

Bayesian Networks

Amy Ratnakaran

Department of Mathematics and Statistics, Applied Statistics Honours

The University of Melbourne

4th November 2005

Abstract

This study investigates the performance of a range of Bayesian Network classifiers by applying the well know classifiers, Tree Augmented Naive Bayes (TAN) and Naive Bayes (NB) classifiers along side intuitively learned classifiers on credit data obtained from Germany of 1000 observations. There was an expectation of the more sophisticated models as TAN and NB to be among the highest performers due to their higher structural complexity. Such models have the ability to measure uncertainty and encompass complexities in the state spaces of related sets of discrete or categorical variables through the use of probability theory and graph theory. All models were applied to the training data set (first 700 observations of the german credit data) for parameter estimation of conditional probabilities. Finally the models were tested on the test data set (last 300 observations of the german credit data) where miss-classification rates on the test data set were a measure of model performance and were a measure of how accurately a model would classify a customer as a defaulter or non-defaulter on a potential loan.

1 Definition of a Graphical Bayesian Network Model

A Bayesian Network (BN) is an directed acyclic graph (DAG) that represents a joint probability distribution over a set of discrete random variables $U = \{X_1, \dots, X_N\}$, where each X_i represents a discrete random variable with a finite set of mutually exclusive states $\{x_{ik} : k = 1, \dots, n\}$. Thus a BN B for U consists of components $B = \langle G, \Theta \rangle$. The graphical component G represents $\{X_i : i = 1, \dots, N\}$ as nodes $i = 1, \dots, N$ such that each node takes on any one state of the set of states $\{x_{ik} : k = 1, \dots, n\}$ of X_i as an event. In conjunction, G assigns directed arcs between any pair of nodes, i.e. from parent node i for variable X_i to child node j for variable $\{X_j : j \neq i\}$. These directed arcs describe direct dependence of the variable X_j on X_i as causal relationships from the immediate parent node i to its' child node j joined by an arc. A directed arc will lead from the causal variable to the effect variable. The lack of an arc between any pair of nodes i & j represents conditional independence between variables X_i & X_j , conditional on the known/unknown state of at least one of the intermediate nodes for X_1, \dots, X_m as a set of the set of nodes S_1, \dots, S_h along any undirected path between nodes i & j . That is

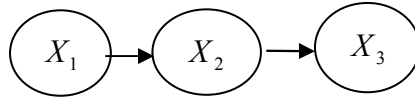
$$P(X_i = x_{ik_i}, X_j = x_{jk_j} | S_h = s_{hk_h}) = P(X_i = x_{ik_i} | S_h = s_{hk_h})P(X_j = x_{jk_j} | S_h = s_{hk_h}).$$

Definition: A serial connection is any connected subgraph H of G consisting of $v = 3$ nodes and $v - 1$ arcs as depicted below. For a serial connection, X_1 & X_3 are

conditionally independent given X_2 . That is if the state of the intermediate node for X_2 is known, then no knowledge of X_1 will alter the probability of X_3 and hence,

$$P(X_3 = x_{3k} | X_1 = x_{1i}, X_2 = x_{2j}) = P(X_3 = x_{3k} | X_2 = x_{2j}).$$

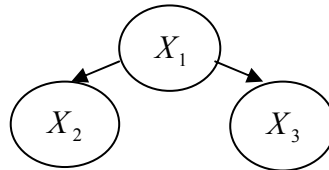
Graph H:



Definition: A diverging connection is any connected subgraph F of G consisting of $v = 3$ nodes and $v - 1$ arcs as depicted below. For a diverging connection, X_2 & X_3 are conditionally independent given X_1 . That is if the state of the intermediate node for X_1 is known, then no knowledge of X_2 will alter the probability of X_3 and hence,

$$P(X_3 = x_{3k} | X_1 = x_{1i}, X_2 = x_{2j}) = P(X_3 = x_{3k} | X_1 = x_{1i}).$$

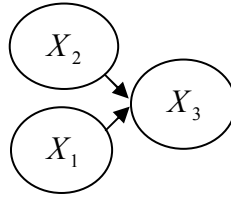
Graph F:



Definition: A converging connection is any connected subgraph E of G consisting of v nodes and $v - 1$ arcs as depicted below. For a converging connection, X_1 & X_2 are conditionally independent given X_3 ,

$$P(X_2 = x_{2j} | X_1 = x_{1i}, X_3 = x_{3k}) = P(X_2 = x_{2j} | X_3 = x_{3k}).$$

Graph E:



Definition: Variables X_i & X_j are *d-separated* if there is in any undirected path between X_i & X_j an intermediate variable X_k with known state, for a serial or diverging connections. And the variables X_i & X_j are *d-separated* if there is in any undirected path between X_i & X_j an intermediate variable of at least one of X_k or descendents of X_k with unknown state, for converging connections (Jensen 1996). Then for all these cases it can be said that X_i & X_j *d-separated* given X_k . *d-separated* is used as conditional independence/dependence terminology and is primarily used in a BN to easily identify and calculate joint probability tables for those nodes representing variables defined by *d-separation*.

1.1 Probability Calculations for the whole network given probabilities for nodes and parents.

Let $\Pi_i = (X_j : j \text{ is an immediate parent of } i)$. The parametric component Θ is the quantitative aspect of B which defines conditional probabilities for each node given their immediate parents as represented by the arcs in G , denoted $P(X_i = x_{ik} | \Pi_i)$, Hence, each node X_i with Π_i in G possess a conditional probability table conditional on Π_i for

each state of X_i and Π_i where the table consists of parameters

$$P(X_i = x_{ik} | \Pi_i) = P(X_i = x_{ik} | X_{j_1} = a_1, \dots, X_{j_m} = a_m) \text{ where } k = 1, \dots, n \text{ and } j_1, \dots, j_m$$

are the immediate parents of i . If $\Pi_i = \emptyset$, then the node for X_i in G has an

unconditional probability table of parameters $P(X_i = x_{ik})$.

In calculating probabilities for the whole network, apart from root node probabilities, we:

Let BN be a Bayesian Network over

$$U = \{X_1, \dots, X_N\}.$$

Then the joint probability distribution $P(U)$ is the product of all conditional probabilities specified in BN.

$$P(U) = \prod_i^N P(X_i = x_{ik} | \Pi_i)$$

Each node for X_i in G possess a conditional probability table that specifies a subset of

$$P(U) = \prod_i^N P(X_i = x_{ik} | \Pi_i) \text{ according to } \Pi_i.$$

More specifically, calculation of probabilities for a BN graph G begins with the root nodes of G where $\Pi_i = \emptyset$. These nodes have probabilities $P(X_i = x_{ik})$ for all states

$\{x_{i1}, \dots, x_{ik}\}$ for X_i , assigned to them by estimation from real world knowledge or

estimation from existing data, where variables represented by the root nodes are

independent. Also, through real world knowledge or estimation from existing data are the non-root node probabilities, $P(X_i = x_{ik} | \Pi_i)$, calculated for G . Once root and non-root node probabilities of G are known, probability theory is used to calculate unconditional probabilities of all nodes i representing X_i for all states $\{x_{i1}, \dots, x_{ik}\}$ in G as follows. Thus using probability theory, the joint probability table $P(X_i = x_{ik}, \Pi_i)$ is calculated for all nodes for X_i via the equation,

$P(X_i = x_{ik}, \Pi_i) = P(X_i = x_{ik} | \Pi_i)P(\Pi_i)$, where when Π_i is a root node, $P(\Pi_i)$ is given as discussed above. And where Π_i is a non-root node $P(\Pi_i)$ is calculated using methods explained below. Then, to calculate total probabilities of all nodes i representing X_i for all states $\{x_{i1}, \dots, x_{ik}\}$ in G , marginalise over each nodes' joint probability table to get $P(X_i = x_{ik})$ for all states $\{x_{i1}, \dots, x_{ik}\}$ for all X_i .

This concept of conditioning on immediate parents provides a less complex representation of joint probability distribution U by reducing the number of probabilities required to be calculated for a BN. To illustrate, an example following a BN with N binary random variables representing the joint $P(X_1 = x_{1k_1}, \dots, X_N = x_{Nk_N})$ would normally need $2^N - 1$ probabilities where

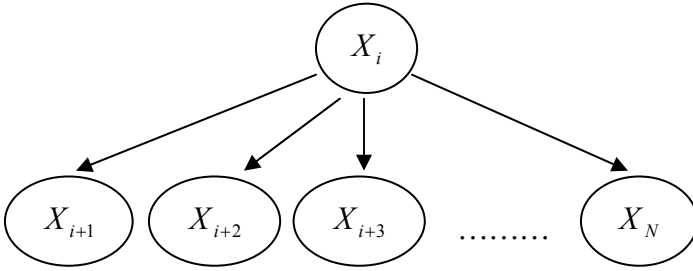
$$P(X_1 = x_{1k_1}, \dots, X_N = x_{Nk_N}) = \prod_{i=1}^N P(X_i | X_{i+1}, \dots, X_N).$$

However, when conditional probability independence assumptions are applied to U , then the joint probability table on U is reduced to requiring $< 2^N$ parameters when $\Pi_i \neq \{X_{i+1}, \dots, X_N\}$, depending on the structure of the BN, since now

$P(X_1 = x_{1k_1}, X_2 = x_{2k_2}, \dots, X_N = x_{Nk_N}) = \prod_{i=1}^N P(X_i | \Pi_i)$. An example of this can be seen

in the below BN where,

$$\begin{aligned}
 P(X_i = x_{ik_i}, \dots, X_N = x_{Nk_N}) &= P(X_{i+1} | X_i, X_{i+2}, \dots, X_N) P(X_i, X_{i+2}, \dots, X_N) \\
 &= P(X_{i+1} | X_i) P(X_{i+2} | X_i, X_{i+3}, \dots, X_N) P(X_i, X_{i+3}, \dots, X_N) \\
 &= P(X_{i+1} | X_i) P(X_{i+2} | X_i) P(X_{i+3} | X_i, X_{i+4}, \dots, X_N) P(X_i, X_{i+4}, \dots, X_N) \\
 &= \dots = P(X_{i+1} | X_i) P(X_{i+2} | X_i) \dots P(X_N | X_i) P(X_i)
 \end{aligned}$$



So for this specific BN the total number of probabilities required reduces to $2(N - 1) + 1$.

