From Continuous to Multiple-valued Data

Denis V. Popel
Computer Science Department
Baker University, Baldwin City, KS 66006
denis.popel@bakeru.edu

Abstract

In modern science, significant advances are typically made at cross-roads of disciplines. Thus, many optimization problems in Multiple-valued Logic Design have been successfully approached using ideas and techniques from Artificial Intelligence. In particular, improvements in multiple-valued logic design have been made by utilizing information/uncertainty measures. In this respect, the paper addresses the problem known as discretization and introduces a method of finding an optimal representation of continuous data in the multiple-valued domain. The paper introduces new information density measures and an optimization criterion. We propose an algorithm that incorporates new measures and is applied to both unsupervised and supervised discretization. The experimental results on continuous-valued benchmarks are given to demonstrate the efficiency and robustness of the algorithm.

1 Introduction

Continuous data analysis has gone through a dramatic change in the last decade and become an essential component in signal processing and knowledge discovery applications. This change and renewed interest in mixed analog/continuous/discrete systems have been facilitated by recent advances in continuous-discrete transformation techniques, where discrete in many cases implies multiple-valued representation. This problem, known as quantization, covers a variety of applications ranging from circuit synthesis to data mining:

Circuit Synthesis: The problem of quantization arises in many circuit applications including adaptive signal processing systems. The essential component of such systems, *long term analog memory*, is designed having quantizers (combinations of A/D and D/A converters) that store the discrete level nearest to analog input data [4]. For example, the current-mode memory proposed in [1] is based on continuous algorithmic converters, which achieve an efficient trade-off between write/read speed and power consumption. The application in circuits also include *field programmable analog arrays* introduced in [6] for continuous, fuzzy, and multi-valued logic. Further extensions in this direction might include the development of multiple-valued, fuzzy and continuous reversible logic.

Logic Design: The transition from continuous valued digits to discrete level multiple-valued digits is also important in the implementation of arithmetic operations using the theory of *analog digits* [11]. The reverse problem of finding continuous representation for discrete data is considered for time series in [5]. Another area of applying discretization techniques and pattern finding for continuous data is the *decomposition* of logic functions. Thus, the method introduced in [10] covers the decomposition of functions whose variables include continuous values.

Data Mining: Substantial amount of work on quantization has been done by the data mining and database research communities. Some examples include the modifications of $ID3$ algorithm [9] and the discretization of continuous attributes [3, 2]. With the set of continuous variables as initial representation of database information, the problem is to find a technique that converts the database information into a multiple-valued function [8].

The purpose of this paper is twofold: (i) to initiate a discussion on continuous vs. multiple-valued representations, particularly concerning their use in circuit synthesis, logic design, and data mining, and (ii) to provide an introduction to some new measures from information theory and highlight their importance to the discretization problem. The research contributes to the pool of already existing techniques for uncertainty measuring. Thus, this paper

Proceedings of the 33rd International Symposium on Multiple-Valued Logic (ISMVL'03)
0195-623X/03 $17.00 © 2003 IEEE

concentrates on impurity measures and introduces information density estimators. In addition, it resolves the problem of finding a multiple-valued representation for continuous valued signals. Here, we utilize information density estimators and develop a unified discretization algorithm.

The breakdown of these efforts is given as follows. Section 2 reviews known discretization techniques emphasizing the background of the research. The new approach to unsupervised and supervised discretization of continuous valued signals is outlined in Section 3. Section 4 discusses the algorithm and demonstrates the results of our study on continuous valued benchmarks. Section 5 concludes the paper.

