Entropy and MDL Discretization of Continuous Variables for Bayesian Belief Networks

Ellis J. Clarke,* Bruce A. Barton
*Maryland Medical Research Institute, 600 Wyndhurst Ave.,
Baltimore, Maryland 21210*

An efficient algorithm for partitioning the range of a continuous variable to a discrete number of intervals, for use in the construction of Bayesian belief networks (BBNs), is presented here. The partitioning minimizes the information loss, relative to the number of intervals used to represent the variable. Partitioning can be done prior to BBN construction or extended for repartitioning during construction. Prior partitioning allows either Bayesian or minimum descriptive length (MDL) metrics to be used to guide BBN construction. Dynamic repartitioning, during BBN construction, is done with a MDL metric to guide construction. The methods are demonstrated with data from two epidemiological studies and these results are compared for all of the methods. The use of the partitioning algorithm resulted in more sparsely connected BBNs, than with binary partitioning, with little information loss from mapping continuous variables into discrete ones. © 2000 John Wiley & Sons, Inc.

I. INTRODUCTION

In the course of conducting clinical trials or epidemiological studies, a large amount of data may be collected. Often, exploratory data analysis is done to get an indication of the interactions of many study variables. This process is not an exhaustive data analysis, but allows the construction of a model of the interactions between study variables to guide further analysis.

This paper develops a general method for exploratory data analysis (EDA) for medical studies, which provides a common scaling for all types of variables. One of the major problems in this context is combining nominal, discrete, and continuous variables in the same model. The following sections develop this theme, using Bayesian belief networks (BBNs), and emphasize methods to convert continuous variables to discrete ones. New methods for discretization of continuous variables, based upon information theory, are presented here. Also,

*Author to whom correspondence should be addressed.

CCC 0884-8173/00/010061-32

these methods are applied using either a Bayesian or a minimum descriptive length (MDL) metric to guide the discretization and the results are compared.

Pearl presents a Bayesian method for constructing a probabilistic network from a database of records.[1,2] The network can provide insight into the probabilistic dependencies that exist among the variables in a database. The computer program searches for a network structure that has a high posterior probability, given the database, and outputs its structure and its probability. A Bayesian belief structure is a directed acyclic graph (DAG) in which nodes represent domain variables and arcs between nodes represent probabilistic dependencies. Variables may be continuous or discrete. The representation of conditional dependencies and independencies is an essential function of a belief network. The belief structure is augmented by conditional probabilities to form a Bayesian belief network. For each node in a belief network, there is a conditional-probability function that relates this node to its immediate predecessors (parents).

In order to develop a general method using BBNs for exploratory data analysis in large medical study databases with continuous measurements, the characteristics of the continuous variable frequency distributions can be empirically used to preprocess the data and to augment BBN construction. This is done to simplify and to clarify the network structure for a given database.

An efficient new algorithm is presented to partition the value range of a continuous random variable. This algorithm uses the characteristics of the information content (entropy) of the continuous variable for the partitioning and does not require comparison with another variable, as do previous methods. Test results show that BBNs, constructed from continuous variables discretized by this algorithm, demonstrate stronger dependencies than comparable BBNs with equal interval partitioning.

The method used above for entropy based discretization is extended to dynamically repartition a continuous variable's values in relation to another variable. An information theoretic metric, which guides BBN construction, is used to guide the repartitioning. BBNs constructed in this way show stronger, and more unexpected, dependencies among variables than those constructed with only entropy partitioning.

A continuous variable from the NHLBI growth and health study[3] (NGHS) study is used to briefly illustrate these methods. The variable is the first year's body mass index measurement. The original continuous frequency distribution is shown in Figure 1(a), accompanied by the discrete distribution resulting from binary equal interval partitioning. The split point for the partitioning is shown on the original distribution.

A better partitioning of the continuous distribution uses equal frequency intervals, which are quintiles in the example in Figure 1(b). However, the resulting uniform distribution is not representative of the original one.

The algorithm for entropy discretization was used to partition the continuous distribution in Figure 2(a). Here, the intervals are of varying lengths and frequencies to minimize the information loss due to discretization. The discrete distribution, in Figure 2(a), gives a very good approximation of the original distribution.

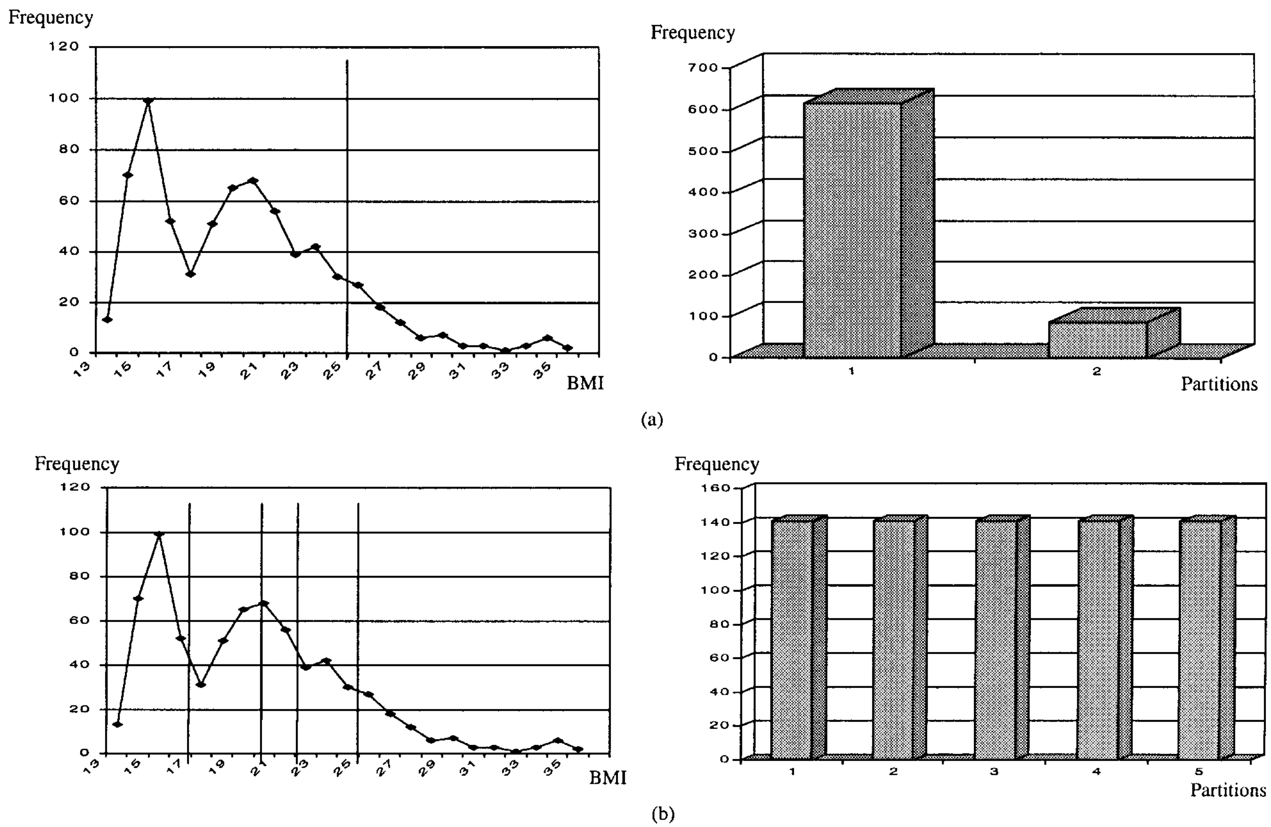

Figure 1. NGHS first year body mass index. Original distribution and equal interval or frequency partitioning. (a) Binary partitioning, (b) quintile partitioning.

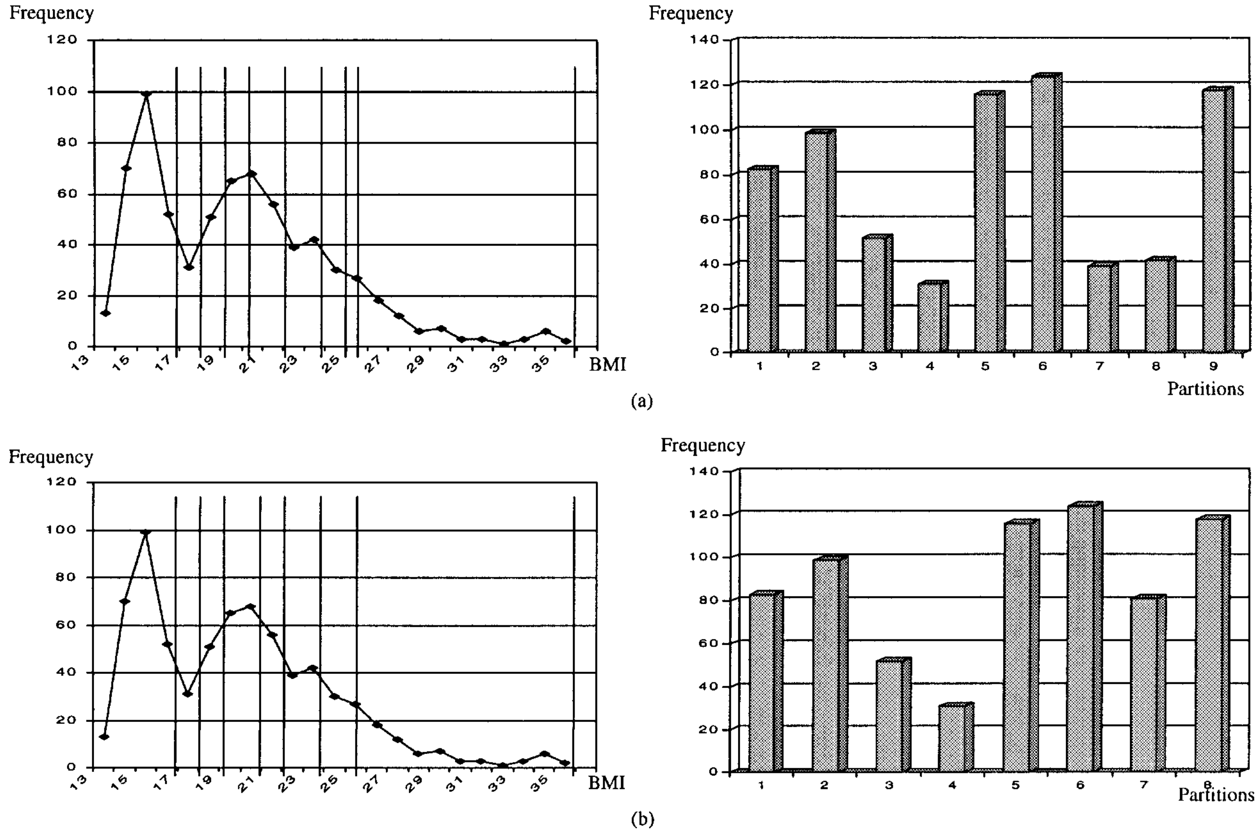

Figure 2. NGHS first year body mass index. Original distribution and entropy or dynamic MDL partitioning. (a) Entropy based partitioning, (b) dynamic MDL based partitioning in relation to second year BMI.