1.2 Given Probabilities for a network and partial observation of nodes, calculation of conditional probabilities for remaining nodes.

At any time a BN graph G can receive information on the certainty of a state x_{ik} of a variable X_i having been observed as a new event. Hence calculations are required to update conditional probability tables of G in light of the new information gathered as the state x'_{ik} for X_i . New information can enter the BN as evidence at any node or as any set of nodes. Hence, upon new evidence discovered on a node i for X_i , the conditional probability table for that node updates from $P(X_i = x_{ik_i} | X_j = x_{jk_j}) = \delta_{ij}$ to

$P(X_i = x'_{ik_i} | X_j = x_{jk_j}) = \delta'_{ij}$. However, here X_i is not a root node. Subsequently, Bayes Rules is used to update conditional probability tables of nodes by systematically recalculating them at each of the ancestor nodes of X_i , following a path from X_i in the opposite direction of the directed arcs, until all unconditional probability tables of root nodes are updated .

Bayes Rule states:

$$P(\Pi_i | X_i = x_{ik}) = \frac{P(X_i = x_{ik} | \Pi_i)P(\Pi_i)}{\sum_i^N P(\Pi_i)P(X_i = x_{ik} | \Pi_i)} = \frac{P(X_i = x_{ik} | \Pi_i)P(\Pi_i)}{P(X_i = x_{ik})}, \text{ where using the}$$

Law of Total Probability simplifies the denominator of this equation. Let $\Phi_i = (X_h : h \text{ is an immediate child of } i)$. Once all $P(\Pi_i | X_i = x_{ik})$ for all root nodes are obtained then follow calculations described in section 1.1 to update unconditional probability tables of remaining nodes sequentially following a path along nodes in the direction of the directed arc from root nodes until all leaf nodes have been updated. Here $\Pi_i = \emptyset$ for all root nodes and $\Phi_i = \emptyset$ for all leaf nodes.

1.3Example

In support of this example, graph F of section 1, provides a graphical representation BN.

Let the root node and non-root node probabilities of graph F be given as,

$$P(X_1 = x_{11}) = 0.7, P(X_1 = x_{12}) = 0.3, P(X_3 = x_{31} | X_1 = x_{11}) = 0.8,$$

$$P(X_3 = x_{31} | X_2 = x_{21}) = 0.1, P(X_2 = x_{21} | X_1 = x_{11}) = 0.8 \text{ and}$$

$$P(X_2 = x_{21} | X_1 = x_{12}) = 0.1, \text{ where all variables are binary. Thus an initial conditional}$$

probability table can be constructed as Table 1. The entries of Table 1 represent

$$P(X_i = x_{ik_i} | X_1 = x_{1k_1}) :$$

Table 1: Initial Conditional Probability Table

	$X_1 = x_{11}$	$X_1 = x_{12}$
$X_2 = x_{21}$	0.8	0.1
$X_2 = x_{22}$	0.2	0.9
$X_3 = x_{31}$	0.8	0.1
$X_3 = x_{32}$	0.2	0.9

Now, using Table 1 and $P(X_i, \Pi_{X_i}) = P(X_i | \Pi_{X_i})P(\Pi_{X_i})$, construct the initial joint probability table below as:

Table 2: Initial Joint Probability Table

	$X_1 = x_{11}$	$X_1 = x_{12}$	Marginals
$X_2 = x_{21}$	$0.8 \times 0.7 = 0.56$	$0.1 \times 0.3 = 0.03$	$0.56 + 0.03 = 0.59$
$X_2 = x_{22}$	$0.2 \times 0.7 = 0.14$	$0.9 \times 0.3 = 0.27$	$0.14 + 0.27 = 0.41$
$X_3 = x_{31}$	$0.8 \times 0.7 = 0.56$	$0.1 \times 0.3 = 0.03$	$0.56 + 0.03 = 0.59$
$X_3 = x_{32}$	$0.2 \times 0.7 = 0.14$	$0.9 \times 0.3 = 0.27$	$0.14 + 0.27 = 0.41$

Thus, $P(X_2 = x_{21}) = 0.59$, $P(X_2 = x_{22}) = 0.41$, $P(X_3 = x_{31}) = 0.59$ &

$P(X_3 = x_{32}) = 0.41$, from the marginals calculated in Table 2. The entries in Table 2

represent $P(X_i = x_{ik_i}, X_1 = x_{1k_1})$.

Let new information enter the BN with probability $P^*(X_3 = x_{31}) = 1$ with

$(P^*(X_3 = x_{31}), P^*(X_3 = x_{32})) = (1, 0)$ and use Bayes Rule to update $P(X_1 = x_{11})$ and

$P(X_1 = x_{12})$.

Hence,

$$\begin{aligned}
P^*(X_1 = x_{1k}) &= P(X_1 = x_{1k} | X_3 = x_{31}) = \frac{P(X_3 = x_{31} | X_1 = x_{1k})P(X_1 = x_{1k})}{P(X_3 = x_{31})} \\
&= \frac{1}{P(X_3 = x_{31})} ((P(X_3 = x_{31} | X_1 = x_{11})P(X_1 = x_{11})), (P(X_3 = x_{31} | X_1 = x_{12})P(X_1 = x_{12}))) \\
&= \frac{1}{0.59} (0.8 \times 0.7, 0.1 \times 0.3) = (0.95, 0.05),
\end{aligned}$$

then $(P(X_1 = x_{11} | X_3 = x_{31}), P(X_1 = x_{12} | X_3 = x_{31})) = (0.95, 0.05)$ and

$$P(X_1 = x_{12} | X_3 = x_{31}) = 0.05, \text{ where } P(X_1 = x_{12} | X_3 = x_{31}) = 1 - P(X_1 = x_{11} | X_3 = x_{31})$$

Now to update $P(X_2 = x_{2i})$ construct the updated joint probability table for node X_2 via

$$\begin{aligned}
P(X_2 = x_{2i}, X_1 = x_{1k}, X_3 = x_{31}) &= P(X_2 = x_{2i} | X_1 = x_{1k}, X_3 = x_{31})P^*(X_1 = x_{1k}) \\
&= P(X_2 = x_{2i} | X_1 = x_{1k})P^*(X_1 = x_{1k}) \text{ by } d\text{-separation. Note: } P^* \neq P
\end{aligned}$$

Table 3: Updated Joint Probability Table for Node X_2 & X_1

	$X_1 = x_{11}, X_3 = x_{31}$	$X_1 = x_{12}, X_3 = x_{31}$	Marginals
$X_2 = x_{21}$	$0.8 \times 0.95 = 0.76$	$0.1 \times 0.05 = 0.005$	0.765
$X_2 = x_{22}$	$0.2 \times 0.95 = 0.19$	$0.9 \times 0.05 = 0.045$	0.235

Thus, $P^*(X_2 = x_{21}) = 0.765$ and $P^*(X_2 = x_{22}) = 0.235$, from the marginals

calculated in Table 3. The entries of Table 3 represent $P(X_2 = x_{2k_2}, X_1 = x_{1k_1}, X_3 = x_{3k_3})$:

New information could also enter the BN as the observed events $X_3 = x_{31}$ and $X_2 = x_{21}$

with $P(X_3 = x_{3j}) = (1,0)$ and $P(X_2 = x_{2i}) = (1,0)$. Hence computations follow as:

$$P^*(X_1 = x_{11}) = P(X_1 = x_{11} \mid X_3 = x_{31}, X_2 = x_{21})$$

$$P(X_1 = x_{11} \mid X_3 = x_{31}, X_2 = x_{21}) = \frac{P(X_1 = x_{11}, X_3 = x_{31}, X_2 = x_{21})}{P(X_3 = x_{31}, X_2 = x_{21})}$$

$$= \frac{P(X_3 = x_{31} \mid X_1 = x_{11})P(X_2 = x_{21} \mid X_1 = x_{11})P(X_1 = x_{11})}{P(X_3 = x_{31}, X_2 = x_{21} \mid X_1 = x_{11})P(X_1 = x_{11}) + P(X_3 = x_{31}, X_2 = x_{21} \mid X_1 = x_{12})P(X_1 = x_{12})}$$

$$= \frac{P(X_3 = x_{31} \mid X_1 = x_{11})P(X_2 = x_{21} \mid X_1 = x_{11})P(X_1 = x_{11})}{P(X_3 = x_{31} \mid X_1 = x_{11})P(X_2 = x_{21} \mid X_1 = x_{11})P(X_1 = x_{11}) + P(X_3 = x_{31} \mid X_1 = x_{12})P(X_2 = x_{21} \mid X_1 = x_{12})P(X_1 = x_{12})}$$

$$= \frac{0.8 \times 0.8 \times 0.7}{0.8 \times 0.8 \times 0.7 + 0.1 \times 0.1 \times 0.3} = 0.9933$$

Then,

$$P(X_1 = x_{12} \mid X_3 = x_{31}, X_2 = x_{21}) = 1 - P(X_1 = x_{11} \mid X_3 = x_{31}, X_2 = x_{21}) = 1 - 0.9933 = 0.0067$$

Thus,

$$P(X_1 = x_{1k} \mid X_3 = x_{3j}, X_2 = x_{2i})P(X_3 = x_{3j}, X_2 = x_{2i}) = P(X_1 = x_{1k}, X_3 = x_{3j}, X_2 = x_{2i})$$

and marginalizing $P(X_1 = x_{1k}, X_3 = x_{3j}, X_2 = x_{2i})$ over $X_3 = x_{3j}$ & $X_2 = x_{2i}$ gives

$$P^*(X_1 = x_{1k}) = \sum_{j=1, i=1}^{n=2, m=2} P^*(X_1 = x_{1k}, X_3 = x_{3j}, X_2 = x_{2i}) = (0.9933, 0.0067)$$