2 Multi-interval Discretization

Discretization is a process of transforming values of a continuous variable into a finite number of intervals, and associating with each interval a discrete numerical value. There are three different dimensions by which discretization approaches can be classified: global vs. local, static vs. dynamic, and unsupervised vs. supervised. We refer to [2] for the review of all discretization approaches. There are two basic techniques of discretization: the one which operates on the single continuous variable at a time is called *local*, in contrast to which, the *global* discretization processes the entire set of continuous variables. Some discretization approaches called *static* require an *a priori* set parameter η of the maximum number of intervals to produce. *Dynamic* approaches search through the space of all possible η values for all continuous variables simultaneously capturing differences in discretization. The first approach is important in multiple-valued representations, because of existing constrains on the number of intervals. In this paper, we work with local static discretization only. *Unsupervised discretization* does not utilize class labels in setting partition boundaries. *Supervised discretization* discretizes variables by taking into account the class labels assigned to examples. In our research, we consider a unified solution for both discretization approaches.

The major drawback of existing discretization techniques is that unsupervised discretization is considered without reference to supervised discretization. In addition to traditional information theory measures[1], some criteria such as *Gini Index* and *Information Gain* [9] have been used. These criteria belong to the class of impurity measures designed to capture some aspects of partitioning relevant to "good" classification. We refer the reader to the paper [13] for detailed theoretical research on impurity measures. This paper suggests employing unified measures for both approaches based on an information density concept. The goal is to develop a fast and efficient discretization algorithm which is applicable to any set of continuous data. The algorithm is supposed to maximize interdependence between the generated discrete intervals and generate the smallest number of intervals for a given continuous variable.

3 Multiple-Valued Functions and Intervals

Let us use the following notation. Denote by s_i the value of a continuous variable x from the set of given values $S = \{s_1, \ldots, s_k\}$ assuming that x can take values in the range $T = [T^{min}, T^{max}]$, where $T^{min} < T^{max}$ and $\Delta T = T^{max} - T^{min}$. A partition $\mathcal{P}$ of the set S into r intervals $\mathcal{P}_u = [\mathcal{P}_u^{min}, \mathcal{P}_u^{max}]$, $u = 0, \ldots, r-1$, can be characterized as

$$\begin{cases} \mathcal{P} = \bigcup_{u=0}^{r-1} P_u, \text{ and } \mathcal{P}_u \neq \oslash \text{ for } u = 0, \ldots, r-1, \\ s_i \notin \mathcal{P}_v, \text{ if } s_i \in \mathcal{P}_u \text{ for } u \neq v. \end{cases}$$

Taking boundaries T^{min} and T^{max} into consideration, the partitions can be described by

$$\begin{cases} \mathcal{P}_0^{min} = T^{min}, \\ \mathcal{P}_u^{min} > T^{min} \text{ and } \mathcal{P}_u^{max} < T^{max}, u = 1, \ldots, r-2, \\ \mathcal{P}_{r-1}^{max} = T^{max}. \end{cases}$$

We study below the transition of the continuous variable x into the r-valued variable χ. Let us define new information measures that are used in unsupervised (with no information about class labels) and supervised (with class labels attached) discretization.

3.1 Unsupervised Discretization

We introduce the following class-independent measures.

[1] In order to quantify the content of information for a finite field of events $A = \{a_1, a_2, \cdots, a_n\}$ with the probability distribution $\{p(a_i)\}$, $i = 1, 2, \cdots, n$, Shannon introduced the concept of entropy [12]. *Entropy* of the finite field A is given by $H(A) = -\sum_{i=1}^{n} p(a_i) \cdot \log p(a_i)$. Suppose there are two finite fields of events A and B with probability distribution $\{p(a_i)\}, i = 1, 2, \cdots, n$, and $\{p(b_j)\}, j = 1, 2, \cdots, m$, respectively. Let $p(a_i, b_j)$ be the probability of the joint occurrence of a_i and b_j. The *joint entropy* of A and B is $H(A, B) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p(a_i, b_j) \cdot \log p(a_i, b_j)$. It can be easily shown that the uncertainty of a joint event is less than or equal to the sum of the distinct uncertainties: $H(A, B) \leq H(A) + H(B)$. The *conditional entropy* of A given B is defined by $H(A|B) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p(a_i, b_j) \cdot \log p(a_i|b_j)$. This quantity determines how uncertain we are of A given B. Using equations for joint and conditional entropy we obtain $H(A, B) = H(B) + H(A|B)$. The *mutual information* between two finite fields A and B is defined by $I(A; B) = H(A) + H(B) - H(A, B)$. One can show that $I(A; B) = H(A) - H(A|B)$. The basic problem tackled by discretization algorithms is how to maximize the conditional entropy of the output A given its input B.

