Journal of Computer Science 10 (5): 869-878, 2014
ISSN: 1549-3636
© 2014 Science Publications
doi:10.3844/jcssp.2014.869.878 Published Online 10 (5) 2014 (http://www.thescipub.com/jcs.toc)

HEURISTIC DISCRETIZATION METHOD FOR BAYESIAN NETWORKS

[1]Mariana D.C. Lima, [1]Silvia M. Nassar, [1]Pedro Ivo R.B.G. Rodrigues, [1]Paulo J. Freitas Filho and [2]Carlos M.C. Jacinto

[1]Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, SC, Brazil
[2]CENPES-Petrobras/PDGP/PCP, Rio de Janeiro, RJ, Brazil

Received 2013-12-06; Revised 2013-12-06; Accepted 2014-01-18

ABSTRACT

Bayesian Network (BN) is a classification technique widely used in Artificial Intelligence. Its structure is a Direct Acyclic Graph (DAG) used to model the association of categorical variables. However, in cases where the variables are numerical, a previous discretization is necessary. Discretization methods are usually based on a statistical approach using the data distribution, such as division by quartiles. In this article we present a discretization using a heuristic that identifies events called peak and valley. Genetic Algorithm was used to identify these events having the minimization of the error between the estimated average for BN and the actual value of the numeric variable output as the objective function. The BN has been modeled from a database of Bit's Rate of Penetration of the Brazilian pre-salt layer with 5 numerical variables and one categorical variable, using the proposed discretization and the division of the data by the quartiles. The results show that the proposed heuristic discretization has higher accuracy than the quartiles discretization.

Keywords: Bayesian Network, Discretization, Global Optimization, Genetic Algorithm, Heuristic

1. INTRODUCTION

A Bayesian Network (BN) allows modeling the knowledge of a domain through a set of usually categorical qualitative) variables and representing relationships and effects among them due to causality and conditional independence. The BN is a Directed Acyclic Graph (DAG) where the nodes are the variables and the arcs represent relation strength expressed in a table of conditional probabilities. Thus, knowledge in a standardized BN is expressed as the ratio structure and the estimation of probabilities. Knowledge can be built from domain experts, a data table, or from a hybrid form between both.

However, there is no guarantee that all variables of an application domain will be categorical, since there will be situations where numerical variables participate directly in the domain context. For these situations, a previous discretization of the variables is recommended, according to some metric or specific criteria. Discretization approaches are usually made by probability distribution or using statistic parameters like the frequency in each class.

The discretization can also be made by the experts on the field in a manual way. However, it can be a complex task: There are cases where the data does not follow any visible pattern and when it does, this pattern may change in different occasions. So, it is necessary to discretize the data based on the data itself, because there is no previous knowledge of its behavior.

Although there are several algorithms for discretization (Mohammed and Shamsuddin, 2011; Alfred, 2009; Ding *et al*., 2010), the majority of them have the ultimate goal of data classification and not the construction and knowledge discovery in a BN. To perform discretization for this domain, it is necessary to consider the conditional distributions of each variable of the process and how they influence the network as a whole.

Corresponding Author: Mariana D.C. Lima, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, SC, Brazil

An important aspect regarding the BNs are on their property inference: The probability distribution of one variable directly influences another. Thus, it is necessary to have global optimization to reduce BN error and increase its accuracy.

In this study, we present a heuristic discretization for Bayesian Networks that seeks to find data patterns and divide the data set according to them. This patterns are identified by two events: Peak and valley which are optimized by a search using Genetic Algorithm. These two events change according to the data set making the proposed discretization more flexible to deal with different application domains.

Although the BN is generally used to estimate the probability vector of the output variable, we show a case of a real application for finding the estimation of the Bit's Rate of Penetration (ROP) on the pre-salt region offshore Brazil. It's a complex domain and depends on different variables, which can be either controlled by the drilling operator or from the geology. The available data comes from previously perforations and maybe not fully represent the new perforation and, besides that, it could have outliers or wrong values from sensor failures.

A Bayesian Network approach for the ROP's problem is relatively recent and publications focus on how to determine a good topology for the network. Rajaieyamchee and Bratvold (2009) shows the use of Influence Diagrams (ID), also known as Bayesian Decision Networks, to have a good quality when faced with real situations involving drilling in the North Sea. Giese and Bratvold (2011) uses ID and interviews with experts in the field to make a topology of a Bayesian network that aim assist in decision making for engineers when designing the treatment of drilling fluids in Saudi Arabia. Al-Yami and Schubert (2012) presents a topology to aid the drilling fluids practice in Saudi Arabia and also shows the Bayesian network as an efficient alternative of the flow charts, since it's not necessary to constantly update them.

The ROP is a quantitative variable, measured on m/s. So, in this problem the objective is not the simple classification of data but finding the knowledge behind it and be able to estimate the numeric value of the output. To accomplish that, we used the result probability distribution of BN to proper inform the expected mean value of the variable.

This study is organized as follows: Section 2 provides necessary background about BN knowledge and terminology. Section 3 presents a brief overview of the optimization technique known as Genetic Algorithms (GA). Section 4 describes the proposed method. Section 5 introduces the optimization problem associated with the method, proposing an approach by GA. Section 6 shows the experimental results of this approach. Section 7 shows the discussion about the results and finally, in Section 8 we conclude the study.