2 Using a Graphical Model as a classifier

BN as classifiers represent the conditional dependence and conditional independence relationships between explanatory variables and hypothesis variables as detailed in previous sections. The hypothesis variable of a BN is an unknown variable of which we wish to ascertain. The explanatory variables are those variables we used to help explain and determine the behaviour of the hypothesis variable. Here, a BN classifier with graph G uses the probability tables of those nodes representing explanatory variables over G to classify the node representing the hypothesis variable. Let X_c be a hypothesis variable and Let $\{X_i : i = 1, \dots, N\}$ where $c \notin \{1, \dots, N\}$ be explanatory variables. BN Classifiers use conditional probability $P(X_i = x_{ik_i}, \dots, X_N = x_{Nk_N} | X_c = x_{ck_c})$ and Bayes Rule to predict the class of the hypothesis variable given known states of the set of remaining explanatory variables in BN such that,

$$P(X_c = x_{ck_c} | X_i = x_{ik_i}, \dots, X_N = x_{Nk_N}) = \frac{P(X_i = x_{ik_i}, \dots, X_N = x_{Nk_N} | X_c = x_{ck_c})P(X_c = x_{ck_c})}{P(X_i = x_{ik_i}, \dots, X_N = x_{Nk_N})}$$

$$= \frac{P(X_i = x_{ik_i}, \dots, X_N = x_{Nk_N}, X_c = x_{ck_c})}{P(X_i = x_{ik_i}, \dots, X_N = x_{Nk_N})}.$$

Here a BN structure is used to encode conditional probabilities between the variables it represents with probabilistic simplicity. This is a result of the structure defining conditional probabilities conditional on Π_i only, rather than all available variables. This reduces the number of probabilities required.

2.1 Preparing the data

A BN represents a joint probability distribution over a set of discrete random variables $U = \{X_1, \dots, X_N\}$, where each X_i represents a discrete random variable with a finite set of mutually exclusive states $\{x_{ik} : k = 1, \dots, n\}$. Consequently, the resulting requirement for any such X_i that possess continuous data is to convert the data from continuous to discrete data by intuitively partitioning it into relevant categories representative of the distinctive states of X_i . Once in discrete form, the data must be segregated into two groups corresponding to a training data set and a test data set. Preferred proportions of data segregation into training and test sets follow 2/3 and 1/3 respectively. Then a BN model needs the use of the training data set for estimation of $P(X_i | \Pi_i)$ and thus $P(X_1 = x_{1k_1}, \dots, X_N = x_{Nk_N})$. The test data set is used to test model accuracy of the BN as a reliable classifier and is also used for performing model selection.

2.2 Estimation of probabilities for training data

When no knowledge is known in the field about the area of study then root node $P(X_i = x_{ik_i})$ and non-root node probabilities $P(X_i = x_{ik_i} | \Pi_i)$ of the BN model need to be estimated from the training data.

Let the entire data set M have size $|M|$. Each observation in M possessing N explanatory attributes, corresponding to the set explanatory variables $\{X_1, \dots, X_N\}$ at states

$\{x_{ik_i} : k = 1, \dots, n\}$, and a class attribute representing the hypothesis variable, X_c such that we redefine $U = \{X_1, \dots, X_N, X_c\}$.

And,

Let R denote the training data set and let T denote the test data set such that $M = R \cup T$ and $|M| = |R| \cup |T|$

So then the frequency of observations that possess identical state space over U are used to calculate estimates of $P(X_c = x_{ck_c} \mid X_1 = x_{1k_1}, \dots, X_N = x_{Nk_N})$ and $P(X_1 = x_{1k_1}, \dots, X_N = x_{Nk_N})$ using R . Thus as an example, for some BN graph, a conditional probability for node 2 can be calculated from the data as:

Here, $\Pi_2 = (X_1 = x_{1k_1}, X_3 = x_{3k_3}, \dots, X_N = x_{Nk_N})$

Then,

$$\hat{P}(X_2 = x_{2k_2} \mid X_1 = x_{1k_1}, X_3 = x_{3k_3}, \dots, X_N = x_{Nk_N}) = \frac{\hat{P}(X_1 = x_{1k_1}, X_3 = x_{3k_3}, \dots, X_N = x_{Nk_N}, X_2 = x_{2k_2})}{\hat{P}(X_1 = x_{1k_1}, X_3 = x_{3k_3}, \dots, X_N = x_{Nk_N})}$$

\approx (the number of observations in R with $X_1 = x_{1k_1}, X_3 = x_{3k_3}, \dots, X_N = x_{Nk_N}, X_2 = x_{2k_2}$) \div

(the number of observations in R with $X_1 = x_{1k_1}, X_3 = x_{3k_3}, \dots, X_N = x_{Nk_N}$)

Also, if $\Pi_2 = \emptyset$, $\hat{P}(X_2 = x_{2k_2}) \approx$ (the number of observations in R with $X_2 = x_{2k_2}$) $\div |R|$

More over, calculating $\hat{P}(X_i = x_{ik_i})$ or $P(X_c = x_{ck_c})$ for root nodes and

$\hat{P}(X_i = x_{ik_i} | \Pi_i)$ or $\hat{P}(X_c = x_{ck_c} | \Pi_i)$ for non-root nodes of the BN model follows as above.

2.3 Testing performance using miss-classification rates on test data

(classification tests)

There are several methods used to test the performance of a BN model B as a reliable classifier, one of which tests miss-classification rates on test data or training data of B . Other BN performance tests include Minimal Description Length (MDL) which is a scoring function that applies a score to B conditional on its' training data set through the use of a function consisting of a log-likelihood and penalty term. However MDL has its' limitation as described by Friedman, Geiger and Goldszmidt 1997. This paper however, will focus on the miss-classification rate test for test data sets in testing the performance of B . Thus the miss-classification rate is an accuracy measure of the ability of B to predict the correct classification of observations in T by producing estimates

$\hat{P}(X_c = x_{ck_c} | X_1 = x_{1k_1}, \dots, X_N = x_{Nk_N}) = \varphi$, $0 \leq \varphi \leq 1$ calculated from R for each

observation in T . These estimates φ are used along with a threshold or cut-off probability p to predict the class of each observation where a class is assigned to each observation in T depending on if $\varphi > p$ or $\varphi < p$. The choice of p can be arbitrary or chosen from optimality tests as detailed in section 2.5. Once a class is predicted for each

observation, the predicted class is then compared to the true class of each observation and a frequency measure is taken of those observations that were miss-classified.

Assume: $X_c = x_{ck_c}$ is binary, where $k \in \{1,2\}$ and $x_{ck_c} \in \{0,1\}$

Let $\hat{P}_T(X_c = 0)$ be the predicted probability of a positive outcome for each observation in T , $\hat{P}_T(X_c = 1)$ be the predicted probability of a negative outcome for each observation in T , $P_T(X_c = 0)$ be the observed probability of a positive outcome for each observation in T & $P_T(X_c = 1)$ be the observed probability of a negative outcome for each observation in T .

In conjunction,

Let $\hat{X}_c = 0$ be the predicted positive outcome for an observation in T , $\hat{X}_c = 1$ be the predicted negative outcome for an observation in T , $X_c = 0$ be the observed positive outcome for an observation in T & $X_c = 1$ be the observed negative outcome for an observation in T .

Then, Let p be a cut-off probability that is a threshold used to predict and assign each observation in T to either class $\hat{X}_c = 0$ or $\hat{X}_c = 1$ such that if

$\hat{P}_T(X_c = 0 | X_1 = x_{1k_1}, \dots, X_N = x_{Nk_N}) > p$ then the class $\hat{X}_c = 0$ is assigned to each T that satisfies as such, otherwise $\hat{X}_c = 1$ is assigned.

Then it follows that $\hat{X}_c = \hat{x}_{ck_c} = X_c = x_{ck_c}$ for correct classifications and $\hat{X}_c = \hat{x}_{ck_c} \neq X_c = x_{ck_c}$ for incorrect classifications or miss-classifications, where terminology follows as $\hat{X}_c = 0 = X_c = 0$ labels true positives, $\hat{X}_c = 1 = X_c = 1$ labels true negatives, $\hat{X}_c = 0 \neq X_c = 1$ labels false positives and $\hat{X}_c = 1 \neq X_c = 0$ labels false negatives. Conveniently, a *confusion matrix* presents the above information in the form of an easily read table as Table 4.

Table 4: Confusion Matrix

OBSERVED	PREDICTED		
	$\hat{X}_c = 0$	$\hat{X}_c = 1$	Total
$X_c = 0$	s	t	$s + t$
$X_c = 1$	d	f	$d + f$
Total	$s + d$	$t + f$	$s + t + d + f$

Here, s is the number of observations in T with a true positive label, f is the number of observations in T with a true negative label, d is the number of observations in T with a false positive label and t is the number of observations in T with a false negative label.

Terminology:

$$\text{Proportion of True Positives} = \frac{s}{s + t}$$

$$\text{Proportion of True Negatives} = \frac{f}{d + f}$$

$$\text{Proportion of False Positives} = \frac{d}{d + f}$$

$$\text{Proportion of False Negatives} = \frac{t}{s + t}$$

$$\text{Proportion of Correct Classifications} = \frac{s + f}{s + t + d + f}$$

Ultimately, a BN model aims to minimise the two types of classification errors, that are

the Proportion of False Positives = $\frac{d}{d + f}$ and Proportion of False Negatives = $\frac{t}{s + t}$, and

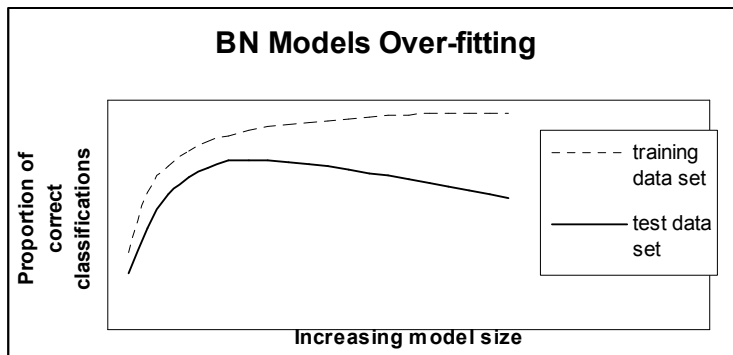
maximise the Proportion of Correct Classifications = $\frac{s + f}{s + t + d + f}$, in its' quest to be a reliable classifier.