Proceedings of the 33rd International Symposium on Multiple-Valued Logic (ISMVL'03)
0195-623X/03 $17.00 © 2003 IEEE

Definition 1 *The Entropy $H(T,S)$ of the partition $\mathcal{P}$ is defined by*

$$H(T,S) = -\sum_{i=0}^{k} \delta_i \cdot \log \delta_i, \quad (1)$$

where the probabilities δ_i are

$$\delta_i = \begin{cases} (s_1 - T^{min})/\Delta T, \ if\ i = 0, \\ (s_{i+1} - s_i)/\Delta T, \ if\ i = 1, \ldots, k-1, \\ (T^{max} - s_k)/\Delta T, \ if\ i = k. \end{cases}$$

The quantity $H(T,S)$ has a number of interesting properties which are utilized in our research.

Property 1 *$H(T,S) = 0$ if and only if the set S has no values: $S = \oslash$. Otherwise, $H(T,S)$ is positive.*

Property 2 *Given k, $H(T,S)$ is maximal and equal to $\log(k+1)$ when all δ_i take equal values, that value being $1/(k+1)$. This is, intuitively, the case of the most uncertainty for the uniform distribution of values.*

Property 3 *Any change toward equalization of probabilities $\delta_0, \ldots, \delta_k$ leads to increasing $H(T,S)$. Thus, if $\delta_0 < \delta_1$ and δ_0 is increasing, δ_1 is decreasing by the same amount, and if δ_0 and δ_1 are nearly equal, then $H(T,S)$ is close to its maximum value.*

Definition 2 *The Information Density $D(T,S)$ of the partition $\mathcal{P}$ is given by*

$$D(T,S) = \begin{cases} H(T,S)/\log(k+1), \ if\ k > 0, \\ 0, \ if\ k = 0. \end{cases} \quad (2)$$

Making the definition consistent, we set $D(T,S) = 0$ for the intervals where $S = \oslash$.

Property 4 *The terminal intervals $[T^{min}, s_1]$ and $[s_k, T^{max}]$ make no contribution to the information density $D(T,S)$. These segments of $D(T,S)$ have values of zero.*

3.2 Supervised Discretization

Here we generalize the definitions presented above to cover supervised discretization. In supervised discretization, the dependencies between the set of class labels $C = \{c_0, \ldots, c_{m-1}\}$, where m is the number of classes, and the set of values $S = \{s_1, \ldots, s_k\}$ can be illustrated as a relation: $S \rightarrow C$. The notation for values in S can be adjusted as follows: $S^c = \{s^c | c \in C\}$, where s^c is the value s together with the associated class label c. Thus, $S = \bigcup_{j=0}^{m-1} S^{c_j}$.

Definition 3 *The Class Entropy $\mathcal{H}(T, S^{c_j})$ of the partition $\mathcal{P}$ is defined by*

$$\mathcal{H}(T, S^{c_j}) = -\sum_{i=0}^{k} \delta_i^{c_j} \cdot \log \delta_i^{c_j}, \quad (3)$$

where the probabilities $\delta_i^{c_j}$ are

$$\delta_i^{c_j} = \begin{cases} (s_1^{c_j} - T^{min})/\Delta T, \ if\ i = 0, \\ (s_{i+1}^{c_j} - s_i^{c_j})/\Delta T, \ if\ i = 1, \ldots, k-1, \\ (T^{max} - s_k^{c_j})/\Delta T, \ if\ i = k. \end{cases}$$

