Relevance and Mutual Information-based Feature Discretization

Artur J. Ferreira[1,3] and Mario A. T. Figueiredo[2,3]
[1]*Instituto Superior de Engenharia de Lisboa, Lisboa, Portugal*
[2]*Instituto Superior Técnico, Lisboa, Portugal*
[3]*Instituto de Telecomunicações, Lisboa, Portugal*

Keywords: Classification, Feature Discretization, Mutual Information, Quantization, Supervised Learning.

Abstract: In many learning problems, *feature discretization* (FD) techniques yield compact data representations, which often lead to shorter training time and higher classification accuracy. In this paper, we propose two new FD techniques. The first method is based on the classical Linde-Buzo-Gray quantization algorithm, guided by a relevance criterion, and is able to work in unsupervised, supervised, or semi-supervised scenarios, depending on the adopted measure of relevance. The second method is a supervised technique based on the maximization of the mutual information between each discrete feature and the class label. For both methods, our experiments on standard benchmark datasets show their ability to scale up to high-dimensional data, attaining in many cases better accuracy than other FD approaches, while using fewer discretization intervals.

1 INTRODUCTION

A typical dataset is composed of categorical and/or numeric features. The categorical features are discrete by nature. The numeric features use real or integer representations. In some cases, these features have noisy values or show minor fluctuations that are irrelevant or even harmful for the learning task at hand. For such features, the performance of machine learning and data mining algorithms can be improved by discretization. Moreover, some learning algorithms require a discrete representation of the data. In order to address these problems, the use of *feature discretization* (FD) (Witten and Frank, 2005) techniques has been extensively considered in the past. FD provides compact representations, with lower memory usage, while at the same time it may reduce the training time and improve the classification accuracy. The literature on FD includes many supervised and unsupervised techniques (*i.e.*, making use, or not, of class labels) (Witten and Frank, 2005; Dougherty et al., 1995; Jin et al., 2009; Liu et al., 2002; Kotsiantis and Kanellopoulos, 2006).

In this paper, we propose two new FD techniques. The first one is based on the classical Linde-Buzo-Gray (LBG) (Linde et al., 1980) quantization algorithm, along with a relevance criterion that guides the discretization process, being able to work in unsupervised, supervised, or semi-supervised learning. The second technique is supervised and is based on the *mutual information* (MI) (Cover and Thomas, 1991) between each (discretized) feature and the class label.

The remainder of this paper is organized as follows. Section 2 reviews the advantages and disadvantages of using FD techniques, describing some unsupervised and supervised techniques. Section 3 presents our proposed methods for FD. The experimental evaluation is carried out in Section 4 and the paper ends in Section 5 with some concluding remarks and directions for future work.

2 FEATURE DISCRETIZATION

In this section, we provide some background on FD techniques, reviewing their benefits and drawbacks. Brief descriptions of unsupervised and supervised FD techniques are provided in Subsection 2.1 and Subsection 2.2, respectively.

Regardless of the type of classifier considered, FD techniques aim at finding a representation of each feature that contains enough information for the learning task at hand, while ignoring minor fluctuations that may be irrelevant for that task. The use of a discretization technique will lead to a more compact (using less memory), and hopefully to a better representation of the data for learning purposes, as compared to the use of the original features.

Ferreira A. and Figueiredo M. (2013).
Relevance and Mutual Information-based Feature Discretization.
In *Proceedings of the 2nd International Conference on Pattern Recognition Applications and Methods*, pages 68-77
DOI: 10.5220/0004268000680077

A typical dataset with numeric features uses real or integer representations. It has been found that the use of FD techniques, with or without a coupled *feature selection* (FS) technique, may improve the results of many learning methods (Dougherty et al., 1995; Witten and Frank, 2005). Although supervised discretization may, in principle, lead to better classifiers, it has been found that unsupervised FD methods perform well on different types of data (see for instance (Yang and Webb, 2001)).

The quality of discretization is usually assessed by two indicators: the *generalization error* and the *complexity*, *i.e.*, the number of intervals or equivalently the number of bits used to represent each instance. A possible drawback of FD is arguably the (time and memory) cost of the discretization procedure.

A detailed description of many FD methods can be found in (Dougherty et al., 1995; Jin et al., 2009; Liu et al., 2002; Kotsiantis and Kanellopoulos, 2006) and the many references therein.

2.1 Unsupervised Methods

The most common techniques for unsupervised FD are (Witten and Frank, 2005): *equal-interval binning* (EIB), which performs uniform quantization; *equal-frequency binning* (EFB) (Chiu et al., 1991), which obtains a non-uniform quantizer with intervals such that, for each feature, the number of occurrences in each interval is the same; *proportional k-interval discretization* (PkID) (Yang and Webb, 2001), which computes the number and size of the discretization intervals as functions of the number of training instances.

EIB is obviously the simplest and easiest to implement, but is sensitive to outliers. In EFB, the quantization intervals are smaller in regions where there are more occurrences of the values of each feature; EFB is thus less sensitive to outliers, as compared to EIB. In the EIB and EFB methods, the user can choose the number of discretization bins. In contrast, the PkID method sets the number and size of the discretization intervals as a function of the number of training instances, seeking a trade-off between bias and variance of the class probability estimate of a naïve Bayes classifier (Yang and Webb, 2001).

Recently, we have proposed an unsupervised scalar FD method (Ferreira and Figueiredo, 2012) based on the LBG algorithm (Linde et al., 1980). For a given number of discretization intervals, LBG discretizes the data seeking the minimum *mean square error* (MSE) with respect to the original representation. This approach, named *unsupervised* LBG (U-LBG 1) and described as Algorithm 1, applies the LBG algorithm to each feature independently and stops when the MSE falls below a threshold Δ or when the maximum number of bits q per feature is reached. A variant of U-LBG1, named U-LBG2, using a fixed number of bits per feature q was also proposed. Both U-LBG1 and U-LBG2 rely on the idea that a discrete representation with low MSE is adequate for learning.