2.4 Over-fitting

Although larger BN models with larger node and arc structures are generally more comprehensive as classifiers than smaller models with simpler node and arc structures, sometimes biggest is not always best. The relationship that exists between increasing model size and increasing proportion of correct classification draws to a limit, where beyond this limit a model any larger will do no better in increasing correct classification proportions and can become decreasing in correct classification proportions as it increases in size. The phenomenon of over-fitting describes this problem, where the model begins to fit unusual data and even usual data which in turn cause the model to overlook the genuine patterns depicted by the data in general. In most cases, over-fitting is the result of a model being too closely designed to fit the training data set. Thus, over-

fitting is most prevalent where it can be seen that a model will fit the training data set perfectly but in the same instance will fail to classify reliably on the test data set

(Cheng&Greiner). The following graph tells this story visually.

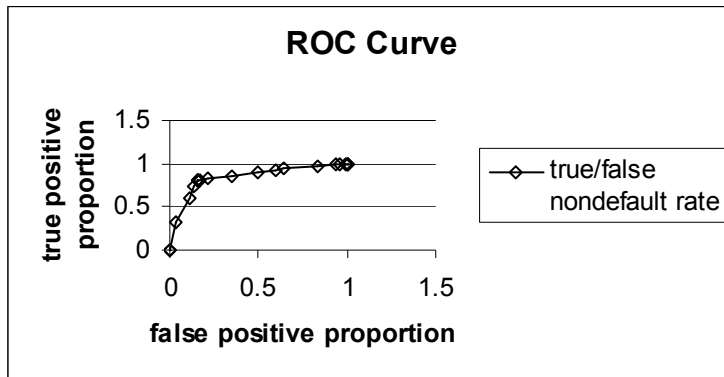


In the above graph it can be seen where the increasing size of a model begins to show major differences in proportion of correct classifications between the training and test data set. Some scoring methods such as MDL help to avoid over-fitting of the training data by regulating the complexity and inturn reducing the number of probability calculations (Friedman,Geiger&Goldszmidt1997).

2.5 Two types of classification error: ROC curves

The Receiver Operating Characteristic (ROC) curve helps to choose optimal cut-off p that gives maximal proportion of correct classifications at minimal proportion of false classifications of a BN model. In doing so, the ROC curve plots for every $0 \leq p \leq 1$, proportions of true positives vs proportions of false positives. The ROC curve can also plot for every $0 \leq p \leq 1$, proportions of true negatives vs proportions of false negatives.

However it is impossible to try to optimize p simultaneously on both curves. Therefore a compromise has to be drawn between which type of the two types of miss-classifications to minimise when optimizing p . Thus, the optimal p is the point on the ROC curve that corresponds to the most minimal miss-classification while coinciding with the most maximal correct classification with a compromise. The following graph gives example of such, with $0 \leq p \leq 1$ at increments of 0.05 for a BN network B using the data set R .



Then as an illustrative example, the most optimal choice of p for T that gives maximal proportion of correct classifications at minimal proportion of false classifications from B is $p \approx 0.4$ which corresponds to the 8th ordered observation on the line of this graphical example.

2.6 Choosing sensitivity to minimise expected cost

The two types of classification errors, false positives and false negatives, can sometimes be equally or unequally weighted in cost. In the case of unequally weighted costs between the two types of classification error, there is usually a greater expense associated

to one type of error than the other. This consequently brings new considerations into choosing the best BN model and the best p . Such considerations can force selection of the best model and best p to compromise priorities in selection on the basis of maximal proportions of correct classifications characteristics in favour of minimizing errors where there are greater costs associated.

3 Model Construction and Selection

The two initial tasks in BN are learning the graphical structure G and then learning the parameters Θ for the structure. The first task when learning a BN is to find the simplest structure G that correctly model explanatory and hypothesis variables of G . This can be done intuitively in some fields of study, such as genetics, where extensive knowledge is already known about conditional dependencies and conditional independencies between variables through historical experimental discovery. However, in the absence of knowledge, the task of learning a BN B with random variables $U = \{X_i : i = 1, \dots, N\}$ of states $\{x_{ik} : k = 1, \dots, n\}$, relies on a training data set R and testing the miss-classification rate on the corresponding test data set T . We explain how this works below. Also, algorithms, in the case of Naive Bayes (NB) and Tree Augmented Naive Bayes (TAN) classifiers, and scoring methods are used on M to search for the best BN that best matches M .

One such scoring method used is MDL, briefly discussed in section 2.3 as a performance test, can also be used to learning BN models and is formulated as:

$$MDL(B | R) = \frac{\log N}{2} |B| - LL(B | R), \text{ where } LL(B | R) = \sum_{i=1}^N \log(P_B(X_i = x_{ik})), \text{ where } B$$

denotes the BN, R denotes the training data set, P_B denotes the probability density over B and $\{X_i : i = 1, \dots, N\}$ are random variables with states $\{x_{ik} : k = 1, \dots, n\}$.

The first term, $\frac{\log N}{2} |B|$, represents the number of probability components that specify B and acts as a regulator to penalizes those models that are too complex and require calculation of large probability density tables. The second term, $LL(B | R)$, is the likelihood of B given R where the higher the likelihood, the closer B is to modelling the probability density of R . Using MDL requires you to construct all possible DAG's of B and then use the MDL score to single out the best of these models, where the smaller the score the better (Friedman, Geiger & Goldszmidt 1997). However, the number of all possible DAG's is usually too large to search through, so only a selection of those are tested by use of algorithms that systematically select structures by selective step procedures.

On the other hand, use of an *Acceptance Measure* to learn a BN model B^* with N variables can also be used (Jensen 1996):

Let Θ be the joint probability table of the true density constructed from R over U and Let Θ^* be the joint probability table over another specified distribution.

Let B^* be the BN for Θ^* . Then, $Acc(\Theta, B^*) = Size(B^*) + k Distance(\Theta, \Theta^*)$, where k is a positive real number. The distance measure can be a choice of either the Euclidean distance or Cross entropy distance functions. While size is a measure of the number conditional probabilities to be calculated for all states over U . It then becomes desirable to find a BN which minimizes the *Acceptance Measure*. Once again, construction of all possible DAG's over U are required, starting from a large model with the maximum set of possible arcs and then successively deleting links until a model with optimal *Acceptance Measure* is found. Such scoring methods pose an expense in the time

required to construct all possible BN structures and then applying the scoring method to each structure in order to search for the single most reliable model. It has also been demonstrated that learning algorithms using confidence intervals overcome the precision problems that scoring-based methods face in learning BNs (Cheng & Greiner). We introduced these scoring methods to control the over-fitting problem. However Cheng says that these scoring methods have problems.

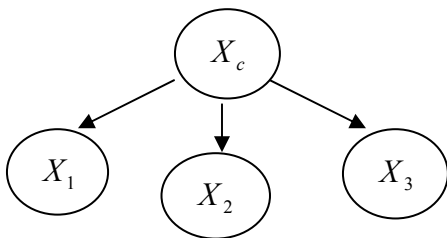
However in the data application of this study, the intuitive approach is used in construction of BN models as well as specific algorithms in the construction of the NB and TAN classifiers.

3.1 Intuitive understanding of causal links

Sometimes knowledge of well known events and their commonly understood causes can shed an intuitive light on causal links and their direction in a BN model. Where the real world is confronted with a change of state as a consequence of the course of an action, we find that causality is the key to explaining such changes. Thus, special correlations can be observed between events of two random variables, X_i & X_j , where having observed X_i will change what we perceive of X_j . In this case it can be said the X_i is the cause of X_j . However if such an observation does not change what you perceive of X_j then you can come to the conclusion that X_i is not the cause of X_j .

3.2 Naive Bayesian Networks (NB Networks)

Naive Bayes networks have a basic universal structure that finds the hypothesis variable, X_c , represented as a sole root node with $\Pi_c = \emptyset$ and all explanatory variables, $\{X_i : i = 1, \dots, N\}$, where $c \neq i$, such that $\Pi_i = \{X_c\}$, are represented as leaf nodes where the root node is the direct parent of all leaf nodes. Here directed arcs are restricted to being directed from the class attribute node to the explanatory attribute nodes only, where no arcs are allowed between the nodes of explanatory attributes. Such a structure has the advantage of possessing a conditional probability density, $P(X_i = x_{ik_i} | X_c = x_{ck_c})$, of the class attribute conditional on all the explanatory attributes in which the classification of the hypothesis variable is able to take all explanatory variables into account. This structure assumes d-separation between all explanatory attributes given the single class attribute. Such conditional independence assumptions of the Naive Bayes network can restrict the model in representing the data comprehensively, where there may be strong correlations between certain explanatory attributes that the model will fail to take into account. However, most notable, is Naive Bayes networks overall reputation as efficient classifiers. The major advantage that Naive Bayes networks have over other BN's is its' easily constructed structure that doesn't require any BN learning procedures involving scoring methods and the like. An example of a NB network is as follows.