In Figure 2(b), the first year's body mass index has been dynamically repartitioned to better predict the value of the second year's body mass index. This is a result of a procedure to dynamically repartition continuous variables during BBN construction. In this case, two partitions have been merged to better represent the value range of the first variable in relation to the second one.

These examples demonstrate the development of more representative methods for the discretization of a continuous variable. A mapping of a continuous variable into discrete values should approximate the continuous frequency distribution with minimal loss of information.

The second section of this paper provides a review of previous methods. Section III provides an overview of the K2 algorithm for BBN construction[4] and the development of the K2 Bayesian metric and of the minimum descriptive length (MDL) metric used to guide BBN construction. Then, new methods for repartitioning a continuous variable, both before and during BBN construction, are presented. Section IV describes partitioning the range of a continuous variable according to its information content, or entropy. The algorithm to find the best balance between the information loss and a low number of partitions is presented. In Section V, this procedure is extended for the dynamic discretization of continuous variables during BBN construction. These procedures are previously unpublished and represent a new contribution to machine learning and to BBN construction methods. Both methods are tested upon data from the NGHS[3] and from the dietary intervention study in children[5] (DISC) epidemiological studies, as described later. The results of the testing are analyzed and discussed. The conclusions for the use of these discretization methods are presented in Section VI.

II. PREVIOUS METHODS FOR DISCRETIZATION OF CONTINUOUS VARIABLES FOR CLASSIFICATION

A number of methods have been used to partition values of a continuous variable into intervals, which can be used as discrete values to represent the variable. The most familiar ones create partitions of K equal lengths (equal intervals) or with $K\%$ of the total data (equal frequencies). In each case, K is chosen to provide a manageable number of discrete values, which give a fair approximation of the continuous frequency distribution. However, these methods may overpartition the distribution, split relevant groupings, or combine separate groupings of values.

Most methods used for discretizing a continuous variable use its relationship to another variable to determine the partitions. This is often found in classification procedures, such as decision trees[6–10] and in naive Bayesian classifiers.[11,12]

An entropy based method, proposed by Fayyad and Irani,[6] chooses the partitioning point(s) in a sorted set of continuous values to minimize the joint entropy ($H(X,Y)$) of the continuous variable and the classification variable. This is applied to the creation of decision tree structures for classification by recursively finding more partitioning points (top-down discretization). The method is expanded to minimize a MDL metric to choose the partitioning points.

Another MDL based method for discretization is described by Pfahringer.[7] A set of the best partitioning points is determined by recursively partitioning the sorted variable values to a depth D ($2^D - 1$ partitions) in a binary tree. Then the MDL metric is used in a best first search in this set to determine the best partitions for decision tree classification.

A method that merges adjacent partitions of sorted variable values, according to the χ^2 statistical test, is described by Liu and Setiono.[8] The variable values are sorted and initially partitioned into, at most, N intervals. The intervals are first recursively merged according to the lowest χ^2 value until a significance level of 0.05 is reached for each partition. The intervals are further merged until a preset error rate with the classification variable is reached. If there is only one resulting interval, the variable is not relevant to the classification problem and is dropped. This method combines the discretization of continuous variables with feature selection for classification.

Dougherty, Kohavi, and Sahami[9] compared several discretization techniques with decision trees and with naive Bayesian classifiers. They found that a MDL metric, similar to that used by Fayyad and Irani,[6] provided slightly better classifications in both methods.

A metric for discretization, based upon a variable's classification in relation to other variables, is described by Hong.[10] This metric is based upon a K nearest neighbor clustering technique and is used to generate decision trees. An interesting feature of this method is that it returns an optimal number of partitions according to the metric. This is done by finding the "knee" of the plotted curve of the score as a function of the number of partitions. The plot is a concave function of the metric; the knee is the point on the plot where the changes in the number of variable values (X axis) become greater than the changes in the metric value (Y axis). The concavity of a plot is exploited in the information theoretic discretization methods developed in Section IV.

Subramonian, Venkata, and Chen,[13] describe a visual framework for interactive discretization for decision tree classification. A user can choose between several algorithms and metrics for a classification problem, instead of being limited to one method and metric. The choice of metrics includes cross-entropy and the $L - 1$ norm between two distributions.

Pazzani[11] describes a technique for iterative discretization of continuous variables for naive Bayesian classifiers. Each continuous variable is initially divided into five partitions. For each variable, two partitions are then merged or a partition is divided into two partitions to find a lower classification error. This procedure is repeated for each continuous variable until the error rate can no longer be reduced.

Another method for constructing naive Bayesian classifiers using a MDL metric is presented by Friedman and Goldszmidt.[12] This method begins by finding the best initial partition of a continuous variable by dividing the range into two partitions and then iterating the partitioning until there is no further improvement in the MDL score (top-down partitioning). The MDL metric includes all of the variables used for classification and is repeated for each continuous variable. Given a BBN structure, this method discretizes each continuous variable in the Markov blanket of each classification variable. This procedure is iterated until there is no improvement in each local MDL score.

Friedman and Goldszmidt propose that this method can be adopted to learning BBN structure by starting with some initial discretization of each continuous variable, learning an initial structure, and then rediscretizing as described above. While this technique optimizes the conditional probabilities, it depends upon the initial approximate discretization of continuous variables to learn the correct network structure.

Extensions to current discretization techniques for classification methods have been proposed for BBNs, but none have currently been published. Methods for both static (done in data preprocessing) and dynamic (done during BBN construction) discretization of continuous variables are presented in Sections IV and V.

Often, a continuous variable has a normal, or Gaussian, frequency distribution. The characteristics of the normal distribution are well understood and a unified heuristic method for finding a BBN structure with both discrete variables and continuous variables with a Gaussian distribution is described by Heckerman and Geiger.[14] This method requires the mean and variance of each continuous variable's values to parameterize the distribution.

The underlying frequency distribution cannot always be assumed to have a normal form. An example is shown in Figure 1, where the distribution is approximately bimodal. In this example, the choice of the number and length of partitions to represent the distribution of body mass index is obviously not the same as when body mass index has a normal distribution.

III. METHOD

A BBN is an easily understood and increasingly popular model, which represents study variables and their conditional dependencies as nodes in a graph connected by directed arcs.[2] A node, or variable, is conditionally dependent upon its parent nodes, i.e., other variables, given the database used for the exploratory data analysis. This model partitions the joint probability distribution over all of the variables investigated into a set of conditional probabilities, as in (1).

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P\left(x_i | \pi_{x_i}\right) \tag{1}$$

where π_{x_i} are the parents of variable x_i.

Since these relationships between the variables are unknown or uncertain, a method is needed to construct the network using only the available data. One widely used method applies the K2 algorithm of Cooper and Herskovits.[4] This algorithm, and many similar ones,[15-17] requires all of the variables to have discrete values. Continuous variables can be transformed into discrete ones by partitioning the range of values into intervals of equal case frequencies, e.g., quartiles,[18] or into equal interval lengths. However, partitions of equal case frequencies are represented as a uniform distribution of variable values in the BBN node, which may not be representative of the underlying continuous frequency distribution of the variable values.

A heuristic method to find the number of equal length partitions, which provides a good mapping of continuous variables to discrete ones for constructing a BBN, is used for comparison with heuristic methods used to find varying length partitions. This method uses either the K2 metric[4] or the minimum descriptive length (MDL) metric[17] to select the number of partitions, which best approximates the underlying probability distribution of a continuously valued variable, to be used in constructing a BBN.

A. Modified K2 Algorithm

The original K2 algorithm, proposed by Cooper and Herskovits,[4] uses a Bayesian measure to pick the most probable parents of each discrete variable from the variable's predecessors in a completely ordered list. Each variable is represented by a node in the BBN.

The basic algorithm used here is essentially the same as K2, but with two modifications. First, a partially ordered list of variables is used. This ordering is based upon a temporal ordering of measurement variables, with measurements taken at the same time placed at the same level in the ordering. Only nodes from preceding levels can be considered as parents of a particular node.

The second modification is the discretization of continuous nodes (variables) during the search for the parents of each node, as described in Section V. This discretization is done to find the best partitioning of a continuous node's value range. The partitions result in discrete values, which may maximize the evaluation metric for this node as a parent. This partitioning is done in relation to the child node and its current set of parents.

The metric used for the K2 algorithm by Cooper and Herskovits,[4] is based upon the theorem for describing the joint probability of a BBN structure, B_s, given a database, D. Thus

$$P(B_s, D) = P(B_s) \prod_{i=1}^{n} \sum_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{2}$$

where i refers to the ith variable (x_i), n is the number of variables, q_i is the set of instantiations the parents, π_i, of variable x_i can have, r_i is the set of values variable x_i can have, N_{ijk} is the number of cases with the kth of the r_i values of

x_i and the jth of the q_i combination of values of the parents of x_i, and

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$$

This theorem is based upon four assumptions: (1) all variables have discrete values; (2) database cases occur independently; (3) there are no missing values; and (4) the prior probabilities of all database structures are uniform.

