Paper SAS474-2017

# Building Bayesian Network Classifiers Using the HPBNET Procedure

Ye Liu, Weihua Shi, and Wendy Czika, SAS Institute Inc.

## ABSTRACT

A Bayesian network is a directed acyclic graphical model that represents probability relationships and conditional independence structure between random variables. SAS® Enterprise Miner™ implements a Bayesian network primarily as a classification tool; it supports naïve Bayes, tree-augmented naïve Bayes, Bayesian-network-augmented naïve Bayes, parent-child Bayesian network, and Markov blanket Bayesian network classifiers. The HPBNET procedure uses a score-based approach and a constraint-based approach to model network structures. This paper compares the performance of Bayesian network classifiers to other popular classification methods such as classification tree, neural network, logistic regression, and support vector machines. The paper also shows some real-world applications of the implemented Bayesian network classifiers and a useful visualization of the results.

## INTRODUCTION

Bayesian network (BN) classifiers are one of the newest supervised learning algorithms available in SAS Enterprise Miner. The HP BN Classifier node is a high-performance data mining node that you can select from the HPDM toolbar; it uses the HPBNET procedure in SAS® High-Performance Data Mining to learn a BN structure from a training data set. This paper show how the various BN structures that are available in PROC HPBNET can be used as a predictive model for classifying a binary or nominal target.

Because of the practical importance of classification, many other classifiers besides BN classifiers are commonly applied. These classifiers include logistic regression, decision tree, support vector machines, and neural network classifiers. Recent research in supervised learning has shown that the prediction performance of the BN classifiers is competitive when compared to these other classifiers. However, BN classifiers can surpass these competitors in terms of interpretability. A BN can explicitly represent distributional dependency relationships among all available random variables; thus it enables you to discover and interpret the dependency and causality relationships among variables in addition to the target's conditional distribution. In contrast, support vector machines and neural network classifiers are black boxes and logistic regression and decision tree classifiers only estimate the conditional distribution of the target. Therefore, BN classifiers have great potential in real-world classification applications, especially in fields where interpretability is a concern.

SAS Enterprise Miner implements PROC HPBNET to build BN classifiers that can take advantage of modern multithreaded distributed-computing platforms. The HPBNET procedure can build five types of BN classifiers: naïve BN, tree-augmented naïve BN, BN-augmented naïve BN, parent-child BN, and Markov blanket BN. This paper introduces the basic structure of these five types of BN classifiers, explains the key programming techniques and outputs of the HPBNET procedure, and demonstrates useful visualization methods for displaying the structures of the output BN classifiers. This paper also compares the prediction performance of BN classifiers to that of the previously mentioned competitor classifiers by using 25 data sets in the UCI Machine Learning Repository (Lichman 2013).

## BAYESIAN NETWORKS

A Bayesian network is a graphical model that consists of two parts, <**G**, **P**>:

- **G** is a directed acyclic graph (DAG) in which nodes represent random variables and arcs between nodes represent conditional dependency of the random variables.
- **P** is a set of conditional probability distributions, one for each node conditional on its parents.

The following example explains these terms in greater detail.

### EXAMPLE OF A SIMPLE BAYESIAN NETWORK

Figure 1 shows a Bayesian network for a house alarm from Russell and Norvig (2010). It describes the following scenario: Your house has an alarm system against burglary. You live in a seismically active area, and the alarm system can be set off occasionally by an earthquake. You have two neighbors, Mary and John, who do not know each other. If they hear the alarm, they might or might not call you.

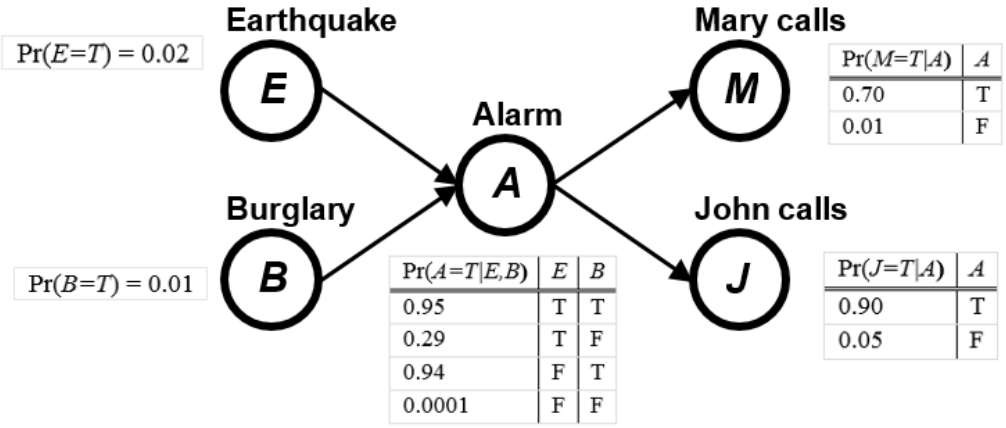


**Figure 1. House Alarm Bayesian Network**

In the house alarm Bayesian network, E, B, A, M, and J are called nodes, and the links between those five nodes are called edges or arcs. Node A is the parent of nodes J and M because the links point from A to J and M; nodes J and M are called the children of node A. Similarly, nodes E and B are the parents of node A; node A is the child of nodes E and B. Those nodes and edges constitute the graph (**G**) part of the Bayesian network model. The conditional probability tables (CPTs) that are associated with the nodes are the probability distribution (**P**) part of the Bayesian network model.

### PROPERTIES OF BAYESIAN NETWORK

Two important properties of a Bayesian network are the following:

- Edges (arcs between nodes) represent "causation," so no directed cycles are allowed.
- Each node is conditionally independent of its ancestors given its parents. This is called Markov property.

According to the Markov property, the joint probability distribution of all nodes in the network can be factored to the product of the conditional probability distributions of each node given its parents. That is,

$$\Pr(\mathrm{G}) = \Pr(X_1, X_2, \ldots, X_p) = \prod_{i=1}^{p} Pr(X_i | \pi(X_i))$$

where $\pi(X_i)$ are the parents of node $X_i$.

In the simplest case, where all the $X_i$ are discrete variables as in the following example, conditional distribution is represented as CPTs, each of which lists the probability that the child node takes on each of its different values for each combination of values of its parents.

In the house alarm example, observe that whether Mary or John calls is conditionally dependent only on the state of the alarm (that is, their parent node). Based on the graph, the joint probability distribution of the events (E,B,A,M, and J) is

$$\Pr(\mathrm{E,B,A,M,J}) = \Pr(\mathrm{J|A}) \cdot \Pr(\mathrm{M|A}) \cdot \Pr(A|E,B) \cdot \Pr(\mathrm{B}) \cdot \Pr(\mathrm{E})$$

The network structure together with the conditional probability distributions completely determine the Bayesian network model.

## SUPERVISED LEARNING USING A BAYESIAN NETWORK MODEL

Now consider this question:

> Suppose you are at work, the house is burglarized ($\mathrm{B} = \mathrm{True}$), there is no earthquake ($\mathrm{E} = \mathrm{False}$), your neighbor Mary calls to say your alarm is ringing ($\mathrm{M} = \mathrm{True}$), but neighbor John doesn't call ($\mathrm{J} = \mathrm{False}$). What is the probability that the alarm went off ($\mathrm{A} = \mathrm{True}$)?

In other words, what is the value of

$$\Pr(\mathrm{A = T|B = T, E = F, M = T, J = F})$$

To simplify the appearance of these equations, $\mathrm{T}$ and $\mathrm{F}$ are used to represent $\mathrm{True}$ and $\mathrm{False}$, respectively.