3.3 Tree Augmented Naive Bayes Networks (TAN)

The TAN is an extension of the Naive Bayes network where like the Naive Bayes network, it is a structure of a single root node representing the class attribute, X_c with $\Pi_c = \emptyset$, and leaf nodes representing explanatory attributes, $\{X_i : i = 1, \dots, N\}$ and $c \notin i$. However, TAN differs from Naive Bayes where d-separation between the explanatory variables no longer holds and arcs are permitted to run between explanatory nodes with limitations. Such limitations are the result of the TAN model building algorithm involving the formation of an augmented spanning tree amongst the explanatory nodes such that each explanatory node has as its set of parents the root node and at most one other explanatory node, $\Pi_i = \{X_c, X_j : j \in i, j \neq i\}$. The arcs between explanatory node pairs represent conditional dependencies between them and are chosen by calculating a mutual information value for any pair of nodes that specify how much information they share by using conditional mutual information between the corresponding explanatory variable pair given the hypothesis variable. The mutual information function for any pair of explanatory variables is as follows:

$$I_{\hat{P}_M}(X_i, X_j | X_c) = \sum_{X_i=x_{ik_i}, X_j=x_{jk_j}, X_c=x_{ck_c}} P(X_i = x_{ik_i}, X_j = x_{jk_j}, X_c = x_{ck_c}) \times$$

$$\log \frac{P(X_i = x_{ik_i}, X_j = x_{jk_j} | X_c = x_{ck_c})}{P(X_i = x_{ik_i} | X_c = x_{ck_c})P(X_j = x_{jk_j} | X_c = x_{ck_c})}$$

The TAN model construction has two phases where the first is to build an augmented tree structure as detail below as Phase 1. And then the second is to build a structure on top as detailed in Phase 2. The augmented tree structure of a TAN network can be constructed in a five step procedure that is a variation of Chow and Lius' four step procedure for constructing a BN tree (Friedman, Geiger & Goldszmidt 1997).

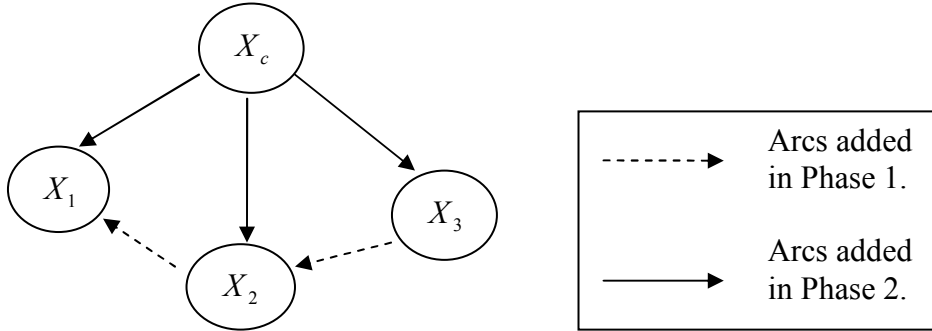
Phase 1:

1. Compute $I_{\hat{P}_M}(X_i, X_j | X_c)$ between each pair of attributes.
2. Build a complete undirected graph with the set of nodes corresponding to all explanatory attributes. Add the weight, $I_{\hat{P}_M}(X_i, X_j | X_c)$, to an arc joining X_i to $X_j \forall i, j$.
3. Build a maximum weight spanning tree using Kruskals' Algorithm as a choice of one of the many maximum weight spanning tree algorithms (Wilson&Watkins).
4. Convert all undirected arcs to directed arcs by choosing a root variable and setting the direction of all arcs to be outwards from it.

Phase 2:

5. Add an extra node for the class attribute, X_c , and then add arcs from X_c to all nodes corresponding to all explanatory attributes, X_i .

The following is an example of a TAN model.



Past studies in classifiers have concluded from results, the superiority of TAN as a high performing classifier in comparison to NB classifiers, while still being equally comparable in with NB network with its computational simplicity (Friedman, Geiger & Goldszmidt 1997).

4 German Credit Data

BN classifiers were applied to German Credit Data. This data was initially published in a paper by Baesens, Egmont-Petersen, Castelo & Vanthienen. For the purposes of this study, the German credit data was obtained from the web address:

<http://www.liacc.up.pt/ML/statlog/datasets/german/german>, where its' originally source was from Professor Dr. Hans Hofmann, Okonometri University, Hamburg. This data set has 1000 observations and 20 variables listed as the 20 attributes, with specific states, in Appendix A.

4.1 Data Preparation

In this study all 20 attributes were used in applying two main tests to the data. The first test involved attributes 7, 8, 12, 16, and 17 as the most relevant variables to included in the structure of Learned BN, A, C, D, E, NB, & TAN classifying models, detailed below. The first test also used attributes 1, 3, 6, 12 and 15 for the structure of BAESSENS model (Egmont-Petersen, Feelders & Baesens 2003). All these classifiers were applied to the data and used to classify the class attribute, attribute 3, as default or non-default. The explanatory attributes used in the first test and their descriptions are as follows:

Attribute 7: time in employment, Attribute 8: instalment rate in percentage of disposable income, Attribute 12: property, Attribute 16: number of existing credits at this bank, Attribute 17: job type, Attribute 1: income, Attribute 6: deposit and Attribute 15: housing.

Also, the class attribute, attribute 3, originally a variable of five states was converted into a binary variable with the states: default and non-default, where the original states as: A30 - no credits taken/all credits paid back duly, A31 - all credits at this bank paid back duly, A32 - existing credits paid back duly till now and A33 - delay in paying off in the past, were combined to form the non-default state. And the remaining state A34 - critical account/other credits existing (not at this bank), was set as the default state. In the final test conducted, all 20 attributes were used in the process of searching for the best set of explanatory attributes to include in a NB model. Since BN require variables represented in the network to be categorical, the continuous data of attributes 2, 5, 8, 11, 13 and 16 were converted to discrete variable, where attributes 2, 5, 11 and 13 were discretized using Chengs' BN software, and attributes 8 and 16 possessed continuous data that naturally followed as categorical data (see Appendix A). In constructing the TAN model, calculations of the mutual information values were made easier by reducing the state space of attributes 7, 8, 12, 16 and 17, where these attributes were converted to represent binary variables by combining their states as follows:

Attribute 7: State 1 = A71 - unemployed, A72 - < 1 year

State 2 = A73 - $1 \leq \dots < 4$ years , A74 - $4 \leq \dots < 7$ years, A75 - ≥ 7 years

Attribute 8: State 1 = 1%, 2%

State 2 = 3%, 4%.

Attribute 12: State 1 = A121 - real estate, A122 - if not A121 : building society savings agreement/life insurance

State 2 = A123 - if not A121/A122 : car or other, not in attribute 6, A124 - unknown / no property .

Attribute 16: let c = Number of existing credits at this bank. Then,

State 1 = $1 \leq c \leq 2$

State 2 = $3 \leq c \leq 4$

Attribute 17: State 1 = A171 - unemployed/ unskilled - non-resident, A172 - unskilled - resident

State 2 = A173 - skilled employee / official, A174 - management/ self-employed/highly qualified employee/ officer.

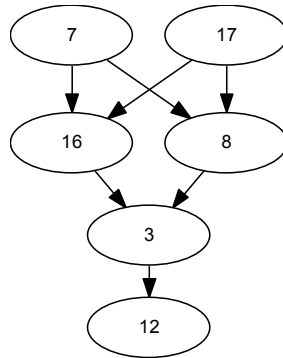
Finally, the data of $M = 1000$ values was split into the training data set consists of the first $R = 700$ data, and the test data set consists of the remaining $T = 300$ data.

4.2 Model Specification

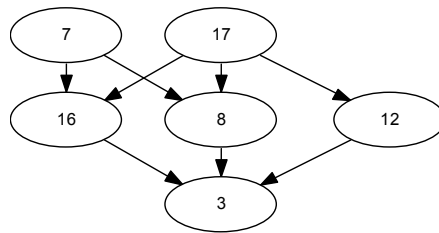
BN Models A, C, D and E were constructed on the basis of intuition alone. While the purpose of the construction of NB models 1-6 was to be able to investigate NB performance on increasing complexity in NB model structure. Starting from NB model 1, an attribute from the set of applicable explanatory attributes 7, 8, 12 and 16 was sequentially added to the structure, according to their perceived increasing importance or association to the class attribute 3. Likewise, TAN Models 2-5 were constructed for the purposes of investigating TAN model performance on increasing complexity in TAN model structure. TAN i is an augmentation of NB i where $i = 2, \dots, 5$. Lastly, BAESSENS Model was taken from the article by Egmont-Petersen, Feelders & Baesens 2003, where

pure curiosity was the reason behind the BAESSENS Model inclusion in analysis, with an interest as to how their model would fair.

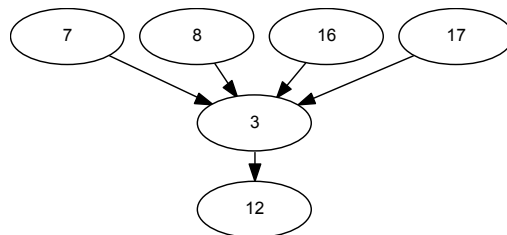
Model A:



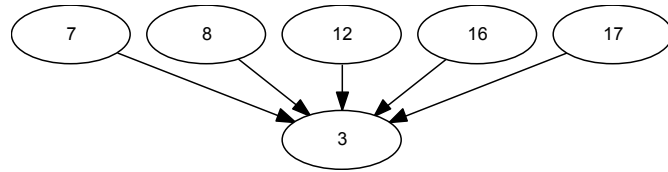
Model C:



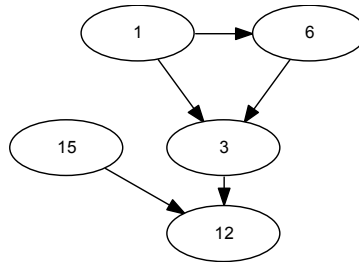
Model D:



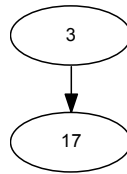
Model E:



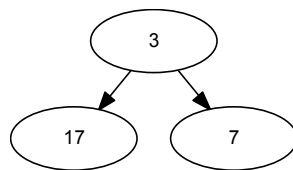
BAESENS Model: (Egmont-Petersen, Feelders, Baesens 2003)



