluated HPB in the domain of preselection of imported goods for human verification using hit curves, RMSE and MCE. In this domain, interactions among attributes have a great influence over the probability of finding a positive instance of misclassification, but due to the high cardinality of the attributes in this domain, exploiting such interactions is challenging.

Even the simpler version of HPB was shown capable of capturing the influence of interactions among high cardinality attributes and achieved performance improvements over standard Bayesian network methods like naïve Bayes and tree augmented naïve Bayes, over Bayesian networks where traditional conditional probability tables were substituted by noisy-OR gates, default tables, decision trees and decision graphs, and over Bayesian networks constructed after a cardinality reduction preprocessing phase using the agglomerative information bottleneck method.

HPB's execution time is exponential in the number of parents of a BN node but independent of their cardinality. Since the number of parents of a BN node is almost always small, for nodes whose parents have high cardinality, HPB, at least when its smoothing coefficients are fixed, is much faster than default tables, decision trees or decision graphs when employed to represent the conditional probability distribution of the node given its parents. This version of HPB uses the training set only through frequencies, thus data can be added dynamically without any retraining procedures other than some frequency increments.

We tested HPB in another classification problem: the prediction of the behavior of two actors when they act together. As a subproblem, this prediction is relevant in several fraud detection domains and, if the general problem is modeled as a BN, generally appears as the the task of representing the CPD of a particular node given its parents. The results, again, favored HPB.

We also provide experimental results over UCI data sets, where Bayesian network classifiers with different CPD representations are compared. Despite the fact that these data sets do not include high cardinality attributes, HPB was the representation that resulted in more winnings than any other representation in three comparison measures. The comparatively large execution times and the fact that most differences in comparison measures were not significant, do not allow us to propose HPB as a general replacement for CPTs. However, we can still conclude that BN nodes whose CPDs given their parents are better represented by HPB than by other methods are not rare. This fact indicates that HPB may have a quite wide applicability.

HPB can be very useful in practice. If specialists are handcrafting a Bayesian network structure, they want it to reflect the structure of the target problem. If this results in a node with high cardinality parents, they can just use HPB as a plug-in replacement for the CPT of the node and keep the structure they want. Without a method like HPB the high cardinality parents could easily result in unreliable probability estimates that could compromise the whole model. The specialists would have to accept a BN structure that would not reflect the target problem as closely as the original one, but which would avoid the use of high cardinality nodes as parents of the same node.

Full hierarchical Bayesian models have been widely used in the marketing community under the name of Hierarchical Bayes (Allenby et al., 1999; Lenk et al., 1996). These models have also been used in medical domains (Andreassen et al., 2003) and robotics (Stewart et al., 2003). However, we are not aware of any hierarchical Bayesian model that can replace conditional probability tables of Bayesian network nodes whose parents have high cardinality. Moreover, HPB deals with a multi level hierarchy recursively and also handles the fact that the population of instances associated to each pattern is contained by several overlapping superpopulations and not by a single one. It would be very difficult to build a full hierarchical Bayesian model that can do the same.

As future work we leave the development of better mechanisms to select HPB coefficients. Both optimization processes and heuristics should be considered. The first for the most reliable predictions and the last for fast and acceptable ones.

The pattern hierarchy employed by HPB is fixed, symmetrical (all attributes are treated the same way) and complete (all subsets of each pattern of interest are considered in the calculation of the probability of a class given the pattern). It is possible that there exists an incomplete, possibly asymmetrical hierarchy that would lead to better results. Developing an algorithm to search for such hierarchy is also left as future work.

We compared HPB against algorithms which employ the best default tables, decision trees and decision graphs chosen using some criterion. If instead of this we employed mixtures of probability tables (Fujimoto and Murata, 2006) where default tables, decision trees or decision graphs were used as category integration tables, results could be better. As a final future work we leave the development of an algorithm that can build such a mixture model and the comparison of its results to HPB's ones.

Acknowledgments

This work is part of the HARPIA project and is supported by Brazil's Federal Revenue.

Appendix A. Detailed Results over UCI Data Sets

In this appendix we detail the results of our tests over UCI data sets (see Section 3.3). To save space we only present results where the number of correct classifications (NC) was used to select the best parameterization for each method. The methods appear in the tables in decreasing order of NC. In some cases, two or more classifiers resulted in exactly the same value for NC. In these cases, we used MCE to decide the order. Results are in Table 9, Table 10 and Table 11.

