A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning

Article *in* IEEE Transactions on Knowledge and Data Engineering · February 2013

DOI: 10.1109/TKDE.2012.35

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning

Salvador García, Julián Luengo, José A. Sáez, Victoria López, and Francisco Herrera

Abstract—Discretization is an essential preprocessing technique used in many knowledge discovery and data mining tasks. Its main goal is to transform a set of continuous attributes into discrete ones, by associating categorical values to intervals and thus transforming quantitative data into qualitative data. In this manner, symbolic data mining algorithms can be applied over continuous data and the representation of information is simplified, making it more concise and specific. The literature provides numerous proposals of discretization and some attempts to categorize them into a taxonomy can be found. However, in previous papers, there is a lack of consensus in the definition of the properties and no formal categorization has been established yet, which may be confusing for practitioners. Furthermore, only a small set of discretizers have been widely considered, while many other methods have gone unnoticed. With the intention of alleviating these problems, this paper provides a survey of discretization methods proposed in the literature from a theoretical and empirical perspective. From the theoretical perspective, we develop a taxonomy based on the main properties pointed out in previous research, unifying the notation and including all the known methods up to date. Empirically, we conduct an experimental study in supervised classification involving the most representative and newest discretizers, different types of classifiers and a large number of data sets. The results of their performances measured in terms of accuracy, number of intervals and inconsistency have been verified by means of nonparametric statistical tests. Additionally, a set of discretizers are highlighted as the best performing ones.

Index Terms—Discretization, continuous attributes, decision trees, taxonomy, data preprocessing, data mining, classification.

1 INTRODUCTION

KNOWLEDGE extraction and Data Mining (DM) are important methodologies to be performed over different databases which contain data relevant to a real application [1], [2]. Both processes often require some previous tasks such as problem comprehension, data comprehension or data preprocessing in order to guarantee the successful application of a DM algorithm to real data [3], [4]. Data preprocessing [5] is a crucial research topic in the DM field and it includes several processes of data transformation, cleaning and data reduction. Discretization, as one of the basic data reduction techniques, has received increasing research attention in recent years [6] and has become one of the preprocessing techniques most broadly used in DM.

The discretization process transforms quantitative data into qualitative data, that is, numerical attributes into discrete or nominal attributes with a finite number of intervals, obtaining a non-overlapping partition of a continuous domain. An association between each interval with a numerical discrete value is then established. In practice, discretization can be viewed as a data reduction method since it maps data from a huge spectrum of numeric values to a greatly reduced subset of discrete values. Once the discretization is performed, the data can be treated as nominal data during any induction or deduction DM process. Many existing DM algorithms are designed to only learn in categorical data, using nominal attributes, while real-world applications usually involve continuous features. Those numerical features have to be discretized before using such algorithms.

In supervised learning, and specifically classification, the topic of this survey, we can define the discretization as follows. Assuming a data set consisting of N examples and C target classes, a discretization algorithm would discretize the continuous attribute A in this data set into m discrete intervals $D = \{[d_0, d_1], (d_1, d_2], \ldots, (d_{m-1}, d_m]\}$, where d_0 is the minimal value, d_m is the maximal value and $d_i < d_{i+i}$, for $i = 0, 1, \ldots, m-1$. Such a discrete result D is called a discretization scheme on attribute A and $P = \{d_1, d_2, \ldots, d_{m-1}\}$ is the set of cut points of attribute A.

The necessity of using discretization on data can be caused by several factors. Many DM algorithms are primarily oriented to handle nominal attributes [7], [6], [8], or may even only deal with discrete attributes. For instance, three of the ten methods considered as the top ten in DM [9] require an embedded or an external

- *S. García is with the Department of Computer Science, University of Jaén, 23071, Jaén, Spain.*
  E-mail: sglopez@ujaen.es
- *J. Luengo is with the Department of Civil Engineering, LSI, University of Burgos, 09006, Burgos, Spain.*
  E-mail: jluengo@ubu.es
- *J.A. Sáez, V. López and F. Herrera are with the Department of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), University of Granada, 18071 Granada, Spain.*
  E-mails: smja@decsai.ugr.es, vlopez@decsai.ugr.es, herrera@decsai.ugr.es

