

# Augmented Naive Bayesian Classifiers for Mixed-Mode Data

Xiao Li

December 15, 2003

## Abstract

Conventional Bayesian networks often require discretization of continuous variables prior to learning. It is important to investigate Bayesian networks allowing mixed-mode data, in order to better represent data distributions as well as to avoid the overfitting problem. However, this attempt imposes potential restrictions to a network construction algorithm, since certain dependency has not been well modeled statistically. This work first introduces parametrical representation for dependencies among mixed-mode variables. It then proposes two associated structure learning algorithms, both intend to augment a naive Bayesian network. Experiments on a medical diagnosis application show that a naive Bayes with parametrical representation works significantly better than the one with pre-discretization, while a clique-augmented naive Bayes makes a slight further improvement.

## 1 Introduction

Bayesian networks have been successfully applied to a great number of classification problems. There has been a surge of interests in learning Bayesian networks from data. The goal is to induce a network that best captures the dependencies among the variables for the given data. This optimization process, however, faces an exponentially large search space. Heuristics, therefore, are usually used in practice to find the best candidate networks in a space narrowed by appropriate constraints.

One important issue associated with Bayesian learning is conditional probability representation. In several domains of interest, such as medical diagnosis and machine diagnosis, variables in the data set often have both discrete and continuous values. As explained in Section 3, the conditional probability representation of such mixed-mode data is not fully defined. People tend to discretize the continuous variables to simplify the learning procedure [13, 7, 4]. However, discretization brings up potential challenges. On one hand, if the number of quantization bins is small, the resolution may not be high enough to capture the variation in data; On the other hand, increasing the number of bins may quickly lead to overfitting.

Lacking an analytically well-defined discretization algorithm driven by classification accuracy, this work aims to investigate Bayesian networks which allow mixed-mode data. Retaining continuous variables, however, implicitly imposes restrictions to a Bayesian network structure. This work explores two construction algorithms which take these restrictions into account and build networks extended from the framework of a *naive Bayes*, one of the most successful classifiers in the Bayesian family.

The rest of the paper is organized as follows. Section 2 reviews naive Bayes and the prior work on its augmentation. Section 3 discusses the probability representation of network allowing hybrid data. Section 4 presents two new algorithms of constructing augmented naive Bayes. Experiments and results are shown in Section 5, followed by conclusion and discussion.

## 2 Prior Work on Augmented Naive Bayes

A naive Bayes [11, 9] assumes conditional independence among all attributes  $A_1, A_2, \dots, A_N$  given the class variable  $C$ . It learns from training data the conditional probability  $P(A_i|C)$  of each attribute given its class label. Domingos gives a good explanation in [8] why a naive Bayes works surprisingly well despite its strong independence assumption. However, since those assumptions rarely ever hold in real applications, it is interesting to explore networks beyond a naive Bayes. One potential solution is to add “augmented edges” among attribute variables. Such a network not only takes all attributes into consideration, as what a naive Bayes does, but also relaxes the strong independence assumption.

Finding the optimal augmented naive Bayes is again an intractable problem. However, by imposing appropriate constraints, we can discover efficient augmentation algorithms. For example, Friedman, Geiger and Goldszmidt proposed in [6] a *tree-augmented naive Bayesian* (TAN) network. The algorithm described constructs a TAN  $B_T$  that maximizes  $LL(B_T|D)$ , which proved to be very successful in various classification applications. There are two foremost assumptions about a TAN network. First, it only allows discrete variables; all continuous variables have to be discretized before learning (otherwise it may result in dependencies which are hardly to represent statistically). Second, each attribute has at parents the class variable and at most one other attributes, thereby setting an upper bound on the clique size of a triangulated graph [5].

Consequently, this algorithm cannot guarantee to find a network supporting mixed-mode data. But with modest assumptions, a TAN is easily extended to tolerate continuous variables. Additionally, the restrictions imposed by a TAN can also be relaxed by allowing a flexible choice of clique size.

## 3 Conditional Probability Representation

Before presenting the augmentation algorithms allowing mixed-mode data, it is necessary to investigate how to model the dependency between two variables with arbitrary types. Table 1 summarizes the probability representation models used in this work. The methods in [1, 5] are referred in this work.