2.2 Supervised Methods

The *information entropy minimization* (IEM) method (Fayyad and Irani, 1993), based on the *minimum description length* (MDL) principle, is one of the oldest and most often used methods for supervised FD. The key idea of using the MDL principle is that the most informative features to discretize are the most compressible ones. The IEM method is based on the use of the entropy minimization heuristic for discretization of a continuous value into multiple intervals. IEM adopts a recursive approach computing the discretization cut-points in such a way that they minimize the amount of bits needed to represent the data. It follows a top-down approach in the sense that it starts with one interval and splits intervals in the process of discretization.

The method termed *IEM variant* (IEMV) (Kononenko, 1995) is also based on an entropy minimization heuristic to choose the discretization intervals. It applies a function, based on the MDL principle, which decreases as the number of different values for a feature increases.

The supervised static *class-attribute interdependence maximization* (CAIM) (Kurgan and Cios, 2004) algorithm aims to maximize the class-attribute interdependence and to generate a (possibly) minimal number of discrete intervals. The algorithm does not

Algorithm 1: U-LBG1.

Input: X, $n \times d$ matrix training set (n patterns, d features).
Δ, q: maximum expected distortion and maximum number of bits/feature.

Output: $\widetilde{X}$: $n \times d$ matrix, discrete feature training set.
$Q^1_{b_1}, \ldots, Q^d_{b_d}$: set of d quantizers (one per feature).

for $i = 1$ to d **do**
  for $b = 1$ to q **do**
    Apply LBG to the i-th feature to obtain a b-bit quantizer $Q_b(\cdot)$;
    Compute $\text{MSE}_i = \frac{1}{n}\sum_{j=1}^{n}(X_{ij} - Q_b(X_{ij}))^2$;
    if ($\text{MSE}_i \leq \Delta$ or $b = q$) **then**
      $Q^i(\cdot) = Q_b(\cdot)$; {/* Store the quantizer. */}
      $\widetilde{X}_i = Q^i(X_i)$; {/* Quantize feature. */}
      break; {/* Proceed to the next feature. */}
    end if
  end for
end for

require a predefined number of intervals, as opposed to some other FD methods. Experimental results reported show that CAIM compares favorably with six other FD discretization algorithms, in that the discrete attributes generated by CAIM almost always have the lowest number of intervals and the highest class-attribute interdependency, achieving the highest classification accuracy (Kurgan and Cios, 2004).

Finally, we mention *class-attribute contingency coefficient* (CACC) (Tsai et al., 2008), which is an incremental, supervised, top-down FD method, that has been shown to achieve promising results regarding execution time, number of discretization intervals, and training time of the classifiers.

3 PROPOSED METHODS

3.1 Relevance-based LBG

As in U-LBG1 (Algorithm 1), our FD proposal, named *relevance-based* LBG (R-LBG) and described in Algorithm 2, uses the LBG algorithm, discretizing data with a variable number of bits per feature. We use a relevance function, denoted *@rel*, and a (non-negative) stopping factor, ε. The relevance function, producing non-negative values, is applied after each discretization. R-LBG behaves differently, depending on the value of ε. If ε is positive, whenever there is an increase below ε on the relevance between two subsequent discrete versions (with b and $b+1$ bits), discretization is halted and b bits are kept, for that feature; otherwise, with a significant (larger than ε) increase on the relevance, it discretizes with one more bit, assessing the new relevance. In summary, it discretizes a feature with an increasing number of bits, stopping only when there is no significant increase on the relevance of the recently discretized feature. If $\varepsilon = 0$, each feature is discretized from 1 up to the maximum q bits and the corresponding relevance is assessed on each discretization. Then, the minimum number of bits that ensures the highest relevance is kept and applied to discretize that feature. Regardless of the value of ε, the method discretizes data with a variable number of bits per feature.

The relevance assessment $r_{ib} = @rel(Q_b^i(X_i);...)$, of feature i with b bits, in line 5 of Algorithm 2, can refer to unsupervised, supervised, or semi-supervised learning. This depends on how the relevance function makes use (or not) of the class labels. The value of ε, when different from zero, should be set according to the range of the *@rel* function.

There are many different choices for the relevance criteria of R-LBG. In the unsupervised case, if we consider $@rel = MSE$ (between original and discrete features) we have the unsupervised U-LBG1 approach. Another relevance criterion is given by the quotient between the variance of the discrete feature and the number of discretization intervals

$$NVAR(\widetilde{X}_i) = \text{var}(\widetilde{X}_i) \;/\; 2^{b_i}, \tag{1}$$

where b_i is the number of bits of the discrete feature.

For the supervised case, we propose to compute relevance by the *mutual information* (MI) (Cover and Thomas, 1991) between discretized features $\widetilde{X}_i$, with b_i bits and the class label vector $\mathbf{y}$

$$\begin{aligned} MI(\widetilde{X}_i;\mathbf{y}) &= H(\widetilde{X}_i) - H(\widetilde{X}_i|\mathbf{y}) \\ &= H(\mathbf{y}) - H(\mathbf{y}|\widetilde{X}_i). \end{aligned} \tag{2}$$

where $H(.)$ and $H(.|.)$ denote entropy and conditional entropy, respectively (Cover and Thomas, 1991).

There are many other (unsupervised and supervised) feature relevance criteria; in fact, all the criteria used in feature selection methods to rank features are

Algorithm 2: R-LBG - Relevance-based LBG.

Input: X: $n \times d$ matrix training set (n patterns, d features).
y: n-length vector with class labels (supervised).
q: maximum number of bits per feature.
$@rel, \varepsilon\ (\geq 0)$: relevance function, stopping factor.