ANNEAL	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-BDG	817.69 ± 123.19	199.99 ± 86.45	175.00 ± 2.54	97.43 ± 1.22	18.21 ± 0.75	0.04 ± 0.00
BN-HC-HPB	820.43 ± 69.84	182.20 ± 34.07	174.80 ± 1.09	97.32 ± 0.46	!314.35 ± 62.57	!6.24 ± 1.72
BN-HC-DF	!1289.10 ± 117.59	!469.69 ± 86.02	!168.60 ± 2.07	!93.87 ± 1.11	2.56 ± 0.08	0.02 ± 0.00
BN-HC-DT	!1486.14 ± 122.19	!610.11 ± 117.16	!166.00 ± 2.34	!92.42 ± 1.22	8.74 ± 0.36	0.04 ± 0.00
NB	!1449.24 ± 131.34	!623.72 ± 158.70	!165.00 ± 1.58	!91.87 ± 0.75	0.06 ± 0.00	0.00 ± 0.00
BN-HC-DE	!1565.84 ± 142.88	!625.42 ± 115.93	!162.60 ± 3.28	!90.53 ± 1.95	2.09 ± 0.10	0.01 ± 0.00
BN-HC-ADE	!1529.27 ± 144.05	!604.07 ± 105.71	!159.19 ± 6.14	!88.63 ± 3.28	2.09 ± 0.09	0.02 ± 0.00
AUDIOLOGY	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-HPB	1062.27 ± 164.74	443.77 ± 168.94	37.40 ± 2.19	82.75 ± 5.01	1207.20 ± 207.25	19.39 ± 4.30
NB	1199.52 ± 141.70	!1034.56 ± 355.41	35.60 ± 2.50	78.76 ± 5.56	0.05 ± 0.00	0.00 ± 0.00
BN-HC-BDG	1200.72 ± 236.04	608.81 ± 271.34	35.40 ± 4.09	78.32 ± 9.18	64.59 ± 6.16	0.06 ± 0.00
BN-HC-DF	!1343.05 ± 121.85	!728.90 ± 167.44	!32.20 ± 2.58	!71.23 ± 5.67	17.60 ± 1.97	0.04 ± 0.00
BN-HC-DE	!1521.41 ± 39.31	!906.80 ± 44.24	!28.00 ± 0.70	!61.95 ± 1.68	14.18 ± 0.44	0.03 ± 0.00
BN-HC-ADE	!1521.01 ± 39.43	!927.73 ± 51.99	!28.00 ± 0.70	!61.95 ± 1.68	14.22 ± 0.51	0.04 ± 0.00
BN-HC-DT	!1559.49 ± 49.27	!970.53 ± 52.04	!26.40 ± 1.67	!58.39 ± 3.43	13.98 ± 1.06	0.07 ± 0.00
AUTOS	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-HPB	2704.17 ± 150.42	2021.03 ± 232.97	26.40 ± 1.67	64.39 ± 4.08	1.26 ± 0.34	0.09 ± 0.01
BN-HC-BDG	2804.60 ± 156.24	!2484.20 ± 367.24	26.40 ± 1.81	64.39 ± 4.43	0.70 ± 0.07	0.00 ± 0.00
NB	2806.48 ± 157.29	2296.88 ± 433.79	!24.80 ± 0.83	!60.48 ± 2.04	0.06 ± 0.00	0.00 ± 0.00
BN-HC-DF	!3037.27 ± 123.34	!2518.42 ± 245.10	!20.60 ± 2.19	!50.24 ± 5.34	0.16 ± 0.01	0.00 ± 0.00
BN-HC-ADE	!3124.67 ± 98.05	!2682.00 ± 226.16	!18.60 ± 2.19	!45.36 ± 5.34	0.12 ± 0.00	0.00 ± 0.00
BN-HC-DE	!3124.67 ± 98.03	!2682.11 ± 226.19	!18.60 ± 2.19	!45.36 ± 5.34	0.12 ± 0.00	0.00 ± 0.00
BN-HC-DT	!3124.67 ± 98.03	!2682.11 ± 226.19	!18.60 ± 2.19	!45.36 ± 5.34	0.20 ± 0.00	0.00 ± 0.00
BREAST-CANCER	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-BDG	4401.36 ± 208.72	4164.95 ± 318.23	42.60 ± 1.81	74.47 ± 3.16	0.18 ± 0.04	0.00 ± 0.00
NB	4429.70 ± 690.84	4467.29 ± 1408.44	42.20 ± 4.32	73.79 ± 7.75	0.05 ± 0.00	0.00 ± 0.00
BN-HC-HPB	4511.98 ± 380.22	4519.62 ± 716.09	42.20 ± 2.77	73.78 ± 5.03	!0.49 ± 0.03	!0.01 ± 0.00
BN-HC-DF	4454.83 ± 217.54	4240.36 ± 327.92	41.80 ± 1.78	73.07 ± 2.96	0.14 ± 0.00	0.00 ± 0.00
BN-HC-DT	4470.69 ± 281.55	4247.27 ± 402.75	40.20 ± 4.32	70.27 ± 7.54	!0.22 ± 0.01	0.00 ± 0.00
BN-HC-DE	4485.58 ± 234.38	4284.39 ± 352.75	39.40 ± 3.78	68.88 ± 6.60	0.12 ± 0.00	0.00 ± 0.00
BN-HC-ADE	4488.31 ± 248.77	4291.26 ± 377.44	39.40 ± 3.78	68.88 ± 6.60	0.12 ± 0.00	0.00 ± 0.00
HORSE-COLIC	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-HPB	3496.97 ± 528.93	2962.44 ± 762.94	62.00 ± 3.39	84.23 ± 4.45	4.35 ± 1.80	0.05 ± 0.00
BN-HC-DT	3580.15 ± 682.03	3689.41 ± 1470.42	62.00 ± 4.41	84.23 ± 5.87	0.51 ± 0.04	0.00 ± 0.00
BN-HC-BDG	3557.76 ± 367.29	3156.62 ± 608.12	61.60 ± 3.20	83.69 ± 4.36	2.49 ± 0.48	0.00 ± 0.00