The K2 metric, which is derived from (2), is used to find a maximally probable set of parents of a variable, based upon the set of conditional probabilities for this structure found from the given data. The metric is

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{3}$$

This is used by the K2 algorithm in a greedy search to find the set of parents of a variable which maximizes the metric's value. Its value is approximately proportional to the probability of each conditional probability distribution, for a node and its parents, in a Dirichlet distribution. The metric value for a node is initialized for the variable by itself, independent of other variables. This is used as a starting point for the discretizing of continuous variables here.

The K2 metric, given a uniform probability distribution on a variable, gives a greater value for a smaller number of partitions [r_i in (2) and (3)] than for more partitions. As the number of partitions decreases, the $\prod N_{ijk}!$ increases at a faster rate than the $(N_{ij} + r_i - 1)!$ term. This results in a higher metric score for fewer partitions of a uniform distribution. In general, the K2 metric favors fewer partitions of a variable, given the same data.

B. MDL Encoding for a BBN

The MDL[19,20] encoding of a BBN combines a measure of the underlying probability distributions of the data sample with a measure of the network complexity. Both of these measures are proportional to the information content of the BBN, in bits. The minimum of the sum of these measures describes a BBN which closely models the underlying data but is not excessively complex. Smaller conditional dependencies between variables, as described in the previous section, will usually not be represented in a BBN induced from the data using the MDL encoding.

The MDL measure of a BBN maximizes the probability of the network structure while minimizing the network complexity. It makes a tradeoff between extreme accuracy, which may be specific to the data sample used for construction, and the BBN model usefulness. The MDL measure minimizes the sum of the encoding lengths, in bits, of both the data and the BBN model. However, finding the network which exactly minimizes these two sums is computationally intractable. Therefore, search heuristics are used which find a low, but not necessarily minimum, MDL encoding. The problem is reduced to using measures that are proportional to the MDL encoding instead of measuring the absolute value of the encoding.

To represent a particular BBN, it is necessary and sufficient to have (a) a list of parents of each node and (b) the set of conditional probabilities associated with each node.

The descriptive length needed to encode these items is $L(B_s, D)$, where B_s is a BBN structure and D is the database. It is defined by Bouckaert[17] as

$$L(B_s, D) = \log_2 P(B_s) - N \cdot H(B_s, D) - \tfrac{1}{2} k \log_2 N \tag{4}$$

where N is the number of cases, as in (2), $H(B_s, D)$ is the mutual information between a node and its parents over all nodes, and k is the cost of encoding the table of conditional probabilities between a node and its parents. $H(B_s, D)$ is defined as

$$\sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} - \frac{N_{ijk}}{N} \log_2 \frac{N_{ijk}}{N_{ij}} \tag{5}$$

where i refers to the ith variable (x_i), n is the number of variables, q_i is the set of values the parents of variable x_i can have, r_i is the set of values variable x_i can have, N_{ijk} is the number of cases with the kth value of x_i and the jth combination of values of the parents of x_i, and

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$$

as in (2). $H(B_s, D)$ increases as arcs are added to a network structure, since q_i increases with each parent node, indicated by an arc.

The k value in (4) is defined as

$$\sum_{i=1}^{n} (r_i - 1) \prod_{x_j \in \pi_i} r_j \tag{6}$$

This term increases when arcs are added, since more conditional probabilities are needed for each parent node ($x_j \in \pi_i$).

The descriptive length equation (4) shows that highly connected networks require longer encodings. Therefore, the MDL principle tends to favor networks in which the nodes have a smaller number of parents (less connected) and in which nodes taking on a large number of values are not parents of nodes that also have a large number of values. This encoding scheme generates a preference for more efficient networks. Since the encoding length of a model is included in the descriptive length, a preference for networks that require the storage of fewer probability parameters is enforced.

The *a priori* probability of a network structure, $P(B_s)$, is assumed to be equal to any other one when there is no prior information, so it is dropped from the metric. The K2 Bayesian metric makes the same assumption.

The MDL measure defined in (4) is similar to the one defined by Lam and Bacchus.[16] However, their algorithm searches a separate space (of possible BBN structures) for each network with the same number of connected nodes to find a BBN structure.

The final MDL metric used by Bouckaert,[17] for finding the local network of a single node and its parents, is reduced to

$$m(i, \pi_i) = \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log_2 \frac{N_{ijk}}{N_{ij}} - \frac{1}{2} q_i \cdot (r_i - 1) \log_2 N \tag{7}$$

This is derived from Eq. (4). The first term of (4) is dropped because $P(B_s)$ is assumed to be equal for all networks, as mentioned above. The first group of terms, in (7), is the conditional entropy between a node and its parents. This is derived from the second group of terms in (4). The N in these terms does not change, so it is dropped. The conditional entropy, for a node and its parents, is proportional to the joint entropy.[21] The second group is the number of bits needed to encode the conditional probability table for the node, which is a restatement of (6). The metric is maximized in the heuristic search, since the usual negation in conditional entropy is reversed.

The metric used by Lam and Bacchus[16] is based upon the mutual information between a node and its parents in a network.[22] Since mutual information can be defined as $I(X;Y) = H(X) - H(X|Y)$ and $H(X)$ does not change, the negated conditional entropy $-H(X|Y)$ is proportional to $I(X;Y)$.

C. Test Data for Discretization Method

A data sample of 704 cases was selected from a subset of an epidemiological health study database to demonstrate this method. The sample is from the NHLBI Growth and Health Study,[3] a study of the development of obesity in young black and white girls. It is not representative of the overall study data since cases selected had no missing values and the data were selected to specifically demonstrate the described method. The variables selected were race (categorical; range 1, 2), maturation stage (categorical; range 1–6), Quetelet index, a measure of body mass (continuous), average daily caloric intake (continuous), measurement age (continuous), and a psychological test measurement of self worth (categorical; range 1–4). The girls were 9 or 10 years of age at the first measurement. All variables used in this analysis, except race, were repeatedly measured over 5 years, with age being the age at each year's measurement.

Another data sample of 466 cases was chosen from another epidemiological database for demonstration. This sample is from the Dietary Intervention Study in Children,[5] a clinical trial designed to assess the efficacy of a lipid lowering diet in 8–10 year old children. The sample is not representative of the overall study, since cases selected had no missing values. One treatment group received several sessions to teach the children and their parents the benefits and menus of a low-fat diet. The other group received only general dietary information.

The variables chosen were gender (categorical; range $1,2$), treatment (categorical; range $1,2$), total cholesterol level (continuous), body mass index (continuous), triglycerides (continuous), activity level (categorical; range 1–6), and measurement age (continuous). Except for gender and treatment, all variables were repeatedly measured at the child's entrance into the study, after 12, 36, and 60 months; again age is the child's age at each measurement.

The modified K2 algorithm was implemented in the SAS©IML programming language on an IBM RS6355 workstation. SAS was used since the study data analysis is done in this environment. This language is interpretive, so the CPU times were excessive. Much shorter execution times would be expected using a compiled language.

IV. ENTROPY BASED DISCRETIZATION OF CONTINUOUS VARIABLES BEFORE BBN CONSTRUCTION

As mentioned above, most methods used for discretizing a continuous variable use its relationship to another variable to determine the partitions. The method proposed here is used to partition a continuous variable by itself. It is based on finding an "optimal" discretization by minimizing both the loss of information or entropy and the number of partitions. This procedure can be used in a variety of machine learning and data mining problems, which require discretization of a continuous variable.

A. Entropy Properties of a Continuous Variable

Initially, the range of a continuous variable, from a database sample, is divided into intervals which contain at least one case each. This is done after sorting on the variable values. At most, there would be m intervals ($O(m)$) for m cases. This converts the continuous variable into a discrete one, with $O(m)$ values.

Entropy, or information, is maximized when the frequency–probability distribution has the maximum number of values.[21] Since there is a discrete partition for every distinct value in the continuous distribution in the database, there is no information or entropy loss from the database sample.

The entropy of a discrete random variable X is defined as

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{8}$$

This can also be written as $H(p)$.

Cover and Thomas[21] define a function $f(x)$ to be convex over an interval (a,b) if for every x_1, x_2 in (a,b) and $0 \leq \lambda \leq 1$,

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2) \tag{9}$$

A function f is concave if $-f$ is convex. Examples of convex and concave functions are shown in Figure 3. $H(p)$ is proven to be a concave function of p.[21] These definitions are applied in the following lemmas.

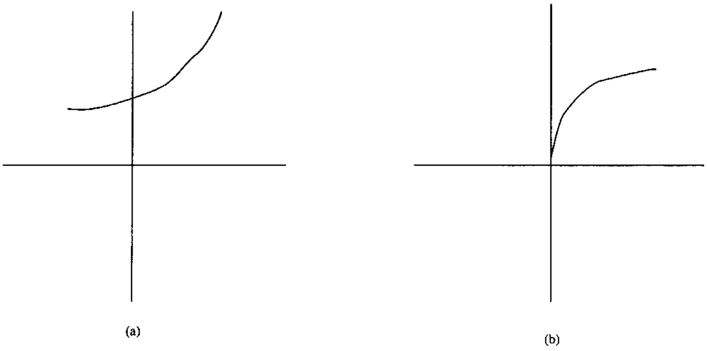

Figure 3. Examples of convex (a) and concave (b) functions.

LEMMA 4.1. *Let each distinct value of a continuous random variable X, in the case database, be represented by a separate interval. Let the number of intervals be k, $1 \leq k \leq m$ for m cases. Let the probability of X in each interval be $p(i)$, $1 \leq i \leq k$. Let the entropy of the distribution of k discrete intervals be $H(p_k)$. If two adjacent intervals, $i, i+1$, are chosen such that $H(p_k) - H(p_{k-1})$ is minimized, then the change in the probability of the combined interval, $(p(i) + p(i+1))$, is monotonically nondecreasing.*

Proof. To minimize $H(p_k) - H(p_{k-1})$, i and $i+1$ are chosen such that the sum $p(i) + p(i+1)$ is minimized. The minimum values of $p(i)$ are set at the initial partitioning of intervals for each distinct value of X. As adjacent intervals are merged, the difference between adjacent interval probabilities can only increase or remain the same. Therefore, the change in each combined $p(i)$ is monotonically nondecreasing. ■

LEMMA 4.2. *If X is a continuous random variable, in the case database, with k distinct values $(1 \leq k \leq m)$ with an interval for each distinct value, then $H(p_k)$ is a concave function over k, when each decrease in the number of intervals is chosen to minimize the change in $H(p_k)$.*

Proof. The maximum entropy of a sorted continuous random variable, in a database of m cases, is the entropy $H(p)$ when each distinct value is represented by a separate interval.