Output: $\widetilde{X}$: $n \times d$ matrix, discrete feature training set.
$Q_{b_1}^1, ..., Q_{b_d}^d$: set of d quantizers (one per feature).

```
1:  for i = 1 to d do
2:      pRel = 0;   {/* Initial/previous rel. for feature i. */}
3:      for b = 1 to q do
4:          Apply LBG to the i-th feature to obtain a b-bit
            quantizer Q_b^i(·);
5:          Compute r_ib = @rel(Q_b^i(X_i);...), relevance of
            feature i with b bits;
6:          if (ε == 0) then
7:              continue;        {/* Discretize up to q bits. */}
8:          end if
9:          if ( (r_ib − pRel) > ε) then
10:             Q^i(·) = Q_b(·); X̃_i = Q_b^i(X_i);        {/* High
                increase. Store quantizer and discretize. */}
11:         else
12:             break;     {/* Non-significant increase. Break
                loop. Move on to the next feature. */}
13:         end if
14:         pRel = r_ib;       {/* Keep previous relevance. */}
15:     end for
16: end for
17: if (ε == 0) then
18:     for i = 1 to d do
19:         Get b_i = arg max_{b∈{1,...,q}} r_{i*} {/* Max. relevance. */}
20:         Q^i(·) ← Apply LBG (b_i bits) to the i-th feature;
21:         X̃_i = Q_{b_i}^i(X_i);          {/* Discretize feature. */}
22:     end for
23: end if
```

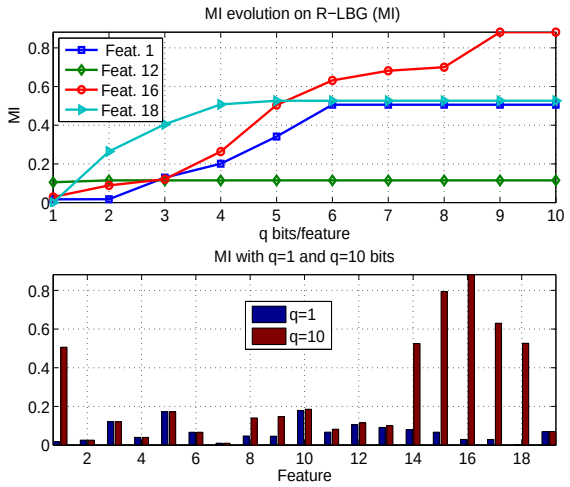

Figure 1: R-LBG (MI) discretization on the Hepatitis dataset. Top: MI as a function of the number of bits $q \in \{1,\ldots,10\}$, for features 1, 12, 16, and 18. Bottom: MI with $q = 1$ and $q = 10$ bits, for all the $d = 19$ features.

potential candidates to serve as a relevance measure in R-LBG. The relevance function can also be chosen such that it only uses the class label for those instances for which it is available, thus being usable in semi-supervised learning.

As an illustration of the supervised case, Figure 1 (top) shows the evolution of the MI between the class label and some of the features discretized by the R-LBG algorithm, using $q \in \{1,\ldots,10\}$ bits per feature, for the Hepatitis dataset. In the bottom plot, we compare the MI values obtained by discretizing with $q = 1$ and $q = 10$ bits, for each of the 19 features of the same dataset. The top plot shows that for features 1, 16, and 18, the MI grows with the number of bits and then it levels off. For feature 12 (which is categorical, thus originally discrete), as obviously expected, an increasing number of bits does not lead to a higher MI. Thus, our method handles both continuous and categorical features and the relevance values provide a feature ranking score. In practice, the choice of adequate values of ε, which depends on the type of data, can be done using these plots, by checking how the MI increases. In the bottom plot, we see that some features, such as 3, 4, 5, and 6, show no significant MI increase, when moving from $q = 1$ to $q = 10$. On the other hand, for features 14 to 18, we have a strong MI increase, which usually corresponds to numeric informative features.

3.2 Mutual Information Discretization

In this subsection we present the proposed supervised FD method, named *mutual information discretization* (MID). Essentially, the MID method consists in discretizing each feature individually, computing the discretization cut-points in order to maximize the MI of the discrete feature with the class label. The key motivation for this FD proposal is that the MI between features and class labels has been extensively used as a FS criterion; see the seminal work in (Battiti, 1994) and (Brown et al., 2012) for a review of MI-based FS methods. It is thus expectable that a good criterion for FS will also be adequate for FD.

The usual argument is based on bounds for the probability of error which depend on the MI between the observations and the class label, namely the Fano, Hellman-Raviv, and Santhi-Vardi bounds (Brown et al., 2012), (Santhi and Vardy, 2006). The Hellman-Raviv bound (Hellman, 1970) on the Bayes risk is given by

$$err_{Bayes}(\widetilde{X}_i) \leq \frac{1}{2}H(\widetilde{X}_i|\mathbf{y}) \tag{3}$$

while the Santhi-Vardy bound (Santhi and Vardy, 2006) is

$$err_{Bayes}(\widetilde{X}_i) \leq 1 - 2^{-H(\widetilde{X}_i|\mathbf{y})}. \tag{4}$$

In order to maximize the MI (2), one must minimize $H(\widetilde{X}_i|\mathbf{y})$, that is, the uncertainty about the feature value, given a known class label. We have $0 \leq H(\widetilde{X}_i|\mathbf{y}) \leq H(\widetilde{X}_i)$, with $H(\widetilde{X}_i|\mathbf{y}) = 0$ meaning deterministic dependence (an ideal feature) and $H(\widetilde{X}_i|\mathbf{y}) = H(\widetilde{X}_i)$ corresponding to independence between the feature and the class label (a useless feature). On the other hand, $H(\mathbf{y})$ does not change with discretization; thus, maximizing (2) is equivalent to minimizing $H(\mathbf{y}|\widetilde{X}_i)$, that is, the uncertainty about the class label given the feature. We have $0 \leq H(\mathbf{y}|\widetilde{X}_i) \leq H(\mathbf{y})$, with $H(\mathbf{y}|\widetilde{X}_i) = 0$ corresponding to deterministic dependence (again, the ideal case) and $H(\mathbf{y}|\widetilde{X}_i) = H(\mathbf{y})$ meaning independence (a useless feature). For an ideal feature (one that is a deterministic injective function of the class label), we have