From the definition of conditional probability,

$$\Pr(\mathrm{A = T|B = T, E = F, M = T, J = F}) = \frac{\Pr(\mathrm{A = T, B = T, E = F, M = T, J = F})}{\Pr(\mathrm{B = T, E = F, M = T, J = F})}$$

According to the equation for $\Pr(\mathrm{E,B,A,M,J})$ from the preceding section and using the values from the conditional probability tables that are shown in Figure 1,

$$\begin{aligned}
&\Pr(\mathrm{A = T, B = T, E = F, M = T, J = F}) \\
&\qquad = \Pr(\mathrm{J = F|A = T})\Pr(\mathrm{M = T|A = T})\Pr(A = T|E = F, B = T)\Pr(\mathrm{B = T})\Pr(\mathrm{E = F}) \\
&\qquad = 0.1 * 0.01 * 0.7 * 0.94 * (1 - 0.02) = 0.00064484
\end{aligned}$$

$$\begin{aligned}
&\Pr(\mathrm{B = T, E = F, M = T, J = F}) = \Pr(\mathrm{A = T, B = T, E = F, M = T, J = F}) + \Pr(\mathrm{A = F, B = T, E = F, M = T, J = F}) \\
&\qquad = 0.00064484 + \Pr(\mathrm{A = F, B = T, E = F, M = T, J = F}) \\
&\qquad = 0.00064484 \\
&\qquad + \Pr(\mathrm{J = F|A = F})\Pr(\mathrm{B = T})\Pr(\mathrm{M = T|A = F})\Pr(A = F|E = F, B = T)\Pr(\mathrm{E = F}) \\
&\qquad = 0.00064484 + (1 - 0.05) * 0.01 * 0.01 * (1 - 0.94) * (1 - 0.02) = 0.000650426
\end{aligned}$$

$$\Pr(\mathrm{A = T|B = T, E = F, M = T, J = F}) = \frac{0.00064484}{0.000650426} \approx 0.99$$

Thus, the conditional probability of the alarm having gone off in this situation is about 0.99. This value can be used to classify (predict) whether the alarm went off.

In general, based on a Bayesian network model, a new observation $X = (x_1, x_2, \dots, x_p)$ is classified by determining the classification of the target Y that has the largest conditional probability,

$$\arg\max_k \Pr(Y = k | x_1, x_2, \dots, x_p)$$

where

$$\Pr\left(Y = k \middle| x_1, x_2, \dots, x_p\right) \propto \Pr\left(Y = k, x_1, x_2, \dots, x_p\right) = \prod_i \Pr(x_i | \pi(X_i)) Pr(Y = k | \pi(Y))$$

Because the target is binary (True or False) in this example, when the value of the preceding equation is greater than 0.5, the prediction is that the alarm went off ($\mathrm{A} = \mathrm{True}$).

## HPBNET PROCEDURE

The HPBNET procedure is a high-performance procedure that can learn different types of Bayesian networks—naïve, tree-augmented naïve (TAN), Bayesian network-augmented naïve (BAN), parent-child Bayesian network (PC), or Markov blanket (MB)—from an input data set. PROC HPBNET runs in either single-machine mode or distributed-computing mode. In this era of big data, where computation performance is crucial for many real-world applications, the HPBNET procedure's distributed-computing mode is very efficient in processing large data sets.

The HPBNET procedure supports two types of variable selection: one by independence tests between each input variable and the target (when PRESCREENING=1), and the other by conditional independence tests between each input variable and the target given any subset of other input variables (when VARSELECT=1, 2, or 3). PROC HPBNET uses specialized data structures to efficiently compute the contingency tables for any variable combination, and it uses dynamic candidate generation to reduce the number of false candidates for variable combinations. If you have many input variables, structure learning can be time-consuming because the number of variable combinations is exponential. Therefore, variable selection is strongly recommended.

To learn a TAN structure, the HPBNET procedure constructs a maximum spanning tree in which the weight for an edge is the mutual information between the two nodes. A maximum spanning tree is a spanning tree of a weighted graph that has maximum weight. If there are K variables in a system, then the corresponding tree structure will have K nodes, and K–1 edges should be added to create a tree structure that connects all the nodes in the graph. Also, the sum of the weights of all the edges needs to be the maximum weight among all such tree structures.

To learn the other BN types, PROC HPBNET uses both of the following approaches:

- The score-based approach uses the BIC (Bayesian information criterion) score to measure how well a structure fits the training data and then tries to find the structure that has the best score. The BIC is defined as

$$\mathrm{BIC}(G, D) = N \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} p(\pi_{ij}) p\left(X_i = v_{ik} \middle| \pi_{ij}\right) \ln p\left(X_i = v_{ik} \middle| \pi_{ij}\right) - \frac{M}{2} \ln N$$

  where $G$ is a network, $D$ is the training data set, $N$ is the number of observations in $D$, $n$ is the number of variables, $X_i$ is a random variable, $r_i$ is the number of levels for $X_i$, $v_{ik}$ is the $k$th value of $X_i$, $q_i$ is the number of value combinations of $X_i$'s parents, $\pi_{ij}$is the $j$th value combination of $X_i$'s parents, and $M = \sum_{i=1}^{n} (r_i - 1) \times q_i$ is the number of parameters for the probability distributions.

- The constraint-based approach uses independence tests (such as a chi-square test or mutual information test) to determine the edges and directions among the nodes as follows: Assume that you have three variables, $X$, $Y$ and $Z$, and that it has been determined (using independence tests) that there are edges between $X$ and $Z$ and $Y$ and $Z$, but no edge between $X$ and $Y$. If $X$ is conditionally independent of $Y$ given any subset of variables $S = \{Z\} \cup S'$, $S' \subseteq \{X, Y, Z\}$, then the directions between $X$ and $Z$ and between $Y$ and $Z$ are $X \rightarrow Z$ and $Y \rightarrow Z$, respectively. Notice that using only independence tests might not be able to orient all edges because some structures are equivalent with respect to conditional independence tests. For example, $X \leftarrow Y \leftarrow Z$, $X \rightarrow Y \rightarrow Z$, and $X \leftarrow Y \rightarrow Z$ belong to the same equivalence class. In these cases, PROC HPBNET uses the BIC score to determine the directions of the edges.

For the PC and MB structures, PROC HPBNET learns the parents of the target first. Then it learns the parents of the input variable that has the highest BIC score with the target. It continues learning the parents of the input variable that has the next highest BIC score, and so on. When learning the parents of a node, it first determines the edges by using independence tests. Then it orients the edges by using both independence tests and the BIC score. PROC HPBNET uses the BIC score not only for orienting the edges but also for controlling the network complexity, because a complex network that has more parents is penalized in the BIC score. Both the BESTONE and BESTSET values of the PARENTING= option try to find the local optimum structure for each node. BESTONE adds the best candidate variable to the parents at each iteration, whereas BESTSET tries to choose the best set of variables among the candidate sets.

### TYPES OF BAYESIAN NETWORK CLASSIFIERS SUPPORTED BY THE HPBNET PROCEDURE

The HPBNET procedure supports the following types of Bayesian network classifiers:

- **Naïve Bayesian network classifier:** As shown in Figure 2, the target node (Y) has a direct edge to each input variable, the target node is the only parent for all other nodes, and there are no other edges. This structure assumes that all input variables are conditionally independent of each other given the target.

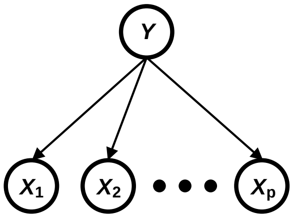


**Figure 2. Naïve Bayesian Network Classifier**

- **Tree-augmented naïve Bayesian network classifier:** As shown in Figure 3, in addition to the edges from the target node Y to each input node, the edges among the input nodes form a tree. This structure is less restrictive than the naïve Bayes structure.

