Supervised non-parametric discretization based on Kernel density estimation

Jose Luis Flores^{a,c,*}, Borja Calvo^a, Aritz Perez^b

^a*Intelligent Systems Group (ISG), Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, Manuel de Lardizabal, Donostia/San Sebastián 20018, Spain*

^b*Basque Center of Applied Mathematics (BCAM), Mazarredo Zumarkalea, Bilbo 48009, Spain*

^c*IK4-Ikerlan Technology Research Centre, Dependable Embedded Systems Area, 20500 Gipuzkoa, Spain*

Abstract

Nowadays, machine learning algorithms can be found in many applications where the classifiers play a key role. In this context, discretizing continuous attributes is a common step previous to classification tasks, the main goal being to retain as much discriminative information as possible.

In this paper, we propose a supervised univariate non-parametric discretization algorithm which allows the use of a given supervised score criterion for selecting the best cut points. The candidate cut points are evaluated by computing the selected score value using kernel density estimation. The computational complexity of the proposed procedure is $O(N \log N)$, where N is the length of the data.

Our proposed algorithm generates a low complexity in discretization policies while retaining the discriminative information of the original continuous variables. In order to assess the validity of the proposed method, a set of real and artificial datasets has been used and the results show that the algorithm provides competitive results in terms of performance, a low complexity in the discretization policies and a high performance.

Keywords: Discretization, Supervised, Non-Parametric, Kernel density

Nowadays, data mining has become one of the most important paradigms to extract relevant information from real datasets in which the preprocess of the data plays a crucial role. In fact, data preprocessing is essential in order to guarantee the success of the process. Among all the preprocessing methods used, discretization (i.e., the transformation of numeric variables into categorical) is one of the most broadly used. In this paper, we propose a supervised discretization algorithm that aims to optimize a supervised performance metric, such as the area under the ROC curve (AUC). In such a context, the discretization must be performed retaining as much discriminative information as possible in a computationally efficient way.

We can highlight three families of supervised dis-

cretization algorithms, taking into account the criteria used to merge or split intervals. The first family is based on the χ^2 statistic and it is guided by a null hypothesis testing criteria to decide whether to split or merge discretized intervals (Kerber, 1992; Tay and Shen, 2002; Su and Hsu, 2005; Abril et al., 2009). The second family is based on information theory where the goal is to find the cut-point that minimizes the entropy of the class variable within an interval (Fayyad and Irani, 1992; Li and Wang, 2002). The third family of algorithms is based on a contingency matrix which contains the statistics per interval of the class variable. The most important representative measures based on quanta matrices are Class-Attribute Interdependence Maximization (CAIM) (Kurgan, 2004) and Class-Attribute Contingency Coefficient (CACC) (Tsai et al., 2008). For a detailed review on discretization algorithms we refer the reader to (Garcia et al.,

*Corresponding author: Tel.: +34-943015034; fax: +34-943015090;
Email address: joseluis.flores@ehu.eus (Jose Luis Flores)

2013).

In this work, we propose a supervised univariate discretization algorithm which allows the use of supervised score criteria for selecting the best cut points. These are obtained by computing the selected score with kernel density estimation (Rosenblatt, 1956; Parzen, 1962). The algorithm estimates the class conditional densities and selects the most promising cut-points according to the densities without requiring any splitting/merging incremental procedure. Using different values for the smoothing parameter for the kernel, the algorithm produces different sets of candidate cut-points, and the best set is selected according to the provided supervised classification performance measure. This flexibility allows the user to obtain the best classification performance without the limitations of using an invariant single measure. In order to test our approach, we have performed an extensive experimentation that includes all the previously cited approaches with the proposed algorithm. The analysis includes a comparison in terms of the AUC and the complexity of the generated discretization policies.

The remainder of the paper is organized as follows: Section 1 introduces the proposed supervised discretization algorithm, in Section 3 we perform an extensive experimentation, and Section 4 summarizes the main contributions of this work and briefly indicates the main future work lines.

1. Discretization based on Kernel

Our discretization algorithm is divided into two parts: i) a criteria for obtaining the discretization policy, i.e., a set of cut-points, given a value of the smoothing parameter for the Kernel, and ii) the selection of the smoothing parameter by optimizing a supervised performance measure.

1.1. Selecting cut-points based on Kernel

In this section, we describe how to determine a set of cut-points, i.e., discretization policy, by using K2 for estimating the class conditional densities of a continuous random variable.

Let us assume that we have a dataset of N instances for the random variables (X, C) i.i.d., according to $\rho(X, C)$, $D = \{(x_1, c_1), \dots, (x_N, c_N)\}$, where X is a continuous random variable, $x \in \Omega$ represents a particular realization

of it, and C is a binary class variable that takes values $c \in \{-, +\}$. In this work we explain the proposed algorithm for binary classification, for the sake of simplicity. In the last paragraph of this subsection we briefly indicate how to extend it to the multi-class scenario. The dataset D can be divided into two subsets according to the class: $D^+ = \{x : (x, +) \in D\} = \{x_1^+, \dots, x_{N^+}^+\}$ and $D^- = \{x : (x, -) \in D\} = \{x_1^-, \dots, x_{N^-}^-\}$, where D^+ contains the N^+ values of x in D that correspond to the class $+$ and D^- contains the N^- values of x that correspond to the class $-$.

With a slight abuse in the notation, let us assume that we have (X, C) distributed according to $\rho(X, C)$, where $X = (X^1, \dots, X^n)$ is an n -dimensional continuous random variable. The Bayes classifier¹ for (X, C) corresponds to $\arg \max_c \rho(x, c)$. In this paper, we present a univariate supervised discretization procedure which implicitly assumes that the unidimensional features are conditionally independent given the class value. Under this conditional independence assumption the Bayes classifier reduces to the naïve Bayes classifier, which is given by $\arg_c \max p(c) \prod f(x^i|c)$. Following this intuition, we propose a discretization algorithm for X^i based on $f(x^i|c)$ rather on $\rho(x^i, c) = p(c) \cdot f(x^i|c)$, for $i = 1, \dots, n$.