$$MI(\widetilde{X}_i;\mathbf{y}) = H(\widetilde{X}_i) = H(\mathbf{y}). \tag{5}$$

Of course, the maximum possible value for $MI(\widetilde{X}_i;\mathbf{y})$ depends on both the number of bits used to discretize X_i and the number of classes c. If we discretize X_i with b_i bits, its maximum entropy is $H_{max}(\widetilde{X}_i) = b_i$ bit/symbol; the maximum value of the class entropy is $H_{max}(\mathbf{y}) = \log_2(c)$ bit/symbol, which corresponds to c equiprobable classes. We thus conclude that the maximum value of the MI between the class label and a discretized feature (with b_i bits) is

$$MI_{max}(\widetilde{X}_i;\mathbf{y}) = \min\{b_i,\ \log_2(c)\}. \tag{6}$$

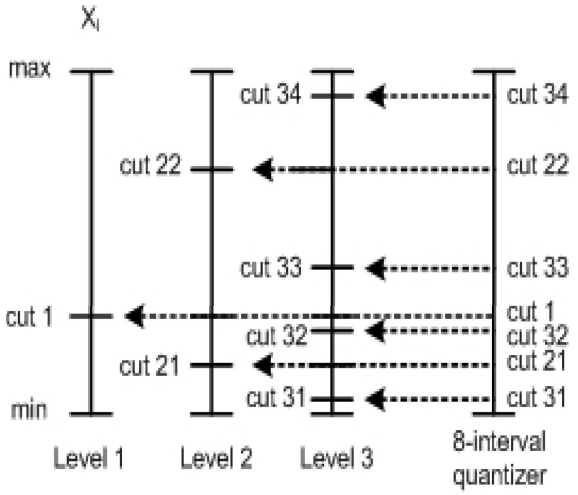

Figure 2: Illustration of the progressive and recursive partition algorithm for feature discretization, using $q = 3$ bits, leading to a 8-interval quantizer.

In the binary case $c = 2$, we have $MI_{max}(\widetilde{X}_i;\mathbf{y}) = 1$ bit. Moreover, to attain the maximum possible value for the MI, one must choose the maximum number of bits q taking into account this expression; this implies that $q \geq \lceil \log_2(c) \rceil$, which is more meaningful for multi-class problems.

3.2.1 Algorithm Outline

At the discretization stage, we search for discretization boundaries such that the resulting discrete feature has the highest MI with the class label. Thus, as described above, by maximizing the MI at each partition and each cut-point we are aiming at leveraging the performance of the discrete feature, leading to higher accuracy. The method works in a recursive way, by successively breaking each feature into intervals, as depicted in Figure 2 with $q = 3$ bits yielding a 8-interval non-uniform quantizer.

We propose two versions of the MID technique. The first, named ***MID fixed***, uses a fixed number of q bits per feature. In summary, given a training set with n instances and d features, $\mathbf{X}$ and a maximum number of bits per feature q, the MID fixed method applies the recursive discretization method just described, using up to q bits per feature, yielding quantizer $Q_i(\cdot)$ for feature i and the discretized feature $\widetilde{X}_i = Q_i(X_i)$.

The second version, named ***MID variable*** allocates *up to* q bits per feature, leading to a variable number of bits per feature. As in R-LBG, we halt the bit allocation for feature X_i with b bits, whenever its discretization with $b+1$ bits does not lead to a significant increase (larger than ε) on the $MI(\widetilde{X}_i;\mathbf{y})$. As a consequence, the MID variable version will produce

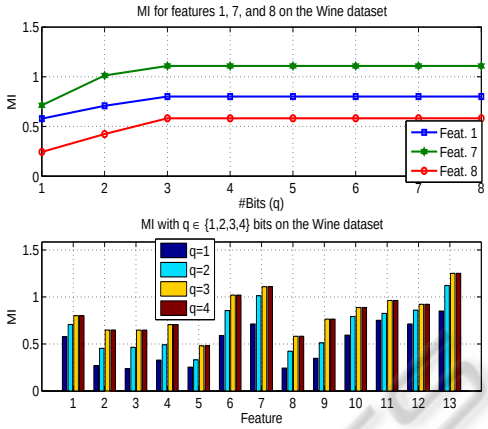

Figure 3: MID for the Wine dataset ($d = 13$ features, $c = 3$ classes). Top: evolution of MI for features 1, 7, and 8, with $q \in \{1,\ldots,8\}$. Bottom: MI between discretized features and the class label, for $q \in \{1,2,3,4\}$.

fewer discretization intervals (and thus fewer bits per instance), as compared to the MID fixed method. By setting $\varepsilon = 0$, MID variable discretizes feature $\widetilde{X}_i$, with maximum MI, using the smallest possible number of bits $b_i \leq q$ (it acts in a similar fashion as R-LBG). The number of discretization intervals depends on the value of ε; larger values will lead to fewer intervals.

Figure 3 (top) plots the evolution of MI for features 1, 7, and 8 for the Wine dataset. We see an increase in the first few bits and then the values of MI level off. The average MI values for all $d = 13$ features are 0.4977, 0.6810, 0.8294, and 0.8294, for $q \in \{1,2,3,4\}$ bits, respectively. The training partition has class entropy $H_{max}(\mathbf{y}) = \log_2(3) = 1.585$ bits. In the bottom plot, we see an overall increase of the MI when moving from 1 to 3 bits; however, using $q = 4$, there is no appreciable increase on the MI.

4 EXPERIMENTAL EVALUATION

This section reports experimental results of our FD techniques on several public domain datasets, for the task of supervised classification. We use a 10-fold *cross validation* (CV) strategy. In each fold, the quantizers are learned in the training partition and then applied to the test partition. We apply linear *support vector machines* (SVM), *naïve Bayes* (NB), and *K*-nearest-neighbors (KNN) (with $K = 3$) classifiers, of the PRTools[1] toolbox (Duin et al., 2007).

Table 1 briefly describes the publicly available datasets that were used in our experiments. We chose

[1] www.prtools.org/prtools.html

Table 1: Datasets, with c classes, n instances, and listed by order of increasing dimensionality d.