1.1. Bayesian Networks

A Bayesian Network (BN) (Pearl, 1988) is a model of representation and reasoning of uncertainty that uses the conditional probability between variables of a specific domain, expressed by Directed Acyclic Graphs (DAG). Its graphical structure can tackle correlations between variables effectively, with appropriate language and efficient resources to represent the joint probability distribution over a set of random variables (Friedman and Goldszmidt, 1996).

Defining formally, a BN is a pair (G,P), where G = (V,E) is a DAG in which the nodes V = $\{v_1,\ldots,v_n\}$ represent the variables and edges E = $\{e_1,\ldots,e_m\}$ represent a direct correlation between each node in V and P is defined as a set of probabilistic parameters expressed through tables: Given a particular variable, a conditional probability distribution is made for each of their classes/values X = $\{x_1,\ldots,x_z\}$ joining each classes/value of their parents.

With that configuration, the network establishes that a variable is independent of all other variables except their descendants in the graph, given the state of its parents. The inference inside the network is done by the Bayes theorem Equation (1):

$$P\left(V = v \middle| X = x\right) = \frac{P(X = x|V = v).P(V = v)}{P(X = x)} \tag{1}$$

The joint probability is determined by the called chain rule and assumes the conditional independence between the variables Equation (2):

$$P\left(v_1,\ldots,v_n\right) = \prod_{i=1}^{n} P\left(V_i \middle| \text{parent}(V_i)\right) \tag{2}$$

where, parent(V_i) determines the set of parent nodes from V_i.

The BN reasoning is established in two distinct scenarios:

$$\begin{cases} \text{if input, then output} \\ \text{if output, then input} \end{cases}$$

1.2. Learning the Conditional Probability

To represent a BN, it is necessary to establish its structure as well as the probability tables (strength of association between variables) through the learning domain to be worked on. There are three ways to accomplish that: By data only (data base), by domain experts only, or by a hybrid form between data and experts.

The naive Bayes topology is therefore a set of mutually independent variables that works as input which collectively has a single parent (output node). One example of naive Bayes topology can be seen on **Fig. 1**. In this case, the node A is the output one and the nodes B, C and D are the inputs.

In addition to BN topology, it is necessary to specify the Conditional Probability Table (CPT) of each node, which lists the probability that the node takes on each of its different values in combination of its parents' values. An example of CPT for this BN is shown in **Table 1**.

1.3. Discretization Based on Frequency

In quantitative cases, the probability of a particular value x_i given a variable in V can be infinitely small. Discretization can circumvent this problem, converting each original quantitative value (x_i) into a qualitative value (x_i^*) under some pre-defined criteria, but information loss may become an issue (Yang and Webb, 2009).

One of the most common approach for discretization of the existing data of quantitative variables is the Equal Frequency Discretization (EFD) (Catlett, 1991; Kerber, 1992; Dougherty *et al*., 1995), that sorts the values on X and divides them into k intervals (user-defined) so that each interval contains approximately the same number of instances. The Algorithm on **Fig. 2** is used for the EFD method.

In Descriptive Statistics, a widely used measure for data separation is the quartile. Quartiles separate data set into four equal parts where each one contains 25% of the data. The first quartile Q_1 is called the lower quartile, the third quartile Q_2 is called the upper quartile and the second quartile is the median itself. The interquartile range is known as the distance between the first and the third quartile.

A possible way to discretize the data comes from the EFD method in combination with the concept of the interquartile range (**Table 2**), here called as QD.

Other techniques, besides the EFD and the QD are also applied in the literature, such as Lazy Discretization (LD) (Hsu *et al*., 2003), Proportional Discretization (PD) (Moore and Neal, 2005) and Fixed Frequency Discretization (FFD) (Yang and Webb, 2009).

1.4. Basic Refence on Genetic Algorithms

Genetic Algorithms (GAs) are function optimizers, i.e., methods for seeking extreme of a given objective function f(x) based on principles of natural selection and population genetics (Goldberg, 1989; Cantu-Paz, 1995; Weile and Michielssen, 1997). The objective function of the problem is typically used to express the fitness function in GA.

Table 1. Conditional Probability Table (CPT) example

A	P(B = state0)	P(B = state1)	P(B = state2)
state0	0.2	0.3 0	0.50
state1	0.1	0.5 0	0.4 0
state2	0.1	0.05	0.85

Table 2. Quartile based discretization

Condition	Result discretization
$x_i \leq Q_1$	"low"
$Q_1 \leq x_i \leq Q_3$	"medium"
$x_i > Q_3$	"high"