Intuitively, we propose to separate regions of Ω_X for which $+$ = $\arg_c \max f(x|c)$ (positive regions) and $-$ = $\arg_c \max f(x|c)$ (negative regions). We define the *candidates* as the set points that satisfy the following equality:

$$\{x \in \mathbb{R} | f(x|C = +) = f(x|C = -)\} \quad (1)$$

where C is a binary class variable that takes values in $\{-, +\}$, and $f(x|C = +)$ and $f(x|C = -)$ are the densities conditioned to the values $+$ and $-$ of the class, respectively. The *cut-points* are a (minimal) subset of the candidates that separate two consecutive positive and negative regions.

One of the most popular non-parametric techniques for estimating probability densities from data is the kernel density estimation (K2) (Silverman, 1986). K2 is defined as:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h} K\left(\frac{x - X_i}{h}\right) \quad (2)$$

¹ the classifier with the minimum classification error.

where K is a kernel function, and h is the *smoothing parameter*, also known as window width or bandwidth. Under mild conditions regarding the smoothing parameter, K2 asymptotically converges to the true distribution (Silverman, 1986; Wand and Jones, 1995). In order to obtain the cut-points, we propose to estimate the conditional densities using K2, and then solve the equality shown in Eq. 1 and select a subset of cut-points.

Based on Eq. 2, the candidates can be obtained by solving the following equality:

$$\frac{1}{N^+} \sum_{x_i^+ \in X^+} K\left(\frac{x - x_i^+}{h^+}\right) - \frac{1}{N^-} \sum_{x_i^- \in X^-} K\left(\frac{x - x_i^-}{h^-}\right) = 0 \quad (3)$$

where h^+ and h^- are the smoothing parameters for the estimated densities conditioned to the class values $+$ and $-$, respectively.

In order to solve Eq. 3 efficiently and in a closed form, we propose to use the Epanechnikov kernel:

$$K(u) = \begin{cases} \frac{3}{4}(1 - u^2), & \text{if } |u| \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

In addition to its interesting asymptotic properties of convergence (Wand and Jones, 1995), we have selected this kernel function mainly because of two interesting properties: i) Epanechnikov kernel can be evaluated locally. That is, in order to determine the value of the estimated density at some point x , $\hat{f}(x)$, we only need to evaluate the kernel function in the subset of points **that fall in** the interval $[x-h, x+h]$, and ii) since K2 using Epanechnikov kernels is an averaging of second order bounded polynomials, it is a second order polynomial defined by intervals. Therefore, by using Epanechnikov, kernel we can find the cut-points in a closed form by solving, at each interval, a second order polynomial whose coefficients determined with subsets of points of D^+ and D^- .

Given the smoothing parameters h^+ and h^- , the procedure for determining the cut-points consists of i) sorting the points in $D^+ \cup D^-$, ii) determining the coefficients of the second order equation defined at each interval, and iii) solving the second order equations of each interval. The computational complexity of step i) is $O(N \log N)$, of step ii) is $O(N)$ and of step iii) is $O(N)$, which leads to a computational complexity of the procedure for determining the cut points of $O(N \log N)$. We would like

to highlight that it is possible to determine the cut points for $O(\log N)$ smoothing parameter values with a computational complexity of $O(N \log N)$ because, once the points are sorted the procedure has a computational complexity of $O(N)$.

In order to extend the procedure from the binary to the multi-class case, given h^c para $c = 1 \dots r$ the cut-points can be determined by following a similar procedure: i) sort the points, ii) determine the coefficients of the second order equation for each interval iii) compute the cut points between the densities $f(x|c)$ conditioned to every pair of classes, iv) select the subset of cut-points $\{x^*\}$ associated to the pair of densities $f(x^*|c_1)$ and $f(x^*|c_2)$ for which the density value is maximum, $c_1, c_2 = \arg_{c \in \{1, \dots, r\}} \max f(x^*|c)$.

2. Selecting the smoothing parameters

The value of smoothing parameter h in K2 is usually selected taking into account the number of available instances, N . For instance, the asymptotic convergence property of K2 requires that the smoothing parameter h and the ratio h/N tend to zero as the number of points N grows to infinity (Silverman, 1986; Wand and Jones, 1995). Fortunately, Pérez et al. (2009) point out that the classifiers based on K2 tends to be insensitive to small changes in the smoothing parameter, and the proposed supervised discretization procedure inherits this robustness.

In order to further reduce the computational cost of the selection of the smoothing parameters h^+ and h^- we set them in relation to a single smoothing parameter h . In particular, we set them such that:

$$h \cdot N = h^+ \cdot N^+ = N^- \cdot h^- \quad (5)$$

Under these constraints, we ensure that the bumps placed at each point of both classes have the same height, while they vary in their width in terms of the proportions of instances of each class.

In order to guide the selection of the smoothing parameters h , we propose to optimize a given supervised performance measure that can be computed with a computational complexity of $O(N)$, e.g., classification error or AUC. To this aim, any optimization heuristic can be used. In this work, we perform a grid search, for the sake of simplicity. First, we determine the interval $[h_{min}, h_{max}]$ for the

grid search, where $h_{min} = \min_{x_i, x_j \in D, x_i \neq x_j} |x_i - x_j|$ and $h_{max} = \max_{x_i, x_j \in D, x_i \neq x_j} |x_i - x_j|$. We obtain a set H of $\log(N)$ equally distributed points in the interval $[h_{min}, h_{max}]$. Then, for each h in H the cut-points of interest are obtained following the procedure described in Section 1.1, which has a total computational complexity of $O(N \log N)$. Next, the performance measure is estimated for the discretization policies determined by each $h \in H$, s_h . In order to obtain the score s_h , we make use of an error estimation procedure with low variance (see (Rodríguez et al., 2013) for further details), e.g., repeated 10-fold cross validation which has a computational complexity of $O(200HI)$ being I the number of intervals and H the grid size. Finally, the discretization algorithm selects the smoothing parameter that maximizes² the estimated performance measure, $h^* = \arg \max_{h \in H} s_h$, and returns the cut-points determined by h^* using all the available data. In summary, the computational complexity of the discretization procedure is $O(N \log N)$.