Dataset	d	c	n	**Type of Data**
Wine	13	3	178	Wine cultivar
Hepatitis	19	2	155	Biological
Ionosphere	34	2	351	Radar return
Colon	2000	2	62	Microarray
SRBCT	2309	4	83	Microarray
AR10P	2400	10	130	Face database
PIE10P	2420	10	210	Face database
Leukemia1	5327	3	72	Microarray
TOX-171	5748	4	171	Microarray
Brain-Tumor1	5920	5	90	Microarray
ORL10P	10304	10	100	Face database
Prostate-Tumor	10509	2	102	Microarray
Leukemia2	11225	3	72	Microarray
GLI-85	22283	2	85	Microarray

several well-known datasets with different kinds of data, problems, classes, and dimensionality, including datasets from the UCI repository[2] (Frank and Asuncion, 2010), face database, and bioinformatics datasets from the *gene expression model selector* (GEMS) project[3] (Statnikov et al., 2005), and from the *Arizona state university* (ASU)[4] repository (Zhao et al., 2010).

The experimental results are organized as follows. We start, in Subsection 4.1, by evaluating the behavior of our supervised FD methods using a variable number of bits per feature. In Subsection 4.2, we compare our methods with existing unsupervised and supervised FD techniques (reviewed in Subsections 2.1 and 2.2, respectively). This evaluation is focused both on the *complexity* and the *generalization error*. Subsection 4.3 provides some discussion on these results.

4.1 Analysis of Our Approaches

For both the R-LBG and *MID variable* algorithms, we use different values of ε and assess the number of discretization intervals and the generalization error. Table 2 reports experimental results with the average number of bits per instance (with $q = 4$ and $\varepsilon \in \{0, 0.1\}$) and the test set error rate for the linear SVM classifier (*No FD* denotes the use of the original features).

On the R-LBG algorithm, $\varepsilon = 0$ usually leads to a larger number of bits per instance, as compared with $\varepsilon = 0.1$. This happens because with $\varepsilon = 0$ we are aiming at finding the maximum relevance, whereas with $\varepsilon > 0$ we halt the discretization process at earlier stages. For the *MID variable* algorithm, $\varepsilon = 0$ leads to the choice of the minimum bits per feature that ensure the maximum MI; for this reason, with $\varepsilon = 0$ we usually have fewer bits per instance as compared with $\varepsilon > 0$. Regarding the classification accuracy, $\varepsilon = 0$ usually attains the best results with a few exceptions.

Table 2: Evaluation of R-LBG (*@rel* = *MI*) and *MID variable* with $q = 4$. For each dataset, the first row displays the total number of bits per instance and the second row the test set error rate (%), of a 10-fold CV for the linear SVM classifier. The best error rate is in bold face.

	R-LBG (MI)		**MID variable**	
D. / No FD	$\varepsilon = 0$	$\varepsilon = 0.1$	$\varepsilon = 0$	$\varepsilon = 0.1$
Wine	52.0	30.6	38.3	26.2
3.9	2.8	**1.7**	3.4	2.8
Hepatitis	46.5	68.8	28.5	65.6
21.3	**15.5**	21.9	18.7	18.1
Ionosphere	129.0	102.4	73.0	85.0
12.8	14.0	12.5	9.4	**5.7**
Colon	7954.6	7564.0	4682.0	6151.9
17.7	19.4	**14.5**	19.4	**14.5**
SRBCT	9222.5	8827.7	7144.2	7180.3
0.0	**0.0**	**0.0**	**0.0**	**0.0**
AR10P	9599.8	9583.2	8620.4	8640.4
0.8	0.8	0.8	**0.0**	**0.0**
PIE10P	9679.9	9662.5	8550.7	8543.4
0.0	**0.0**	**0.0**	**0.0**	**0.0**
Leukemia1	21248.2	19818.7	14636.9	15555.9
8.3	**4.2**	5.6	8.3	6.9
TOX-171	22988.5	21439.2	19012.4	20070.8
14.6	**2.3**	2.9	4.1	4.1
B-Tumor1	23649.6	22174.4	17531.0	17436.5
11.1	**8.9**	10.0	10.0	10.0
ORL10P	41215.6	41195.1	37410.3	37410.2
1.0	**1.0**	**1.0**	2.0	2.0
P-Tumor	41735.0	40431.3	25493.1	36598.8
10.8	**7.8**	**7.8**	**7.8**	**7.8**
Leukemia2	44300.1	40072.9	31124.0	30255.4
5.6	**1.4**	**1.4**	**1.4**	**1.4**
GLI-85	88561.7	84364.2	54906.9	72131.8
10.6	**8.2**	**8.2**	**8.2**	**8.2**

We have assessed the statistical significance of our results with the Friedman test (Friedman, 1940), to the average test set error rates, as suggested by (Demsar, 2006) and (Garcia and Herrera, 2008). For this purpose, we have used the JAVA tool of (Garcia and Herrera, 2008)[5]. For the results in Table 2, the Friedman test reported a p-value of $0.04164 < 0.05$, stating that these results are statistically significant.

In order to assess how discretization is affected by the values of ε, Figure 4 shows the test set error rate for a 10-fold CV of the NB classifier on data discretized by R-LBG and *MID variable* with $q = 5$ bits and ε ranging in the real interval from 0 to 0.2, on the Wine dataset. For the small values of ε, the R-LBG algorithm leads to discrete features with lower test set error rate than those obtained by the *MID variable* algorithm. On both algorithms, by choosing values

[2] archive.ics.uci.edu/ml/datasets.html

[3] www.gems-system.org

[4] featureselection.asu.edu/datasets.php

[5] sci2s.ugr.es/keel/multipleTest.zip

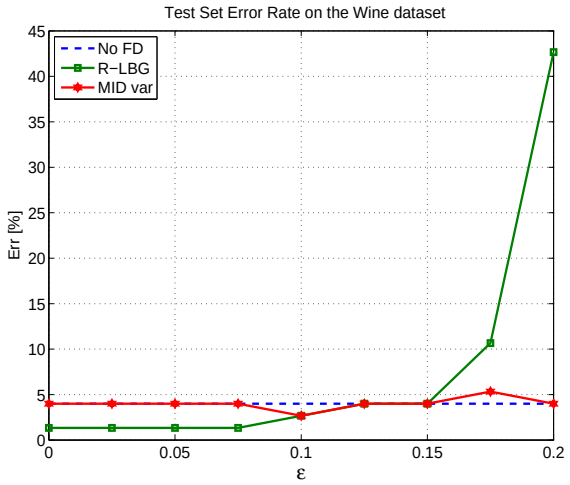

Figure 4: R-LBG (MI) and *MID variable* discretization on the Wine dataset with $q = 5$ bits. Test set error rate (%) of the NB classifier for a 10-fold CV, as function of the ε parameter, ranging in the real interval from 0 to 0.2.