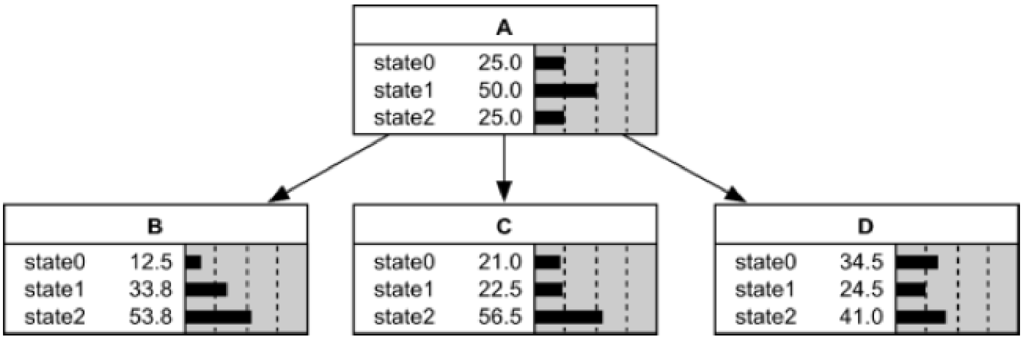
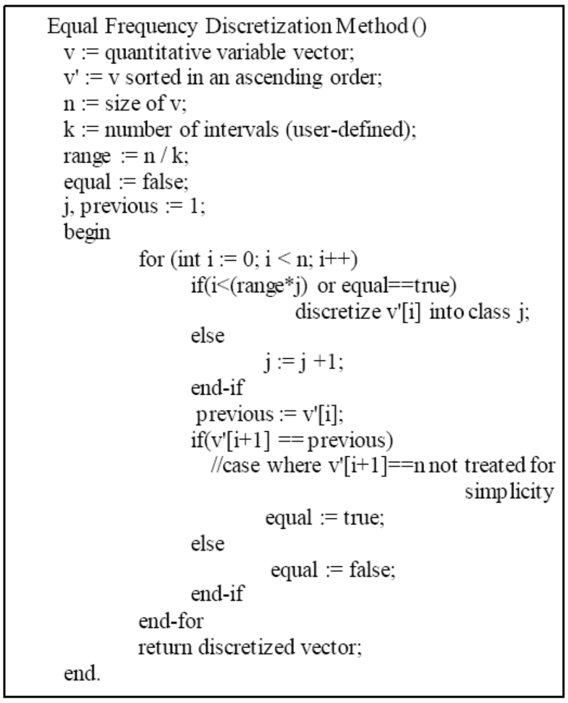


Fig. 1. Algorithm for EFD method



```
Equal Frequency Discretization Method ()
  v := quantitative variable vector;
  v' := v sorted in an ascending order;
  n := size of v;
  k := number of intervals (user-defined);
  range := n / k;
  equal := false;
  j, previous := 1;
  begin
        for (int i := 0; i < n; i++)
              if(i<(range*j) or equal==true)
                    discretize v'[i] into class j;
              else
                    j := j +1;
              end-if
              previous := v'[i];
              if(v'[i+1] ==previous)
                 //case where v'[i+1]==n not treated for
                                                  simplicity
                    equal := true;
              else
                    equal := false;
              end-if
        end-for
        return discretized vector;
  end.
```

Fig. 2. Algorithm for EFD method

An important aspect of the fitness function is its responsibility to measure the performance of the solution (objective function) as a way to generate an allocation of resources to reproduction (Whitley, 1994).

An individual is defined as a valid candidate solution in GA, expressed by either a binary string or a vector of float numbers (Janikow and Michalewicz, 1991; Wright, 1991), where a set of individuals is considered a population. Three operators are commonly used: Selection, crossover and mutation.

The selection operator uses the fitness of each individual to choose the most adapted ones of the current population to result in a new generation. There are several ways to accomplish this selection of individuals, but it always ensures that the better adapted individuals (best fitness) have a higher probability to be selected.

The reproduction is made by operators of crossover and mutation. The first one is the primary exploration mechanism of GA: It randomly chooses a pair of preselected individuals and exchanges information (substring in a binary representation) between them to create new individuals.

The mutation operator is generally considered as a secondary operator and is used to prevent the solution from becoming stagnant at some local minima. Mutation is done by selecting a random substring in an individual and changing its value. The percentage defined for this operator is usually much smaller than the crossover operator.

The GA starts with a current population and then selection is applied to create an intermediate population. Recombination (mutation and crossover) is then used to create the next population. The process between the current population to the next population is called a generation in the execution of GA (Whitley, 1994).

The GA convergence tends to evolve through successive generations until the fitness of the best individual and the average fitness of a population approach the global optimal (Beasley *et al*., 1993). Genetic Algorithms do not guarantee finding the optimal solution and its effectiveness is determined by the size of the population n. The time required for a GA to converge is O(nlogn) function evaluations (Goldberg, 1989).

2. MATERIALS AND METHODS

The proposed method for discretization in Bayesian Networks, Peak-Valley Discretization method (PVD), assumes that numeric variables V have a range where intermediate values are inserted and in a complementary way, analyzing this range of intermediate values makes it possible to obtain the range of extreme values and establish their conditional probabilities, as well as the relations of cause and effect: “What caused this behavior? What does it entail?”

Observing the behavior of the variable, is possible to tell where a value x_i is out of a given range, positively (high) or negatively (low). The delimitation of range uses two cut points expressed in percentile: The peak one is restricted to the area where values are considered “high”‘ and the second one, valley, covers the area where the data is considered “low”. The range of intermediate values is defined by the interval between two cuts. The use of percentile as cut point’s measures incorporates the frequency distribution of variables on the method (following the line of EFD).

However, the behavior of a numeric variable is unknown and it is not possible to assume that it has higher values as well as lower ones. Considering this prerogative, it is possible to characterize data in two or three behaviors-defined as classes in a BN. In other words, a variable can have a negligible valley or peak cut value, or these points can be so close to each other that an intermediate range is irrelevant.