BN-HC-DF	3488.76 ± 560.79	3072.05 ± 1039.40	61.20 ± 4.08	83.15 ± 5.60	0.31 ± 0.01	0.00 ± 0.00
BN-HC-DE	3630.07 ± 699.88	3907.53 ± 1591.14	61.20 ± 3.96	83.14 ± 5.22	0.21 ± 0.00	0.00 ± 0.00
BN-HC-ADE	3648.28 ± 683.27	3900.65 ± 1545.75	60.80 ± 3.70	82.60 ± 4.92	0.21 ± 0.00	0.00 ± 0.00
NB	3951.10 ± 596.02	!4951.78 ± 1805.65	60.40 ± 3.57	82.06 ± 4.81	0.05 ± 0.00	0.00 ± 0.00
CREDIT-RATING	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-BDG	3115.11 ± 216.95	2423.92 ± 284.79	120.80 ± 3.19	87.53 ± 2.31	0.69 ± 0.17	0.00 ± 0.00
BN-HC-HPB	3245.36 ± 289.67	2578.43 ± 364.32	120.20 ± 4.14	87.10 ± 3.00	1.05 ± 0.46	!0.06 ± 0.02
BN-HC-DE	3220.08 ± 205.71	2571.82 ± 353.12	119.20 ± 2.94	86.37 ± 2.13	0.15 ± 0.00	0.00 ± 0.00
BN-HC-ADE	3221.82 ± 212.88	2577.11 ± 357.98	119.20 ± 2.94	86.37 ± 2.13	0.16 ± 0.00	0.00 ± 0.00
BN-HC-DF	3224.01 ± 122.88	2527.08 ± 166.69	118.60 ± 1.67	85.94 ± 1.21	0.19 ± 0.00	0.00 ± 0.00
BN-HC-DT	3213.24 ± 224.72	2529.39 ± 292.68	118.60 ± 3.28	85.94 ± 2.38	0.36 ± 0.02	0.00 ± 0.00
NB	3304.10 ± 195.52	2754.44 ± 278.50	118.60 ± 1.67	85.94 ± 1.21	0.06 ± 0.00	0.00 ± 0.00
GERMAN-CREDIT	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
NB	4167.54 ± 164.17	3789.33 ± 258.83	149.80 ± 7.39	74.89 ± 3.69	0.06 ± 0.00	0.00 ± 0.00
BN-HC-DF	4166.08 ± 96.36	3788.58 ± 106.86	148.60 ± 6.14	74.30 ± 3.07	!0.40 ± 0.02	!0.00 ± 0.00
BN-HC-HPB	4244.67 ± 165.79	3934.53 ± 320.17	147.80 ± 2.94	73.90 ± 1.47	!4.69 ± 1.83	!0.13 ± 0.03
BN-HC-DT	4216.53 ± 111.46	3851.92 ± 172.64	147.60 ± 4.87	73.80 ± 2.43	!1.06 ± 0.03	!0.00 ± 0.00
BN-HC-BDG	4228.17 ± 121.41	3882.09 ± 227.78	146.60 ± 5.22	73.30 ± 2.61	!1.10 ± 0.18	0.00 ± 0.00
BN-HC-ADE	4230.48 ± 126.66	3876.35 ± 199.32	145.60 ± 6.02	72.80 ± 3.01	!0.37 ± 0.01	0.00 ± 0.00
BN-HC-DE	4223.58 ± 123.27	3855.98 ± 192.26	145.40 ± 6.84	72.70 ± 3.42	!0.36 ± 0.01	0.00 ± 0.00
CLEVELAND-14-HEART-DISEASE	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-DF	2339.91 ± 113.39	1296.49 ± 119.09	49.80 ± 1.78	82.17 ± 2.72	0.09 ± 0.00	0.00 ± 0.00
BN-HC-BDG	2366.36 ± 154.85	1333.08 ± 182.49	49.60 ± 2.07	81.84 ± 3.28	!0.12 ± 0.01	0.00 ± 0.00
BN-HC-HPB	2381.80 ± 136.95	1299.65 ± 95.18	49.40 ± 1.14	81.51 ± 1.44	!0.20 ± 0.00	!0.05 ± 0.02
NB	2354.95 ± 182.24	1334.42 ± 181.36	49.40 ± 2.40	81.50 ± 3.61	0.06 ± 0.00	0.00 ± 0.00
BN-HC-DT	2362.61 ± 152.32	1338.52 ± 166.44	49.00 ± 2.54	80.84 ± 3.70	!0.13 ± 0.00	0.00 ± 0.00
BN-HC-DE	!2487.86 ± 110.45	!1495.27 ± 134.98	47.60 ± 2.70	78.54 ± 4.22	0.09 ± 0.00	0.00 ± 0.00
BN-HC-ADE	!2488.44 ± 116.62	1428.83 ± 142.14	47.40 ± 2.88	78.20 ± 4.46	0.09 ± 0.00	0.00 ± 0.00
HUNGARIAN-14-HEART-DISEASE	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-HPB	2186.65 ± 256.16	1278.47 ± 198.23	49.20 ± 2.58	83.69 ± 4.82	0.30 ± 0.05	0.04 ± 0.00
NB	2184.21 ± 410.39	1163.94 ± 327.13	48.20 ± 4.43	82.01 ± 8.05	0.05 ± 0.00	0.00 ± 0.00
BN-HC-ADE	2289.01 ± 475.44	1251.44 ± 402.26	47.60 ± 4.39	80.98 ± 7.93	0.08 ± 0.00	0.00 ± 0.00
BN-HC-DE	2299.81 ± 486.99	1275.80 ± 426.23	47.60 ± 4.39	80.98 ± 7.93	0.08 ± 0.00	0.00 ± 0.00
BN-HC-BDG	2290.08 ± 331.60	1319.64 ± 275.60	47.40 ± 3.50	80.62 ± 6.13	0.26 ± 0.02	0.00 ± 0.00
BN-HC-DF	2305.03 ± 411.21	1275.65 ± 332.30	46.60 ± 4.15	79.28 ± 7.55	0.08 ± 0.00	0.00 ± 0.00
BN-HC-DT	2330.69 ± 360.65	1278.32 ± 273.29	46.40 ± 4.87	78.94 ± 8.72	0.12 ± 0.00	0.00 ± 0.00