Starting at the point of maximum entropy and maximum number of k intervals $(2 \leq k \leq m)$, the two adjacent intervals are merged which result in the smallest change in $H(p_k)$ to give $H(p_{k-1})$. This smallest change in $H(p_k)$ results from merging the two adjacent intervals with the smallest difference between $p(i)$ and $p(i+1)$. If this procedure is applied repeatedly, the size of this sum is monotonically nondecreasing.

Since $H(p)$ is a concave function of p, $H(p_k)$ is a concave function of p_k in this procedure. Since each p_k represents k intervals in this procedure, $H(p_k)$ is a concave function of the number of decreasing intervals when the p_k is monotonically nondecreasing. ■

The procedure of merging adjacent intervals, described above, is continued until a stopping point is reached. The determination of this point is described next.

As seen in Figure 3(b), a concave function, such as entropy, is monotonically increasing with an increase in X. However, its rate of increase is always decreasing as it approaches the maximum value of X. The entropy over all partitionings of the NGHS variable year 2 body mass index is shown in Figure 4.

As stated above, the maximum entropy occurs with the maximum number of partitions. The best tradeoff between maximum information and a manageable number of partitions is reached when the change in X becomes greater than the change in the entropy of X. This is at the knee of the function plot.

If a chord is drawn from the origin of the graph to the point of maximum entropy and number of partitions, all of the points on the concave function plot will be above the chord. The change in X becomes greater than the change in the entropy of X at the point on the curve which is furthest from the chord. This is displayed as the vertical line from the chord to the curve in Figure 4.

The maximum height of a point on the function curve above the chord is proportional to $x_{\max}y - y_{\max}x$. Therefore, the stopping point for the merging of adjacent intervals is reached just before the decrease in this score. A similar method for determining discretization intervals relative to a classification variable, using a metric based upon a sum of squares distance, is described by

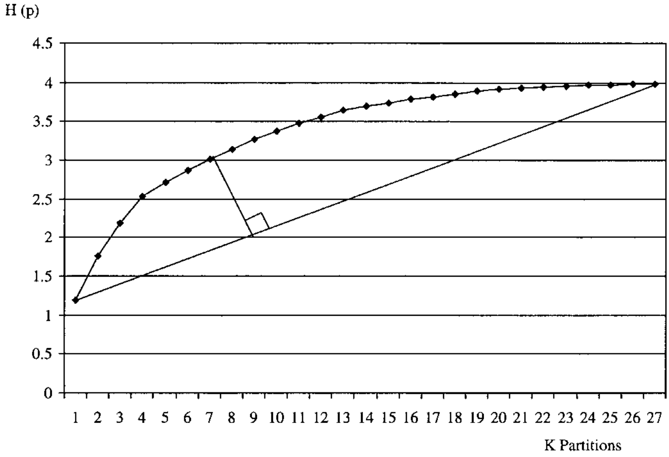

Figure 4. Entropy over all partitionings of NGHS variable year 2 body mass index. Chord shown with perpendicular line to indicate optimal number of partitions.

Hong.[10] However, this method, as well as all of the others referenced here, bases discretization upon one variable's relationship to another. Also, it makes no use of the entropy measure.

Repeatedly merging adjacent intervals, as described above, gives the least decrease in entropy and allows intervals with low frequency difference to be merged prior to those with higher frequency differences. This method preserves obvious groupings or clusters.

B. Discretization Procedure

This entropy discretization method was applied to the continuous variables in both the NGHS and DISC datasets. For each continuous variable in a dataset, the dataset was sorted on that variable and only the values of that variable were input to the program.

The two adjacent values with the smallest entropy difference were found and merged. If there was more than one pair of adjacent values with the smallest entropy difference, one pair was randomly chosen. This procedure was iterated until the stopping criteria, finding the knee of the concave function plot, was met. The initial variable values, which defined the partitions (the split points), were then stored. After every continuous variable was partitioned in this manner, the stored split points were written to an output file.

The procedure is implemented in the following algorithm:

```
procedure ENTD;
/* Input: a database, $D$, of $m$ cases and $n$ continuous variables;
   Output: a file of partition split points for every continuous variable; */
for $i = 1$ to $n$ do;      /* do for each continuous variable */
   sort $D$ by continuous variable $i$;
   read $m$ cases of variable $i$ into array $X$;
   accumulate frequency counts and save split points for each distinct $k$ values
   of $X$;
   oldent = entropy of $X$ with $k$ partitions;
   initent = oldent;
   init$k$ = $k$;
   oldchek = 0;
   oktogo = TRUE;
   while (oktogo);      /* loop to find knee of entropy curve */
      find the 2 adjacent intervals, $a, a + 1$, with the minimum frequency difference;
      merge intervals $a, a + 1$;
      newent = entropy of $X$ with intervals $a, a + 1$, merged ($k - 1$ partitions);
      newchek = (initk$*$ newent) − (initent$*$ ($k - 1$));      /* calculate max$(x)y$
      $-$max$(y)x$ */
      if (newchek > oldchek)                              /* check to continue */
         then do;
            reset split points for $X$ to reflect merger of $a, a + 1$;
            reset interval frequencies;
            oldent = newent;
            $k = k - 1$;
            oldchek = newchek;
            end;
         else oktogo = FALSE;
      end while;
   save split points for $X$;
   end $i$ loop;
output to file split points for all continuous variables;
end ENTD.
```

Sorting the dataset has $O(m \log_2 m)$ computational complexity for m cases. Assuming a distinct value for each case, iteratively selecting and merging the partitions takes $2(m + (m-1) + (m-2) + \cdots 1)$ operations in the worst case. The number of operations is $2(m(m-1))/2$ or $O(m^2)$. With n continuous variables, the worst case complexity is $O(n(m \log_2 m + m^2))$.

The partitioning of the NGHS year 1 body mass index is shown in Figure 2(a). The frequency distribution of the initial values is shown, with the split points indicated, along with the discrete value frequency distribution.

The programs to implement the entropy discretization, for each dataset, were written in SAS©IML for use by statisticians. The execution times for partitioning were 20 s for NGHS data and 10 s for DISC data.

C. Results for Entropy Discretization with K2 Bayesian Metric

The NGHS and DISC datasets, with their accompanying files of split points for discretization of continuous variables, were input to the BBN construction program, which used the K2 algorithm. The continuous variables were initially partitioned according to their split points and the BBN was constructed.

The number of partitions for each NGHS continuous variable, after entropy discretization, is shown in Table I. In 85% of the variables, the entropy discretization resulted in 10 or fewer partitions over the range of continuous variable values.

To provide bases for comparison, BBNs were constructed with equal length interval partitioning of all continuous variables. The number of partitions, for each continuous variable, was chosen to maximize the current metric score (K2 Bayesian or MDL) for the variable alone. The partitions were further divided into twice as many equal length intervals if the metric score was increased with the continuous variable as a parent of a particular other variable. The resulting BBNs, according to metric and test data set, are compared with the BBNs constructed with entropy partitioning.

The network structure, shown in Figure 6, is more sparsely connected then for the BBN with initial and dynamic equal length partitions (Fig. 5). After the partitioning of year 1 age into six intervals, it no longer was a parent of any year 1 measurements, as it was previously.

The K2 metric scores are presented in Table II, with the metric scores for the initial and dynamic equal interval partitioning for comparison. For all of the entropy partitioned continuous variables, the scores were much lower than the

Table I. NGHS, number of partitions for entropy based discretization of continuous variables.

Variable	Number of Partitions Using Entropy
Race	(2) categorical
Age (1)	6
Blood pressure (1)	5
Body mass index (1)	9
Maturation stage (1)	(6) categorical
Daily calories (1)	8
Self worth (1)	(4) categorical
Age (2)	8
Blood pressure (2)	9
Body mass index (2)	6
Maturation stage (2)	(6) categorical
Daily calories (2)	10
Age (3)	7
Blood pressure (3)	9
Body mass index (3)	10
Maturation stage (3)	(6) categorical
Daily calories (3)	12
Self worth (3)	(4) categorical
Age (4)	7
Blood pressure (4)	9
Body mass index (4)	8
Maturation stage (4)	(6) categorical
Daily calories (4)	11
Age (5)	9
Blood pressure (5)	6
Body mass index (5)	11
Maturation stage (5)	(6) categorical
Daily calories (5)	8
Self worth (5)	(4) categorical

previous ones. This was caused by the increase in the number of partitions, which brings a corresponding decrease in the K2 metric score.

The number of partitions for each DISC continuous variable, after entropy discretization, is shown in Table III. In this dataset, all of the continuous variables were partitioned into 10 or fewer intervals.

The network structure, shown in Figure 8, is again more sparsely connected than the one for initial and dynamic equal interval partitioning (Fig. 7). One of the main study interactions, the effect of treatment on the second total cholesterol measurement, is not seen because of the partitioning of the value range into five intervals instead of two.