2.1. PVD Properties

To elucidate the properties of PVD, the following concepts are defined in the context of a variable v_i:

- p(x) as a function that takes a value x as input and returns the percentile in which it is located

- $p^{-1}(y)$ as the inverse function of $p(x)$: Takes a percentile y as input and returns the value x that it represents
- valley as the percentile expressed by the valley cut point
- peak as the percentile expressed by the peak cut point
- p(xmin) as the percentile that represent the lowest value x_{min} in v_i
- $p(x_{max})$ as the percentile that represent the highest value x_{max} in v_i
- $X^* = \{x_1^*,\ldots,x_n^*\}$ as the discretized vector of classes from node v $(X = \{x_1\ldots x_z\})$

It is possible to merge or despise cut points if they are not relevant to the solution. The relevance of the cut points and its proximity to the boundary values are expressed by the coefficient of relevance α $(0<\alpha<1)$ defined by user, that determines how close these values are.

It is necessary, however, to apply a correction in α to ensure that the cuts always have a range of values to be considered relevant independent of the proximity of x_{min} to x_{max}. The adjusted coefficient α' goes by Equation (3):

$$\alpha' = ((1-\delta)\cdot\alpha) + \delta \tag{3}$$

where, δ is the boundary coefficient between x_{min} and x_{max} defined by:

$$\delta = \sqrt{\frac{x_{min}}{x_{max}}} \tag{4}$$

Which implies that the limit of Equation (4) when $\delta \to 0$ is as follow Equation (5):

$$\lim_{\delta\to 0}((1-\delta)\cdot\alpha) + \delta = \alpha \tag{5}$$

The relevance of cuts through the proximity of each with x_{min} and x_{max} is determined by α'. The lowest relevant value of valley is given by:

$$p^{-1}(\text{valley}_{min}) = \frac{x_{min}}{\alpha'} \tag{6}$$

And the highest relevant value of peak is:

$$p^{-1}(\text{peak}_{max}) = x_{max}\cdot\alpha' \tag{7}$$

Through the Equation (6 and 7) and considering that both cuts have different definitions, it is possible to define the following hierarchy Equation (8):

$$p_{x_{min}} \le \text{valley} \le \gamma \le \text{peak} \le p_{x_{max}} \tag{8}$$

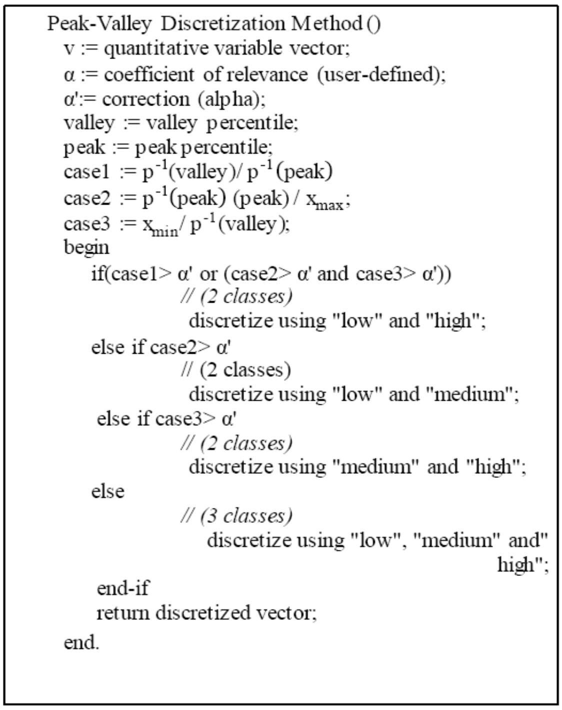
where Equation (9):

$$\gamma = \frac{\text{valley}_{min} + \text{peak}_{max}}{2} \tag{9}$$

Represents the limit between peak and valley. The following criteria are used to despise or merge cuts:

$$\begin{cases} \text{if } \dfrac{p^{-1}(\text{valley})}{p^{-1}(\text{peak})} > \alpha', \text{then merge by} \dfrac{\text{valley}+\text{peak}}{2} \\ \text{if } \dfrac{p^{-1}(\text{peak})}{x_{max}} > \alpha' \text{then ignor the peak cut} \\ \text{if } \dfrac{x_{min}}{p^{-1}(\text{valley})} > \alpha' \text{then ignor the peak cut} \\ \text{if ignor both the peak and the valley cuts, then merge} \\ \text{by} \dfrac{\text{valley}+\text{peak}}{2} \end{cases} \tag{10}$$

The BN feature of explicitly representing knowledge creates a concern regarding class names in X^*, which should be intuitive and express their features. To supply for such requirement, class names were chosen based on Equation 10 and expressed by the Algorithm expressed on **Fig. 3**.



```
Peak-Valley Discretization Method()
  v := quantitative variable vector;
  $\alpha$ := coefficient of relevance (user-defined);
  $\alpha'$:= correction (alpha);
  valley := valley percentile;
  peak := peak percentile;
  case1 := $p^{-1}$(valley)/ $p^{-1}$(peak)
  case2 := $p^{-1}$(peak) (peak) / $x_{max}$;
  case3 := $x_{min}$/ $p^{-1}$(valley);
  begin
     if(case1> $\alpha'$ or (case2> $\alpha'$ and case3> $\alpha'$))
                  // (2 classes)
                   discretize using "low" and "high";
     else if case2> $\alpha'$
                  // (2 classes)
                   discretize using "low" and "medium";
      else if case3> $\alpha'$
                  // (2 classes)
                   discretize using "medium" and "high";
     else
                  // (3 classes)
                     discretize using "low", "medium" and"
                                                      high";
      end-if
      return discretized vector;
  end.
```