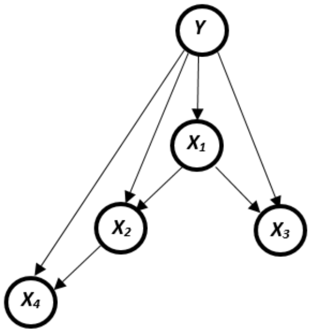


**Figure 3. Tree-Augmented Naïve Bayesian Network Classifier**

- **Bayesian network-augmented naïve Bayesian network classifier:** As shown in Figure 4, the target node Y has a direct edge to each input node, and the edges among the input nodes form a Bayesian network.

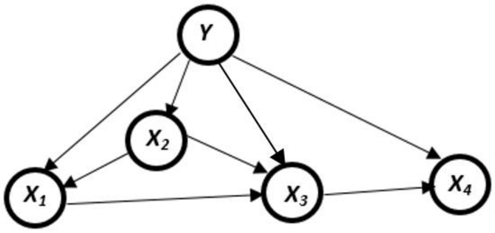


**Figure 4. Bayesian Network-Augmented Naïve Bayesian Network Classifier**

- **Parent-child Bayesian network classifier:** As shown in Figure 5, input variables can be the parents of the target variable Y. In addition, edges from the parents of the target to the children of the target and among the children of the target are also possible.

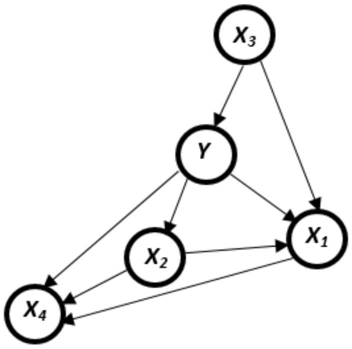


**Figure 5. Parent-Child Bayesian Network Classifier**

- **Markov blanket Bayesian network classifier:** As shown in Figure 6, the Markov blanket includes the target's parents, children, and spouses (the other parents of the target's children).

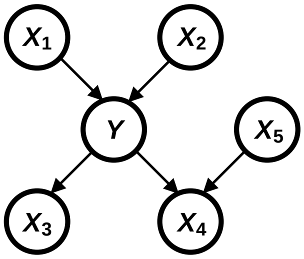


**Figure 6. Markov Blanket Bayesian Network Classifier**

One advantage of PROC HPBNET is that you can specify all the structures that you want to consider for training and request (by specifying the BESTMODEL option) that the procedure automatically choose the best structure based on each model's performance on validation data.

## EXAMPLE OF USING PROC HPBNET TO ANALYZE DATA

This example uses PROC HPBNET to diagnose whether a patient has breast cancer, based on the Breast Cancer Wisconsin data set from the UCI Machine Learning Repository (Lichman 2013).

Table 1 lists the details of the attributes found in this data set.

| Variables | Attribute | Domain | Description of Benign Cells | Description of Cancerous Cells |
|---|---|---|---|---|
| 1 | Sample code number | ID number | N/A | N/A |
| 2 | Clump thickness | 1–10 | Tend to be grouped in monolayers | Often grouped in multiple layers |
| 3 | Uniformity of cell size | 1–10 | Evenly distributed | Unevenly distributed |
| 4 | Uniformity of cell shape | 1–10 | Evenly distributed | Unevenly distributed |
| 5 | Marginal adhesion | 1–10 | Tend to stick together | Tend not to stick together |
| 6 | Single epithelial cell size | 1–10 | Tend to be normal-sized | Tend to be significantly enlarged |
| 7 | Bare nuclei | 1–10 | Typically nuclei are not surrounded by cytoplasm of benign cells | Nuclei might be surrounded by cytoplasm |
| 8 | Bland chromatin | 1–10 | Uniform "texture" of nucleus | Coarser "texture" of nucleus |
| 9 | Normal nucleoli | 1–10 | Very small, if visible | More prominent, and greater in number |
| 10 | Mitoses | 1–10 | Grade of cancer determined by counting the number of mitoses (nuclear division, the process by which the cell divides and replicates) | |
| 11 | Class | 2 or 4 | 2 | 4 |

**Table 1. Attributes of Breast Cancer Wisconsin Data Set**

The RENAME statement in the following DATA step enables you to assign a name to each variable so that you can understand it more easily:

```
data BreastCancer;
set BreastCancer;
rename var1=ID
       var2=Clump_Thickness
       var3=Uniformity_of_Cell_Size
       var4=Uniformity_of_Cell_Shape
       var5=Marginal_Adhesion
       var6=Single_Epithelial_Cell_Size
       var7=Bare_Nuclei
       var8=Bland_Chromatin
       var9=Normal_Nucleoli
       var10=Mitoses
       var11=Class;
run;
```

The following SAS program shows how you can use PROC HPBNET to analyze the BreastCancer data set:

```
proc hpbnet data=BreastCancer nbin=5 structure=Naive TAN PC MB bestmodel;
target Class;
id ID;
input Clump_Thickness Uniformity_of_Cell_Size  Uniformity_of_Cell_Shape
Marginal_Adhesion Single_Epithelial_Cell_Size Bare_Nuclei Bland_Chromatin
Normal_Nucleoli Mitoses/level=INT;
output network=net validinfo=vi varselect=vs
     varlevel=varl parameter=parm fit=fitstats pred=prediction;
partition fraction(validate=0.3 seed=12345);
code file="c:\hpbnetscorecode.sas" ;
run;
```

The TARGET statement specifies **Class** as the target variable. The ID statement specifies **ID** as the ID variable. The INPUT statement specifies that all the other variables are to be used as interval inputs. The NBIN= option in the PROC HPBNET statement specifies 5 for the number of equal-width bins for interval inputs. Four different structures are specified in the STRUCTURE= option (so each structure is trained), and the BESTMODEL option requests that PROC HPBNET automatically choose the best model to minimize the validation misclassification rate. The FRACTION option in the PARTITION statement requests that 30% of the data be used for validation (leaving 70% to be used for training). The OUTPUT statement specifies multiple output tables to be saved in the Work directory. The CODE statement specifies a filename (**hpbnetscorecode.sas**) where the generated score code is to be stored.

After you run PROC HPBNET, you can visualize the final model by using the **%createBNCdiagram** macro in the Appendix to view the selected Bayesian network structure. This macro takes the target variable and the output network data as arguments.

Figure 7 shows the generated diagram, which indicates that the naïve Bayes network is selected as the best structure for this data set, because the input variables are all conditionally independent of each other given the target.