	Disc. Child	Cont. Child
Cont. Parent	Conditional probabilities	Gaussian mixtures
Disc. Parent	–	Inverse covariance matrices

Table 1: Dependency probability representations in a Bayesian network

### 3.1 Discrete parent, discrete child (DPDC)

In this work, we let  $K_i$  denote discrete variables and  $X_i$  denote continuous ones. For a DPDC edge, a conditional probability table is used to represent the dependency.

$$P(K_2 = k_2 | K_1 = k_1) = p_{k_1 k_2}. \quad (1)$$

If the parent and the child have cardinality  $M_1$  and  $M_2$  respectively, the table will have  $M_1 \cdot M_2$  entries.

Learning  $p_{k_1 k_2}$  is straightforward by counting the samples in the corresponding table entry.

### 3.2 Discrete parent, continuous child (DPCC)

The dependency between a discrete parent and a continuous can be represented parametrically using a single Gaussian or a Gaussian mixture.

$$P(X = x | K = k) = \sum_{i=1}^M w_{k,i} \mathcal{N}(x; \mu_{k,i}, \sigma_{k,i}), \quad (2)$$

where  $M$  is the number of Gaussian components,  $\mu_{k,i}$ ,  $\sigma_{k,i}$  and  $w_{k,i}$  are the mean, variance and weight of the  $i^{\text{th}}$  component respectively.

EM algorithm is used in learning the parameters of the Gaussian mixtures.

### 3.3 Continuous parent, continuous child (CPCC)

The dependency between a continuous parent and a continuous child (CPCC) can be specified implicitly using inverse covariance matrices. Assuming  $X_1, X_2, \dots, X_L$  are  $L$  continuous variables in the network and they are jointly distributed as a  $L$ -variate Gaussian, we have

$$p(\vec{x}) = (2\pi)^{-L/2} |\Sigma|^{-1/2} e^{-\frac{1}{2}(\vec{x}-\vec{\mu})' \Sigma^{-1} (\vec{x}-\vec{\mu})}. \quad (3)$$

The conditional probability  $p(X_2 = x_2 | X_1 = x_1)$  can be obtained by marginalization and conditioning using Equation (3). In [12], it is proved that the conditional independence properties of the distribution are determined by the location of zeros in  $K = \Sigma^{-1}$ . For example,  $k_{12} = 0$  indicates that  $X_1$  and  $X_2$  are independent given all other variables.

A factorization procedure is present in [2] to efficiently learn the linear coefficients. The basic idea is that  $K$  can be Cholesky factored as  $K = R'R$ , and  $R$  can be further decomposed as  $R = D^{1/2}U$ , where  $D^{1/2} = \text{diag}(R)$ . The exponent part of Equation (3), therefore, can be represented as

$$(\vec{x} - \vec{\mu})' \Sigma^{-1} (\vec{x} - \vec{\mu}) = (\vec{x} - B\vec{x} - \vec{\mu}^*)' D (\vec{x} - B\vec{x} - \vec{\mu}^*), \quad (4)$$

where  $B = I - U$  is an upper triangular matrix with zeros along the diagonal. In this way,  $K$  is transformed into a linear regression on  $x$ , and a non-linear optimization is no longer required within each EM iteration.

### 3.4 Continuous parent, discrete child (CPDC)

Representing the conditional probability of a CPDC edge is a hard problem. In this work, we avoid using such an edge. In other words, we always use a DPCC edge to represent dependency between a discrete and a continuous variable. However, the TAN construction algorithm in [6] may result in such a situation that a DPCC is indispensable in the directed tree. For example, if a continuous attribute variable is connected to two discrete attribute variables in an undirected tree, a DPCC edge is always needed. Consequently, the TAN construction procedure is no longer valid for a network with mixed-mode data. We therefore propose two alternate algorithms in the next section dealing with this issue.

## 4 Proposed Augmentation Algorithms

This section investigates two augmentation algorithms, both allowing mixed-mode data. The first one is a modified version of the TAN construction algorithm. The second one removes the restriction imposed by TAN by allowing arbitrary clique size.

### 4.1 Extended tree-augmented naive Bayes

As mentioned the previous sections, the TAN construction algorithm only allow discrete variables. A *extend tree-augmented naive Bayesian* (ETAN) network is a modified version of TAN, where we allow all edge types except for CPDC. The construction algorithm is as follows,

1. Build a complete directed graph for all attribute nodes, where each edge has two directions except for CPDC edges. Annotate the edge weights between  $A_i$  and  $A_j$  as  $I(A_i, A_j | C)$ ;
2. Build a maximum weighted spanning tree on the directed graph according to the algorithm presented in [14];
3. Construct ETAN by adding the class node and its outgoing edges to the attributes.