Table 9: Comparisons over UCI data sets

HEPATITIS	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-ADE	3337.13 ± 410.16	2722.43 ± 691.25	26.60 ± 0.89	85.80 ± 2.88	0.15 ± 0.01	0.00 ± 0.00
BN-HC-DT	3343.89 ± 668.03	2800.25 ± 1027.57	26.60 ± 1.67	85.80 ± 5.39	!0.26 ± 0.01	0.00 ± 0.00
BN-HC-DE	3369.93 ± 385.56	2767.97 ± 708.74	26.40 ± 1.51	85.16 ± 4.89	0.14 ± 0.00	0.00 ± 0.00
NB	3605.14 ± 587.26	3782.49 ± 1160.16	26.40 ± 1.51	85.16 ± 4.89	0.06 ± 0.00	0.00 ± 0.00
BN-HC-HPB	3354.12 ± 303.11	2587.67 ± 413.88	26.20 ± 0.83	84.51 ± 2.69	!0.65 ± 0.07	!0.01 ± 0.00
BN-HC-BDG	3580.20 ± 601.23	3797.96 ± 1617.52	26.20 ± 1.30	84.51 ± 4.20	!0.33 ± 0.01	0.00 ± 0.00
BN-HC-DF	3540.69 ± 455.63	3069.03 ± 1017.49	25.80 ± 0.83	83.22 ± 2.69	0.16 ± 0.01	0.00 ± 0.00
HYPOTHYROID	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-HPB	1894.74 ± 13.90	1117.79 ± 33.44	696.20 ± 0.44	92.28 ± 0.05	395.15 ± 41.22	4.80 ± 0.36
BN-HC-DF	1895.51 ± 6.60	1130.20 ± 25.31	696.20 ± 0.44	92.28 ± 0.05	4.25 ± 0.04	0.04 ± 0.00
BN-HC-ADE	1898.35 ± 7.58	1135.40 ± 16.14	696.20 ± 0.44	92.28 ± 0.05	4.25 ± 0.14	0.04 ± 0.00
BN-HC-DE	1898.35 ± 7.58	1135.44 ± 16.11	696.20 ± 0.44	92.28 ± 0.05	4.25 ± 0.12	0.03 ± 0.00
BN-HC-DT	1906.34 ± 14.12	!1159.86 ± 26.54	696.20 ± 0.44	92.28 ± 0.05	22.10 ± 0.86	0.09 ± 0.00
BN-HC-BDG	1888.97 ± 20.46	1106.31 ± 51.51	696.00 ± 0.70	92.25 ± 0.07	35.19 ± 1.43	0.07 ± 0.00
NB	1895.38 ± 11.11	1134.41 ± 31.89	696.00 ± 0.70	92.25 ± 0.11	0.06 ± 0.00	0.00 ± 0.00
KR-VS-KP	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-BDG	1301.83 ± 192.75	505.85 ± 125.79	624.60 ± 4.61	97.71 ± 0.75	161.41 ± 4.41	0.05 ± 0.00
BN-HC-HPB	1404.34 ± 203.37	574.37 ± 157.82	624.00 ± 4.74	97.62 ± 0.69	!2241.20 ± 90.05	!15.30 ± 0.95
BN-HC-DF	!1780.81 ± 276.91	!894.29 ± 172.35	!615.20 ± 8.64	!96.24 ± 1.35	20.74 ± 1.43	0.04 ± 0.00
BN-HC-DT	!1696.09 ± 153.76	!727.22 ± 133.38	!614.79 ± 2.68	!96.18 ± 0.46	!187.80 ± 11.61	!0.10 ± 0.00
BN-HC-DE	!1889.40 ± 142.39	!959.52 ± 142.96	!612.00 ± 6.00	!95.74 ± 0.90	14.04 ± 0.32	0.03 ± 0.00
BN-HC-ADE	!1888.02 ± 135.00	!931.64 ± 141.25	!611.20 ± 4.26	!95.61 ± 0.62	14.05 ± 0.34	0.03 ± 0.00