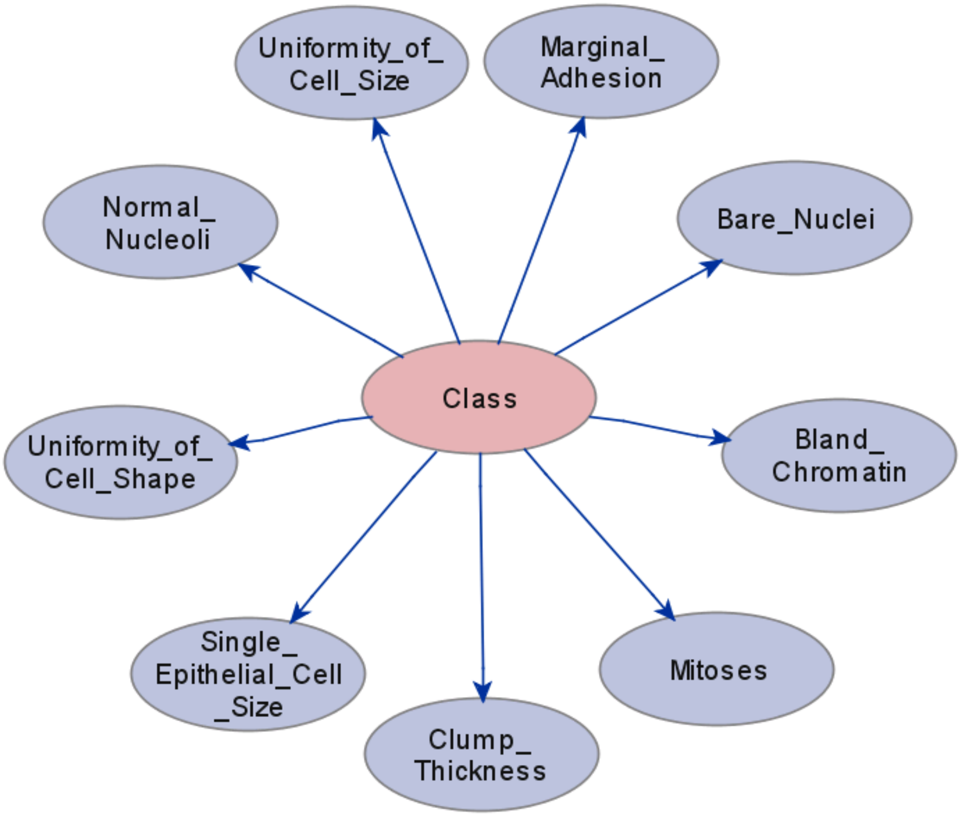
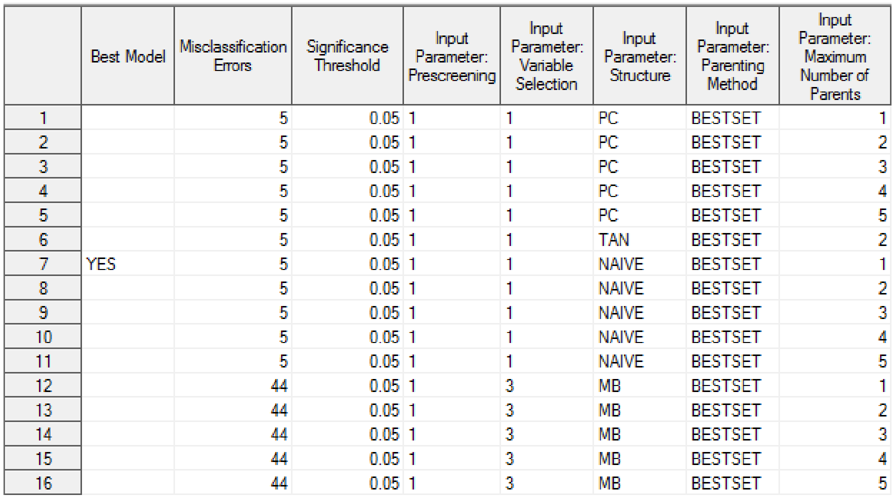


**Figure 7. Bayesian Network Diagram**

Table 2 through Table 7 show all the other output tables, which are stored in the Work directory.

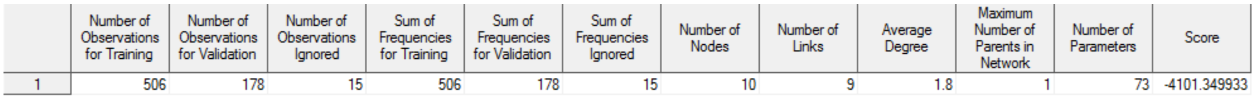
The Best Model column in Table 2 shows that a naïve Bayesian network model with a maximum of one parent is selected, and the Misclassification Errors column shows that five validation observations are misclassified.



| | Best Model | Misclassification Errors | Significance Threshold | Input Parameter: Prescreening | Input Parameter: Variable Selection | Input Parameter: Structure | Input Parameter: Parenting Method | Input Parameter: Maximum Number of Parents |
|---|---|---|---|---|---|---|---|---|
| 1 | | 5 | 0.05 | 1 | 1 | PC | BESTSET | 1 |
| 2 | | 5 | 0.05 | 1 | 1 | PC | BESTSET | 2 |
| 3 | | 5 | 0.05 | 1 | 1 | PC | BESTSET | 3 |
| 4 | | 5 | 0.05 | 1 | 1 | PC | BESTSET | 4 |
| 5 | | 5 | 0.05 | 1 | 1 | PC | BESTSET | 5 |
| 6 | | 5 | 0.05 | 1 | 1 | TAN | BESTSET | 2 |
| 7 | YES | 5 | 0.05 | 1 | 1 | NAIVE | BESTSET | 1 |
| 8 | | 5 | 0.05 | 1 | 1 | NAIVE | BESTSET | 2 |
| 9 | | 5 | 0.05 | 1 | 1 | NAIVE | BESTSET | 3 |
| 10 | | 5 | 0.05 | 1 | 1 | NAIVE | BESTSET | 4 |
| 11 | | 5 | 0.05 | 1 | 1 | NAIVE | BESTSET | 5 |
| 12 | | 44 | 0.05 | 1 | 3 | MB | BESTSET | 1 |
| 13 | | 44 | 0.05 | 1 | 3 | MB | BESTSET | 2 |
| 14 | | 44 | 0.05 | 1 | 3 | MB | BESTSET | 3 |
| 15 | | 44 | 0.05 | 1 | 3 | MB | BESTSET | 4 |
| 16 | | 44 | 0.05 | 1 | 3 | MB | BESTSET | 5 |

**Table 2. Validation Information Table**

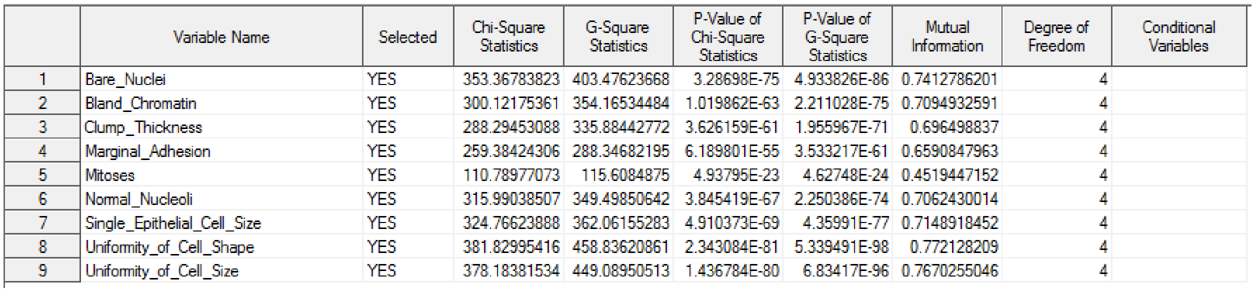
Table 3 shows that the number of observations for validation is 178. Together with the misclassification errors shown in Table 2, you can calculate the validation accuracy as 1 – 5/178 = 97.19%. In PROC HPBNET, continuous variables are binned to equal-width discrete levels in order to simplify the model. If you want to improve this accuracy, you can discretize the interval inputs differently. For example, you could use entropy binning instead of equal-width binning.



| | Number of Observations for Training | Number of Observations for Validation | Number of Observations Ignored | Sum of Frequencies for Training | Sum of Frequencies for Validation | Sum of Frequencies Ignored | Number of Nodes | Number of Links | Average Degree | Maximum Number of Parents in Network | Number of Parameters | Score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 506 | 178 | 15 | 506 | 178 | 15 | 10 | 9 | 1.8 | 1 | 73 | -4101.349933 |