3. Experimentation

The objective of the experiments is to analyze the behaviour of the proposed algorithm and compare it with the state-of-the-art methods. The analysis has been carried out in terms of i) the AUC obtained by different classifiers using the discretized features, and ii) the number of intervals in which the features are discretized. The classifiers considered in the experimentation are based on Bayesian networks: naïve Bayes (NB, Maron (1961)), tree augmented naïve Bayes (TAN, Friedman et al. (1997)), and k -dependence Bayesian classifier (k DB, Sahami (1996)). The set of representative state-of-the-art discretization algorithms (Garcia et al., 2013) considered in the experiments can be found in Table 3. The parameter values of the discretization algorithms considered have been taken from (Garcia et al., 2013).

We begin the experiments by illustrating the benefits of using the proposed discretization algorithm compared to the state-of-the-art algorithms using two artificial data

²Depending on the nature of the score function we have to find for the h value that maximizes or minimizes the score.

Table 1: Discretization algorithms

ID	Name	Complete name
0	NODISC	No discretization
1	Ameva	Ameva
2	CACC	Class-Attribute Contingency Coefficient
3	CAIM	Class-Attribute Interdependency Maximization
4	Chi2	Chi Square ($\alpha = 0.5, \delta = 0.05$)
5	ChiMerge	ChiSquare Merge ($\alpha = 0.05$)
6	EF	Equal Frequency ($\log_2(n) + 1$)
7	EW	Equal Width ($\log_2(n) + 1$)
8	ExtendChi2	Extended Chi Square ($\alpha = 0.5$)
9	K2	Kernel based discretization
10	MDLP	Minimum description length (N/R)
11	MODCHI2	Modified Chi Square ($\alpha = 0.5$)

sets. Next, we compare the behaviour of the proposed discretization algorithm using 25 data sets from the UCI repository. In order to analyze the robustness of the discretization algorithms we have simulated classification problems with noisy labels (Frénay and Verleysen, 2014) using a subset of 12 datasets.

3.1. Artificial datasets

The (one-dimensional) artificial datasets of this section are used to illustrate the importance of considering the class information when discretizing a continuous variable for dealing with supervised classification problem.

The first dataset is formed by several groups of instances regularly separated by empty intervals. Each group of instances contains instances of both classes that can be completely separated by means of a single cut point (see Fig. 1). The AUC has been estimated using stratified 10-fold cross-validation.

$$X^+ = \{ \overbrace{1..1}^{50}, \overbrace{4..4}^{50}, \overbrace{7..7}^{50}, \overbrace{10..10}^{50}, \overbrace{13..13}^{50}, \overbrace{16..16}^{50}, \overbrace{19..19}^{50}, \overbrace{22..22}^{50}, \overbrace{25..25}^{50}, \overbrace{28..28}^{50} \}$$

$$X^- = \{ \overbrace{2..2}^{50}, \overbrace{5..5}^{50}, \overbrace{8..8}^{50}, \overbrace{11..11}^{50}, \overbrace{14..14}^{50}, \overbrace{17..17}^{50}, \overbrace{20..20}^{50}, \overbrace{23..23}^{50}, \overbrace{26..26}^{50}, \overbrace{29..29}^{50} \}$$

Figure 1: First artificial data set

The results with the first dataset are summarized in Fig. 2. The experiment shows that K2 is the discretization algorithm that make the most effective use of the class information and generates the optimal discretization policy.

Figure 2: The estimated AUC of NB classifier using the discretized variable using different discretization algorithms.

The second artificial dataset illustrates the benefit of using the class information for generating a discretized variable with an appropriate number of intervals. This data set is formed by sets of points falling at intervals regularly distributed where the points falling in the interval $[0, 15]$ belong to class + while points falling in the interval $[15, 30]$ belong to class - (see Fig. 3).

$$X^+ = \{ \overset{50}{1..1}, \overset{50}{2..2}, \overset{50}{4..4}, \overset{50}{5..5}, \overset{50}{7..7}, \overset{50}{8..8}, \overset{50}{10..10}, \overset{50}{11..11}, \overset{50}{13..13}, \overset{50}{14..14}, \}$$

$$X^- = \{ \overset{50}{16..16}, \overset{50}{17..17}, \overset{50}{19..19}, \overset{50}{20..20}, \overset{50}{22..22}, \overset{50}{23..23}, \overset{50}{20..20}, \overset{50}{25..25}, \overset{50}{26..26}, \overset{50}{28..28}, \overset{50}{29..29}, \}$$

Figure 3: Artificial Dataset 02

The summary of the results obtained with the second artificial data set are shown in Fig. 4. K2 and Chi2 found an optimal discretization policy. This experiment suggests that both algorithm have a good self-regulatory behavior that, by using class information, allows to control the number of cut-points required for discretizing continuous random variables.

Figure 4: The number of cut points used to discretize the variable of the second artificial data set. The optimal discretization policies requires a single cut point.

3.2. Real datasets

The objective of the next experiments is to evaluate the performance and the complexity of the proposed discretization algorithm using real datasets. The datasets have been taken from the UCI repository and Table 3.2 summarizes their main features. We have performed three experiments: i) Discretize the datasets with the set of algorithms in Table 3 and estimate the AUC for NB, ii) analyze the average number of cut points generated by the discretization algorithms for each data set, and iii) analyze the robustness of the discretization algorithm by simulating classification problems with noisy labels.

3.2.1. Supervised classification

Table 3 summarizes the AUC estimated with NB in the selected datasets (see Table 3.2), where the black colour represents the best value of the row, that is, the best value of the corresponding dataset. The AUC has been estimated using the stratified 10-fold cross-validation.