NB	!3022.02 ± 171.19	!2104.82 ± 196.02	!560.60 ± 10.23	!87.70 ± 1.54	0.06 ± 0.00	0.00 ± 0.00
LABOR	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-DT	2592.18 ± 1284.37	2068.30 ± 1725.01	10.40 ± 0.89	91.36 ± 8.73	0.09 ± 0.00	0.00 ± 0.00
BN-HC-HPB	2914.13 ± 937.77	2208.65 ± 1031.56	10.40 ± 0.89	91.36 ± 8.73	!0.26 ± 0.05	0.00 ± 0.00
BN-HC-DF	2695.25 ± 1514.59	2493.34 ± 2032.41	10.40 ± 0.89	91.36 ± 8.73	0.08 ± 0.00	0.00 ± 0.00
BN-HC-ADE	3005.58 ± 1228.36	2617.90 ± 1746.11	10.40 ± 0.89	91.36 ± 8.73	0.08 ± 0.02	0.00 ± 0.00
BN-HC-BDG	2836.04 ± 1381.80	2624.67 ± 2036.46	10.40 ± 0.89	91.36 ± 8.73	!0.14 ± 0.01	0.00 ± 0.00
BN-HC-DE	2763.72 ± 1490.36	2699.29 ± 2374.92	10.40 ± 0.89	91.36 ± 8.73	0.07 ± 0.00	0.00 ± 0.00
NB	2526.04 ± 1401.66	1968.36 ± 1567.61	10.00 ± 1.00	87.87 ± 9.65	0.06 ± 0.00	0.00 ± 0.00
LYMPHOGRAPHY	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-HPB	2512.72 ± 387.81	1824.09 ± 591.88	25.80 ± 2.16	87.12 ± 6.69	3.33 ± 1.62	0.06 ± 0.02
NB	2380.47 ± 450.03	1486.82 ± 544.14	25.60 ± 1.67	86.43 ± 4.36	0.06 ± 0.00	0.00 ± 0.00
BN-HC-BDG	2662.61 ± 659.37	2700.19 ± 1378.11	25.20 ± 2.48	85.10 ± 7.77	0.62 ± 0.12	0.00 ± 0.00
BN-HC-ADE	2754.23 ± 276.46	1944.44 ± 368.85	24.40 ± 1.94	82.36 ± 5.28	0.23 ± 0.06	0.00 ± 0.00
BN-HC-DE	2687.32 ± 344.42	1864.19 ± 420.07	24.20 ± 2.38	81.67 ± 6.86	0.18 ± 0.01	0.00 ± 0.00
BN-HC-DF	2718.45 ± 502.66	2131.58 ± 942.79	24.20 ± 1.92	81.70 ± 5.41	0.22 ± 0.01	0.00 ± 0.00
BN-HC-DT	2681.11 ± 276.46	1879.54 ± 408.83	!23.60 ± 1.14	!79.70 ± 2.65	0.41 ± 0.03	0.00 ± 0.00
MUSHROOM	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-DE	0.01 ± 0.01	0.00 ± 0.00	1624.80 ± 0.44	100.00 ± 0.00	10.50 ± 0.23	0.05 ± 0.00
BN-HC-ADE	0.01 ± 0.02	0.00 ± 0.00	1624.80 ± 0.44	100.00 ± 0.00	10.52 ± 0.26	0.05 ± 0.00
BN-HC-DF	0.08 ± 0.09	0.00 ± 0.00	1624.80 ± 0.44	100.00 ± 0.00	!14.70 ± 0.16	!0.07 ± 0.00
BN-HC-HPB	0.10 ± 0.13	0.00 ± 0.00	1624.80 ± 0.44	100.00 ± 0.00	!240.65 ± 50.90	!2.20 ± 0.20
BN-HC-BDG	!0.41 ± 0.24	!0.03 ± 0.01	1624.80 ± 0.44	100.00 ± 0.00	!76.26 ± 4.26	!0.09 ± 0.00
BN-HC-DT	28.28 ± 54.93	0.80 ± 1.36	1624.60 ± 0.54	99.98 ± 0.02	!114.42 ± 4.39	!0.22 ± 0.00
NB	!876.14 ± 56.96	!216.17 ± 36.18	!1609.00 ± 2.91	!99.02 ± 0.17	0.06 ± 0.00	0.00 ± 0.00