### Table 3. Fit Statistics Table

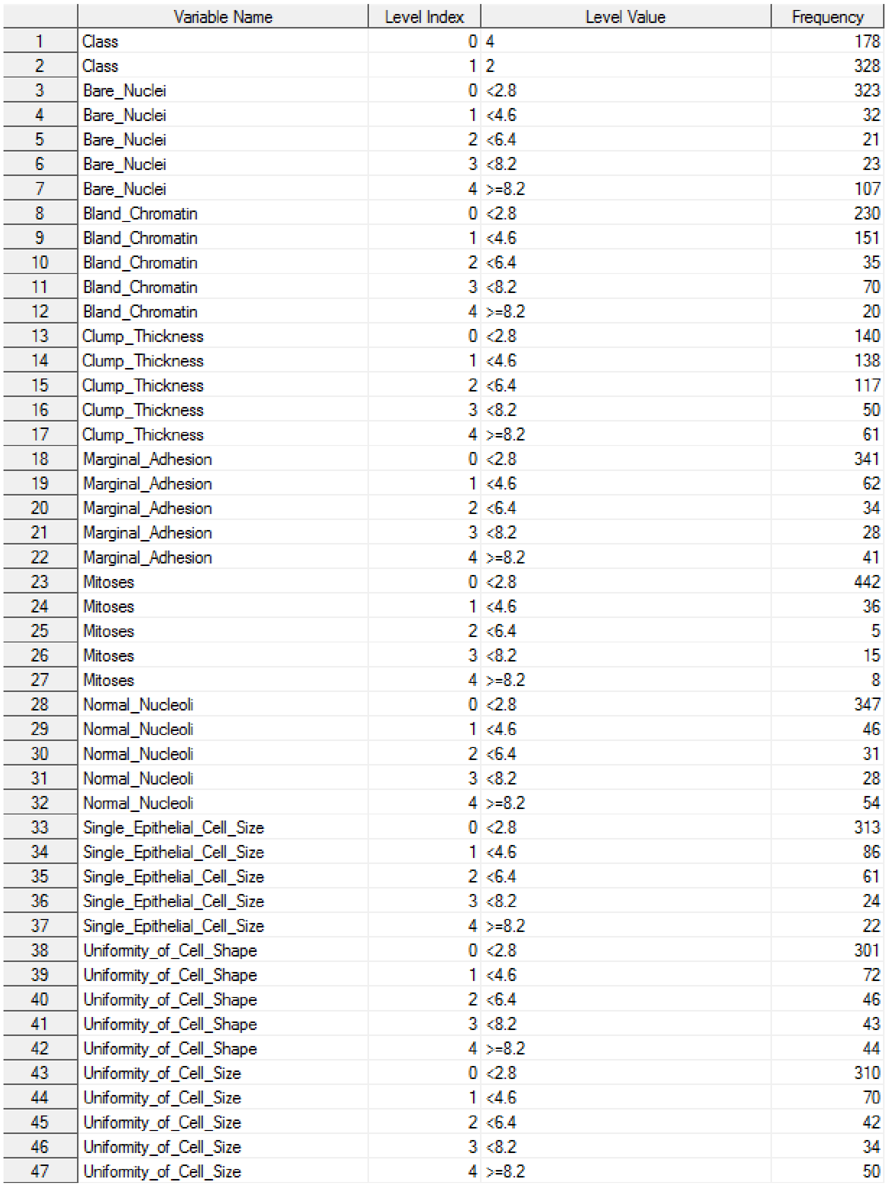
Table 4 shows the variable selection results. In the preceding PROC HPBNET call, the VARSELECT= option is not specified in the PROC statement, so its default value is applied. By default, each input variable is tested for conditional independence of the target variable given any other input variable, and only the variables that are conditionally dependent on the target given any other input variable are selected. Table 4 shows that all the nine input variables are selected into the model.



| | Variable Name | Selected | Chi-Square Statistics | G-Square Statistics | P-Value of Chi-Square Statistics | P-Value of G-Square Statistics | Mutual Information | Degree of Freedom | Conditional Variables |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Bare_Nuclei | YES | 353.36783823 | 403.47623668 | 3.28698E-75 | 4.933826E-86 | 0.7412786201 | 4 | |
| 2 | Bland_Chromatin | YES | 300.12175361 | 354.16534484 | 1.019862E-63 | 2.211028E-75 | 0.7094932591 | 4 | |
| 3 | Clump_Thickness | YES | 288.29453088 | 335.88442772 | 3.626159E-61 | 1.955967E-71 | 0.696498837 | 4 | |
| 4 | Marginal_Adhesion | YES | 259.38424306 | 288.34682195 | 6.189801E-55 | 3.533217E-61 | 0.6590847963 | 4 | |
| 5 | Mitoses | YES | 110.78977073 | 115.6084875 | 4.93795E-23 | 4.62748E-24 | 0.4519447152 | 4 | |
| 6 | Normal_Nucleoli | YES | 315.99038507 | 349.49850642 | 3.845419E-67 | 2.250386E-74 | 0.7062430014 | 4 | |
| 7 | Single_Epithelial_Cell_Size | YES | 324.76623888 | 362.06155283 | 4.910373E-69 | 4.35991E-77 | 0.7148918452 | 4 | |
| 8 | Uniformity_of_Cell_Shape | YES | 381.82995416 | 458.83620861 | 2.343084E-81 | 5.339491E-98 | 0.772128209 | 4 | |
| 9 | Uniformity_of_Cell_Size | YES | 378.18381534 | 449.08950513 | 1.436784E-80 | 6.83417E-96 | 0.7670255046 | 4 | |

### Table 4. Selected Variables Table

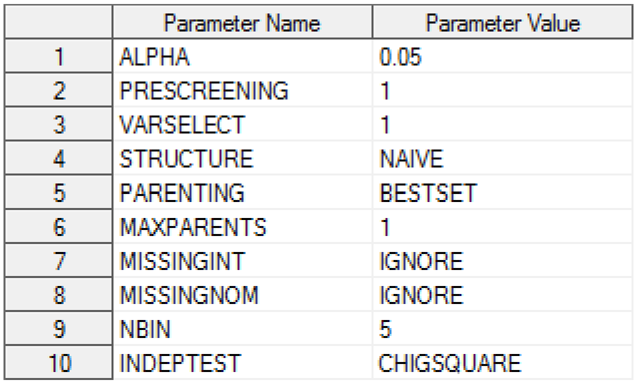
Table 5 shows the details for each level of the target and input variables. The values of 0–4 in the Level Index column indicate that PROC HPBNET bins each interval input variable into five equal-width levels The number of bins can be specified in the NBIN= option; by default, NBIN=5.