Analyzing the results K2 of the Bayesian is better than discretization methods: Ameva, CACC, CAIM, Chi2, Chi2Merge, extendChi2, and ModChi2, it is even better than classifying without discretizing data. Equal Width shows competitive results but it is not good enough because the differences are much higher when K2 is better than Equal Width compared to the inverse case. On the other hand, Equal Frequency and MDLP are the most competitive algorithms showing a high probability of being equivalent under these conditions, this is related with the nature of K2 which merges the properties of Equal Frequency and MDLP.

Table 2: Datasets from UCI

ID	Description	Categorical	Numerical	Instances
01	Horse Colic	15	7	368
02	Credit Approval	8	6	690
03	German Credit	17	3	1000
04	Pima Indians Diabetes Database	0	8	768
05	Haberman's Survival Data	1	2	306
06	Ionosphere	0	32	351
07	BUPA Liver disorders	0	6	345
08	Sonar (Mines and Rocks)	0	60	208
09	SPECT Heart	0	44	349
10	Banknote authentication	0	4	1372
11	Blood transfusion	0	4	748
12	Climate simulation crashes	0	19	540
13	Planning relax	0	12	182
14	Appendicitis	0	7	106
15	SA Heart	0	9	462
16	Musk1	3	163	476
17	Parkinsons	1	22	195
18	Badges	4	7	294
19	Glass2	0	9	163
20	Indian Liver Patient	0	9	583
21	Vertebral Column	0	6	310
22	Mamographic Mass	1	4	961
23	Cylinder Bands	21	18	459
24	Heart Disease Hungarian	0	13	294
25	Leukemia Haslinger	0	50	100

In addition to this, we have performed a statistical assessment of the results for all discretization algorithms, but we have selected only the most competitive algorithms, that is, Equal Frequency, MDLP but also the results compared to the dataset without discretizing and we only show only the results for these, the rest of the results have been provided as supplementary material (Flores (2019)).

To this aim, we have used the Bayesian approach presented in Benavoli et al. (2017), as it provides more useful information than the classical statistical test methodology. In particular, as the results obtained by some algorithms can hardly be considered Gaussian, we have used the Bayesian alternative to the Wilcoxon test. The rest of the analysis are available at Flores (2019).

The methodology proposed in Benavoli et al. (2017) requires defining the concept of ‘region of practical equivalence’, the rope, which is a range of differences between algorithms that are considered irrelevant. In the paper, the authors propose using a value of 0.01 and so we have considered this value in our experimentation. In other words, we consider that if the difference in AUC between two algorithms is smaller than 0.01 we can consider them as equivalent.

ID	NODISC	Ameva	CACC	CAIM	Chi2	Chi2Merge	EF	EW	ExtendChi2	K2	MDLP	ModChi2
1	0.84	0.83	0.81	0.85	0.84	0.85	0.84	0.84	0.85	0.87	0.85	0.85
2	0.90	0.84	0.83	0.84	0.89	0.82	0.82	0.82	0.81	0.91	0.92	0.91
3	0.79	0.76	0.72	0.75	0.77	0.76	0.79	0.79	0.77	0.82	0.78	0.77
4	0.82	0.45	0.50	0.27	0.58	0.81	0.83	0.83	0.56	0.82	0.81	0.54
5	0.85	0.81	0.58	0.51	0.68	0.84	0.88	0.88	0.68	0.74	0.65	0.68
6	0.94	0.07	0.08	0.09	0.60	0.95	0.94	0.93	0.60	0.93	0.95	0.60
7	0.84	0.42	0.47	0.44	NA	NA	0.71	0.67	0.45	0.69	0.52	0.46
8	0.80	0.17	0.27	0.14	NA	NA	0.86	0.86	0.50	0.84	0.87	0.50
9	0.87	0.72	0.54	0.75	NA	NA	0.81	0.80	0.50	0.92	0.89	0.50
10	0.84	0.10	0.12	0.07	0.75	0.95	0.86	0.87	0.79	0.96	0.92	0.82
11	0.72	0.89	0.47	0.32	0.56	0.69	0.70	0.69	0.56	0.69	0.69	0.56
12	0.87	0.79	0.72	0.45	0.51	0.66	0.78	0.77	0.55	0.82	0.67	0.55
13	0.41	0.54	0.48	0.49	0.50	0.45	0.88	0.40	0.50	0.32	0.50	0.50
14	0.82	0.19	0.20	0.18	0.50	0.85	0.87	0.87	0.50	0.85	0.78	0.50
15	0.75	0.59	0.83	0.48	0.59	0.66	0.74	0.73	0.71	0.74	0.72	0.71
16	0.85	0.20	0.14	0.22	0.59	0.93	0.87	0.91	0.62	0.92	0.93	0.67
17	0.86	0.12	0.12	0.13	0.40	0.91	0.80	0.91	0.51	0.91	0.88	0.50
18	1.00	1.00	1.00	1.00	1.00	NA	1.00	1.00	1.00	1.00	1.00	1.00
19	0.76	0.21	0.15	0.23	NA	NA	0.87	0.82	0.67	0.85	0.86	0.69
20	0.73	0.26	0.28	0.28	NA	NA	0.74	0.70	0.68	0.75	0.73	0.69
21	0.88	0.20	0.20	0.19	0.75	0.80	0.86	0.82	0.79	0.84	0.84	0.72
22	0.90	0.88	0.88	0.88	0.89	0.90	0.90	0.90	0.89	0.89	0.90	0.89
23	0.79	0.71	0.57	0.76	NA	NA	0.83	0.80	0.81	0.82	0.80	0.81
24	0.89	0.24	0.25	0.15	NA	NA	0.90	0.90	0.79	0.83	0.89	0.78
25	0.81	0.06	0.07	0.06	0.51	0.82	0.83	0.82	0.50	0.91	0.93	0.50

Table 3: Estimated AUC of NB using the discretized variables in 25 datasets from the UCI repository.

In order to carry out the analysis, first, the average AUC for each algorithm and each dataset was computed. Then, our proposal (K2) was compared with all the alternatives. For every pair, the Bayesian model was used to obtain a sample of the posterior distribution of the probability of win/tie/lose (Benavoli et al. (2017)).