PRIMARY-TUMOR	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-HPB	1786.01 ± 53.39	1218.08 ± 52.72	32.60 ± 2.70	48.09 ± 4.08	11.06 ± 8.75	0.72 ± 0.04
NB	1792.85 ± 50.38	1296.85 ± 122.63	32.20 ± 2.28	47.48 ± 3.27	0.06 ± 0.00	0.00 ± 0.00
BN-HC-DF	1810.11 ± 71.59	1290.67 ± 124.59	29.80 ± 2.16	43.93 ± 2.93	0.40 ± 0.06	0.01 ± 0.00
BN-HC-BDG	!1933.29 ± 66.50	!1978.04 ± 173.90	!28.60 ± 3.50	!42.16 ± 4.98	7.59 ± 0.67	0.03 ± 0.00
BN-HC-ADE	!1983.01 ± 14.42	!1561.17 ± 33.06	!17.00 ± 0.70	!25.07 ± 1.05	0.46 ± 0.11	0.01 ± 0.00
BN-HC-DT	!2011.13 ± 2.24	!1670.29 ± 10.79	!16.80 ± 0.44	!24.78 ± 0.71	1.04 ± 0.07	0.03 ± 0.00
BN-HC-DE	!1984.37 ± 15.07	!1565.26 ± 39.21	!16.60 ± 1.14	!24.48 ± 1.71	0.47 ± 0.10	0.01 ± 0.00
SICK	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-BDG	2273.43 ± 31.56	1364.06 ± 55.35	708.40 ± 0.54	93.90 ± 0.09	34.73 ± 3.78	0.03 ± 0.00
BN-HC-HPB	2268.98 ± 21.24	1338.45 ± 23.39	708.20 ± 0.83	93.87 ± 0.10	!250.84 ± 98.29	!2.08 ± 0.85
BN-HC-ADE	2279.90 ± 16.33	1351.24 ± 43.58	708.20 ± 0.44	93.87 ± 0.05	4.15 ± 0.18	0.02 ± 0.00
BN-HC-DE	2280.38 ± 15.09	1354.35 ± 44.87	708.20 ± 0.44	93.87 ± 0.05	4.18 ± 0.13	0.01 ± 0.00
BN-HC-DF	2287.45 ± 30.30	1372.02 ± 53.92	708.20 ± 0.44	93.87 ± 0.05	4.24 ± 0.10	0.02 ± 0.00
BN-HC-DT	2283.19 ± 26.33	1346.01 ± 45.16	707.60 ± 0.89	93.79 ± 0.10	20.38 ± 0.78	!0.05 ± 0.00
NB	!2380.36 ± 50.23	!1432.11 ± 53.36	!704.60 ± 2.07	!93.39 ± 0.32	0.06 ± 0.00	0.00 ± 0.00
SOYBEAN	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-BDG	549.80 ± 52.43	75.57 ± 12.73	131.80 ± 1.30	96.48 ± 0.80	25.50 ± 1.25	0.10 ± 0.00
BN-HC-HPB	!648.62 ± 46.64	!115.11 ± 10.06	!129.40 ± 1.34	!94.72 ± 0.95	!525.54 ± 202.49	!17.23 ± 4.09
BN-HC-DF	!682.22 ± 91.60	!152.13 ± 82.49	!129.00 ± 2.34	!94.43 ± 1.60	3.42 ± 0.19	0.06 ± 0.00
NB	!730.32 ± 112.99	!267.33 ± 103.20	!128.80 ± 2.77	!94.28 ± 1.76	0.05 ± 0.00	0.00 ± 0.00
BN-HC-DE	!808.74 ± 139.12	!312.35 ± 137.43	!126.40 ± 3.28	!92.52 ± 2.17	1.49 ± 0.03	0.05 ± 0.00
BN-HC-ADE	!808.01 ± 138.81	!322.17 ± 135.10	!126.40 ± 3.28	!92.52 ± 2.17	1.48 ± 0.02	0.05 ± 0.00
BN-HC-DT	!792.47 ± 50.65	!157.82 ± 18.19	!125.20 ± 2.38	!91.65 ± 1.53	14.10 ± 0.95	!0.15 ± 0.00