Based on the proof in [6], it is easy to show that this algorithm builds a ETAN  $B_T$  which maximizes  $LL(B_T|D)$ . This work applies an efficient implementation algorithm described in [10].

## 4.2 Clique-augmented naive Bayes

Recall the second constraint of a TAN stated in Section 2, that the network has a clique size at most 3. This may mitigate the overfitting problem. However, for a train set with sufficient data, this restriction may fail to capture the intertwined dependencies among more than 2 attribute variables. A *clique-augmented naive Bayesian* (CAN) network is therefore presented in the following text. It not only allows mixed data types, but also allows arbitrary clique size, where the largest clique size is controlled by a threshold.

1. Build a complete directed graph for all attribute nodes. Annotate the edge weights between  $A_i$  and  $A_j$  as  $I(A_i, A_j|C)$ ;
2. Prune the edges according to a thresholding weight  $w^*$ , where the edges with weight less than  $w^*$  are removed from the graph;
3. Find the maximum clique (the clique with the largest size) in the remaining graph. If there exist multiple maximum cliques with the same size, choose the with the highest average weight. Remove this clique from the graph. Repeat this step until there is no edge exists.
4. Within each clique selected in Step 3, locally build a complete directed acyclic graph (DAG) without a CPDC.
5. Construct a CAN by adding the class node and its outgoing edges to the attributes.

Note that Step 3 is implemented using a depth-first search algorithm. Additionally, it is always possible to find such a complete DAG without any CPDC edges. In this work, it is realized by iteratively choosing a continuous node and setting all its undirected edges to be outward.

## 5 Experiments and Results

### 5.1 Application and data preparation

The application used to evaluate these classifiers is *thyroid disease diagnosis*. Specifically, there are three disease classes, normal-, hyper-, and hypo-functioning of the thyroid gland, and there are a score of measurements available to serve as attributes. The measurements have both discrete and continuous types. Because this work does not deal with missing values, the database was “cleaned” before learning:

1. Remove all attributes with a significant number of missing values;
2. Remove redundant attributes (For example, “whether TSH is measured?” and “the TSH value” are redundant attributes, where the former is removed.);
3. Remove all instances with missing attributes

The cleaned train set consists of 3772 instances, but over 92% of them belong to the normal-functioning class. This may easily cause overfitting problem, since either the hyper- or the hypo- class has only around 100 instances. The cleaned test set consists of 3428 instances.

### 5.2 Experimental methodology and results

I ran three sets of experiments on this database. First I applied a maximum entropy discretization to the continuous variables, leading to a database with only discrete attributes. A naive Bayesian network was then learned from the discretized data. Second, I built a naive Bayes for the original mixed-mode data, where I tried both Gaussians and Gaussian mixtures in DPCC probability representation. The results of all the experiments above are shown in Table 2. Finally, I applied my proposed augmentation

algorithms to the naive Bayes. Specifically, the conditional mutual information was calculated for each pair of attributes. The ETAN and CAN construction algorithms were then applied respectively to the naive Bayes to build the augmented naive Bayesian networks.

Note that the Bayesian Network was implemented using *Graphical Model Toolkit* developed by Bilmes and Zweig [3]. The mutual information calculation was based on the *Mutual Information Toolkit* written by Filali. The ETAN and CAN construction algorithms were implemented by myself using C++.

Disc-NB, 2-Bin	Disc-NB, 3-B	Mixed-NB, SG	Mixed-NB GM
92.3%	92.0%	95.2%	95.2%

Table 2: Accuracy using Naive Bayes with and without discretization.

Table 2 shows the classification results of Naive Bayesian networks. “Disc-NB, 2-Bin” means a naive Bayes with continuous attributes discretized using 2 bins, similarly for “Disc-NB, 3-Bin”; “Mixed-NB, SG” means a naive Bayes using single Gaussians to represent DPCC, whereas “Mixed-NB, GM” uses Gaussian mixtures each with 2 components. Note that the second and third networks have the same number of parameters. Obviously the Naive Bayes allowing mixed-mode data outperforms its discretized counterpart by nearly 3%.