Figure 5 shows the posterior samples projected in Simplex plots. Briefly, the higher the probability of one of the three alternatives, the closer the points are to its corresponding vertex. If we represent the probabilities as triplets $[P(K2 \text{ win}), P(K2 \text{ tie}), P(K2 \text{ lose})]$, the vertex labeled as ‘K2’ corresponds to the point $[1, 0, 0]$ and the vertex labeled as ‘Rope’ corresponds to the point $[0, 1, 0]$; the central point corresponds to $[0.33, 0.33, 0.33]$.

Note that the dashed lines divide the area into regions where the highest probability is that of the corresponding vertex. For more details of this type of representation, the reader is referred to Benavoli et al. (2017).

Besides the graphical representation, the plots include the number of samples in each region (upper-right part) and the expected posterior probabilities of win/tie/lose (upper-left part). The figures provide us with two key pieces of information. On the one hand, the location of the points tells us whether our approach is better, equal or worse than the other method in the comparison. On the other hand, the spreadness of the points provides and in-

Figure 5: Bayesian analysis using the estimated AUC of the most competitive discretization algorithms: Kernel, EF and MDLP.

dication of the uncertainty about the conclusions we can derive from the results. If all the points are concentrated in a small region, there will be a low uncertainty about the probabilities estimated, while if the points are spread in a large area, we will have a high uncertainty about the estimated posterior probabilities.

Analyzing the results the probability that K2 is better than MDLP is higher than 75%, in the case of taking the decision of not discretizing the probability of being better is higher than 79%. Finally, the results show under normal situations that the probability of being equivalents are higher than saying that K2 is better than EF or EF better than K2.

Figure 6: Complexity vs Accuracy

Based on this figure several conclusions can be inferred:

- K2 is a very competitive alternative and reasonable discretization policies.
- The performance of MDLP is competitive although sometimes it provides the worst results (upper left).

3.2.2. Complexity of the discretization policies

In this section we measure the complexity of the discretization policies generated by discretization algorithms in terms of the number of intervals. A low number intervals is particularly important in domains where the number of instances is small.

Following (Garcia et al., 2013), in the experiment we analyze the accuracy and complexity of the discretization policies generated by EF, EW and MDLP. The complexity has been calculated as the normalized average value of the number of intervals for all attributes and for all datasets. The results are shown in Figure 6 using a scatter plot, where the size of each point represents the AUC, and their position in the horizontal axes is associated with the complexity of the discretization policy.

3.2.3. Noisy labels

In this case, the experiments have involved the use different classifiers: a naive Bayes, a Tree Augmented naive Bayes (Friedman et al. (1997)) and a k-dependence Bayesian classifiers (Sahami (1996)). The results are shown in Table 4 where each row represents a dataset and groups the results for different noise levels (10,20 and 30), and the columns are grouped at first level by the discretization algorithm, and at second level by the classifier. Cells in black represent the best value for the specific combination of discretization algorithm, classifier,

noise rate, and dataset. Empty cells represent situations where the discretization algorithms were not able to find cut points. The table show only the results with a subset of the datasets (01, 02, 04, 05, 07, 10, 11, 15, 18, 20, 21 and 22) and the most competitive algorithms due to space reasons, the rest of the results are in supplementary material (Flores (2019)).

As we can see in Table 4 several conclusions can be inferred:

- Our approach is the discretization approach that provides the best behavior providing the best AUC values while increasing the noise compared to other algorithms.
- Our approach provides competitive results with different classifiers such as the k-DB.
- Equal Frequency and Equal Width lose progressively performance as the noise rate increases.
- K2 is competitive not only under normal situation but in difficult situations with different classifiers. Particularly, it is important to remark that in all cases K2 shows never shows worse performance but better or equal.

4. Conclusions

This paper presents results of experiments in which different techniques and different datasets (real and artificial) were used for discretization. The results show that the proposed discretization algorithm, K2, provides competitive results not only in classification power but also in terms of complexity and high performance in standard supervised data and on the presence of noisy labels. A remarkable feature of the K2 algorithm is that it maintains better the performance even with more complex classifiers when learning from datasets with noisy labels whereas the rest of the most competitive algorithms decreases the performance.

The computational complexity of the algorithm is $O(N \log N)$, that is, the highest cost is sorting the data, whereas if the data are sorted the algorithm is linear.

In the future, we plan to extend the proposed procedure from the univariate case to the multivariate one by using more sophisticated optimization meta-heuristics.

Acknowledgments

Aritz Pérez is supported by the Basque Government through the BERC 2018-2021 program and through ELKARTEK program, and by Spanish Ministry of Economy and Competitiveness MINECO through BCAM Severo Ochoa excellence accreditation SEV-2013-0323 and through project TIN2017-82626-R funded by AEI/FEDER, UE.