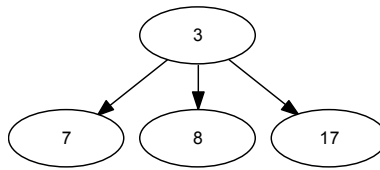
NB Model 1:



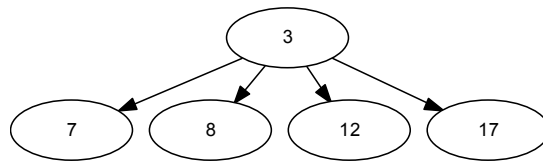
NB Model 2:



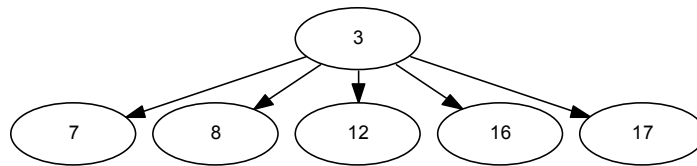
NB Model 3:



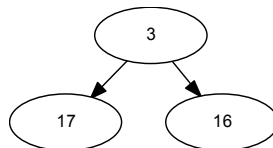
NB Model 4:



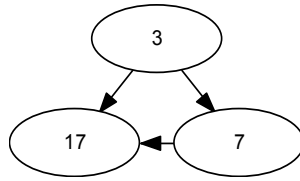
NB Model 5:



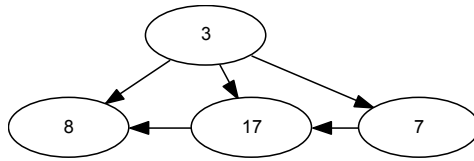
NB Model 6:



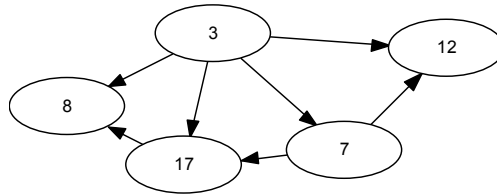
TAN Model 2:



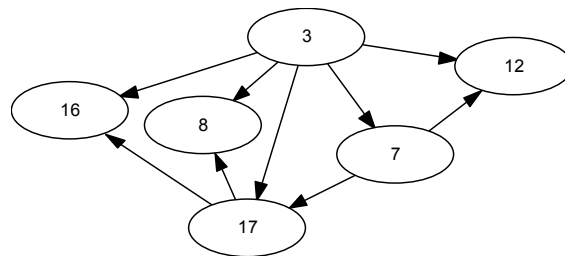
TAN Model 3:



TAN Model 4:



TAN Model 5:



4.3 Software

Hugin .

Chengs' Bayesian Network Program.

This package automatically discretizes continuous data.

4.4 Comparative Results

In searching for the model that best classifies the data M , the statistics used as described in Table 4 were:

Proportion of Correctly Classified Defaulter = $\frac{s}{s+t}$, denoted η .

Proportion of Correctly Classified Non-Defaulter = $\frac{f}{d+f}$, denoted θ .

Proportion of Incorrectly Classified Defaulters = $\frac{d}{d+f}$, denoted α .

Proportion of Incorrectly Classified Non-Defaulters = $\frac{t}{s+t}$, denoted β .

Proportion of Correct Classifications = $\frac{s+f}{s+t+d+f}$, denoted λ .

For each model the estimates $\hat{\alpha}$, $\hat{\beta}$ and $\hat{\lambda}$ (calculated using the test set) were used to compare models. However it is not possible to simultaneously optimise $\hat{\alpha}$ and $\hat{\beta}$.

Relative values of $\hat{\alpha}$ and $\hat{\beta}$ depend on the threshold p used. In this example there is a greater cost weighting of a false positive than there is of a false negative since a financial institution will incur an expensive loss if credit is provided to an incorrectly classified non-defaulter who is actually a defaulter. While the loss of business to the financial institution after denying credit to a credible customer who is a genuine non-defaulter and has been incorrectly classified as a defaulter is a loss not as expensive as the former. In this case small $\hat{\alpha}$ is more important than small $\hat{\beta}$.

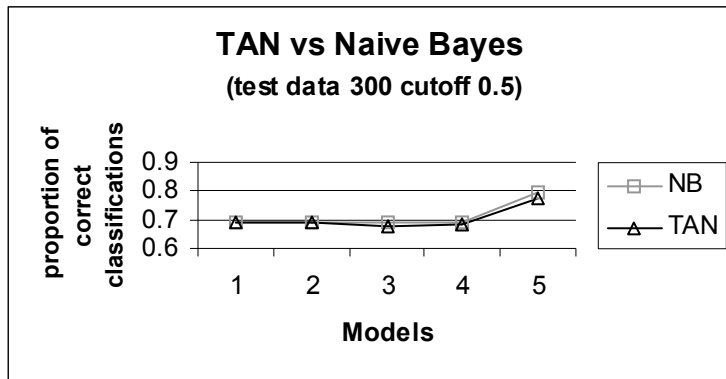
In constructing TAN Model 5, Phase 1 - Step 4 of the TAN model constructing algorithm required the arbitrary choice of a root node from which to organise arc directions of the tree augmented structure in a TAN model. Therefore since the choice of this root node was without specification, tests were conducted to select the root node from the set of explanatory attributes 7,8,12,16 & 17, that would give TAN Model 5 optimal $\hat{\alpha}$, $\hat{\beta}$ and $\hat{\lambda}$. Results in the below table confirmed the node for attribute 7 was the most optimal root node to choose in constructing the TAN model 5, with largest $\hat{\lambda} = 0.803333$ and $\hat{\alpha} = 0.193237$ and $\hat{\beta} = 0.204301$.

Table 5: Choosing the Optimal Root Node of the Tree Augmented TAN Structure.

	correct classification proportion	false non- defaulter proportion	false defaulter proportion
TAN model 5 (root 7,Algorithm Phase 1:Step 4)	0.803333	0.204301	0.193237
TAN model 5 (root 8,Algorithm Phase 1:Step 4)	0.796667	0.204301	0.202899
TAN model 5 (root 12,Algorithm Phase 1:Step 4)	0.796667	0.215054	0.198068
TAN model 5 (root 16,Algorithm Phase 1:Step 4)	0.796667	0.204301	0.202899
TAN model 5 (root 17,Algorithm Phase 1:Step 4)	0.796667	0.204301	0.202899

Extensive studies have gone into demonstrating the significance of TAN and NB models. Some have claimed they lead the class of all classifiers in their computational simplicity and classification accuracy. Some studies have conducted comparative experiments that show TAN models to be better classification performers than NB models (Friedman, Geiger & Goldszmidt), while the contrary has been indirectly shown in other studies (Baesens, Egmont-Pertersen, Castelo & Vanthienen 2002). Thus, in consideration of past comparative studies, tests were conducted to observe the performance of TAN models and NB models. Also included in this experiment and of further interest was how both models fared as their structural complexity increased. Results of this experiment are detailed in the Figure 2 (also see Appendices C & D). Initially classification tests were applied to the test data set while the training data set was used to calculate network probability estimates. The results of this experiment are depicted in Figure 1, where the selected cut-off probability was $p = 0.5$.

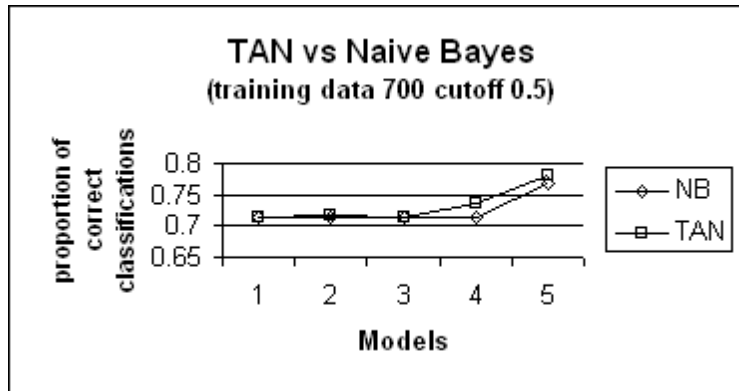
Figure 1



These results on the test data set showed the NB models outperforming the TAN models where TAN Model 5 is the best of the TAN models and NB Model 5 is the best model overall. Surprisingly, the TAN models 3 & 4 were poorer in classification performance than TAN models 1 & 2 despite structural complexity. What would normally be expected would be to see models with greater complexity in their structure to be better classifiers than those of simpler structure. Similarly, the NB models show an almost identical pattern, where the inclusion of attributes 7, 8 and 12 successively did not seem to provide the expected improvement anticipated for the NB model. However, attribute 16 brings a significantly better improvement to both the TAN and NB models as a classifier. This suggests over-fitting that could not be seen in the training data.

Consequently, these results prompted the experiment to re-perform classification tests on the training data, after having learned network probability estimates from the same training data set, in order to check how well the models would perform. The graph below gives details of these results.

Figure 2



Now results on the training data show the TAN models surpassing the NB models in classification precision. TAN Models 2-5 overall do a better job at fitting the training data set than the test data set. Hence, the prevalent signs of over-fitting immerge in the TAN models especially. Also once again, we see the increasing complexity in model structure becoming an insignificant feature in improving classification precision for both TAN and NB models. Here it can be see that the TAN model does no better as a classifier with the inclusion of attribute 8, while the inclusion of attribute 12 in its' structure does provide an improvement when classifying the training data only and conclude that the model is just fitting noise and not stucture. Equivalently, the NB model depicts the same result. But the point is exactly that an improvement in modelling the training data doesn't make a model the best classifier in classifying correctly the test data set, where the obvious over-fitting suggests that however perfectly these models fit the training data it would be erroneous to rely on these models as capable classifiers since they were incapable of doing so on the test data set.