Fig. 3. Algorithm for PVD method

2.2. The Optimization Problem for Quantitative Output

The following concepts are defined as:

- $vout$ as the output variable in V
- $V^* = \{v_1^*,\ldots,v_n^*\}$ as the vector of all discretized variables in V: Pre-discretized or by PVD
- $vout^*$ as the output variable in V^*
- $\tilde{X} = \tilde{x}_1,....,\tilde{x}_n$ as the predicted values of v^*_{out} by BN

When the output variable is quantitative the algorithm goal goes beyond the classification: It is necessary that the mean estimated by the probability vector reflects the behavior of the variable. The follow function returns the expected quantitative value of the v^*_{out} node in BN, based on current beliefs and a list of real numbers that represent each class in v^*_{out} Equation (11):

$$ev(x) = \sum_{i=1}^{n} belief_i \cdot midpoint_i \qquad (11)$$

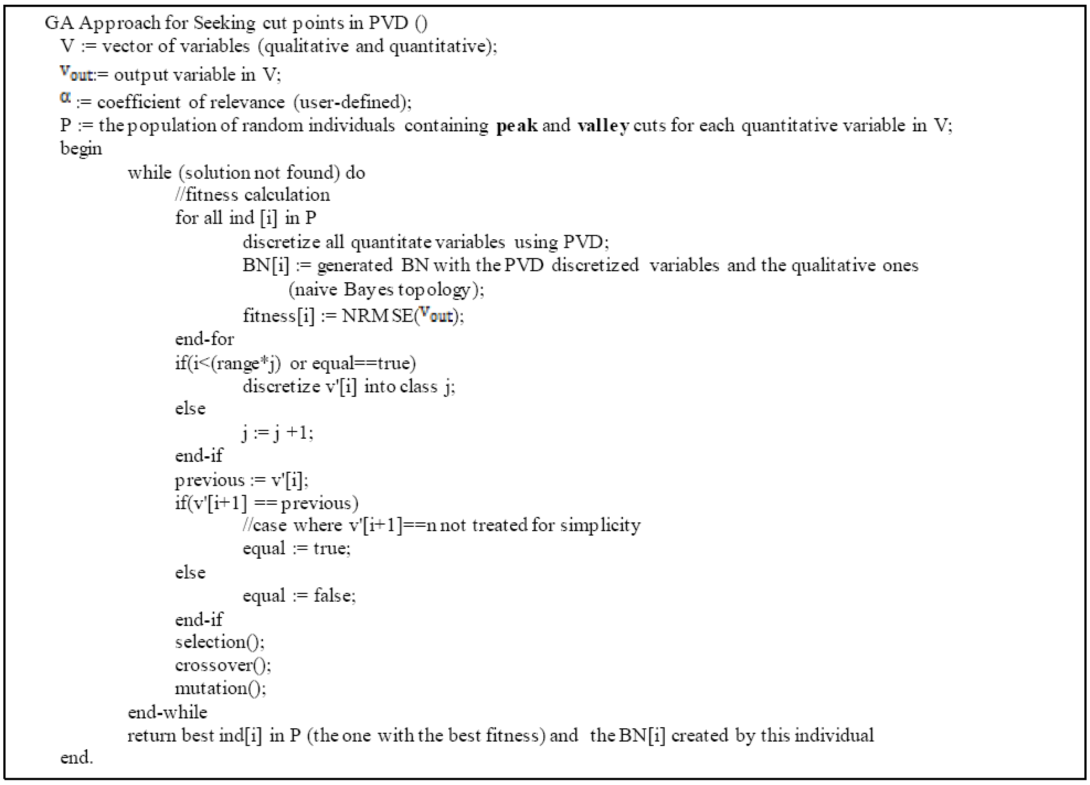
where the list of real numbers is handled as the respective midpoints of each class in v^*_{out} in relation to v_{out}.

Discretization of a variable v_i in PVD depends on two cut points: Peak and valley and a pre-defined coefficient of relevance α. However, probability distribution of v_i influences the inference process of the entire BN Equation (1).

Thus, it is required to discretize all variables simultaneously, which generates a Global Optimization problem (Horst *et al*., 2002), that is, finding the best set of acceptable conditions to achieve an objective formulated in mathematical terms.

The objective of such optimization problem consists in choosing an output variable in BN and discretizing all the other quantitative variables so that the Bayesian classification of the output values is as close to the actual value as possible. Assuming that v_{out} is quantitative, the objective function is given by the minimization of the Normalized Root Mean Square Error (NRMSE) between the estimated mean value and the actual value of the variable:

$$\text{find } V^* = \min NRMSE\left(v_{out}\right) \qquad (12)$$



```
GA Approach for Seeking cut points in PVD ()
  V := vector of variables (qualitative and quantitative);
  vout:= output variable in V;
  α := coefficient of relevance (user-defined);
  P := the population of random individuals containing peak and valley cuts for each quantitative variable in V;
  begin
        while (solution not found) do
              //fitness calculation
              for all ind [i] in P
                    discretize all quantitate variables using PVD;
                    BN[i] := generated BN with the PVD discretized variables and the qualitative ones
                          (naive Bayes topology);
                    fitness[i] := NRMSE(vout);
              end-for
              if(i<(range*j) or equal==true)
                    discretize v'[i] into class j;
              else
                    j := j +1;
              end-if
              previous := v'[i];
              if(v'[i+1] == previous)
                    //case where v'[i+1]==n not treated for simplicity
                    equal := true;
              else
                    equal := false;
              end-if
              selection();
              crossover();
              mutation();
        end-while
        return best ind[i] in P (the one with the best fitness) and the BN[i] created by this individual
  end.
```