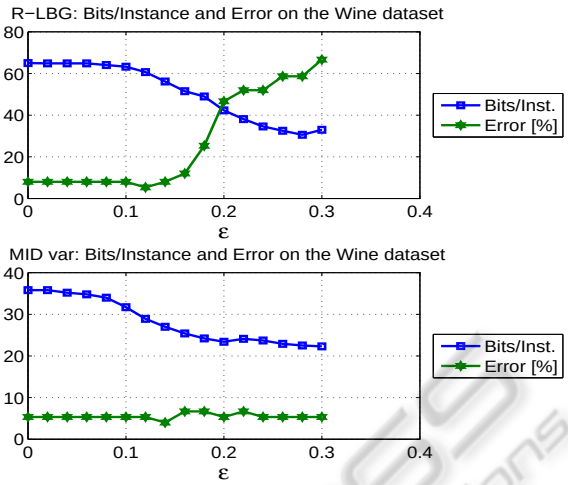

Figure 5: Number of bits/instance and the test set error rate (%) for a 10-fold CV of the NB classifier on data discretized by R-LBG and MID variable with $q = 5$ bits and ε in the real interval from 0 to 0.3, on the Wine dataset for R-LBG (top) and MID variable (bottom).

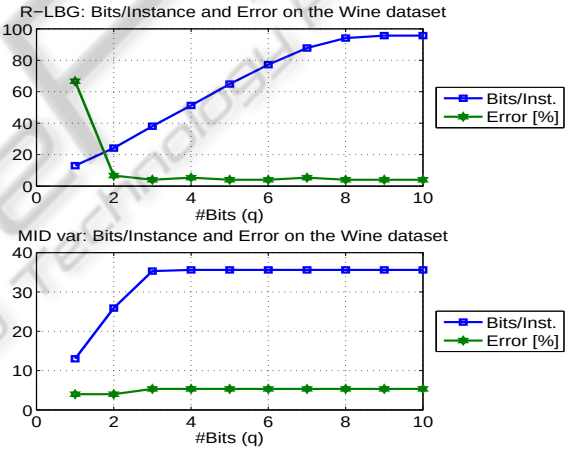

Figure 6: Number of bits/instance and the test set error rate (%) for a 10-fold CV of the NB classifier on data discretized by R-LBG and MID variable with $\varepsilon = 0.05$, and $q \in \{1, \ldots, 10\}$ bits, on the Wine dataset.

of ε from zero roughly up to 0.15, we get a generalization error equal or better than the baseline error (without FD). On the R-LBG algorithm, with ε close to 0.2, the number of discretization intervals per feature drops yielding poor discretizations and the test error rate degrades seriously. The MID algorithm is less sensitive to the increase of the ε parameter; in the *MID variable* approach, we have a wide range of values of ε that lead to low generalization error.

Figure 5 shows the evolution of both the number of bits/instance and the test set error rate for a 10-fold CV of the NB classifier on data discretized by R-LBG and *MID variable* with $q = 5$ bits and ε in the real interval from 0 to 0.3, on the Wine dataset. As ε increases, the number of discretization intervals and thus the number of bits per instance decreases. The test set error rate only becomes higher at larger values of ε. Again, the R-LBG algorithm shows higher sensitivity with respect to the increase of this parameter, since the test set error rate becomes unacceptably high for ε close to 0.15. On the other hand, the *MID variable* algorithm exhibits a more stable behavior as compared to R-LBG; the corresponding test set error rate does not increase so fast as in R-LBG, whenever the number of bits per instance decreases.

Figure 6 shows the effect of varying the maximum number of bits for discretization, $q \in \{1, \ldots, 10\}$, for both R-LBG and *MID variable*, keeping a fixed value $\varepsilon = 0.05$ (again with both the NB classifier and the Wine dataset). By increasing the maximum number of bits per feature, *MID variable* uses fewer bits per instance as compared to the R-LBG algorithm. The classification accuracies are similar and they both exhibit a stable behavior in the sense that an (excessive) increase on the maximum number of bits per feature q does not degrade the test set error rate.

4.2 Comparison with Existing Methods

4.2.1 Unsupervised Discretization

First, we assess the behavior of R-LBG in unsupervised mode, comparing it with five existing unsupervised FD methods (see Subsection 2.1). We evaluate the average number of discretization intervals and the average 10-fold CV error (%), attained by each FD method, with $q = 3$ bits. R-LBG uses *@rel* = *NVAR*

Table 3: Evaluation of unsupervised FD. For each dataset, the first row presents the average total number of bits per instance and the second row has the average test set error rate (%), using a 10-fold CV for the linear SVM classifier. The best error rate is in bold face. We have used $q = 3$ bit/feature, $\Delta = 0.05\text{range}(X_i)$ for U-LBG1, *@rel* = *NVAR*, and $\varepsilon = 0.25$, for R-LBG.