Declaration of interest

None

BBDD	Noise	Anova				CACC				CAM				CH2				CIM				EF				EW				extraCH2				K2				MDLP				ModCh2	
		NS	TAN	NB	KOB	CACC	TAN	NB	KOB	CAM	TAN	NB	KOB	CIM	TAN	NB	KOB	EF	TAN	NB	KOB	EW	TAN	NB	KOB	extraCH2	TAN	NB	KOB	K2	TAN	NB	KOB	MDLP	TAN	NB	ModCh2						
01	10%	0.7319	0.7263	0.7425	0.6589	0.6850	0.6920	0.5373	0.8153	0.7125	0.6559	-	0.7653	0.7580	0.7375	0.7748	0.7895	0.8383	0.7784	0.7624	0.8261	-	-	-	-	-	-	0.8001	0.7920	0.7958	-	-	-	-	-	-	-	-					
	20%	0.6953	0.6989	0.7166	0.6790	0.6677	0.6672	0.6814	0.5353	0.6552	-	-	-	0.7400	0.6981	0.7201	0.7607	0.7180	0.7418	0.7677	0.7447	0.7677	-	-	-	-	-	0.7570	0.7322	0.7514	-	-	-	-	-	-	-	-					
	30%	0.6299	0.6299	0.6467	0.6077	0.5987	0.5991	0.6298	0.5151	0.7760	0.6676	0.5683	0.6573	0.7071	0.6492	0.6566	0.7213	0.6454	0.7335	0.7293	0.6797	0.7021	-	-	-	-	-	0.7595	0.6966	0.7179	-	-	-	-	-	-	-	-					
02	10%	0.6500	0.5501	0.5331	0.5584	0.6859	0.5502	0.5260	0.8209	0.6178	0.6676	0.7428	0.7368	0.7830	0.7884	0.8641	0.9145	0.8788	0.8562	0.8968	0.9128	-	-	-	-	-	0.8498	0.8416	0.8312	0.8136	0.8581	0.8108	0.8156	0.8124	0.8801	-	-						
	20%	0.5155	0.5404	0.5713	0.5586	0.6224	0.5607	0.5283	0.8131	0.7757	0.7818	0.7492	0.7429	0.7649	0.7648	0.7974	0.8469	0.7974	0.8478	0.8162	0.8913	0.8108	0.7910	0.8350	0.8421	0.8415	0.8530	0.8441	0.8583	0.8709	-	-	-	-	-	-	-						
	30%	0.4322	0.4553	0.4667	0.4279	0.5359	0.4923	0.4254	0.7370	0.7146	0.6337	0.6317	0.7271	0.6980	0.6583	0.7279	0.6733	0.6873	0.6828	0.7551	0.6506	-	-	-	-	-	0.8372	0.7653	0.7955	-	-	-	-	-	-	-	-						
03	10%	0.5414	0.5814	0.6247	0.3945	0.4492	0.4384	0.3338	0.7275	0.3959	0.4317	0.5835	0.6850	0.6460	0.6762	0.7387	0.6508	0.7201	0.7559	0.7201	0.6980	0.6598	0.5526	0.6588	0.6748	0.7386	0.7595	0.8372	0.7653	0.7955	-	-	-	0.8534	0.8355	0.8548	-						
	20%	0.3545	0.3958	0.4348	0.3949	0.4464	0.5036	0.3492	0.3859	0.5841	0.4152	0.4831	-	0.4541	0.4510	0.6244	0.4774	0.5608	0.5752	0.7356	0.6987	0.7825	-	-	-	0.7766	0.7708	0.7472	-	-	-	-	-	-	-	-							
	30%	0.2732	0.3124	0.4042	0.4166	0.4383	0.5057	0.4402	0.4106	0.2815	0.4503	0.4871	0.5320	0.4910	0.5728	0.5191	0.7099	0.5994	0.7288	0.6811	0.6987	0.7427	-	-	-	0.7154	0.6715	0.7474	-	-	-	-	-	-	-	-							
04	10%	0.2697	0.3172	0.4565	0.4537	0.5677	0.5553	0.2683	0.4348	0.5154	0.4712	0.4885	0.5076	0.4820	0.5226	0.7109	0.7009	0.6451	0.7310	0.7290	0.9269	-	-	-	-	-	0.7422	0.7022	0.8174	-	-	-	-	-	-	-	-						
	20%	0.3731	0.5059	0.5059	0.4584	0.4779	0.5354	0.2726	0.4545	0.4584	0.3503	0.5351	0.6267	0.4607	0.5417	0.7100	0.6505	0.6176	0.6978	0.9003	0.9569	0.9100	0.8971	0.9617	0.9594	0.6933	0.6986	0.7033	-	-	-	-	-	-	-	-	-						
	30%	0.4059	0.4852	0.4852	0.4834	0.4834	0.4840	0.2965	0.4574	0.4543	0.6144	0.5489	0.6470	0.6877	0.6116	0.5077	0.5974	0.4895	0.5307	0.6797	0.8348	0.8546	-	-	-	0.7327	0.6290	0.5550	-	-	-	-	-	-	-	-	-						
05	10%	0.2171	0.3181	0.4037	0.4141	0.5076	0.4971	0.2426	0.5700	0.4522	0.5029	0.4949	-	0.6082	0.5941	0.5993	0.4824	0.5981	0.6506	0.6192	0.6201	-	-	-	-	-	0.6136	0.6116	0.6201	-	-	-	-	-	-	-	-						
	20%	0.4932	0.5026	0.4742	0.5202	0.5119	0.5144	0.4398	0.5378	0.4458	0.5434	0.5395	-	0.5907	0.5860	0.5739	0.5110	0.5986	0.5892	0.5928	0.6900	-	-	-	-	-	0.6019	0.5985	0.5975	-	-	-	-	-	-	-	-						
	30%	0.4847	0.5008	0.4747	0.5295	0.5119	0.4752	0.4332	0.5378	0.4458	0.5322	0.5214	0.5995	0.5503	0.5442	0.5518	0.5567	0.5566	0.5549	0.5784	0.5861	0.6000	-	-	-	0.5687	0.5593	0.5714	-	-	-	-	-	-	-	-							
06	10%	0.2226	0.3415	0.4165	0.2789	0.4210	0.3495	0.1719	0.4148	0.3924	0.2441	0.5091	0.7146	0.7865	0.7492	0.8875	0.8582	0.8597	0.8223	0.6685	0.8300	0.5657	0.8453	0.7448	0.8401	0.9095	0.8512	-	-	-	-	-	-	-	-	-							
	20%	0.2952	0.4477	0.5141	0.3328	0.4821	0.3635	0.2671	0.4174	0.4616	0.5995	0.9393	0.9148	0.8529	0.8247	0.9379	0.9373	0.9894	0.8510	0.8441	0.9321	0.8935	0.8723	0.8404	0.8698	0.8678	0.9508	-	-	-	-	-	-	-	-	-							