Fig. 4. GA Approach Algorithm for PVD method

Where:

$$\mathrm{NRMSE}\left(v_{out}\right)=\frac{100\cdot\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i-ev(\breve{x}_i)\right)}}{x_{max}-x_{min}} \qquad (13)$$

The algorithm expressed on **Fig. 4** shows the workflow that satisfies the objective function (Equation 12), by utilizing the technique of Genetic Algorithm (GA).

3. RESULTS

The proposed method was tested in a data set of Bit's Rate of Penetration Problem (Section 6.1), which was randomly separated so that (0.7n) of the data belongs to the training set and (0.3n) to the test set. The output variable is the "ROP" and α (coefficient of relevance) adopted is 0.8.

3.1. Bit's Rate of Penetration Problem

Environments of high complexity and risk, such as the pre-salt region of Brazil, aim to optimize the cost of drilling wells. The minimization of these costs is directly related to the maximization of Rate of Penetration (ROP).

However, each operation has unique properties that make this task highly difficult. Many properties vary, such as rock type, rock porosity, gas presence, pressure, drill bit wear rate, among others. All these properties affect the ROP, as well as many other parameters which are controlled by a drilling operator.

There are 277 data points listed in the data set used about a specific type of drilling bit, using the value of ROP in a quantitative way (m/s). The input parameters have quantitative values, named: Revolutions Per Minute (RPM), Weight on Bit (WOB), HSI (bit hydraulic horsepower per square inch) and accumulated meters.

The first three variables are intrinsic to the drilling process, the last one (accumulated meters) has a linear and accumulative behavior bringing information about the bit wear.

There is also a qualitative parameter, discretized by domain experts: Unconfined Compressive Strength (UCS) related to soil geology.

3.2. Generated Bayesian Networks

The training set were discretized according to each one of the methods (PVD and QD) and then created the Bayesian networks (**Fig. 5 and 6**).

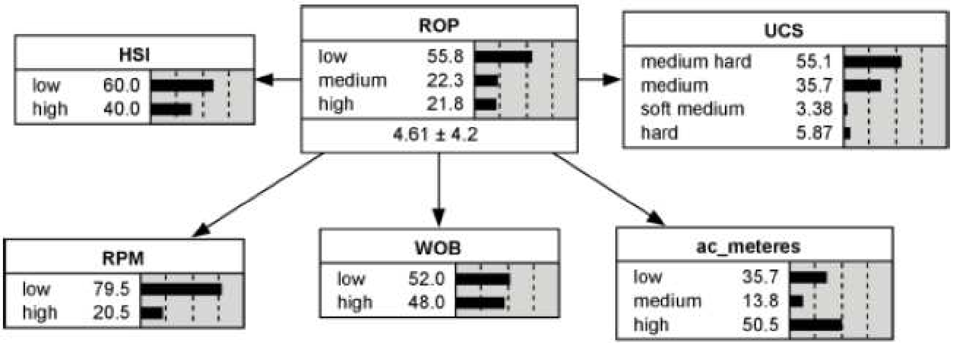


Fig. 5. Trained BN by PVD for Bit's ROP

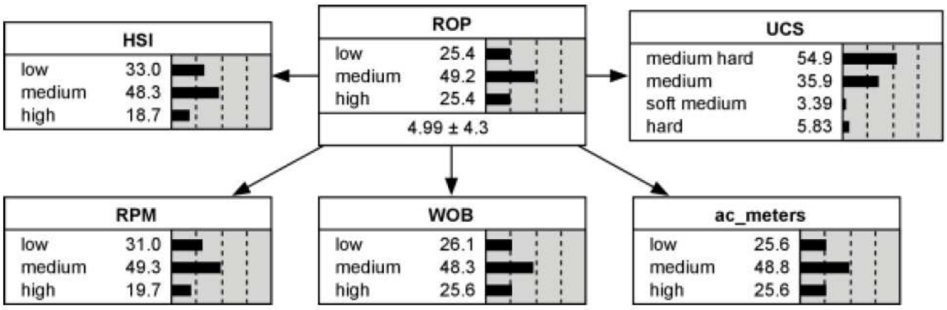


Fig. 6. Trained BN by QD for Bit's ROP

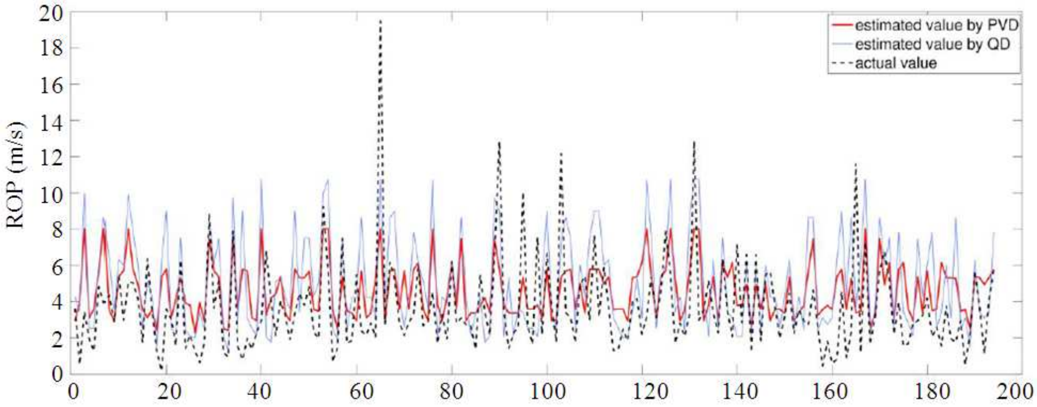


Fig. 7. Actual and estimated values of the Bit's ROP (training set)