The K2 metric scores are presented in Table IV, with the scores for the initial and dynamic equal interval partitioning for comparison. The entropy discretization scores are much lower, except for the third total cholesterol measurement (eight partitions) and the fourth high density lipoprotein measure-

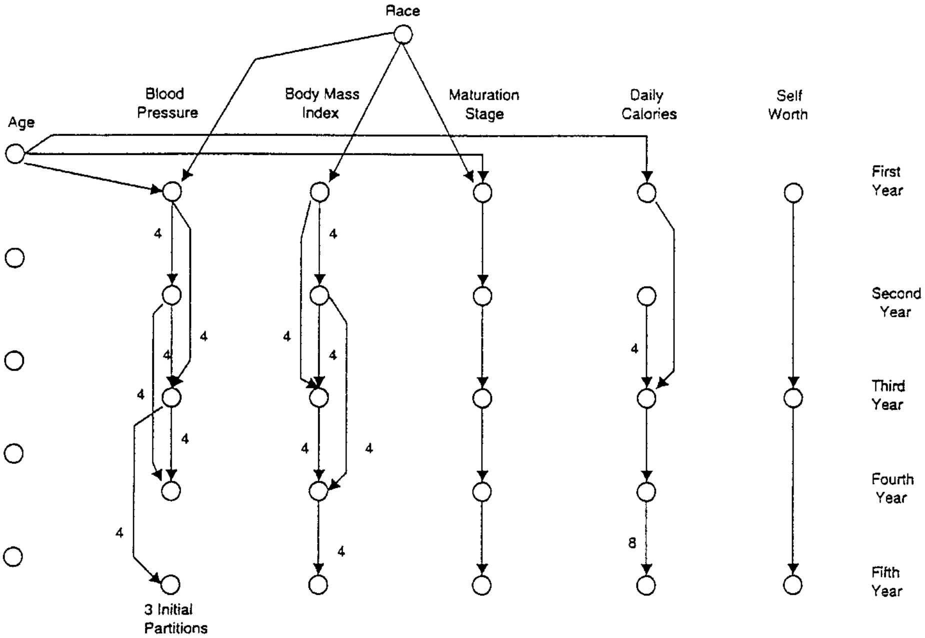

Figure 5. NGHS network, K2 Bayesian metric, with all continuous variables initially partitioned into two equal length intervals (year 5 blood pressure three intervals). Dynamically repartitioned into equal length intervals, as indicated by numbers adjacent to directed arcs.

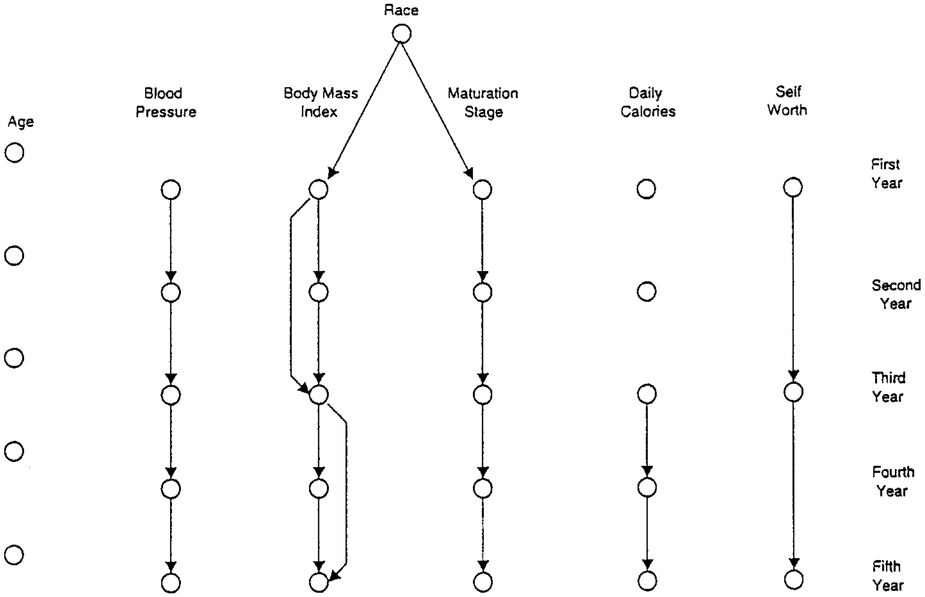

Figure 6. NGHS network, K2 Bayesian metric, with all continuous variables initially partitioned using entropy.

Table II. NGHS K2 metric scores for nodes with parents for equal interval and initial entropy based discretization.[a]

Variable	All Continuous Initially and Dynamically Partitioned with Equal Length Intervals	All Continuous Partitioned Using Entropy
Race	—	—
Age (1)	—	—
Blood pressure (1)	−191	—
Body mass index (1)	−110	−**644**
Maturation stage (1)	−351	−**366**
Daily calories (1)	−77	—
Self worth (1)	—	—
Age (2)	—	—
Blood pressure (2)	−176	−**498**
Body mass index (2)	−58	−**272**
Maturation stage (2)	−392	−392
Daily calories (2)	—	—
Age (3)	—	—
Blood pressure (3)	−151	−**418**
Body mass index (3)	−52	−**462**
Maturation stage (3)	−388	−388
Daily calories (3)	−83	—
Self worth (3)	−289	−289
Age (4)	—	—
Blood pressure (4)	−171	−**418**
Body mass index (4)	−38	−**238**
Maturation stage (4)	−339	−339
Daily calories (4)	−25	−**456**
Age (5)	—	—
Blood pressure (5)	−183	−**300**
Body mass index (5)	−37	−**436**
Maturation stage (5)	−268	−268
Daily calories (5)	−33	−**252**
Self worth (5)	−310	−310

[a] Changes from previous implementation are in bold face.

ment (10 partitions). The more precise partitioning of these variables and their parents overcame the scoring handicap of more partitions.

The CPU execution times for network construction was 1320 min for NGHS data and 176 min for DISC data. This great disparity is due to the higher number of partitions of continuous variables for NGHS and to the greater number of cases, as well as to slightly greater network complexity.

D. Results for Entropy Discretization with MDL Metric

Again, BBNs constructed with equal length interval partitioning of continuous variables, as described in Section IV.C, were used for comparison.

The NGHS network constructed, using the MDL metric with entropy discretization, is shown in Figure 10. It is more sparsely connected than the

Table III. DISC, number of partitions for entropy based discretization of continuous variables.

Variable	Number of Partitions Using Entropy
Gender	(2) categorical
Treatment	(2) categorical
Age (1)	8
Total cholesterol (1)	2
High density lipoprotein (1)	4
Triglycerides (1)	8
Body mass index (1)	6
Activity level (1)	(5) categorical
Age (2)	7
Total cholesterol (2)	5
High density lipoprotein (2)	4
Triglycerides (2)	8
Body mass index (2)	7
Activity level (2)	(5) categorical
Age (3)	8
Total cholesterol (3)	2
High density lipoprotein (3)	5
Triglycerides (3)	8
Body mass index (3)	8
Activity level (3)	(5) categorical
Age (4)	10
Total cholesterol (4)	7
High density lipoprotein (4)	2
Triglycerides (4)	9
Body mass index (4)	7
Activity level (4)	(5) categorical

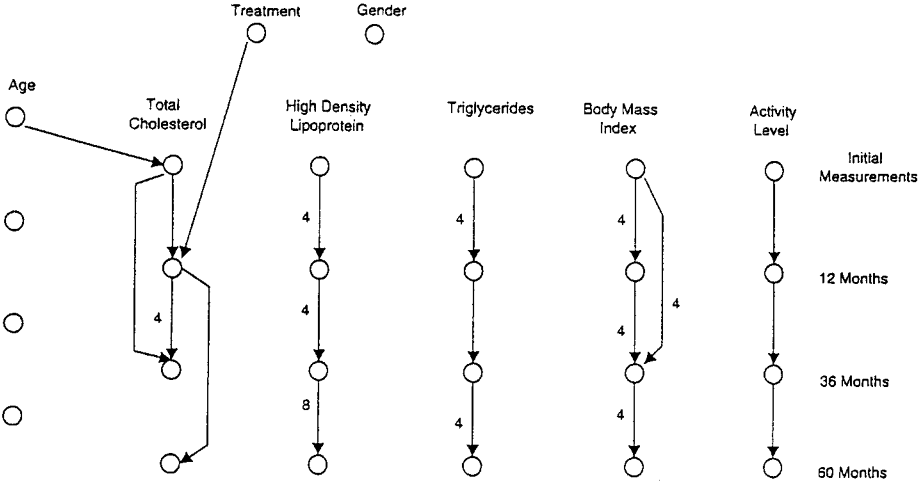

Figure 7. DISC network, K2 Bayesian metric, with all continuous variables initially partitioned into two equal length intervals. Dynamically repartitioned into equal length intervals, as indicated by numbers adjacent to directed arcs.

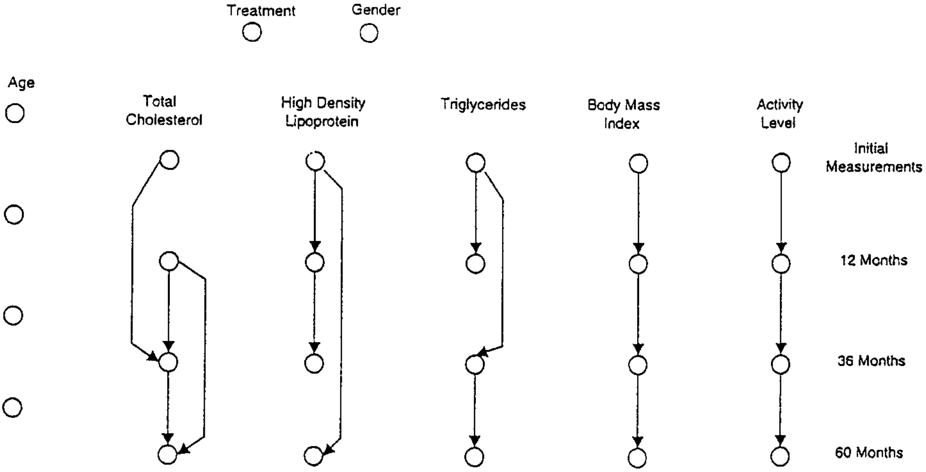

Figure 8. DISC network, K2 Bayesian metric, with all continuous variables initially partitioned using entropy method.

BBN with initial and dynamic equal interval partitioning (Fig. 9). Major differences are the lack of year to year dependencies in blood pressure measurements and maturation stage being the only race dependency.

The MDL metric scores are presented in the middle column of Table V, with the scores of initial and dynamic equal interval partitioning for comparison. The scores are much lower, due to the greater number of partitions.