The class entropy $\mathcal{H}(T, S^{c_j})$ employs Properties 1–3. Hence, for the entropy of the partition $\mathcal{P}$ comprised of m classes, we have

$$\mathcal{H}(T,S) = \begin{cases} \frac{1}{m} \cdot \sum_{j=0}^{m-1} \mathcal{H}(T, S^{c_j}), \ if\ \exists j\ |S^{c_j}| > 0, \\ H(T,S), \ if\ \forall j\ |S^{c_j}| = 0. \end{cases}$$

Definition 4 *The Class Information Density $\mathcal{D}(T,S)$ of the partition $\mathcal{P}$ is given by*

$$\mathcal{D}(T,S) = \begin{cases} \mathcal{H}(T,S)/\log(k+1), \ if\ k > 0, \\ 0, \ if\ k = 0. \end{cases} \quad (4)$$

Property 5 *For a single class of labels, $m = 1$, the measures of the entropy/information density and the class entropy/class information density are equal: $H(T,S) = \mathcal{H}(T,S)$ and $D(T,S) = \mathcal{D}(T,S)$.*

The measures of entropy/information density and class entropy/class information density can be classified as impurity measures: they are minimum if $\exists i$ such that $\delta_i = 1$; they are maximum if $\forall i$: $i = 0, \ldots, k$, $\delta = 1/(k+1)$; they are symmetric with respect to components of δ; and they are differentiable everywhere in their ranges.

3.3 Conditional Information Density and Information Density Gain

For the given set of values S with the associated set of class labels C, an arbitrary cut point T^{cut} splits the partition $\mathcal{P}$ into two partitions $\mathcal{P}_1$ and $\mathcal{P}_2$. It has been proved in [3] that when searching for the best binary split by choosing a single cut point, we can restrict our attention to boundary points. Thus, the cut point T^{cut} preserves the following:

$$\begin{cases} \mathcal{P}_1^{min} = T^{min}, \\ \mathcal{P}_1^{max} = \mathcal{P}_2^{min} = T^{cut}, \\ \mathcal{P}_2^{max} = T^{max}. \end{cases}$$

Definition 5 *The Conditional Information Density $\mathcal{D}(T, S|T^{cut})$ of the partition $\mathcal{P}$ given the cut point T^{cut} is defined by*

$$\mathcal{D}(T, S|T^{cut}) = p_1 \cdot \mathcal{D}(T_1, S_1) + p_2 \cdot \mathcal{D}(T_2, S_2), \quad (5)$$

where $\mathcal{D}(T,S)$ is the class information density (see Equation (4)), p_1 and p_2 are the following probabilities:

$$p_1 = (T^{cut} - T^{min})/\Delta T, \ p_2 = (T^{max} - T^{cut})/\Delta T.$$

Definition 6 *The Information Density Gain* $\mathcal{I}(T, S; T^{cut})$ *as the measure of "goodness" of the partitioning process is given by*

$$\mathcal{I}(T, S; T^{cut}) = \mathcal{D}(T, S|T^{cut}) - \mathcal{D}(T, S). \quad (6)$$

Property 6 *If the discretization process leads to increasing the information density of the formed partitions* $\mathcal{P}_1$ *and* $\mathcal{P}_2$*, then information density gain* $\mathcal{I}(T, S; T^{cut}) > 0$*. Otherwise,* $\mathcal{I}(T, S; T^{cut})$ *is not positive, and the cut point* T^{cut} *is worthless.*

Property 7 *The information density gain* $\mathcal{I}(T, S; T^{cut})$ *is scale and shift invariant.*

3.4 Optimization Criterion

Here we introduce a criterion based on the previously defined information density measures to control the number of intervals produced over the continuous space. The cut points are evaluated according to the *information density criterion* maximizing the distinction between classes assuming no gaps between intervals, intuitively,

$$T^{cut} = argmax\{\mathcal{D}(T, S|T^{cut}) - \mathcal{D}(T, S)\}. \quad (7)$$

Obviously, the lesser the interval, the lesser the density of values.