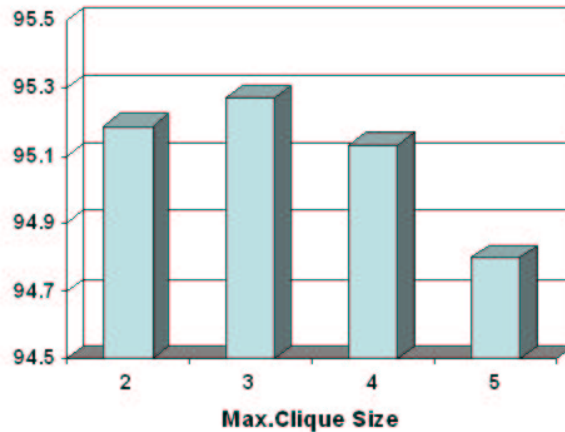


Figure 1: Accuracy using CAN network with different maximum clique size

Plot 1 shows the classification results of CAN networks. The threshold  $w^*$  was gradually decreased, leading to networks with an increasing maximum clique size. Note that there may exist multiple CANs with the same maximum clique size, and this plot only reports those with the empirically best results. The first column, a CAN with maximum clique size 2, is simply a naive Bayes. The second actually has only two augmented edges added, resulting in a slight improvement in accuracy. Further decreasing the threshold obviously degrades the classification performance.

It is worth noting that the mutual information calculation shows that only 3 pairs of attributes have an  $I(A_i, A_j|C)$  greater than 0.01 bits. And these three pairs, (TSH, TT4), (T3, T4U) and (TT4, T4U), are all from continuous-valued measurements, where (TSH, TT4) and (T3, T4U) are the two augmented edges added to the CAN with the maximum clique size 3.

Due to the relative independence properties displayed in this data set, I did not implement the network resulting from the ETAN construction algorithm, considering that an ETAN has 20 augmented edges, which is unlikely to have good performance for this task.

## 6 Conclusion and Discussion

This work compared a naive Bayes classifier with and without discretization on data. The naive Bayes allowing mixed-mode data beats the one with maximum entropy discretization by an almost 3% absolute increase in accuracy. Based on the best naive Bayes, two augmentation algorithms were applied to construct augmented naive Bayesian networks. The clique-augmented one obtained a slight increase in accuracy when the maximum clique size is 3, where only two extra edges were added. As the clique size increases, overfitting problem becomes more and more pronounced since the model complexity is exponential in the clique size. The extended tree-augmented naive Bayes is not expected to work well for this task for the following reasons. First, by inspecting the conditional mutual information among attribute variables, we found that most of the attributes were quite independent of each other. In other words, the naive Bayes is quite accurate in making its independence assumption, while adding many edges, as in a tree-structure, is unnecessary. Second, lacking sufficient training data, especially for the hyper- and hypo-functioning cases, adding many augmented edges will cause overfitting. Therefore, a naive Bayes, or a lightly augmented naive Bayes, will work very well for this particular data set. For other applications where training data is abundant, the ETAN and CAN may manifest their advantages, but that requires evaluations on a significant number of databases.

## References

- [1] D.Heckermann and D.Geiger. Learning Bayesian networks: a unification for discrete and Gaussian domains. In *Proc.Uncertainty Artificial Intelligence*, 1995.
- [2] J.Bilmes. Factored sparse inverse covariance matrices. In *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2000.
- [3] J.Bilmes and G.Zweig. Graphical model toolkit. <http://ssli.ee.washington.edu/people/bilmes/pubs-frame.html>.
- [4] L.Kurgan and K.J.Cios. Discretization algorithm that uses class-attribute interdependence maximization. In *Proc. Intl. Conf. on Artificial Intelligence*, 2001.
- [5] M.Jordan and C.Bishop. *Introduction to Graphical Models*. Pre-print, 2000.
- [6] N.Friedman, D.Geiger, and M.Goldszmidt. Bayesian network classifiers. *Machine Learning*, 1997.
- [7] N.Friedman and M.Goldszmidt. Discretization of continuous attributes while learning Bayesian networks. In *Proc. Intl. Conf. on Machine Learning*, 1996.
- [8] P.Domingos and M.Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 1997.
- [9] P.Langley, W.Iba, and K.Thompson. An analysis of Bayesian classifiers. In *Proc. on Intl. Conf. on Artificial Intelligence*, 1992.
- [10] R.E.Tarjan. Finding optimum brachings. *Networks*, 1977.
- [11] R.O.Duda and P.E.Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.
- [12] S.L.Lauritzen. *Graphical Models*. Oxford Science Publications, 1996.
- [13] U.M.Fayyad and K.B.Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Artificial Intelligence*, 1993.
- [14] Y.J.Chu and T.H.Liu. On the shortest arborescence of a directed graph. *Science Sinica*, 1965.