	30%	0.3437	0.5242	0.6171	0.4172	0.4417	0.3265	0.4613	0.4175	0.4740	0.5915	0.9598	0.9386	0.7798	0.7620	0.8597	0.8597	0.9702	0.8572	0.8593	0.8621	0.8442	0.8952	0.8682	0.7568	0.8598	0.8740	0.9200	-	-	-	-	-	-	-	-							
07	10%	0.3732	0.5209	0.6171	0.4172	0.4417	0.3265	0.4613	0.4175	0.4740	0.5915	0.9598	0.9386	0.7798	0.7620	0.8597	0.8597	0.9702	0.8572	0.8593	0.8621	0.8442	0.8952	0.8682	0.7568	0.8598	0.8740	0.9200	-	-	-	-	-	-	-	-							
	20%	0.5431	0.7585	0.8456	0.3970	0.4941	0.4851	0.2715	0.5507	0.5370	0.6955	0.9898	0.9730	0.7952	0.6899	0.5954	0.7175	0.6796	0.6550	0.7257	0.7500	0.6330	-	-	-	0.7425	0.7388	0.8595	-	-	-	-	-	-	-	-							
	30%	0.4747	0.5511	0.6383	0.4850	0.4850	0.4850	0.2715	0.5507	0.5370	0.6955	0.9898	0.9730	0.7952	0.6899	0.5954	0.7175	0.6796	0.6550	0.7257	0.7500	0.6330	-	-	-	0.7425	0.7388	0.8595	-	-	-	-	-	-	-	-							
08	10%	0.4747	0.4539	0.5333	0.3750	0.4588	0.5254	0.3464	0.4392	0.4899	0.4852	0.5021	-	0.7044	0.6420	0.5785	0.6850	0.6128	0.6289	0.6802	0.6741	-	-	-	-	-	0.7359	0.7278	0.8465	-	-	-	-	-	-	-							
	20%	0.3572	0.4713	0.5104	0.4165	0.4688	0.5045	0.3461	0.4674	0.4588	0.4985	0.4985	-	0.9060	0.9020	0.8481	0.8189	0.8388	0.8576	0.8653	0.8128	0.9380	-	-	-	0.7412	0.6476	0.7962	-	-	-	-	-	-	-	-							
	30%	0.2715	0.4713	0.4948	0.4685	0.4811	0.5075	0.3461	0.4477	0.4474	0.4956	0.5102	0.5172	0.5999	0.5999	0.5894	0.5591	0.6298	0.6394	0.6520	0.6096	-	-	-	0.8592	0.8522	0.9503	-	-	-	-	-	-	-	-								
09	10%	0.5870	0.3442	0.3448	0.4850	0.4850	0.4749	0.3068	0.1583	0.9542	0.4411	0.4510	-	0.9382	0.9437	-	-	0.8657	0.8448	0.8647	0.9807	0.8816	-	-	-	0.9478	0.9502	0.9818	-	-	-	-	-	-	-	-							
	20%	0.4882	0.3742	0.3748	0.5419	0.4682	0.7719	0.4309	0.2022	0.4634	0.6156	0.5710	-	0.9111	0.9263	-	-	0.9451	0.9262	0.9736	0.9587	0.9847	-	-	-	0.9475	0.9580	0.9770	-	-	-	-	-	-	-	-							
	30%	0.5150	0.4872	0.4767	0.5930	0.4768	0.6742	0.4505	0.1711	0.5244	0.5792	0.5798	0.6655	0.6070	0.6168	0.7862	0.7485	0.8303	0.8638	0.7910	0.9188	-	-	-	-	-	0.8993	0.8811	0.9160	-	-	-	-	-	-	-	-						
10	10%	0.3812	0.4165	0.2851	0.2851	0.4439	0.4226	0.3490	0.3449	0.3978	0.4202	-	0.9318	0.9368	0.9368	0.7382	0.6699	0.6568	0.7198	0.6668	0.6578	0.9811	-	-	-	0.8765	0.6646	0.7333	-	-	-	-	-	-	-	-							
	20%	0.4151	0.4613	0.4243	0.4243	0.4761	0.4837	0.3265	0.4767	0.3749	0.4579	0.4915	-	0.9599	0.9613	0.9613	0.6590	0.5202	0.6385	0.6247	0.6522	-	-	-	-	-	0.8331	0.6395	0.7103	-	-	-	-	-	-	-							
	30%	0.4432	0.4834	0.3863	0.4333	0.4760	0.4211	0.3264	0.5130	0.3214	0.6003	0.5662	0.9296	0.9194	0.9397	0.6119	0.6513	0.5662	0.6568	0.6288	0.5956	0.9151	-	-	-	0.8118	0.5984	0.5995	-	-	-	-	-	-	-	-							
11	10%	0.4325	0.5377	0.2131	0.4109	0.5396	0.3174	0.3478	0.4189	0.2282	0.4245	0.6145	0.4347	0.6716	0.6574	0.6773	0.7386	0.6890	0.7141	0.7405	0.8113	0.7496	0.7052	0.7362	0.7899	0.8384	0.8243	0.8169	0.8096	0.8948	0.6968	-	-	-	-	-							
	20%	0.4704	0.4646	0.2794	0.3961	0.5364	0.4396	0.3174	0.2599	0.2452	0.6319	0.5880	0.9398	0.9186	0.9474	0.6237	0.7463	0.6661	0.7591	0.7141	0.6926	0.7805	0.7148	0.7747	0.7502	0.7770	0.7527	0.7651	0.7577	0.7750	-	-	-	-	-								
	30%	0.4246	0.5284	0.4645	0.4297	0.5493	0.4982	0.3307	0.3717	0.2574	0.5914	0.5552	0.9690	0.9518	0.9548	0.5852	0.7177	0.6151	0.7154	0.6922	0.6298	0.7117	-	-	-	0.7103	0.6199	0.6866	-	-	-	-	-	-	-	-							
12	10%	0.4048	0.7095	0.8725	0.8734	0.8738	0.8788	0.8505	0.8018	0.8777	0.7656	0.8870	0.9205	0.9113	0.8887	0.8525	0.8167	0.8054	0.8282	0.8282	0.8282	0.9016	0.9159	0.9004	0.8847	0.8328	0.8163	0.8599	0.8244	0.8169	0.9966	0.8054	0.8011	0.8855	-	-							
	20%	0.4041	0.7971	0.8033	0.7716	0.8757	0.8454	0.7997	0.6749	0.7642	0.7092	0.8390	0.9186	0.8985	0.8940	0.8512	0.8133	0.8076	0.8242	0.8120	0.9997	0.9192	0.9052	0.8807	0.8685	0.8620	0.8130	0.8577	0.8222	0.8167	0.9842	0.7959	0.7959	0.8497	-	-							
	30%	0.2903	0.7892	0.6992	0.7203	0.7732	0.8522	0.801																																			