Table 10: Comparisons over UCI data sets

VOTE	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-HPB	1852.76 ± 389.88	954.62 ± 345.06	83.80 ± 1.92	96.32 ± 2.21	7.96 ± 4.79	0.08 ± 0.03
BN-HC-BDG	2078.01 ± 379.00	!1806.33 ± 890.10	83.00 ± 1.58	95.40 ± 1.81	2.85 ± 0.33	0.00 ± 0.00
BN-HC-DT	1985.64 ± 632.81	1190.27 ± 623.88	82.80 ± 2.38	95.17 ± 2.74	1.82 ± 0.12	0.00 ± 0.00
BN-HC-ADE	2143.81 ± 744.24	1376.41 ± 831.41	82.20 ± 3.27	94.48 ± 3.75	0.40 ± 0.02	0.00 ± 0.00
BN-HC-DE	2119.88 ± 719.66	1417.61 ± 816.58	82.20 ± 2.58	94.48 ± 2.97	0.40 ± 0.01	0.00 ± 0.00
BN-HC-DF	2261.07 ± 621.48	1494.57 ± 821.78	81.59 ± 2.70	93.79 ± 3.10	0.46 ± 0.03	0.00 ± 0.00
NB	!2979.38 ± 540.11	!4398.98 ± 1905.74	!78.59 ± 2.60	!90.34 ± 2.99	0.05 ± 0.00	0.00 ± 0.00
ZOO	RMSE($\times 10^4$)	MCE($\times 10^4$)	NC	PC(%)	TR(s)	TS(s)
BN-HC-HPB	1076.29 ± 416.51	321.46 ± 226.17	19.20 ± 0.83	95.04 ± 3.53	1.15 ± 0.43	0.05 ± 0.01
BN-HC-DF	1000.05 ± 537.16	309.72 ± 236.93	19.00 ± 1.00	94.04 ± 4.19	0.17 ± 0.00	0.00 ± 0.00
NB	1106.38 ± 494.76	369.25 ± 238.98	19.00 ± 0.70	94.09 ± 4.07	0.05 ± 0.00	0.00 ± 0.00
BN-HC-BDG	1088.29 ± 614.67	436.05 ± 358.95	19.00 ± 1.00	94.04 ± 4.19	0.51 ± 0.04	0.00 ± 0.00
BN-HC-ADE	1007.12 ± 410.85	285.87 ± 166.44	18.80 ± 0.83	93.09 ± 4.39	0.14 ± 0.00	0.00 ± 0.00
BN-HC-DE	1108.85 ± 335.92	361.89 ± 143.52	18.80 ± 0.83	93.09 ± 4.39	0.13 ± 0.00	0.00 ± 0.00
BN-HC-DT	1303.27 ± 358.63	431.96 ± 241.95	!18.00 ± 1.00	!89.09 ± 4.21	0.40 ± 0.01	0.00 ± 0.00

Table 11: Comparisons over UCI data sets

References

Greg M. Allenby, Robert P. Leone, and Lichung Jen. A dynamic model of purchase timing with application to direct marketing. *Journal of the American Statistical Association*, 94(446):365–374, 1999.