A more extensive comparative test was also performed to compare all the models detailed in Table 6, where for each model Chengs' BN software chose the optimal cut-off p that would give the model optimal classification accuracy (see Appendix B for p values). In this experiment probability estimates were calculated using the training data while the classification tests were performed on the test data set.

Table 6: Comparing classification performance of all models with optimal cut-off p

	Model A	Model C	Model E	NB Model 5
Proportion of correct classifications	0.69	0.69	0.69	0.813333333
Proportion of incorrectly classified non-defaulters	1	1	1	0.161290323
Proportion of incorrectly classified defaulters	0	0	0	0.198067633

	Model D	TAN Model 5	BAESENS Model	NB Model 6
Proportion of correct classifications	0.69	0.803333333	0.67	0.813333
Proportion of incorrectly classified non-defaulters	1	0.204301075	1	0.16129
Proportion of incorrectly classified defaulters	0	0.193236715	0.028985507	0.198068

The results of Table 6 show the best performing classifiers are equally NB Models 5 & 6 with the largest $\hat{\lambda}$ and smallest $\hat{\alpha}$ and $\hat{\beta}$. An even more favourable feature of the NB Models 5 & 6 are the smallest $\hat{\beta}$ where higher cost weighting associated are set to a minimum. The TAN Model 5, although almost as large in $\hat{\lambda}$ is incomparable in minimizing $\hat{\beta}$ compared with NB Models 5 & 6. The worst performing models were

those models that were constructed on the basis of intuition. These intuitively designed Models A, C, E and D possess the most undesirable $\hat{\beta}$ being all at the 100% rate which means that all these models fail completely in identifying all defaults and miss-classifies them all as non-defaulters. In fact these models classify everyone as non-defaulters. Surprisingly, however the BAESENS Model was included in the group of worst performing models and was almost equally as bad as all the intuitively designed Models.

As a matter of interest the experiment of Table 6 was repeated and the optimal p being replaced by a designated constant $p = 0.5$ for all graphs with results showing in Table 7.

Table 7: Comparing classification performance of all models with constant cut-off $p = 0.5$

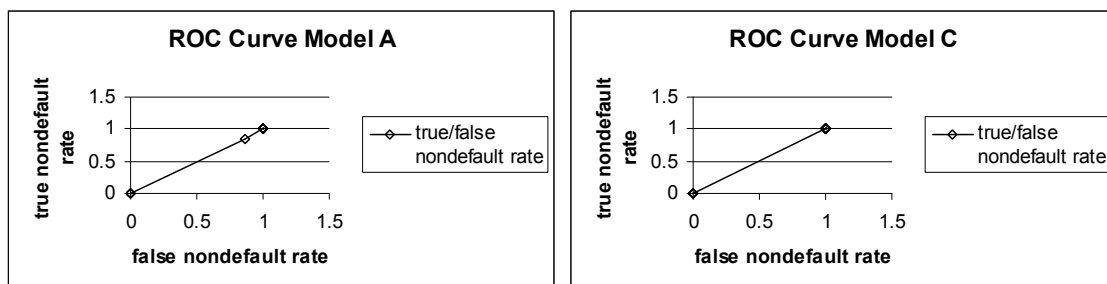
	Model A	Model C	Model E	NB Model 5
Proportion of correct classifications	0.69	0.69	0.69	0.796667
Proportion of incorrectly classified non-defaulters	1	1	1	0.344086
Proportion of incorrectly classified defaulters	0	0	0	0.140097

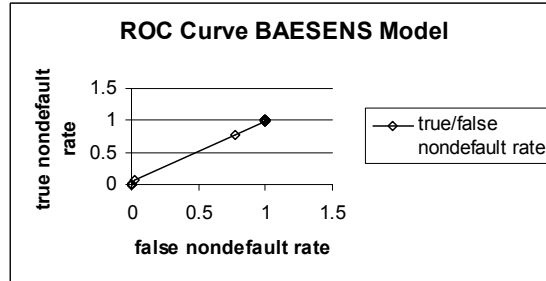
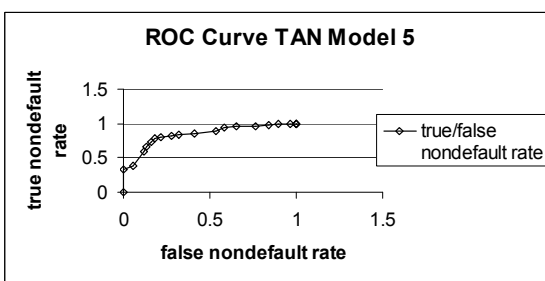
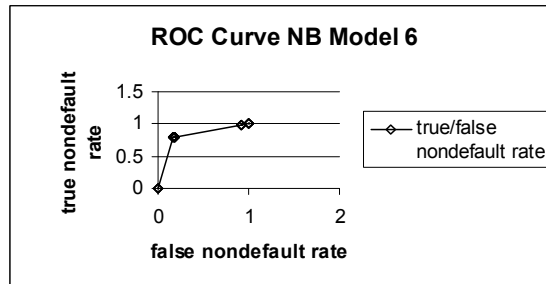
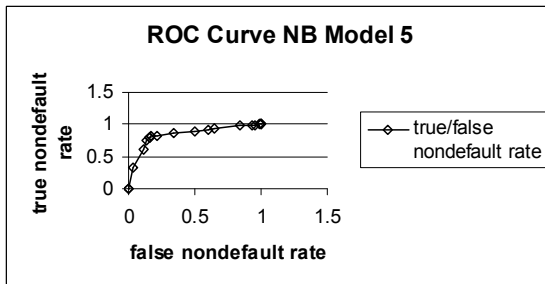
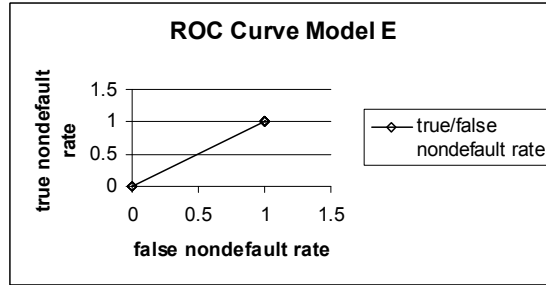
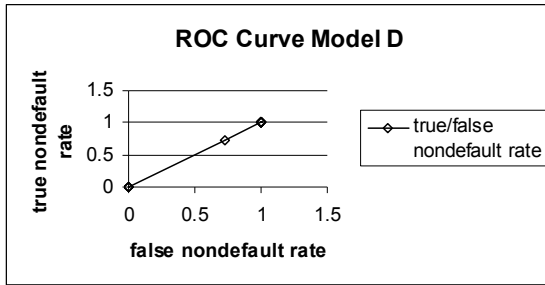
	Model D	TAN Model 5	BAESENS Model	NB Model 6
Proportion of correct classifications	0.69	0.776667	0.69	0.806667
Proportion of incorrectly classified non-defaulters	1	0.408602	1	0.182796
Proportion of incorrectly classified defaulters	0	0.140097	0	0.198068

The results from Table 6 show similar general results to that of Table 7 with the exception of the single outstanding performance of NB Model 6 above all other models.

TAN Model 5 and NB Model 5 now have an unfavorable increase in $\hat{\beta}$. Overall, this then gives reason to re-invent the NB model structure to include only those attributes that seem to make a significant improvement in the classification capabilities of these models, that is, attributes 17 and 16 giving rise to NB Model 6.

For each model ROC curves were constructed to be able to select the most optimal p where the turning point of the ROC curve would graph towards the most top left corner of the graph. As well as this the ROC curves were used to confirm the results of Tables 6 and 7 by providing a rough visual comparison between models where the best performing model would have an ROC curve as far above the $y=x$ line which would typify a good classifying model. Thus, the ROC curves overall confirmed the general results of Table 6 and 7 where the curves of TAN Model 5, BN Model 5 and 6 showed curves coherent with the expected patterns usually possessed by good classifiers. The following graphs are all the ROC curves produced.





Finally, tests were performed to investigate the potential to improve NB Model 6 by adding any other attributes not originally included to the model (Appendix E). Results showed that there was no attribute to include to the NB Model 6 to improve its' successful classification rate.

4.5 Conclusion

As classifier this study found NB models to be the best. The TAN model did not do as well as NB models. However the failure of the TAN model to be the best model may have been due to the small size of the data set. It may be that larger models need larger data set to model on in order to show results that you would expect of a more comprehensive model. The failure of increasing model complexity to perform across both TAN and NB shows the models strongly over-fitting. It would be of interest to investigate whether the same pattern prevails for a larger data set. On the other hand you would expect to see the equivalent but simpler NB model 6 lagg in performance to NB model 5 when modelling a large data set. But this is an expectation to be tested and observed. The poor performance of the intuitive models was an expected result. But the poor performing BAESSENS model was expected to perform a lot better considering it's background.

Appendix A

Hypothesis Variable (class attribute)

Attribute 3: Credit history (qualitative)

A30 : no credits taken/all credits paid back duly

A31 : all credits at this bank paid back duly

A32 : existing credits paid back duly till now

A33 : delay in paying off in the past

A34 : critical account/other credits existing (not at this bank)

Explanatory Variables (explanatory attributes)

Attribute 1: Status of existing checking account (qualitative)

A11 : ... < 0 DM

A12 : 0 <= ... < 200 DM

A13 : ... >= 200 DM /salary assignments for at least 1 year

A14 : no checking account

Attribute 2: Duration in month (numerical)

Attribute 4: Purpose (qualitative)

A40 : car (new)

A41 : car (used)

A42 : furniture/equipment

A43 : radio/television

A44 : domestic appliances

A45 : repairs

A46 : education

A47 : (vacation - does not exist?)