| | Variable Name | Level Index | Level Value | Frequency |
|---|---|---|---|---|
| 1 | Class | 0 | 4 | 178 |
| 2 | Class | 1 | 2 | 328 |
| 3 | Bare_Nuclei | 0 | <2.8 | 323 |
| 4 | Bare_Nuclei | 1 | <4.6 | 32 |
| 5 | Bare_Nuclei | 2 | <6.4 | 21 |
| 6 | Bare_Nuclei | 3 | <8.2 | 23 |
| 7 | Bare_Nuclei | 4 | >=8.2 | 107 |
| 8 | Bland_Chromatin | 0 | <2.8 | 230 |
| 9 | Bland_Chromatin | 1 | <4.6 | 151 |
| 10 | Bland_Chromatin | 2 | <6.4 | 35 |
| 11 | Bland_Chromatin | 3 | <8.2 | 70 |
| 12 | Bland_Chromatin | 4 | >=8.2 | 20 |
| 13 | Clump_Thickness | 0 | <2.8 | 140 |
| 14 | Clump_Thickness | 1 | <4.6 | 138 |
| 15 | Clump_Thickness | 2 | <6.4 | 117 |
| 16 | Clump_Thickness | 3 | <8.2 | 50 |
| 17 | Clump_Thickness | 4 | >=8.2 | 61 |
| 18 | Marginal_Adhesion | 0 | <2.8 | 341 |
| 19 | Marginal_Adhesion | 1 | <4.6 | 62 |
| 20 | Marginal_Adhesion | 2 | <6.4 | 34 |
| 21 | Marginal_Adhesion | 3 | <8.2 | 28 |
| 22 | Marginal_Adhesion | 4 | >=8.2 | 41 |
| 23 | Mitoses | 0 | <2.8 | 442 |
| 24 | Mitoses | 1 | <4.6 | 36 |
| 25 | Mitoses | 2 | <6.4 | 5 |
| 26 | Mitoses | 3 | <8.2 | 15 |
| 27 | Mitoses | 4 | >=8.2 | 8 |
| 28 | Normal_Nucleoli | 0 | <2.8 | 347 |
| 29 | Normal_Nucleoli | 1 | <4.6 | 46 |
| 30 | Normal_Nucleoli | 2 | <6.4 | 31 |
| 31 | Normal_Nucleoli | 3 | <8.2 | 28 |
| 32 | Normal_Nucleoli | 4 | >=8.2 | 54 |
| 33 | Single_Epithelial_Cell_Size | 0 | <2.8 | 313 |
| 34 | Single_Epithelial_Cell_Size | 1 | <4.6 | 86 |
| 35 | Single_Epithelial_Cell_Size | 2 | <6.4 | 61 |
| 36 | Single_Epithelial_Cell_Size | 3 | <8.2 | 24 |
| 37 | Single_Epithelial_Cell_Size | 4 | >=8.2 | 22 |
| 38 | Uniformity_of_Cell_Shape | 0 | <2.8 | 301 |
| 39 | Uniformity_of_Cell_Shape | 1 | <4.6 | 72 |
| 40 | Uniformity_of_Cell_Shape | 2 | <6.4 | 46 |
| 41 | Uniformity_of_Cell_Shape | 3 | <8.2 | 43 |
| 42 | Uniformity_of_Cell_Shape | 4 | >=8.2 | 44 |
| 43 | Uniformity_of_Cell_Size | 0 | <2.8 | 310 |
| 44 | Uniformity_of_Cell_Size | 1 | <4.6 | 70 |
| 45 | Uniformity_of_Cell_Size | 2 | <6.4 | 42 |
| 46 | Uniformity_of_Cell_Size | 3 | <8.2 | 34 |
| 47 | Uniformity_of_Cell_Size | 4 | >=8.2 | 50 |

**Table 5. Variable Levels Table**

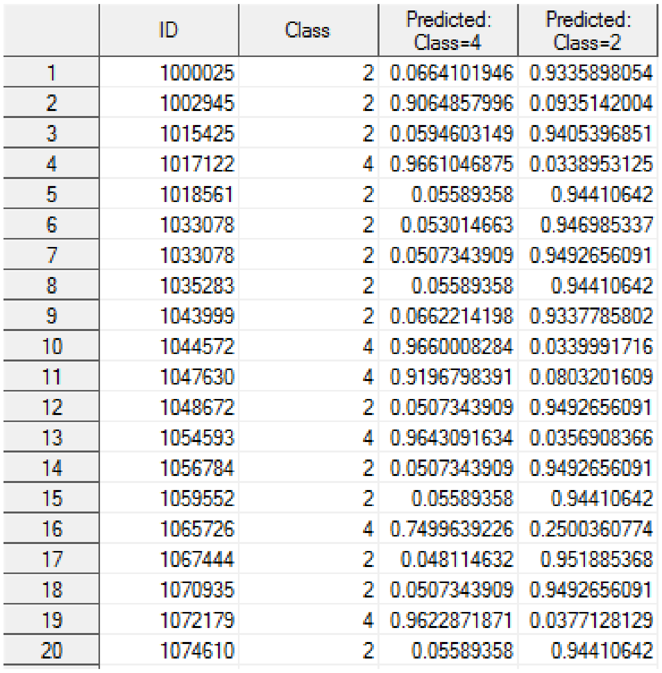
Table 6 shows the parameter values for the resulting model.



| | Parameter Name | Parameter Value |
|---|---|---|
| 1 | ALPHA | 0.05 |
| 2 | PRESCREENING | 1 |
| 3 | VARSELECT | 1 |
| 4 | STRUCTURE | NAIVE |
| 5 | PARENTING | BESTSET |
| 6 | MAXPARENTS | 1 |
| 7 | MISSINGINT | IGNORE |
| 8 | MISSINGNOM | IGNORE |
| 9 | NBIN | 5 |
| 10 | INDEPTEST | CHIGSQUARE |

**Table 6. Parameter Table**

Table 7 shows the prediction results for the first 20 observations of the training data. The Predicted: Class= columns contain the conditional probabilities for the **Class** variable, where Class=2 indicates a benign cell and Class=4 indicates a malignant cell. The conditional probabilities are then used to predict the target class. Here the target is known because these are the training data, but you can use this information to see how well the model is performing. The model is considered to perform well when the actual target class matches the target class that is predicted based on the conditional probabilities.



| | ID | Class | Predicted: Class=4 | Predicted: Class=2 |
|---|---|---|---|---|
| 1 | 1000025 | 2 | 0.0664101946 | 0.9335898054 |
| 2 | 1002945 | 2 | 0.9064857996 | 0.0935142004 |
| 3 | 1015425 | 2 | 0.0594603149 | 0.9405396851 |
| 4 | 1017122 | 4 | 0.9661046875 | 0.0338953125 |
| 5 | 1018561 | 2 | 0.05589358 | 0.94410642 |
| 6 | 1033078 | 2 | 0.053014663 | 0.946985337 |
| 7 | 1033078 | 2 | 0.0507343909 | 0.9492656091 |
| 8 | 1035283 | 2 | 0.05589358 | 0.94410642 |
| 9 | 1043999 | 2 | 0.0662214198 | 0.9337785802 |
| 10 | 1044572 | 4 | 0.9660008284 | 0.0339991716 |
| 11 | 1047630 | 4 | 0.9196798391 | 0.0803201609 |
| 12 | 1048672 | 2 | 0.0507343909 | 0.9492656091 |
| 13 | 1054593 | 4 | 0.9643091634 | 0.0356908366 |
| 14 | 1056784 | 2 | 0.0507343909 | 0.9492656091 |
| 15 | 1059552 | 2 | 0.05589358 | 0.94410642 |
| 16 | 1065726 | 4 | 0.7499639226 | 0.2500360774 |
| 17 | 1067444 | 2 | 0.048114632 | 0.951885368 |
| 18 | 1070935 | 2 | 0.0507343909 | 0.9492656091 |
| 19 | 1072179 | 4 | 0.9622871871 | 0.0377128129 |
| 20 | 1074610 | 2 | 0.05589358 | 0.94410642 |

**Table 7. Prediction Results Table**

## PREDICTION ACCURACY COMPARISON

This section compares the prediction accuracy of Bayesian classifiers to that of their four popular competitor classifiers (decision tree, neural network, logistic regression, and support vector machines) for 25 data sets that were downloaded from the UCI Machine Learning Repository (Lichman 2013). Table 8 summarizes these data sets.