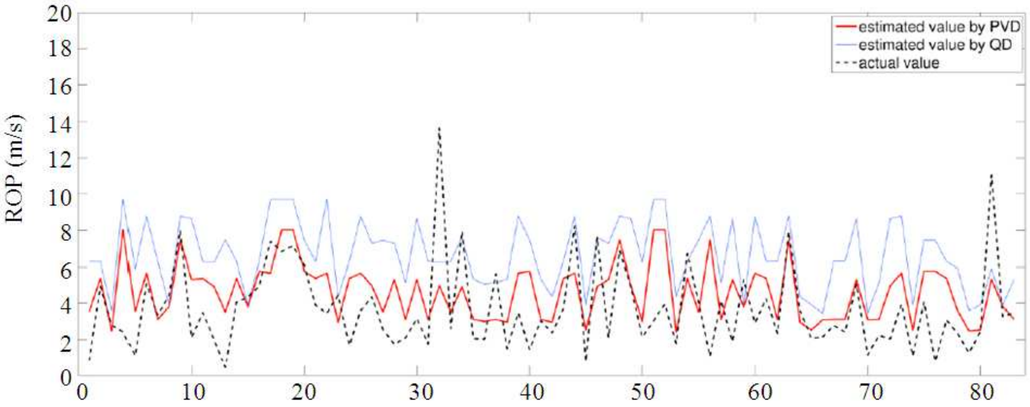


Fig. 8. Actual and estimated values of the Bit's ROP (test set)

Table 3. Classification matrix for Bit's ROP Problem (training set)

	Predicted					
Approach	Actual	Low	Medium	High	Total	Accuracy
QD	Low	20	27	2	49	51.03%
	Medium	19	53	24	126	
	High	5	18	26	49	
	Low	100	5	4	109	63.91%
PVD	Medium	24	14	5	43	
	High	27	5	10	42	

Table 4. Classification matrix for Bit's ROP Problem (test set)

	Predicted					
Approach	Actual	Low	Medium	High	Total	Accuracy
QD	Low	0	34	15	49	
	Medium	0	17	17	34	20.48%
	High	0	0	0	0	
	Low	38	2	3	43	
PVD	Medium	18	6	1	25	59.03%
	High	8	2	5	15	

Table 5. Obtained NRMSE of Bit's ROP Problem

	NRMSE	
Approach	Training (%)	Test (%)
QD	15.46	29.69
PVD	10.66	18.44

Each class of the ROP output node had its midpoint value calculated in this process (Equation 11).

The test set were then discretized using the same cut points found in the training set. Classification matrices of the training and test set are shown on **Table 3 and 4**.

3.3. Estimated Mean Values

The probability distribution of the output node was used to estimate the values of the variable. With the estimated mean values and the actual value the NRMSE was calculated for each one of the methods (**Table 5**).

The actual and estimated values of the methods are shown in **Fig. 6** (training set) and **Fig. 7** (test set).

4. DISCUSSION

The PVD method uses a heuristic that aims to minimize the NRMSE of the training set through the search for two cut points (peak and valley) by Genetic Algorithm. An experiment was conducted using a real data set of Bit's ROP in order to estimate the mean values of the output variable in two different methods: The PVD proposed method and the QD method.

The data set is derived from a drilling process under the influence of various factors, such as equipment operators, geology and sensors measure. Therefore, the data is not always reliable and the application domain is considered a complex domain.

In the training set there is a greater accuracy when the PVD method is used (**Table 3**). The division between the classes of the output variable is not the same in PVD and QD methods, since the QD divides the data in a proportional way and PVD shows an asymmetric division in this experiment.

The proportional behavior of the frequency distribution is kept for the entire training set when using the QD method (**Fig. 6**), however in the PVD method each variable get a particular distribution (**Fig. 5**).

In relation to the NRMSE on the Bit's ROP problem, the PVD method shows a lower error in both training and test sets which reinforces its generalization capacity (**Table 5**). When looking at the training set the PVD has an error approximately 31% lower than the QD. In the test set this difference is even more evident, the PVD has an error approximately 38% lower than the QD method.

In relation to the NRMSE on the Bit's ROP problem, the PVD method shows a lower error in both training and test sets which reinforces its generalization capacity (**Table 5**). When looking at the training set the PVD has an error approximately 31% lower than the QD. In the test set this difference is even more evident, the PVD has an error approximately 38% lower than the QD method.

The PVD's generalization capability is also demonstrated by the graph expressed in **Fig. 7** showing a greater adherence to the actual data curve from the PVD over the QD method in the training set. In the test set, the PVD shows a significantly greater adherence to the actual data than the QD (**Fig. 8**). The QD method also tends to overestimate the estimated values in the test set.