A48 : retraining

A49 : business

A410 : others

Attribute 5: Credit amount (numerical)

Attribute 6: Savings account/bonds (qualitative)

A61 : ... < 100 DM

A62 : 100 <= ... < 500 DM

A63 : 500 <= ... < 1000 DM

A64 : .. >= 1000 DM

A65 : unknown/ no savings account

Attribute 7: Present employment since (qualitative)

A71 : unemployed

A72 : ... < 1 year

A73 : 1 <= ... < 4 years

A74 : 4 <= ... < 7 years

A75 : .. >= 7 years

Attribute 8: Installment rate in percentage of disposable income (numerical)

Attribute 9: Personal status and sex (qualitative)

A91 : male : divorced/separated

A92 : female : divorced/separated/married

A93 : male : single

A94 : male : married/widowed

A95 : female : single

Attribute 10: Other debtors / guarantors (qualitative)

A101 : none
A102 : co-applicant
A103 : guarantor

Attribute 11: Present residence since (numerical)

Attribute 12: Property (qualitative)

A121 : real estate
A122 : if not A121 : building society savings agreement/life insurance
A123 : if not A121/A122 : car or other, not in attribute 6
A124 : unknown / no property

Attribute 13: Age in years (numerical)

Attribute 14: Other installment plans (qualitative)

A141 : bank
A142 : stores
A143 : none

Attribute 15: Housing (qualitative)

A151 : rent
A152 : own
A153 : for free

Attribute 16: Number of existing credits at this bank (numerical)

Attribute 17: Job (qualitative)

A171 : unemployed/ unskilled - non-resident
A172 : unskilled - resident
A173 : skilled employee / official
A174 : management/ self-employed/highly qualified employee/ officer

Attribute 18: Number of people being liable to provide maintenance for (numerical)

Attribute 19: Telephone (qualitative)

A191 : none
A192 : yes, registered under the customers name

Attribute 20: (qualitative)

foreign worker
A201 : yes
A202 : no

Appendix B

Model A

ROC cutoff:

	default	nondefault
default	0	0.47104
non-default	0.46688	0

PREDICTED

OBSERVED	default	non-default	Total
default	0	93	93
non-default	0	207	207
Total	0	300	300

Model C

ROC cutoff:

	default	non-default
Default	0	0.4887
non-default	0.5113	0

PREDICTED

OBSERVED	default	non-default	Total
default	0	93	93
non-default	0	207	207
Total	0	300	300

Model E

ROC cutoff:

	default	non-default
default	0	0.4887
non-default	0.5113	0

PREDICTED

OBSERVED	default	non-default	Total
default	0	93	93
non-default	0	207	207
Total	0	300	300

NB Model 5

ROC cutoff:

	default	non-default
default	0	0.84406
non-default	0.84458	0

PREDICTED

OBSERVED	default	non-default	Total
default	78	15	93
non-default	41	166	207
Total	119	181	300

Model D

ROC cutoff:

	default	non-default
default	0	0.47104
non-default	0.46688	0

PREDICTED

OBSERVED	default	non-default	Total
default	0	93	93
non-default	0	207	207
Total	0	300	300

TAN Model 5

ROC cutoff:

	default	non-default
default	0	0.83206
non-default	0.83206	0

PREDICTED

OBSERVED	default	non-default	Total
default	74	19	93
non-default	40	167	207
Total	114	186	300

BAESENS Model

ROC cutoff:

	default	non-default
default	0	0.48756
non-default	0.4726	0

PREDICTED

OBSERVED	default	non-default	Total
default	0	93	93
non-default	6	201	207
Total	6	294	300

NB Model 6

ROC cutoff:

	default	non-default
Default	0	0.81617
non-default	0.80032	0

PREDICTED

OBSERVED	default	non-default	Total
default	78	15	93
non-default	41	166	207
Total	119	181	300

Appendix C

NB Model 1 test set 300 Attributes 3&17 cutoff=0.5:

Proportion of correct classification 0.295714

NB Model 2 test set 300 Attributes 3,17&7 cutoff=0.5:

Proportion of correct classification 0.295714

NB Model 3 test set 300 Attributes 3,7,8&17 cutoff=0.5:

Proportion of correct classification 0.295714

NB Model 4 test set 300 Attributes 3,7,8,12&17 cutoff=0.5:

Proportion of correct classification 0.295714

NB Model 5 test set 300 Attributes 3,7,8,12,16&17 cutoff=0.5:

Proportion of correct classification 0.341429

TAN Model 2 test set 300 Attributes 3,17&7 cutoff=0.5:

Proportion of correct classification 0.295714

TAN Model 3 test set 300 Attributes 3,7,8&17 cutoff=0.5:

Proportion of correct classification 0.29

TAN Model 4 test set 300 Attributes 3,7,8,12&17 cutoff=0.5:

Proportion of correct classification 0.292857

TAN Model 5 test set 300 Attributes 3,7,8,12,16&17 cutoff=0.5:

Proportion of correct classification 0.332857

Appendix D

NB Model 1 TRAINING set 700 Attributes 3&17 cutoff=0.5:

Proportion of correct classification 0.714286

NB Model 2 TRAINING set 700 Attributes 3,17&7 cutoff=0.5:

Proportion of correct classification 0.714286

NB Model 3 TRAINING set 700 Attributes 3,7,8&17 cutoff=0.5:

Proportion of correct classification 0.714286

NB Model 4 TRAINING set 700 Attributes 3,7,8,12&17 cutoff=0.5:

Proportion of correct classification 0.714286

NB Model 5 TRAINING set 700 Attributes 3,7,8,12,16&17 cutoff=0.5:

Proportion of correct classification 0.767143

TAN Model 2 TRAINING set 700 Attributes 3,17&7 cutoff=0.5:

Proportion of correct classification 0.715714

TAN Model 3 TRAINING set 700 Attributes 3,7,8&17 cutoff=0.5:

Proportion of correct classification 0.714286

TAN Model 4 TRAINING set 700 Attributes 3,7,8,12&17 cutoff=0.5:

Proportion of correct classification 0.735714

TAN Model 5 TRAINING set 700 Attributes 3,7,8,12,16&17 cutoff=0.5:

Proportion of correct classification 0.78

Appendix E

NB Model 6: Attributes 3,16&17

OBSERVED	PREDICTED	
	default	non-default
default	78	15
nondefault	41	166
correct classification proportion	0.813333	

NB Model 6.1: Attributes 3,16,17&12

OBSERVED	PREDICTED	
	default	non-default
default	77	16
nondefault	41	166
correct classification proportion	0.81	

NB Model 6.2: Attributes 3,16,17&1

OBSERVED	PREDICTED	
	default	non-default
default	77	16
nondefault	41	166
correct classification proportion	0.81	

NB Model 6.3: Attributes 3,16,17&2

OBSERVED	PREDICTED	
	default	non-default
default	77	16
nondefault	41	166
correct classification proportion	0.81	

NB Model 6.4: Attributes 3,16,17&4

	PREDICTED	

OBSERVED	default	non-default
default	76	17
nondefault	41	166
correct classification proportion	0.806667	

NB Model 6.5: Attributes 3,16,17&5

	PREDICTED	
OBSERVED	default	non-default
default	78	15
nondefault	41	166
correct classification proportion	0.813333	

NB Model 6.6: Attributes 3,16,17&6

	PREDICTED	
OBSERVED	default	non-default
default	78	15
nondefault	41	166
correct classification proportion	0.813333	

NB Model 6.7: Attributes 3,16,17&7

	PREDICTED	
OBSERVED	default	non-default
default	78	15
nondefault	41	166
correct classification proportion	0.813333	

NB Model 6.8: Attributes 3,16,17&8

	PREDICTED	
OBSERVED	default	non-default
default	78	15
nondefault	41	166
correct classification proportion	0.813333	

NB Model 6.9: Attributes 3,16,17&9

OBSERVED	PREDICTED	
	default	non-default
default	78	15
nondefault	41	166
correct classification proportion	0.813333	

NB Model 6.10: Attributes 3,16,17&10

OBSERVED	PREDICTED	
	default	non-default
default	78	15
nondefault	41	166
correct classification proportion	0.813333	

NB Model 6.11: Attributes 3,16,17&11

OBSERVED	PREDICTED	
	default	non-default
default	78	15
nondefault	41	166
correct classification proportion	0.813333	

NB Model 6.12: Attributes 3,16,17&13

OBSERVED	PREDICTED	
	default	non-default
default	78	15
nondefault	41	166
correct classification proportion	0.813333	

NB Model 6.13: Attributes 3,16,17&14

OBSERVED	PREDICTED	
	default	non-default
default	77	16
nondefault	41	166
correct classification proportion	0.81	

NB Model 6.14: Attributes 3,16,17&15

OBSERVED	PREDICTED	
	default	non-default
default	78	15
nondefault	41	166
correct classification proportion	0.813333	

NB Model 6.15: Attributes 3,16,17&18

OBSERVED	PREDICTED	
	default	non-default
default	78	15
nondefault	41	166
correct classification proportion	0.813333	

NB Model 6.16: Attributes 3,16,17&19

OBSERVED	PREDICTED	
	default	non-default
default	77	16
nondefault	41	166
correct classification proportion	0.81	

NB Model 6.17: Attributes 3,16,17&20

OBSERVED	PREDICTED	
	default	non-default
default	78	15
nondefault	41	166
correct classification proportion	0.813333	

References

1. Jensen, Finn V, (1996) *An introduction to Bayesian Networks*, Springer.
2. Baesens, B., Egmont-Petersen, M., Castelo, R., Vanthienen, J., *Learning Bayesian Network classifiers for credit scoring using Markov Chain Monte Carlo Search* pp.49-52. (2002)
3. Egmont-Petersen, M., Feelders, A., Baesens, B., *Probabilistic network classifiers and probability confidence intervals illustrated by an application in credit scoring*, Elsevier Science, (2003)
4. Friedman, N., Geiger, D., Goldszmidt, M., *Bayesian Network Classifiers*
5. Cheng, J., Greiner, R., *Learning Bayesian Belief Network Classifiers: Algorithms and System*
6. Stephenson, T. A., *An Introduction To Bayesian Network Theory and Usage* (2002)
7. Wilson, K.J., Watkins, J.J., *Graph Theory An Introductory Approach*.