Example 1 *Let us consider an example of the discretization process for a continuous variable* x *specified by the set of arbitrary values* $S = \{0.022, 0.376, 0.443, 0.519, 0.598, 0.704, 0.837, 0.841, 0.899, 0.953, 0.954\}$ *and the range* $T = [0, 1]$*. Figure 1 illustrates two steps of the unsupervised discretization. Shown on the left are the distributions of the information density gain* $\mathcal{I}(T, S; T^{cut})$ *for different cut points. Depicted on the right are the final cut points* $T^{cut} = \{0.080, 0.807\}$ *according to the optimization criterion (Equation (7)). Figure 2 illustrates several steps of the supervised discretization for the continuous variable* x *specified by the set of values and class labels* $S = \{0.022^0, 0.376^0, 0.443^0, 0.519^1, 0.598^0, 0.704^1, 0.837^1, 0.841^2, 0.899^2, 0.953^2, 0.954^2\}$ *and the range* $T = [0, 1]$*. Eight cut points are detected* $T^{cut} = \{0.097, 0.512, 0.559, 0.607, 0.833, 0.840, 0.842, 0.938\}$*. The result of the discretization process is the set of partitions and corresponding values for their discrete representation (the radix of the multiple-valued variable* χ *is* $r = 2$ *for the unsupervised discretization and* $r = 8$ *for the supervised discretization).*

4 Discretization Algorithm and Experimental Results

Our discretization algorithm works in a greedy top-down manner: it starts with the single partition $\mathcal{P}$ and splits it recursively. A sketch of the algorithm for the continuous variable x is shown below:

Step1. Sort all given values from the set $S = \{s_1, \ldots, s_k\}$ in ascending order. Initialize the first partition: interval boundaries T^{min}, T^{max}; and the set of class labels C.

Step 2. For the current partition, generate a set of possible cut points with a kernel $\Delta T/100$. Calculate the class information density: $\mathcal{D}(T, S) = \mathcal{H}(T, S)/\log(k + 1)$.

Step 3. Form two partitions $\mathcal{P}_1$ and $\mathcal{P}_2$. Calculate the resulting conditional information density $\mathcal{D}(T, S|T^{cut}) = p_1 \cdot \mathcal{D}(T_1, S_1) + p_2 \cdot \mathcal{D}(T_2, S_2)$.

Step 4. Find a potential cut point according to the optimization criterion: $T^{cut} = argmax\{\mathcal{D}(T, S|T^{cut}) - \mathcal{D}(T, S)\}$.

Step 5. If the resulting conditional information density of two partitions $\mathcal{P}_1$ and $\mathcal{P}_2$ given the potential cut point T^{cut} is greater than the information density of the initial partition $\mathcal{P}$ (see Property 6), then accept the cut point T^{cut}. Otherwise, terminate the recursion.

Step 6. Execute Steps 2-5 for the partitions $\mathcal{P}_1$ and $\mathcal{P}_2$ recursively.

Step 7. Sort all cut points, assign discrete values to formed partitions.

The algorithm is coded in Java and the experimental results are obtained on a Pentium III workstation with 256Mb of memory (the same hardware settings are used for other experiments reported in this paper). The main purpose of our experimental study is to compare different outcomes of the unsupervised and supervised discretization. The experiments are not intended to estimate learning criteria or any restrictions for continuous data. The question of applying certain restrictions of multiple-valued logic to assign values to variables is beyond the scope of this paper (it can be resolved via algebraic rules to refer to suitable systems of multiple-valued logic).

The experiments have been carried out on *Machine Learning* benchmarks[2] each of which has at least one continuous variable. A fragment of our results is shown in Table 1, where r denotes the maximal radix of input variables, m denotes the radix of a single-output function, [continuous] indicates the number of continuous variables, [discrete] indicates the number of discrete variables. The columns labeled [Av. Radix] list the average radix of continuous variables after discretization. The columns labeled [Error (%)] give the error rate of discretization algorithms. It is calculated as the ratio of the number of distinguished combinations to the total number of combinations.