The network constructed, with entropy discretization of DISC data, is shown in Figure 12. It has almost half of the dependencies shown for continuous variables, compared to the BBN for initial and dynamic equal length interval partitions (Fig. 11). Most of the missing connections are for triglyceride measurements, which are completely independent in this implementation.

The MDL metric scores are presented in the middle column of Table VI, with the initial and dynamic equal interval partitioning scores for comparison. All of the scores are much lower, except for the third total cholesterol measurement and the fourth high density lipoprotein measurement. This decrease in the MDL scores is consistent with the decrease in the K2 metric scores for these variables.

The CPU execution time for network construction was 468 min for NGHS data and 97 min for DISC data. These execution times for each dataset is roughly proportional to the times using the K2 metric.

E. Discussion of Results

The entropy discretization method provides an optimal balance between information loss and a manageable number of values for continuous variables. Also, it can be done efficiently without comparison to another variable. In the

Table IV. DISC K2 metric scores for nodes with parents for equal interval and initial entropy based discretization.[a]

Variable	All Continuous Initially and Dynamically Partitioned with Equal Length Intervals	All Continuous Partitioned Using Entropy
Gender	—	—
Treatment	—	—
Age (1)	—	—
Total cholesterol (1)	−134	—
High density lipoprotein (1)	—	—
Triglycerides (1)	—	—
Body mass index (1)	—	—
Activity level (1)	—	—
Age (2)	—	—
Total cholesterol (2)	−122	—
High density lipoprotein (2)	−78	**−199**
Triglycerides (2)	−59	**−391**
Body mass index (2)	−23	**−191**
Activity level (2)	−226	−226
Age (3)	—	—
Total cholesterol (3)	−122	**−39**
High density lipoprotein (3)	−60	**−250**
Triglycerides (3)	−17	**−389**
Body mass index (3)	−122	**−287**
Activity level (3)	−242	−242
Age (4)	—	—
Total cholesterol (4)	−20	**−357**
High density lipoprotein (4)	−76	**−61**
Triglycerides (4)	−26	**−388**
Body mass index (4)	−50	**−167**
Activity level (4)	−243	−243

[a] Changes from previous implementation are in bold face.

two datasets used for testing, there were generally 10 or fewer partitions used to represent the value range of a continuous variable.

The increased number of partitions led to more sparsely connected networks than those with equal length partitions for both the NGHS and DISC datasets. The number of dependencies shown, involving continuous variables, was nearly halved for both datasets. Also, the higher number of partitions usually resulted in lower metric scores, as well as increased execution times for network construction.

This method of discretization of continuous variables brings out the strongest conditional dependencies in the data. These dependencies are stronger since they exist through a more precise partitioning of data value ranges, than for equal length interval partitioning.

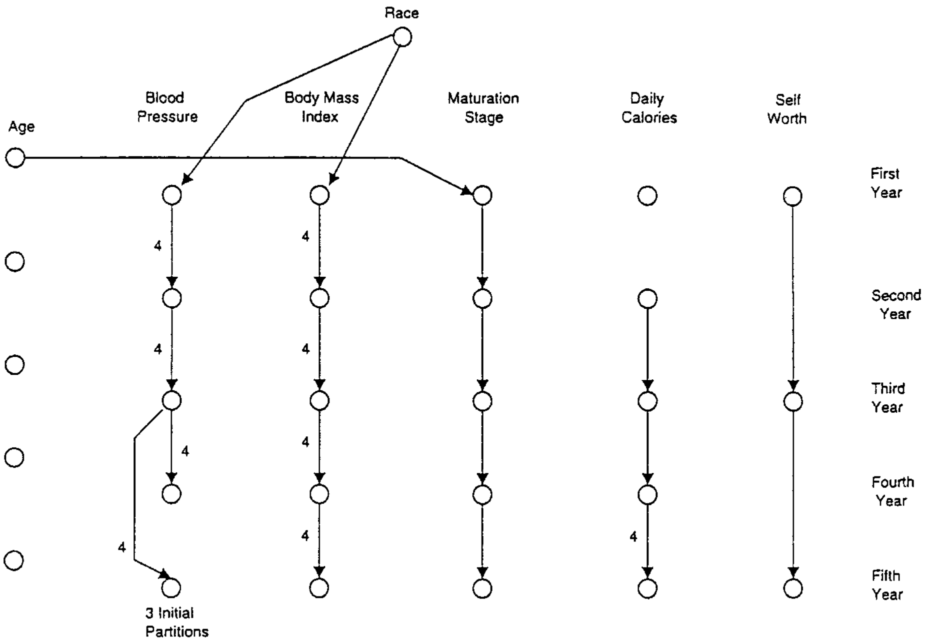

Figure 9. NGHS network, MDL metric, with all continuous variables initially partitioned into two equal length intervals (year 5 blood pressure three intervals). Dynamically repartitioned into equal length intervals, as indicated by numbers adjacent to directed arcs.

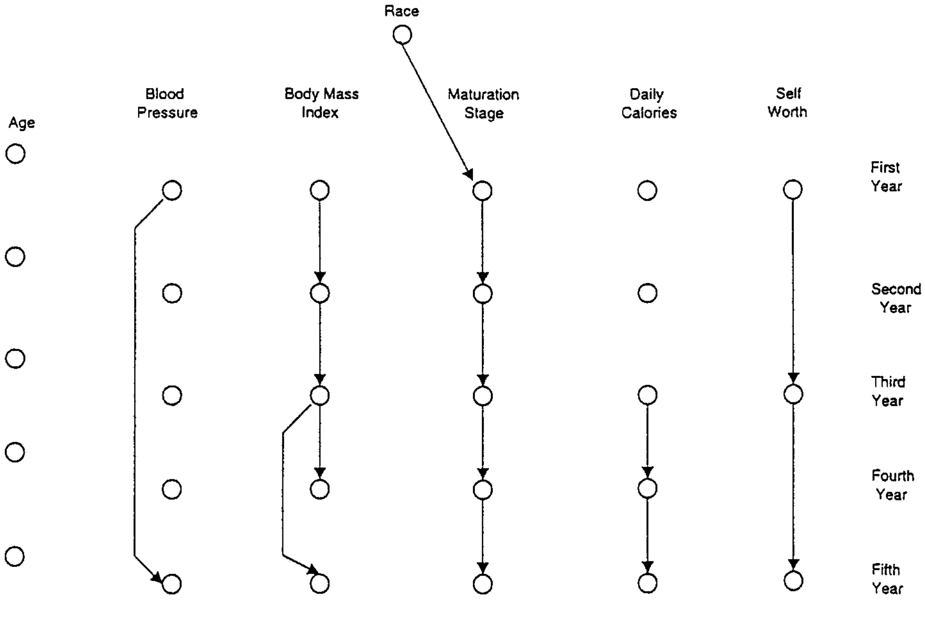

Figure 10. NGHS network, MDL metric, with all continuous variables initially partitioned using the entropy method.

Table V. NGHS MDL metric scores for nodes with parents for equal length interval and entropy based discretization.[a]

Variable	All Continuous Initially and Dynamically Partitioned with Equal Length Intervals	All Continuous Partitioned Using Entropy	All Continuous Partitioned Using Entropy; Dynamically Repartitioned Using MDL
Race	—	—	—
Age (1)	—	—	—
Blood pressure (1)	−637	—	—
Body mass index (1)	−367	—	—
Maturation stage (1)	−1221	−1222	−1222
Daily calories (1)	—	—	—
Self worth (1)	—	—	—
Age (2)	—	—	—
Blood pressure (2)	−591	—	—
Body mass index (2)	−191	**−953**	**−935**
Maturation stage (2)	−1359	−1359	−1359
Daily calories (2)	—	—	—
Age (3)	—	—	—
Blood pressure (3)	−518	—	—
Body mass index (3)	−175	**−1642**	**−1620**
Maturation stage (3)	−1340	−1340	−1340
Daily calories (3)	−289	—	—
Self worth (3)	−980	−980	−980
Age (4)	—	—	—
Blood pressure (4)	−585	—	**−1277**
Body mass index (4)	−134	**−879**	**−861**
Maturation stage (4)	−1165	−1165	−1165
Daily calories (4)	−82	**−1832**	**−1752**
Age (5)	—	—	—
Blood pressure (5)	−618	**−1068**	**−1031**
Body mass index (5)	−121	**−1698**	**−1667**
Maturation stage (5)	−928	−928	−928
Daily calories (5)	−133	**−1000**	**−974**
Self worth (5)	−1053	−1053	−1053

[a] Changes from previous implementation are in bold face.

V. DYNAMIC REPARTITIONING FOR BBNS BASED ON THE MDL METRIC

A. Introduction

To further improve the accuracy of the BBNs constructed using entropy based discretization of continuous variables, a method for dynamic repartitioning using the MDL metric was developed. Since higher metric scores result from fewer variable values (partitions) and the initial discretization was not done in relation to any other variables, this repartitioning merged existing partitions of continuous variables to find more definitive conditional dependencies. This

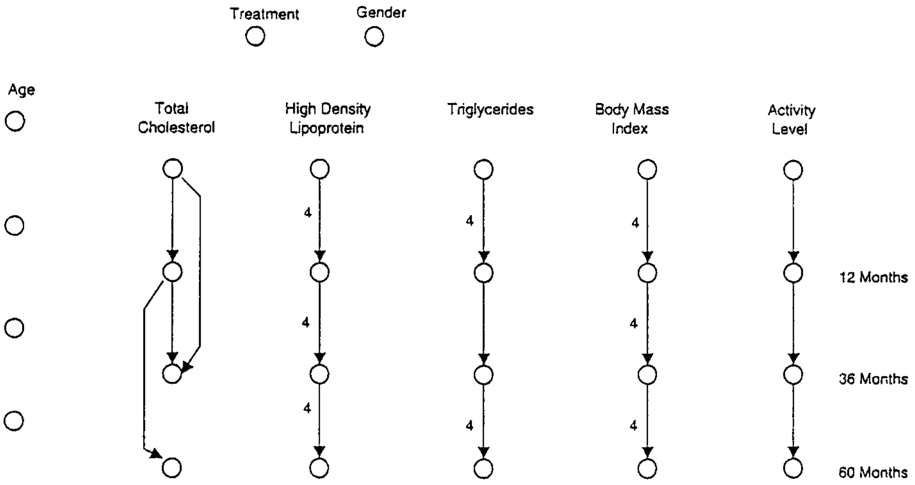

Figure 11. DISC network, MDL metric, with all continuous variables initially partitioned into two equal length intervals. Dynamically repartitioned into equal length intervals, as indicated by numbers adjacent to directed arcs.

procedure for dynamically repartitioning continuous variables during BBN construction, using the MDL metric, is completely new. It could have wide use in data mining applications using BBN models.