| | Data Set | Attributes | Target Levels | Number of Observations | |
|---|---|---|---|---|---|
| | | | | Total | Validation |
| 1 | **Adult** | 13 | 2 | 48,842 | 16,116 |
| 2 | **Statlog (Australian Credit Approval)** | 14 | 2 | 690 | CV-5 |
| 3 | **Breast Cancer Wisconsin (Original)** (Mangasarian and Wolberg 1990) | 9 | 2 | 699 | CV-5 |
| 4 | **Car Evaluation** | 6 | 4 | 1,728 | CV-5 |
| 5 | **Chess (King-Rook vs. King-Pawn)** | 36 | 2 | 3,196 | 1,066 |
| 6 | **Diabetes** | 8 | 2 | 768 | CV-5 |
| 7 | **Solar Flare** | 10 | 2 | 1,066 | CV-5 |
| 8 | **Statlog (German Credit Data)** | 24 | 2 | 1,000 | CV-5 |
| 9 | **Glass Identification** | 9 | 6 | 214 | CV-5 |
| 10 | **Heart Disease** | 13 | 2 | 270 | CV-5 |
| 11 | **Hepatitis** | 19 | 2 | 155 | CV-5 |
| 12 | **Iris** | 4 | 3 | 150 | CV-5 |
| 13 | **LED Display Domain + 17 Irrelevant Attributes** | 24 | 10 | 3,190 | 1,057 |
| 14 | **Letter Recognition** | 16 | 26 | 20,000 | 4,937 |
| 15 | **Lymphography** | 18 | 4 | 148 | CV-5 |
| 16 | **Nursery** | 8 | 5 | 12,960 | 4,319 |
| 17 | **Statlog (Landsat Satellite)** | 36 | 6 | 6,435 | 1,930 |
| 18 | **Statlog (Image Segmentation)** | 19 | 7 | 2,310 | 770 |
| 19 | **Soybean (Large)** | 35 | 19 | 683 | CV-5 |
| 20 | **SPECT Heart** | 22 | 2 | 267 | CV-5 |
| 21 | **Molecular Biology (Splice-Junction Gene Sequences)** | 60 | 3 | 3,190 | 1,053 |
| 22 | **Tic-Tac-Toe Endgame** | 9 | 2 | 958 | CV-5 |
| 23 | **Statlog (Vehicle Silhouettes)** | 18 | 4 | 846 | CV-5 |
| 24 | **Congressional Voting Records** | 16 | 2 | 435 | CV-5 |
| 25 | **Waveform Database Generator (Version 1)** | 21 | 3 | 5,000 | 4,700 |

**Table 8 Summary of 25 UCI Data Sets**

For the larger data sets, the prediction accuracy was measured by the holdout method (that is, the learning process randomly selected two-thirds of the observations in the data set for building the classifiers, and then evaluated their prediction accuracy on the remaining observations in the data set). For smaller data sets, the prediction accuracy was measured by five-fold cross validation (CV-5). Each process was repeated five times. Observations that have missing values were removed from the data sets. All continuous variables in the data set were discretized with a tree-based binning method. The final average prediction accuracy values and their standard deviations are summarized in Table 9. The best accuracy values for each data set are marked in bold in each row of the table. You can see that PC and TAN in the five BN structures claim most of the wins and are competitive to the other classifiers.

| Data Set | BN Classifiers | | | | | Competitor Classifiers | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Naïve Bayes | BAN | TAN | PC | MB | Logistic | NN | Tree | SVM* |
| 1 Adult | 78.06+- 0.24 | 80.93+- 0.34 | 79.81+- 0.42 | 85.00+- 0.25 | 49.61+- 0.37 | 81.17+- 6.24 | **85.84+- 0.27** | 85.28+- 0.13 | 85.73+- 0.29 |
| 2 Statlog (Australian Credit Approval) | **86.43+- 0.33** | 86.29+- 0.30 | 85.88+- 0.33 | 86.20+- 0.54 | 85.51+- 0.00 | 82.38+- 4.71 | 85.59+- 0.78 | 84.96+- 0.42 | 85.65+- 0.27 |
| 3 Breast Cancer Wisconsin (Original) (Mangasarian and Wolberg 1990) | **97.42+- 0.00** | **97.42+- 0.00** | 96.65+- 0.39 | 97.17+- 0.12 | 96.88+- 0.40 | 95.82+- 0.57 | 96.54+- 0.45 | 94.11+- 0.40 | 96.42+- 0.20 |
| 4 Car Evaluation | 80.01+- 0.21 | 86.56+- 1.03 | 87.52+- 0.10 | 88.24+- 0.90 | 86.52+- 1.27 | 77.26+- 0.26 | 93.07+- 0.49 | **96.89+- 0.36** | |
| 5 Chess (King-Rook vs. King-Pawn) | 90.41+- 0.72 | 95.31+- 0.38 | 95.12+- 0.38 | 95.01+- 0.56 | 92.25+- 0.91 | 52.25+- 0.00 | 96.92+- 0.56 | **99.04+- 0.39** | 97.17+- 0.54 |
| 6 Diabetes | 76.07+- 0.67 | 76.02+- 0.69 | 74.97+- 1.17 | **78.10+- 0.70** | 72.71+- 1.22 | 75.86+- 2.98 | 77.29+- 1.03 | 75.94+- 0.95 | 77.63+- 0.89 |
| 7 Solar Flare | 73.58+- 0.79 | 73.94+- 0.92 | 73.60+- 0.78 | 80.02+-1.08 | 77.60+- 1.81 | 81.54+- 0.22 | 81.69+- 0.56 | 81.07+- 0.45 | **82.18+- 0.42** |
| 8 Statlog (German Credit Data) | 71.60+- 0.55 | 71.28+- 1.02 | 71.94+- 1.29 | **76.18+- 0.37** | 66.40+- 1.47 | 75.24+- 0.50 | 75.04+- 0.34 | 72.18+- 0.59 | 75.86+- 0.76 |
| 9 Glass Identification | 65.61+- 2.28 | 65.61+- 2.28 | **71.68+- 1.02** | 69.53+- 1.42 | 69.53+- 1.42 | 62.80+- 3.70 | 70.37+- 3.54 | 69.81+- 1.43 | |
| 10 Heart Disease | 82.89+- 1.21 | 83.56+- 1.35 | 82.74+- 1.07 | 83.33+- 0.69 | 80.52+- 1.19 | 83.26+- 2.05 | **84.67+- 1.30** | 81.41+- 1.32 | 84.15+- 1.66 |
| 11 Hepatitis | 86.60+- 1.86 | 86.61+- 1.20 | 88.73+- 2.60 | 90.56+- 1.34 | 92.11+- 1.94 | 88.69+- 3.25 | 91.59+- 1.85 | **92.12+- 1.35** | 91.06+- 1.22 |
| 12 Iris | **95.86+- 0.30** | **95.86+- 0.30** | 95.19+- 0.74 | **95.86+- 0.30** | **95.86+- 0.30** | 80.37+- 0.72 | 94.92+- 1.40 | 94.53+- 0.86 | |
| 13 LED Display Domain + 17 Irrelevant Attributes | 73.96+- 1.22 | 73.96+- 1.22 | 74.25+- 0.88 | 74.27+-1.17 | **74.70+- 1.21** | 19.79+- 0.73 | 73.25+- 0.39 | 74.08+- 0.92 | |
| 14 Letter Recognition | 68.33+- 0.58 | 73.19+- 0.77 | **78.75+- 0.63** | 72.07+- 0.63 | 70.80+- 5.37 | 10.98+- 0.27 | 78.69+- 0.46 | 77.66+- 0.43 | |
| 15 Lymphography | 80.81+- 1.56 | 81.49+- 1.83 | 79.32+- 0.77 | **83.78+- 1.51** | 74.19+- 3.71 | 61.62+- 3.89 | 81.35+- 1.56 | 74.86+- 0.88 | |
| 16 Nursery | 82.92+- 0.65 | 86.46+- 0.69 | 89.25+- 0.39 | 91.45+- 0.63 | 91.02+- 0.25 | 90.86+- 0.34 | 92.27+- 0.47 | **97.41+- 0.16** | |
| 17 Statlog (Landsat Satellite) | 81.39+- 0.73 | 86.36+- 0.51 | 86.31+- 0.79 | 86.58+- 0.49 | 84.56+- 0.65 | 72.78+- 0.29 | **87.84+- 0.60** | 85.55+- 0.38 | |
| 18 Statlog (Image Segmentation) | 89.45+- 0.71 | 91.09+- 1.71 | 93.04+- 0.81 | 91.09+- 1.71 | 67.01+- 2.34 | 58.83+- 3.24 | 92.78+- 0.90 | **93.56+- 0.74** | |
| 19 Soybean (Large) | 89.78+- 0.35 | 89.78+- 0.35 | **92.97+- 0.99** | 89.43+- 0.44 | 60.97+- 2.80 | 44.22+- 3.67 | 91.80+- 0.51 | 91.65+- 1.01 | |
| 20 SPECT Heart | 72.06+- 1.65 | 75.36+- 1.04 | 73.41+- 1.38 | 80.60+- 1.25 | 69.96+- 2.74 | 78.35+- 1.66 | **82.25+- 1.20** | 79.33+- 1.51 | 81.95+- 1.97 |
| 21 Molecular Biology (Splice-Junction Gene Sequences) | 95.31+- 0.51 | 95.38+- 0.47 | 95.71+- 0.71 | **96.05+- 0.16** | 92.61+- 7.13 | 80.46+- 1.61 | 95.48+- 0.70 | 94.17+- 0.62 | |
| 22 Tic-Tac-Toe Endgame | 66.08+- 1.49 | 79.04+- 1.58 | 72.03+- 0.70 | 77.14+- 0.82 | 75.03+- 3.02 | 77.10+- 0.80 | 98.10+- 0.09 | 93.28+- 0.67 | **98.33+- 0.00** |
| 23 Statlog (Vehicle Silhouettes) | 62.01+- 0.84 | 70.26+- 1.29 | **71.25+- 0.80** | 70.26+- 1.39 | 58.96+- 5.60 | 63.55+- 1.77 | 70.09+- 0.91 | 69.36+- 0.48 | |
| 24 Congressional Voting Records | 94.80+- 0.53 | 95.17+- 0.16 | 95.13+- 0.72 | 94.90+- 0.10 | 94.99+- 0.38 | 93.79+- 2.11 | **95.82+- 0.99** | 95.08+- 0.42 | 95.40+- 0.43 |
| 25 Waveform Database Generator(Version 1) | 78.31+- 1.48 | 78.31+- 1.48 | 73.68+- 1.77 | 78.35+- 1.33 | 78.62+- 1.50 | 62.43+- 3.43 | **81.78+- 0.85** | 70.27+- 3.06 | |