		Existing Methods					Proposed
Dataset	**No FD**	**EIB**	**EFB**	**PkID**	**U-LBG1**	**U-LBG2**	**R-LBG**
Wine		39.0	39.0	52.0	22.0	39.0	14.8
	4.5	**3.4**	4.5	3.9	3.9	**3.4**	7.9
Hepatitis		57.0	57.0	46.0	28.9	57.0	38.6
	22.6	20.0	20.6	21.9	20.0	19.4	**18.7**
Ionosphere		99.0	99.0	145.0	44.9	99.0	75.4
	12.8	11.1	13.1	16.2	**10.8**	14.0	11.4
Colon		6000.0	6000.0	6000.0	6000.0	6000.0	2534.4
	19.4	16.1	**11.3**	12.9	14.5	14.5	**11.3**
SRBCT		6924.0	6924.0	6924.0	2825.3	6924.0	2641.2
	0.0	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	1.2
AR10P		7200.0	7200.0	9267.8	7200.0	7200.0	6568.6
	0.8	**0.0**	0.8	0.8	0.8	0.8	1.5
PIE10P		7260.0	7260.0	9680.0	7260.0	7260.0	3774.4
	0.0	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Leukemia1		15981.0	15981.0	15981.0	15981.0	15981.0	5733.0
	5.6	**2.8**	4.2	4.2	4.2	4.2	**2.8**
TOX-171		17244.0	17244.0	22992.0	17244.0	17244.0	5847.6
	9.9	**1.2**	1.8	**1.2**	1.8	1.8	8.2
Brain-Tumor1		17760.0	17760.0	23680.0	17760.0	17760.0	6085.4
	13.3	**8.9**	11.1	11.1	**8.9**	**8.9**	11.1
ORL10P		30912.0	30912.0	41216.0	30912.0	30912.0	19385.4
	1.0	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Prostate-Tumor		31527.0	31527.0	42035.4	31520.0	31527.0	11394.0
	10.8	**8.8**	**8.8**	**8.8**	**8.8**	**8.8**	**8.8**
Leukemia2		33675.0	33675.0	33675.0	33675.0	33675.0	12431.4
	4.2	**2.8**	**2.8**	**2.8**	**2.8**	**2.8**	4.2
GLI-85		66849.0	66849.0	66849.0	66849.0	66849.0	25118.7
	14.1	10.6	**8.2**	**8.2**	9.4	9.4	**8.2**

and $\varepsilon = 0.25$. Table 3 shows these values for the linear SVM classifier.

For all datasets, the use of a FD technique leads to equal or better results as compared to the use of the original features. R-LBG leads to the best test set error rate in 7 out of 14 tests. Moreover, in the majority of these tests, R-LBG computes fewer discretization intervals, as compared to the other techniques. This difference on the discretization intervals is most noticed in the higher-dimensional datasets.

For the test set error rates in Table 3, the Friedman test has reported a p-value of $0.04287 < 0.05$, stating that these results have statistical significance.

4.2.2 Supervised Discretization

We now assess the behavior of our methods for supervised FD. The *MID fixed*, *MID variable*, and R-LBG methods, with $q = 3$ bits, are compared against the four supervised FD techniques described in Subsection 2.2. R-LBG uses MI as the relevance measure. For both R-LBG and *MID variable* we use $\varepsilon = 0.1$. Table 4 reports linear SVM results for a 10-fold CV and the average number of bits per instance.

Again, the use of a FD technique improves on the test set error rate, as compared to the use of the original features, for the 14 datasets considered in this experimental evaluation. The CAIM and CACC algorithms are not suitable for the higher-dimensional datasets, since they both take a prohibitive running time (hours) as compared to other approaches. One of our approaches usually attains the best result, except in three cases (two in which IEMV is the best and one where CAIM attains the best result). Within our approaches, the R-LBG and *MID variable* methods attain the best results, which suggests:

i) the adequacy of MI between features and class labels for discretization purposes;

ii) a variable number of bits per feature is preferable to the use of a fixed number, regarding both complexity and generalization error.

The p-value of the Friedman test for the error rates of Table 4 (excluding both CAIM and CACC) is $0.00461 < 0.05$, showing statistical significance.

Table 4: Evaluation of supervised FD. For each dataset, the first row presents the average total number of bits per instance and the second row has the average test set error rate (%), using a 10-fold CV for the linear SVM classifier. The best error rate is in bold face. We have used $q = 3$ bit/feature, $@rel = MI$, and $\varepsilon = 0.1$ for both R-LBG and MID variable.

		Existing Methods				**Proposed Methods**		
Dataset	**No FD**	**IEM**	**IEMV**	**CAIM**	**CACC**	**R-LBG**	**MID fixed**	**MID variable**
Wine		19.8	21.3	39.0	39.0	27.4	39.0	23.0
	5.1	2.2	**1.7**	2.8	2.8	3.9	2.8	**1.7**
Hepatitis		45.5	42.9	34.6	43.3	52.3	57.0	50.0
	19.4	22.6	21.3	17.4	20.0	21.9	**16.8**	19.4
Ionosphere		84.4	83.0	65.0	96.9	85.0	99.0	73.3
	12.5	11.7	9.4	10.5	12.5	10.8	7.7	**6.0**
Colon		11341.3	11089.3	4000.0	7431.2	5765.4	6000.0	5121.5
	17.7	**16.1**	**16.1**	**16.1**	**16.1**	17.7	19.4	**16.1**
SRBCT		12476.1	9691.4	6924.0	6924.0	6248.1	6924.0	6553.5
	0.0	**0.0**	**0.0**	1.2	**0.0**	**0.0**	**0.0**	**0.0**
AR10P		12903.6	7138.4	7200.0	7200.0	7145.6	7200.0	7266.3
	0.8	2.3	20.0	0.8	**0.0**	**0.0**	**0.0**	**0.0**
PIE10P		9103.4	5264.0	7260.0	7260.0	7077.3	7260.0	7154.7
	0.0	**0.0**	1.9	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Leukemia1		28435.3	26034.7	*	*	14656.1	15981.0	14278.0
	5.6	40.3	56.9	*	*	4.2	**2.8**	**2.8**
TOX-171		36134.8	28253.7	*	*	15725.6	17244.0	15824.2
	15.2	5.8	**2.9**	*	*	**2.9**	3.5	4.7
Brain-Tumor1		32808.3	27133.5	*	*	15674.3	17760.0	16343.6
	14.4	20.0	35.6	*	*	11.1	10.0	**8.9**
ORL10P		26475.7	24176.8	*	*	30863.0	30912.0	30786.9
	2.0	9.0	**1.0**	*	*	2.0	2.0	2.0
Prostate-Tumor		54395.6	51964.7	*	*	30695.0	31527.0	28506.3
	8.8	12.7	11.8	*	*	**5.9**	6.9	7.8
Leukemia2		48380.1	40447.3	*	*	28857.3	33675.0	28670.8
	5.6	8.3	6.9	*	*	**2.8**	**2.8**	**2.8**
GLI-85		135866.9	130689.1	*	*	64065.0	66849.0	58633.4
	10.6	11.8	12.9	*	*	**9.4**	**9.4**	10.6