In the experimental study, the unsupervised and supervised discretization approaches are compared based on the information density criterion discussed in Section 2. Not surprisingly, the experiments reveal that supervised discretization produces more accurate results than unsupervised discretization. The average radix of continuous variables is bigger for the supervised algorithm. For many benchmarks, the supervised discretization algorithm outperforms the algorithm based on information gain measures (the column [Inf. Gain]) reported in [9]. Additionally, we supply the output of the supervised discretization

[2] http://www.ics.uci.edu/mlearn/MLRepository.html

Table 1. Results on the UCI machine learning benchmark set (continuous variables)

			Variables		Unsupervised		Supervised		Inf. Gain ♭		FCM ♯
Dataset	r	m	continuous	discrete	Av. Radix	Error (%)	Av. Radix	Error (%)	Av. Radix	Error (%)	
adult	41	2	6	11	3.4	2.843	4.8	2.572	4.5	2.238	-
allbp	6	2	7	22	2.5	10.250	5.1	6.560	4.7	8.121	-
auto-mpg	14	5	5	2	4.0	4.117	7.0	1.005	5.5	1.899	121
bupa	7	2	6	0	5.3	8.023	6.6	6.087	4.7	11.950	-
clean1	476	2	166	2	5.3	0.935	6.9	0.210	7.3	4.741	-
crx	14	2	6	9	3.8	4.822	5.0	2.174	4.4	3.338	-
dis	6	2	7	22	2.9	12.540	5.1	6.560	4.0	10.502	-
glass	7	7	9	0	3.1	15.567	5.9	10.748	3.6	22.747	62
ionosphere	7	2	34	0	2.6	4.643	4.7	1.139	4.9	1.250	89
iris	7	3	4	0	4.1	5.007	6.0	4.667	6.5	3.667	8
letter	7	26	16	0	4.8	1.782	6.4	0.025	7.2	4.093	-
pima-indians-diabetes	7	2	8	0	4.0	15.163	5.3	6.771	6.0	1.903	-
post-operative	3	3	1	7	2.0	8.801	3.0	7.78	3.0	7.780	38
sick	7	2	7	22	3.9	5.186	5.3	4.546	4.3	9.153	-
spambase	7	2	57	0	2.1	2.257	3.6	1.304	4.8	2.020	-
waveform	7	3	21	0	5.5	4.513	7.0	1.660	6.1	3.677	-
wdbc	7	2	30	0	3.1	2.962	4.53	1.757	4.8	1.535	-
yeast	7	10	8	0	3.7	5.320	5.3	5.054	5.0	10.011	30

algorithm to the program which creates multiple-valued decision diagrams [7]. The obtained results (the number of nodes) are given in the column [FCM].

5 Conclusion

This paper investigated the use of information measures to resolve the problem, known as discretization, of finding radices for continuous variables in a multiple-valued domain. We introduced new information density measures and formulated their properties. An efficient discretization algorithm is developed which utilizes the information density criterion. We demonstrated the ability of the algorithm to deal with unsupervised and supervised discretization. It is worth noting that the suggested discretization algorithm is robust, unaffected by data shift and scaling.

Acknowledgements

Insightful comments from an anonymous reviewer were helpful during revision of this paper.

References

[1] I. Baturone, S. Sanchez-Solano, and J. Huertas. A self-checking current-mode analogue memory. *Electronics Letters*, 33(16):1349–1350, 1997.

[2] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Proc. Int. Conference on Machine Learning*, pages 194–202, 1995.

[3] U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. Int. Conference on Artificial Intelligence*, pages 1022–1027, 1993.

[4] P. Heim and M. Jabri. Long-term CMOS static storage cell performing AD/DC conversion for analog neural network implementations. *Electronics Letters*, 30(25):2124–2125, 1994.