*SVM for binary target only

**Table 9. Classification Accuracy on 25 UCI Machine Learning Data Sets**

## CONCLUSION

This paper describes Bayesian network (BN) classifiers, introduces the HPBNET procedure, and shows how you can use the procedure to build BN classifiers. It also compares the competitive prediction power of BN classifiers with other state-of-the-art classifiers, and shows how you can use a SAS macro to visualize the network structures.

## REFERENCES

Lichman, M. 2013. “UCI Machine Learning Repository.” School of Information and Computer Sciences, University of California, Irvine. http://archive.ics.uci.edu/ml.

Russell, S., and Norvig, P. 2010. *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, New Jersey: Pearson.

Mangasarian, O. L., and Wolberg, W. H. 1990. "Cancer Diagnosis via Linear Programming", *SIAM News*, 23 (September): 1, 18.

## ACKNOWLEDGMENTS

The authors would like to thank Yongqiao Xiao for reviewing this paper and Anne Baxter for editing it.

The breast cancer database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg.

The lymphography domain was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. Thanks go to M. Zwitter and M. Soklic for providing the data.

The Vehicle Silhouettes data set comes from the Turing Institute, Glasgow, Scotland.

The creators of heart disease data set are:

1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.

2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.

3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.

4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Ye Liu
ye.liu@sas.com

Weihua Shi
weihua.shi@sas.com

Wendy Czika
wendy.czika@sas.com

## APPENDIX

```
%macro createBNCdiagram(target=Class, outnetwork=net);

   data outstruct;
        set &outnetwork;
        if strip(upcase(_TYPE_)) eq 'STRUCTURE' then output;
        keep _nodeid_   _childnode_  _parentnode_;
   run;

   data networklink;
       set outstruct;
        linkid = _N_;
        label linkid ="Link ID";
   run;

   proc sql;
      create table work._node1 as
         select distinct  _CHILDNODE_ as  node
         from networklink;
      create table work._node2  as
         select distinct _PARENTNODE_  as node
         from networklink;
   quit;

   proc sql;
      create table work._node as
         select node
         from work._node1
         UNION
         select node
         from work._node2;
   quit;

   data bnc_networknode;
       length NodeType $32.;
       set work._node;
       if strip(upcase(node)) eq strip(upcase("&target")) then do;
         NodeType = "TARGET";
         NodeColor=2;
       end;
       else  do;
         NodeType = "INPUT";
         NodeColor = 1;
       end;
       label NodeType ="Node Type" ;
       label NodeColor ="Node Color" ;

   run;

   data parents(rename=(_parentnode_ = _node_)) children(rename=(_childnode_
= _node_)) links;
       length _parentnode_ _childnode_ $ 32;
       set networklink;
       keep _parentnode_ _childnode_ ;
   run;
   /*get list of all unique nodes*/
   data nodes;
       set parents children;
   run;

   proc sort data=nodes;
       by _node_;
   run;

   data nodes;
       set nodes;
       by _node_;
       if first._node_;
      _Parentnode_ = _node_;
      _childnode_ = "";
   run;

   /*merge node color and type */
   data nodes;
       merge nodes bnc_
 networknode (rename=(node=_node_ nodeColor=_nodeColor_
nodeType=_nodeType_));
       by _node_;
   run;

   /*sort color values to ensure consistent color mapping across networks */
   /*note that the color mapping is HTML style dependent though */
   proc sort data=nodes;
       by  _nodeType_;
   run;

   /*combine nodes and links*/
   /* need outsummaryall for model report*/
   data bnc_networksummary(drop=_shape_ _nodecolor_ _nodepriority_ _shape_
_nodeID_ _nodetype_ _linkdirection_) bnc_networksummaryall;
       length _parentnode_ _childnode_ $ 32;
       set nodes links;
       drop _node_;
       if _childnode_ EQ "" thendo;
               _nodeID_ = _parentnode_;
               _nodepriority_ = 1;
               _shape_= "OVAL";
           end;
       else do;
         _linkdirection_ = "TO";
         output bnc_networksummary;
       end;
       output bnc_networksummaryall;
       label _linkdirection_="Link Direction";
   run;

    proc datasets lib=work nolist nowarn;
         delete _node _node1 _node2 nodes links parents children;
   run;

   quit;
   proc template;
      define statgraph bpath;
         begingraph / DesignHeight=720 DesignWidth=720;
            entrytitle "Bayesian Network Diagram";
            layout region;
              pathdiagram fromid=_parentnode_ toid=_childnode_ /
              arrangement=GRIP
              nodeid=_nodeid_
              nodelabel=_nodeID_
              nodeshape=_shape_
              nodepriority=_nodepriority_
              linkdirection=_linkdirection_
              nodeColorGroup=_NodeColor_
                        textSizeMin = 10
               ;
            endlayout;
         endgraph;
      end;
   run;

   ods graphics;
   proc sgrender data=bnc_networksummaryall template=bpath;
   run;

%mend;

%createBNCdiagram;
```