Bouckaert[23] has shown that the MDL metric used here, Eq. (7), is a concave function in the number of partitions used in the conditional dependen-

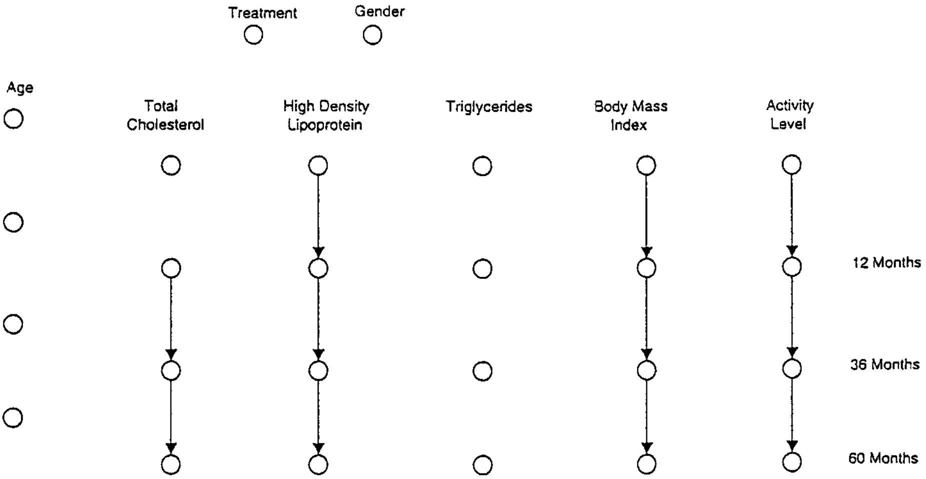

Figure 12. DISC network, MDL metric, with all continuous variables initially partitioned using entropy method.

Table VI. DISC MDL metric scores for nodes with parents for equal interval and entropy based discretization.[a]

Variable	All Continuous Initially and Dynamically Partitioned with Equal Length Intervals	All Continuous Partitioned Using Entropy	All Continuous Partitioned Using Entropy; Dynamically Repartitioned Using MDL
Gender	—	—	—
Treatment	—	—	—
Age (1)	—	—	—
Total cholesterol (1)	—	—	—
High density lipoprotein (1)	—	—	—
Triglycerides (1)	—	—	—
Body mass index (1)	—	—	—
Activity level (1)	—	—	—
Age (2)	—	—	—
Total cholesterol (2)	−407	—	—
High density lipoprotein (2)	−264	−**680**	−**673**
Triglycerides (2)	−200	—	—
Body mass index (2)	−76	−**685**	−**673**
Activity level (2)	−787	−787	−787
Age (3)	—	—	—
Total cholesterol (3)	−409	−**140**	−**136**
High density lipoprotein (3)	−203	−**853**	−**864**
Triglycerides (3)	−56	—	—
Body mass index (3)	−149	−**1055**	−**984**
Activity level (3)	−839	−839	−839
Age (4)	—	—	—
Total cholesterol (4)	−66	−**1220**	−1220
High density lipoprotein (4)	−265	−**37**	−**47**
Triglycerides (4)	−92	—	−**1332**
Body mass index (4)	−168	−**623**	−**609**
Activity level (4)	−842	−842	−842

[a] Changes from previous implementation are in bold face.

cies between parent and child nodes. This is based upon the concavity of the conditional entropy used in the metric. Finding the knee of this function was used as a stopping point for merging the partitions of continuous variables. This was done by a method similar to the one used to stop the entropy based partitioning of a single continuous variable. As described in Section IV.A, the knee of the plot of the concave function is used as a stopping point for merging partitions.

If there were no stopping points for mergers, all continuous variables would reach a binary partitioning. This is a result of the MDL metric giving a higher score to variables with fewer values. Two values were the minimum for a measurement to remain a variable and not to become a constant.

B. Method

As the set of parents of a variable was being found during BBN construction, each continuous variable was dynamically repartitioned to find the highest MDL metric for this variable as a parent. If a continuous variable was being considered as a parent, the two adjacent partitions with the lowest frequency (probability) difference were merged. After this merging, a new MDL metric value was found which used this variable, the child variable, and any other previously established parent variables. The merging, of the adjacent partitions with the lowest frequency difference, was based upon the heuristic that adjacent variable values, with very similar frequencies, belong to the same value cluster for predicting the value of another variable.

This merging was iterated until the knee of the MDL function was reached. The resulting metric score was considered to be the best possible for this continuous variable and was used for selecting it as a parent variable. If the variable was not chosen as a parent, it retained its initial partitioning.

After the child variable and its parents were written to the output file, a continuous parent variable reverted to its initial partitioning. In this way, each dynamic repartitioning was independent of any others for the same variable and the partitioning of the parent as a child node remained unchanged. In the network structure, a new node can be created between the original node and the child to represent this repartitioning. This new node is a direct parent of this child node and only of this one child node.

C. Procedure

As described in Section IV.B, the NGHS or DISC dataset, with its accompanying file of continuous variable split points, was read into the program with the basic K2 BBN construction algorithm. As each continuous variable was considered as a parent, it was repartitioned, as described above, to find the best partitioning relative to a child variable. If the continuous variable was selected as a parent, it was listed in the output file as one, along with its current split points and the number of partitions. The partitioning of a continuous variable was always reset after it was considered as a parent. This procedure was used in a modified version of the K2 algorithm, with the repartitioning done according by merging the adjacent partitions with the lowest frequency difference, instead of in the manner of the repartition function in the algorithm.

Aside from the complexity of the initial entropy discretization of the data, this method of dynamic discretization significantly adds to the computational complexity of the K2 based algorithm. With m cases, n variables, r values for a variable, and a maximum of c parents, this method uses $c[(rm) + ((r-1)m) + \cdots (2m)]$ operations in the worst case. This comes to $O((r(r-1)/2)cm)$ or $O(r^2cm)$ operations for each variable. When combined with the initial complexity for the K2 algorithm of $O(mn^4r)$, this gives a worst case complexity of $O(m^2n^4r^3c)$.

D. Results for Dynamic Repartitioning with the MDL Metric

The network constructed using dynamic repartitioning with the MDL metric, for NGHS data, is shown in Figure 13. The only changes in network structure, from using only entropy discretization (Fig. 10), are between blood pressure measurements. There are two additional arcs and one arc that has been deleted.

The MDL metric scores are presented in the third column of Table V. These scores are higher than the scores using only initial entropy discretization, since they represent fewer partitions.

The network constructed using dynamic repartitioning with the MDL metric for DISC data is shown in Figure 14. There are two arcs added to the network constructed with initial entropy discretization (Fig. 12), while one arc was deleted.

The metric scores are presented in the third column of Table VI. They are higher than the scores using only initial entropy discretization, except for the last two measurements of high-density lipoprotein.

The execution times were 1101 min for NGHS data and 210 min for DISC data.

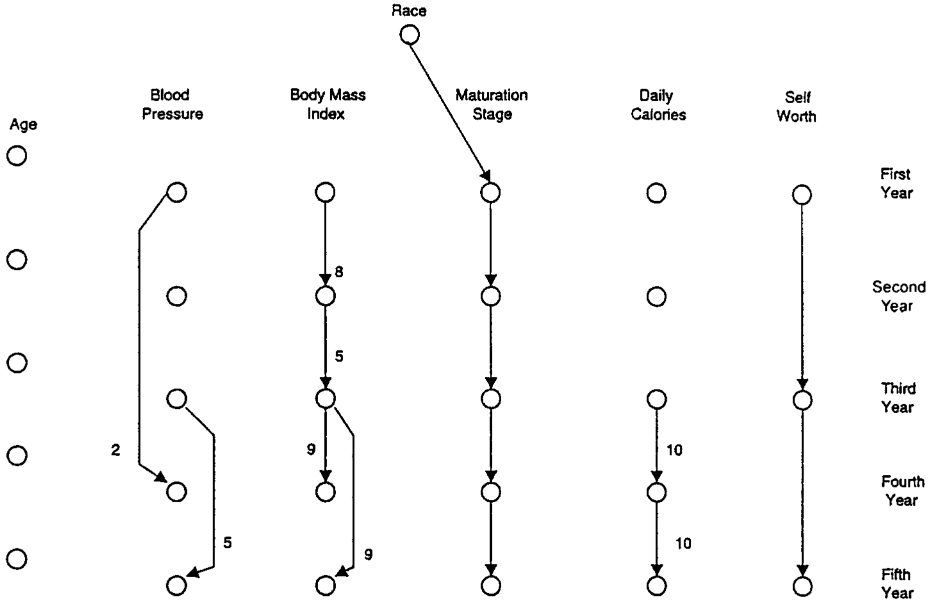

Figure 13. NGHS network, MDL metric, with all continuous variables initially partitioned using the entropy method, then dynamically repartitioned using MDL metric. Number of partitions indicated by numbers adjacent to directed arcs.