Steen Andreassen, Brian Kristensen, Alina Zalounina, Leonard Leibovici, Uwe Frank, and Henrik C. Schonheyder. Hierarchical dirichlet learning - filling in the thin spots in a database. In Michel Dojat, Elpida T. Keravnou, and Pedro Barahona, editors, *Proceedings of the 9th Conference on Artificial Intelligence in Medicine (AIME)*, volume 2780 of *Lecture Notes in Computer Science*, pages 204–283. Springer, 2003.

Paul N. Bennett. Assessing the calibration of naive bayes' posterior estimates. Technical Report CMU-CS-00-155, School of Computer Science, Carnegie Mellon University, 2000.

Marc Boullé. A bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research*, 6:1431–1452, 2005.

Bojan Cestnik. Estimating probabilities: a crucial task in machine learning. In *Proceedings of the European Conference on Artificial Intelligence*, pages 147–149, 1990.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence and Research*, 16: 321–357, 2002.

David Maxwell Chickering, David Heckerman, and Christopher Meek. A bayesian approach to learning bayesian networks with local structure. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 80–89, San Franscisco, CA, 1997. Morgan Kaufman.

Pedro Domingos and Michael J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.

Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

James P. Egan. *Signal Detection Theory and Roc Analysis*. Academic Press, New York, 1975.

Nir Friedman and Moises Goldszmidt. Building classifiers using bayesian networks. In *Proceedings of the American Association for Artificial Intelligence (AAAI)/Innovative Applications of Artificial Intelligence (IAAI)*, volume 2, pages 1277–1284, 1996a.

Nir Friedman and Moises Goldszmidt. Learning bayesian networks with local structure. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Inteligence (UAI)*, pages 252–262, San Francisco, CA, 1996b. Morgan Kaufmann Publishers.

Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.

Yu Fujimoto and Noboru Murata. Robust estimation for mixture of probability tables based on beta-likelihood. In Joydeep Ghosh, Diane Lambert, David B. Skillicorn, and Jaideep Srivastava, editors, *Proceedings of the Sixth SIAM International Conference on Data Mining*. SIAM, 2006.

Andrew B. Gelman, John S. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall, 2. edition, 2003.

V. Hamine and P. Helman. Learning optimal augmented bayes networks. Technical Report TR-CS-2004-11, Computer Science Department, University of New Mexico, 2004.

Jorge Jambeiro Filho and Jacques Wainer. Using a hierarchical bayesian model to handle high cardinality attributes with relevant interactions in a classification problem. In *Proceedings of the International Joint Conference of Artificial Intelligence (IJCAI)*. AAAI Press, 2007.

Eamonn J. Keogh and Michael J. Pazzani. Learning augmented bayesian classifiers: A comparison of distribution-based and classification-based approaches. In *Proceeeding of the Seventh International Workshop on Artificial Intelligence and Statistics*, pages 225–230, Ft. Lauderdale, FL, 1999.

Peter Lenk, Wayne DeSarbo, Paul Green, and Martin Young. Hierarchical bayes conjoint analysis: recovery of part worth heterogeneity from reduced experimental designs. *Marketing Science*, 15: 173–191, 1996.

Daniele Micci-Barreca. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *SIGKDD Explor. Newsl.*, 3(1):27–32, 2001.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988. ISBN 1558604790.

Irina Rish, Joseph Hellerstein, and Jayram Thathachar. An analysis of data characteristics that affect naive bayes performance. Technical Report RC21993, Watson Research Center, 2001.

Noam Slonim and Naftali Tishby. Agglomerative information bottleneck. In *Advances in Neural Information Processing Systems 12 (NIPS)*, pages 617–623, Denver, Colorado, USA, 1999. The MIT Press. ISBN 0-262-19450-3.

Benjamin Stewart, Jonathan Ko, Dieter Fox, and Kurt Konolige. The revisiting problem in mobile robot map building: A hierarchical bayesian approach. In Christopher Meek and Uffe Kjærulff, editors, *Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 551–558, Acapulco, Mexico, 2003. Morgan Kaufmann. ISBN 0-127-05664-5.

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers Inc., 1999.

Bianca Zadrozny. Reducing multiclass to binary by coupling probability estimates. In *Proceedings of the Advances in Neural Information Processing Systems 14 (NIPS)*, Cambridge, MA, 2001. MIT Press.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 694–699. ACM Press, 2002.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 609–616, MA, USA, 2001. Morgan Kaufmann. ISBN 1-55860-778-1.

Harry Zhang and Jiang Su. Naive bayesian classifiers for ranking. *Lecture Notes in Computer Science*, 3201:501–512, 2004.

Mu Zhu. Recall, precision and average precision. Technical Report 09, Department of Statistics & Actuarial Science, University of Waterloo, 2004.