5. CONCLUSION

The proposed method performs discretization using two cut points which identify valley events ("low"), peak events ("high") and intermediate events ("medium") and was applied in a real domain of Bit's Rate of Penetration.

The PVD method makes discretization independent from the frequency distribution of each variable. By observing the generated BN, it is possible to infer that the class probability distribution ("low", "medium" and "high") can either tend towards symmetry or asymmetry. The frequency distribution of the classes variables found by the PVD reinforces the idea that a symmetrical distribution of the classes on discretization does not necessarily lead to a better performance of the network.

The estimated mean from probability distribution of the BN generated by PVD reflected the data behavior well, although it was not able to accurately reproduce the actual extreme values of the variable, but does not tends to overestimate like the QD method. The PVD also had a better accuracy in classification, lower NRMSE on the estimated values and a better generalization of the problem when compared with the QD.

With the presented results, we conclude that the proposed discretization method is more effective and has a better knowledge representation of the problem than a conventional approach for discretization like the QD that uses a proportional division of the data based on the quartiles.

6. ACKNOWLEDGEMENT

This study was supported by Petrobras and the Universidade Federal de Santa Catarina (UFSC) as part of a research project.

7. REFERENCES

Alfred, R., 2009. Discretization numerical data for relational data with one-to-many relations. J. Comput. Sci., 5: 519-528. DOI: 10.3844/jcssp.2009.519.528

Al-yami, A.S. and J. Schubert, 2012. Using bayesian network to model drilling fluids practices in Saudi Arabia. Soc. Petroleum Eng.

Beasley, D., R.R. Martin and D.R. Bull, 1993. An overview of genetic algorithms: Part 1. Fundamentals. Univ. Comput., 15: 58-58.

Cantu-Paz, E., 1995. A summary of research on parallel genetic algorithms. CiteSeerX.

Catlett, J., 1991. On changing continuous attributes into ordered discrete attributes. Mach. Learn. EWSL, 482: 164-178. DOI: 10.1007/BFb0017012

Ding, Y., L.M. Zhu, X.J. Zhang and H. Ding, 2010. A full-discretization method for prediction of milling stability. Int. J. Mach. Tools Manufact., 50: 502-509. DOI: 10.1016/j.ijmachtools.2010.01.003

Dougherty, J., R. Kohavi and M. Sahami, 1995. Supervised and unsupervised discretization of continuous features. Proceedings of the Machine Learning-International Workshop Then Conference, (IWC' 95), Morgan Kaufmann, pp: 194-202.

Friedman, N. and M. Goldszmidt, 1996. Discretizing continuous attributes while learning Bayesian networks. Proceedings of the Machine Learning-International Workshop then Conference (IWC' 96), Morgan Kaufmann, pp: 157-165.

Giese, M. and R.B. Bratvold, 2011. Probabilistic modeling for decision support in integrated operations. SPE Econ. Manage.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. 1st Edn., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA., ISBN-10: 0201157675, pp: 432.

Horst, R., P.M. Pardalos and H.E. Romeijn, 2002. Handbook of Global Optimization. 1st Edn., Kluwer Academic, Dordrecht, Boston, ISBN-10: 1402006322, pp: 572.

Hsu, C.N., H.J. Huang and T.T. Wong, 2003. Implications of the dirichlet assumption for discretization of continuous variables in naive Bayesian classifiers. Mach. Learn., 53: 235-263. DOI: 10.1023/A:1026367023636

Janikow, C.Z. and Z. Michalewicz, 1991. An experimental comparison of binary and floating point representations in genetic algorithms. Proceedings of the 4th International Conference on Genetic Algorithms, (CGA' 91), San Diego, CA, USA.

Kerber, R., 1992. ChiMerge: discretization of numeric attributes. Proceedings of the 10th National Conference on Artificial Intelligence, (CAI' 92), ACM Press, pp: 123-128.

Mohammed, B.O. and S.M. Shamsuddin, 2011. Feature discretization for individuality representation in twins handwritten identification. J. Comput. Sci., 7: 1080-1087. DOI : 10.3844/jcssp.2011.1080.10874

Moore, D.S. and D.K. Neal, 2005. Introduction to the Practice of Statistics TI-83 Graphing Calculator Manual. 1st Edn., W. H. Freeman, ISBN-10: 0716763648, pp: 160.

Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. 1st Edn., Morgan Kaufmann, ISBN-10: 1558604790, pp: 552.

Rajaieyamchee, M.A. and R. Bratvold, 2009. Real time decision support in drilling operations using bayesian decision networks. Proceedings of SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers, (ESPE' 09) pp: 1-17.

Weile, D.S. and E. Michielssen, 1997. Genetic algorithm optimization applied to electromagnetics: A review. IEEE Trans. Antennas Propagat., 45: 343-353. DOI: 10.1109/8.558650

Whitley, D., 1994. A genetic algorithm tutorial. Stat. Comput., 4: 65-85. DOI: 10.1007/BF00175354

Wright, A.H., 1991. Genetic algorithms for real parameter optimization. Foundat. Genetic Algorithms, 1: 205-218.

Yang, Y. and G.I. Webb, 2009. Discretization for naive-bayes learning: Managing discretization bias and variance. Mach. Learn., 74: 39-74. DOI: 10.1007/s10994-008-5083-5