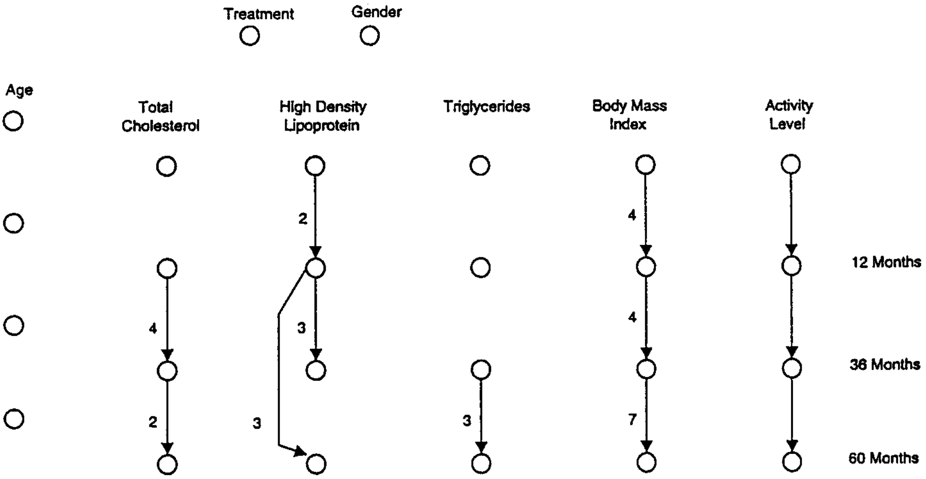

Figure 14. DISC network, MDL metric, with all continuous variables initially partitioned using the entropy method, then dynamically repartitioned using MDL metric. Number of partitions indicated by numbers adjacent to directed arcs.

E. Discussion

The dynamic discretization of continuous variables, using the MDL metric, resulted in minor changes in BBN structure. However, metric scores were generally higher and additional dependencies were found.

The use of dynamic repartitioning, in conjunction with entropy discretization, brought out dependencies which were hidden in coarser representations of continuous variables. The finding of unexpected results with a more flexible discretization method justifies the increase in computational complexity.

VI. CONCLUSION

Methods for discretizing continuous variables for BBN construction were developed here. An algorithm was developed for discretizing continuous variables according to the decreasing entropy, or information, contained in fewer partitions. The partitioning is optimal in that it represents the best compromise between information loss and a manageable number of partitions. This method can be applied efficiently since it requires no other variable, or classifier, for comparison during the discretization.

The resulting partitioning from entropy discretization was further modified, during BBN construction, using the MDL scoring metric since it is a concave function of the number of partitions. Adjacent partitions of a continuous variable were merged to achieve an optimal metric score for local network structure by adding that variable. The score was optimal in that it provided the best balance of information loss and the number of partitions.

These methods were demonstrated using selected data from two epidemiological studies, NGHS and DISC. Two metrics to determine conditional dependencies, K2 Bayesian and MDL, were applied for all applicable discretization methods. The results were compared across methods and across metrics.

Dynamic repartitioning of continuous variables led to more sparsely connected networks with equal length interval partitioning. However, dynamic repartitioning of continuous variables, initially partitioned using entropy discretization, led to more highly connected networks. All methods of dynamic repartitioning led to better representations of underlying dependencies among variables in both NGHS and DISC data.

Pearson correlation coefficients were found for the continuous NGHS variables with dependencies shown in Figure 13 (Table VII). These are presented to compare the efficacy of the discretization methods presented here. The high correlations between the initial continuous values of the variables, which have no information loss from discretization, verify the dependencies. For seven of eight pairs of variables, the correlations are much higher with entropy discretization or with dynamic MDL repartitioning than with binary equal interval discretization.

Table VII. NGHS Pearson correlation coefficients for pairs of continuous variables with relationships shown in Figure 4.20 (BBN with dynamic MDL repartitioning).[a]

Correlation Variables	Initial Continuous Values	All Continuous Binary with Equal Length Intervals	All Continuous Partitioned Using Entropy	All Continuous Partitioned Using Entropy Dynamically Repartitioned Using MDL
Body mass index (1) × Body mass index (2)	0.956	0.785	0.870	0.840
Body mass index (2) × Body mass index (3)	0.957	0.762	0.888	0.891
Body mass index (3) × Body mass index (4)	0.960	0.821	0.773	0.792
Body mass index (3) × Body mass index (5)	0.920	0.703	0.877	0.877
Blood pressure (1) × Blood pressure (4)	0.470	0.319	0.391	0.334
Blood pressure (3) × Blood pressure (5)	0.541	0.356	0.447	0.362
Daily calories (3) × Daily calories (4)	0.845	0.540	0.815	0.802
Daily calories (4) × Daily calories (5)	0.930	0.653	0.877	0.872

[a] Coefficients shown for initial continuous values, and partitioning with binary equal interval, entropy, and dynamic MDL repartitioning.

While entropy discretization generally shows slightly higher correlations than dynamic MDL repartitioning, the differences are not significant. Both correlations are generally within one-tenth of the correlation of the continuous values, with significantly fewer values. This shows that entropy discretization provides a very good approximation of the underlying continuous distribution of values in the sample data.

For both sets of data and for both metrics, the most complex BBNs were found using the simplest methods for partitioning continuous variables. This reflected the bias, in both metrics, in favor of fewer variable values. Dynamic repartitioning of continuous variables, with the MDL metric, led to more highly connected BBNs than with only entropy partitioning. This was the opposite of the results for dynamic repartitioning with equal length interval partitioning. The new methods presented here, for converting continuous variables into discrete ones, led to better representations of the dependencies among variables for data from both medical studies.

The methods presented here are well suited for exploratory data analysis. Starting with the simplest discretization, relationships between variables can be approximated. More computationally complex methods can be applied when justified by more rigorous analysis requirements. The method for dynamic repartitioning, using the MDL metric, can also be applied to discrete variables to better represent their frequency distribution.

The use of entropy and MDL based partitioning of continuous variables resulted in the clarification and simplification of the BBNs by providing an optimal number of values to represent continuous variables.

This research was funded in part by National Heart, Lung, and Blood Institute contract NO1-HC-55023-26 and by National Heart, Lung, and Blood Institute Grant U01-HL-37948. The authors thank Timothy Finin and Yun Peng for their comments on an earlier version of this work.

References

1. Pearl J, Verma T. The logic of representing dependencies by directed graphs. Proc American Association for Artificial Intelligence. 1987; p 374–379.
2. Pearl J. Probabilistic reasoning in intelligent systems: Networks of plausible inference, San Mateo, CA: Morgan Kaufmann; 1988.
3. National Heart, Lung, and Blood Institute Growth and Health Study Research Group, Obesity and cardiovascular disease risk factors in black and white girls: The NHLBI growth and health study. Amer J Public Health 1992;82:1613–1620.
4. Cooper G, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. Machine Learning 1992;9:309–347.
5. Dietary Intervention Study in Children Collborative Research Group, Dietary intervention study in children (DISC) with elevated low-density-lipoprotein cholesterol: Design and baseline characteristics. Ann Epidemiology 1993;3(4):393–402.
6. Fayyad U, Irani K. Multi-interval discretization of continuous-valued attributes for classification learning. Proc 13th International Joint Conference on Artificial Intelligence. Vol. 2, San Mateo, CA: Morgan Kaufmann; 1993. p 1022–1027.
7. Pfahringer B. Compression-based discretization of continuous variables. Proc 12th International Conference on Machine Learning. Prieditis A, Russell S, editors. San Francisco, CA: Morgan Kaufmann; 1995. p 456–463.

8. Liu H, Setiono R. Chi2: Feature selection and discretization of numeric attributes. IEEE International Conference on Tools with Artificial Intelligence. (http://www.iscs.nus.sg/ ~ liuh/tai95.ps); 1995.
9. Dougherty J, Kohavi R, Sahami M. Supervised and unsupervised discretization of continuous features. Proc 12th International Conference on Machine Learning. Prieditis A, Russel S, editors. San Francisco, CA: Morgan Kaufmann; 1995. p 194–202.
10. Hong S. Use of contextual information for feature ranking and discretization. IEEE Trans Knowledge Data Eng 1997;9(5):718–730.
11. Pazzani M. An iterative improvement approach for the discretization of numeric attributes in Bayesian classifiers. Proc 1st International Conference on Knowledge Discovery and Data Mining. Fayyad U, Uthurusamy R, editors. Menlo Park, CA: AAAI Press; 1995. p 228–233.
12. Friedman N, Goldszmidt M. Discretizing continuous attributes while learning Bayesian networks. Proc 13th International Conference on Machine Learning. Saitta L, editor. San Francisco, CA: Morgan Kaufmann; 1996. p 157–165.
13. Subramonian R, Venkata R, Chen J. A visual interactive framework for attribute discretization. Proc 3rd International Conference on Knowledge Discovery and Data Mining. Heckerman D, Mannila H, Pregibon D, editors. Menlo Park, CA: AAAI Press; 1997. p 82–88.
14. Heckerman D, Geiger D. Learning Bayesian networks: A unification for discrete and Gaussian domains. Proc 11th Conference in Uncertainty in Artificial Intelligence. Besnard P, Hanks S, editors. San Francisco, CA: Morgan Kaufmann, 1995. p 274–284.
15. Spirtes P, Meek C. Learning Bayesian networks with discrete variables from data. Proc 1st International Conference on Knowledge Discovery and Data Mining. Fayyad U, Uthurusamy R, editors. Menlo Park, CA: AAAI Press; 1995. p 294–299.
16. Lam W, Bacchus F. Learning Bayesian belief networks: An approach based on the MDL principle. Comput Intell 1994;10(3):269–293.
17. Bouckaert R. Properties of Bayesian belief network learning algorithms. Proc 10th Conference in Uncertainty in Artificial Intelligence. deMantaras R, Poole D, editors. San Francisco, CA: Morgan Kaufmann; 1994. p 102–109.
18. Clarke E, Barton B. A SAS macro for exploratory analysis using a Bayesian belief network. Contr Clinical Trials 1996;17:2S, 110S.
19. Rissanen J. Stochastic complexity and modeling. Ann Statist 1986;14(3):1080–1100.
20. Rissanen J. Stochastic complexity. J Roy Statist Soc B 1987;49(3):223–239; 252–265.
21. Cover T, Thomas J. Elements of information theory. New York: John Wiley & Sons; 1991.
22. Chow C, Liu C. Approximating discrete probability distributions with dependence trees. IEEE Trans Inform Theory 1968;14(3):462–467.
23. Bouckaert R. Probabilistic network construction using the minimum description length principle. Proc European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty. Clarke M, Kruse R, Moral S, editors. New York: Springer-Verlag; 1993. p 41–48.