4.3 Summary and Discussion

In the comparison with state-of-the-art techniques for unsupervised and supervised FD, our methods improve on the test set error rate, in the majority of the tests. Our proposals scale well for high-dimensional data, contrary to other approaches. The adequacy of MI between features and class labels for discretization has been shown. Discretizing with a variable number of bits per feature is preferable in terms of both complexity and generalization error, as compared to a fixed number of bits, since it allows to attain the best trade-off between the number of discretization intervals and the generalization error. Our methods show stability regarding the variation of their input parameters q and ε. An excessive value on q does not lead to an excessive number of discretization intervals, since both R-LBG and *MID variable* stop allocating bits, whenever the relevance criterion is not fulfilled. The choice of ε is set by the maximum possible value of the MI between the feature and the class label; setting ε from 0 to 10% of the maximum MI seems adequate for different kinds of data.

5 CONCLUSIONS

In many machine learning and data mining tasks, FD is a useful pre-processing step. Even in cases where FD is not required, it may be used to attain compact and adequate representations of the data. Often, the use of FD techniques improves on the generalization error and lowers the training time.

In this paper, we have proposed two FD techniques. The first one is based on the unsupervised Linde-Buzo-Gray algorithm with a relevance criterion to monitor the discretization process. Depending on the relevance criterion, this technique works in unsupervised, supervised, or semi-supervised problems. The second technique is supervised and is based on mutual information maximization between the discrete feature and the class label. It uses a recursive approach that finds the optimal cut points in the mutual information sense, being able to work with a fixed or a variable number of bits per feature.

The experimental evaluation of these techniques was carried out on publicly available binary and multi-class, medium and high-dimensional datasets,

with different types of data. Under a supervised learning evaluation task with different classifiers, both techniques have shown improvement as compared to unsupervised and supervised FD approaches. The first technique has obtained similar or better results, when compared to its unsupervised counterparts. For the supervised FD tests, the second technique has proved to be more effective regarding the number of discretization intervals and the generalization error. For both techniques, the classifiers learned on discrete features usually attain better accuracy than those learned on the original ones. Both techniques scale well for high-dimensional and multi-class problems.

As future work, we will explore the embedding of feature selection in the discretization process.

REFERENCES

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5:537–550.

Brown, G., Pocock, A., Zhao, M., and Luján, M. (2012). Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *J. Machine Learning Research*, 13:27–66.

Chiu, D., Wong, A., and Cheung, B. (1991). Information discovery through hierarchical maximum entropy discretization and synthesis. In *Proc. of the Knowledge Discovery in Databases*, pages 125–140.

Cover, T. and Thomas, J. (1991). *Elements of Information Theory*. John Wiley & Sons.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.

Dougherty, J., Kohavi, R., and Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. In *Int. Conf. Mac. Learn. (ICML)*, pages 194–202.

Duin, R., Juszczak, P., Paclik, P., Pekalska, E., Ridder, D., Tax, D., and Verzakov, S. (2007). PRTools4.1: A Matlab Toolbox for Pattern Recognition. Technical report, Delft Univ. Technology.

Fayyad, U. and Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. Int. Joint Conf. on Art. Intell. (IJCAI)*, pages 1022–1027.

Ferreira, A. and Figueiredo, M. (2012). An unsupervised approach to feature discretization and selection. *Pattern Recognition*, 45:3048–3060.

Frank, A. and Asuncion, A. (2010). UCI machine learning repository, available at http://archive.ics.uci.edu/ml.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92.

Garcia, S. and Herrera, F. (2008). An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694.

Hellman, M. (1970). Probability of error, equivocation, and the Chernoff bound. *IEEE Transactions on Information Theory*, 16(4):368–372.

Jin, R., Breitbart, Y., and Muoh, C. (2009). Data discretization unification. *Know. Inf. Systems*, 19(1):1–29.

Kononenko, I. (1995). On biases in estimating multi-valued attributes. In *Proc. Int. Joint Conf. on Art. Intell. (IJCAI)*, pages 1034–1040.

Kotsiantis, S. and Kanellopoulos, D. (2006). Discretization techniques: A recent survey. *GESTS Int. Trans. on Computer Science and Engineering*, 32(1).

Kurgan, L. and Cios, K. (2004). CAIM discretization algorithm. *IEEE Trans. on Know. and Data Engineering*, 16(2):145–153.

Linde, Y., Buzo, A., and Gray, R. (1980). An algorithm for vector quantizer design. *IEEE Trans. on Communications*, 28:84–94.

Liu, H., Hussain, F., Tan, C., and Dash, M. (2002). Discretization: An Enabling Technique. *Data Mining and Knowledge Discovery*, 6(4):393–423.

Santhi, N. and Vardy, A. (2006). On an improvement over Rényi's equivocation bound. In *44-th Annual Allerton Conference on Communication, Control, and Computing*.

Statnikov, A., Tsamardinos, I., Dosbayev, Y., and Aliferis, C. F. (2005). GEMS: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. *Int J Med Inf.*, 74(7-8):491–503.

Tsai, C.-J., Lee, C.-I., and Yang, W.-P. (2008). A discretization algorithm based on class-attribute contingency coefficient. *Inf. Sci.*, 178:714–731.

Witten, I. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, Morgan Kauffmann.

Yang, Y. and Webb, G. (2001). Proportional k-interval discretization for naïve-Bayes classifiers. In *12th Eur. Conf. on Machine Learning*, (ECML), pages 564–575.

Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., and Liu, H. (2010). Advancing feature selection research - ASU feature selection repository. Technical report, Arizona State University.