Table 4: Noisy Labels Results with different classifiers and rates

References

- Abril, L.G., Cuberos, F.J., V., F., Ortega, J.A., 2009. Ameva: An autonomous discretization algorithm. *Expert Syst. Appl.* 36, 5327–5332.
- Benavoli, A., Corani, G., Demsar, J., Zaffalon, M., 2017. Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis. *Journal of Machine Learning Research* 18, 1–36.
- Butterworth, R., Simovici, D.A., Santos, G.S., Ohno-Machado, L., 2004. A greedy algorithm for supervised discretization. *Journal of Biomedical Informatics* 37, 285–292.
- Catlett, J., 1991. On changing continuous attributes into ordered discrete attributes, in: *Proceedings of the 5th European Working Session on Learning*, Springer-Verlag, Porto (Portugal). pp. 164–178.
- Ching, J.Y., Wong, A.C., Chan, K.C.C., 1995. Class-dependent discretization for inductive learning from continuous and mixed-mode data. *IEEE Transactions on Pattern Analysis and Machine Learning* 17, 641–651.
- Domingos, P., Pazzani, M., 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* 29, 103–130.
- Fayyad, U., Irani, K., 1992. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning* 8, 87–102.
- Flores, J., 2019. Complete Bayesian analysis of discretization algorithms. <https://github.com/isg-ehu/joseluis.flores/tree/master/kernel.discretization/>. [Online; accessed 20-June-2019].
- Frénay, B., Verleysen, M., 2014. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems* 25, 845–869. doi:10.1109/tnnls.2013.2292894.
- Friedman, N., Geiger, D., Goldszmidt, M., 1997. Bayesian network classifiers. *Machine Learning* 29, 131–163.
- Garcia, S., Luengo, J., Saez, J., Lopez, V., Herrera, F., 2013. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transaction on Knowledge and Data Engineering* 25, 734–750.
- Giráldez, R., Aguilar-Ruiz, J.S., Santos, J.C.R., 2003. Natural coding: a more efficient representation for evolutionary learning, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, Springer-Verlag, Chicago (Illinois,USA). pp. 979–990.
- Grabczewski, K., 2004. SSV criterion based discretization for naive Bayes classifiers., in: *Proceedings of the International Conference on Artificial Intelligence and Soft Computing*, Springer-Verlag, Zakopane (Poland). pp. 574–579.
- Kerber, R., 1992. Chimerge: Discretization of numeric attributes, in: *Proceedings of the 10th National Conference on Artificial Intelligence*, San Jose (California,USA). pp. 123–128.
- Kotkanen, P., Myllymaki, P., Tirri, H., 1997. A Bayesian approach to discretization, in: *Proceedings of the European Symposium on Intelligent Techniques*, Bari (Italy). pp. 265–268.
- Kozlov, A.V., Koller, D., 1997. Nonuniform dynamic discretization in hybrid networks, in: *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*, Providence (Rhode Island,USA). pp. 314–325.
- Kurgan, L., 2004. Discretization algorithm that uses class attribute interdependence maximization. *IEEE Transactions on Knowledge and Data Engineering* 16, 145–163.
- Li, R.P., Wang, Z.O., 2002. An entropy-based discretization method for classification rules with inconsistency checking, in: *Proceedings. International Conference on Machine Learning and Cybernetics*, pp. 243–246.
- Liu, L., Wong, A.K.C., Wang, Y., 2004. A global optimal algorithm for class-dependent discretization of continuous data. *Intelligent Data Analysis* 8, 151–170.
- Maron, M.E., 1961. Automatic indexing: An experimental inquiry. *Journal of the ACM* 8, 404–417.

- Pérez, A., Inza, I., Larrañaga, P., 2009. Bayesian classifiers based on kernel density estimation: Flexible classifiers. *International Journal of Approximate Reasoning* 52, 341–362.
- Pfahring, B., 1995. Compression-based discretization of continuous attributes, in: *International Conference on Machine Learning*, Tahoe City (California, USA). pp. 456–463.
- Rodríguez, J.D., Pérez, A., Lozano, J.A., 2013. A general framework for the statistical analysis of the sources of variance for classification error estimators. *Pattern Recognition* 46, 855–864.
- Sahami, M., 1996. Learning limited dependence bayesian classifiers, in: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, AAAI Press. pp. 335–338. URL: <http://dl.acm.org/citation.cfm?id=3001460.3001537>.
- Setiono, R., Liu, H., 1995. Chi2: Feature selection and discretization of numeric attributes, in: *Proceedings of the 7th International Conference on Tools with Artificial Intelligence*, Washington DC (USA). pp. 388–391.
- Silverman, B.W., 1986. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall.
- Su, C.T., Hsu, J.H., 2005. An extended chi2 algorithm for discretization of real value attributes. *IEEE Transactions on Knowledge and Data Engineering* 17, 437–441.
- Tay, F., Shen, L., 2002. A modified chi2 algorithm for discretization. *Knowledge and Data Engineering, IEEE Transactions on* 14, 666–670.
- Tsai, C.J., Lee, C.I., Yang, W.P., 2008. A discretization algorithm based on class-attribute contingency coefficient. *Information Sciences* 178, 714 – 731.
- Wand, M.P., Jones, M.C., 1995. *Kernel smoothing*. Monographs on statistics and applied probability, Chapman & Hall/CRC, Boca Raton (Fla.), London, New York.
- Wang, M., Geisser, S., 2005. Optimal dichotomization of screening test variables. *Journal of Statistical Planning and Inference* 131, 191–206.
- Yang, Y., 2003. *Discretization for Naive-Bayes Learning*. Master's thesis. Monash University.