[5] M. Koga, A. Uchiyama, and M. Sampei. A transformation of discrete-time systems into continuous-time systems using the general solution of a matrix equation. *A Publication of Elect., Infor. and Systems Society*, 119-C(12):1561, 1999.

[6] E. Pierzchala, M. Perkowski, and S. Grygiel. A filed programmable analog array for continuous, fuzzy, and multi-valued logic applications. In *Proc. IEEE Int. Symposium on Multiple-Valued Logic*, pages 148–155, 1994.

[7] D. Popel and R. Drechsler. Efficient minimization of multiple-valued decision diagrams for incompletely specified functions. In *Proc. IEEE Int. Symposium on Multiple-Valued Logic*, 2003.

[8] D. Popel and N. Hakeem. Improving web database access using decision diagrams. In *Proc. IEEE Int. Conference on Computer Systems and Applications*, pages 519 – 525, 2001.

[9] J. Quinlan. Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.

[10] T. Ross, J. Goldman, D. Gadd, M. Noviskey, and M. Axtell. The decomposition of real-valued functions. In *Proc. Int. Workshop on Post-Binary Ultra-Large Scale Integration*, pages 186–193, 1994.

[11] A. Saed, M. Ahmadi, and G. Jullien. Arithmetic circuits for analog digits. In *Proc. IEEE Int. Symposium on Multiple-Valued Logic*, pages 186–193, 1999.

[12] C. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423, 623–656, 1948.

[13] D. Simovici and S. Jaroszewicz. An axiomatization of generalized entropy of partitions. In *Proc. IEEE Int. Symposium on Multiple-Valued Logic*, pages 259–264, 2001.

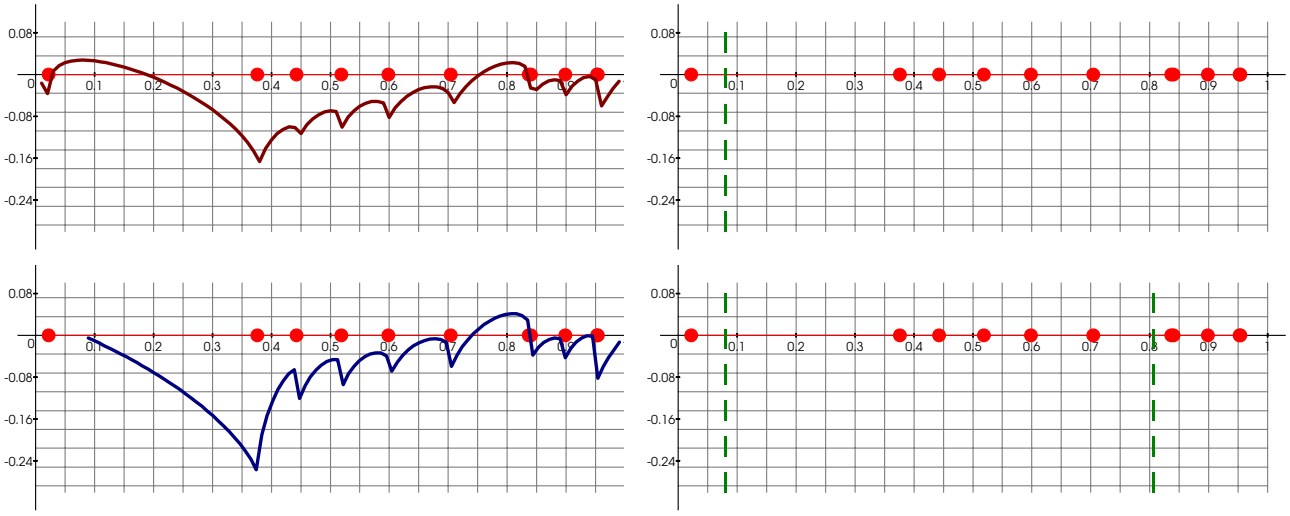

Figure 1. Unsupervised discretization for the set of values given in Example 1

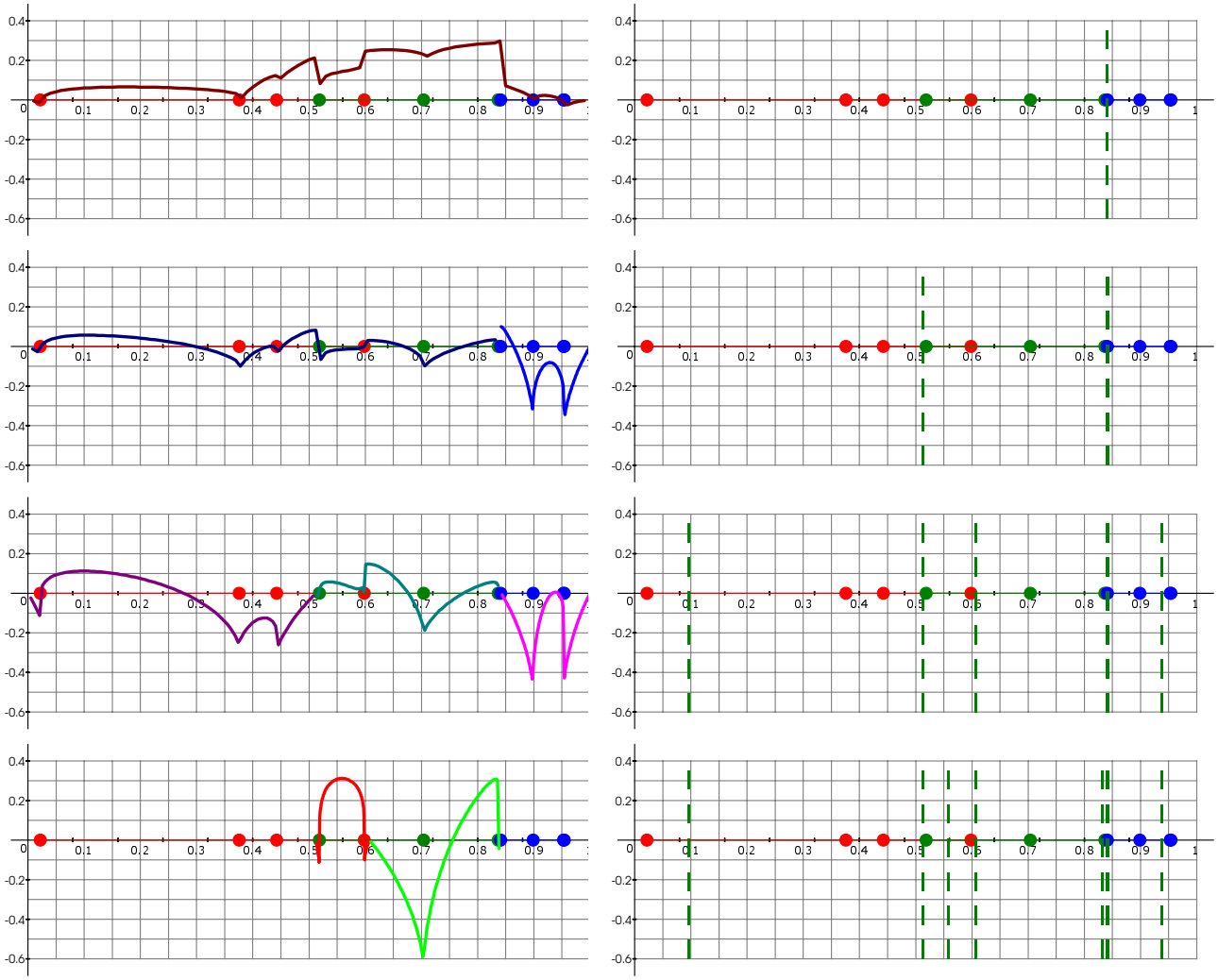

Figure 2. Supervised discretization for the set of values and class labels given in Example 1