Digital Object Indentifier 10.1109/TKDE.2012.35 1041-4347/12/$31.00 © 2012 IEEE

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

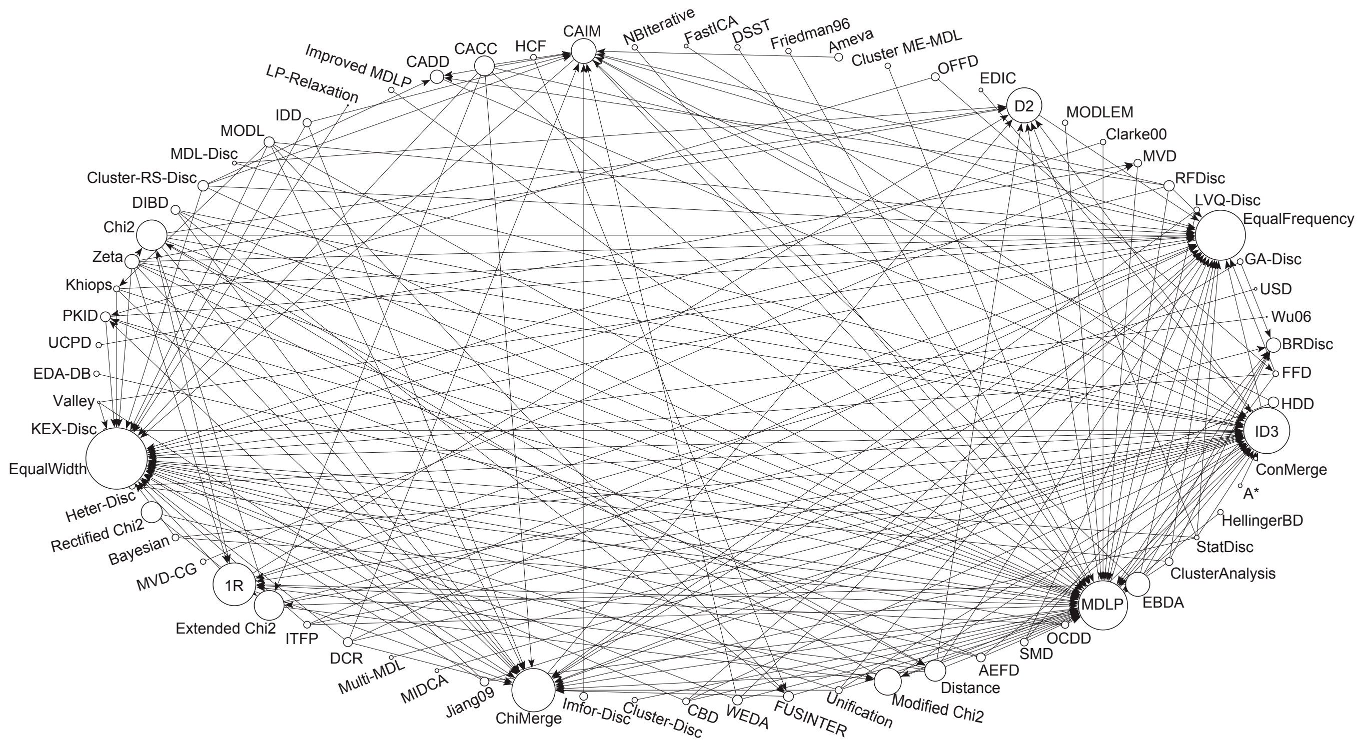


Fig. 1: Comparison Network of discretizers. Later, the methods will be defined in Table 1.

discretization of data: C4.5 [10], Apriori [11] and Naive Bayes [12], [13]. Even with algorithms that are able to deal with continuous data, learning is less efficient and effective [14], [5], [4]. Other advantages derived from discretization are the reduction and the simplification of data, making the learning faster and yielding more accurate, compact and shorter results; and noise possibly present in the data is reduced. For both researchers and practitioners, discrete attributes are easier to understand, use, and explain [6]. Nevertheless, any discretization process generally leads to a loss of information, making the minimization of such information loss the main goal of a discretizer.

Obtaining the optimal discretization is NP-complete [15]. A vast number of discretization techniques can be found in the literature. It is obvious that when dealing with a concrete problem or data set, the choice of a discretizer will condition the success of the posterior learning task in accuracy, simplicity of the model, etc. Different heuristic approaches have been proposed for discretization, for example, approaches based on information entropy [16], [7], statistical χ^2 test [17], [18], likelihood [19], [20], rough sets [21], [22], etc. Other criteria have been used in order to provide a classification of discretizers, such as univariate/multivariate, supervised/unsupervised, top-down/bottoum-up, global/local, static/dynamic and more. All these criteria are the basis of the taxonomies already proposed and they will be deeply elaborated upon in this paper. The identification of the best discretizer for each situation is a very difficult task to carry out, but performing exhaustive experiments considering a representative set of learners and discretizers could help to decide the best choice.

Some reviews of discretization techniques can be found in the literature [7], [6], [23], [8]. However, the characteristics of the methods are not studied completely, many discretizers, even classic ones, are not mentioned, and the notation used for categorization is not unified. For example, in [7], the static/dynamic distinction is different from that used in [6] and the global/local property is usually confused with the univariate/multivariate property [24], [25], [26]. Subsequent papers include one notation or other, depending on the initial discretization study referenced by them: [7], [24] or [6].

In spite of the wealth of literature, and apart from the absence of a complete categorization of discretizers using a unified notation, it can be observed that, there are few attempts to empirically compare them. In this way, the algorithms proposed are usually compared with a subset of the complete family of discretizers and, in most of the studies, no rigorous empirical analysis has been carried out. Furthermore, many new methods have been proposed in recent years and they are going unnoticed with respect to the discretizers reviewed in well-known surveys [7], [6]. Figure 1 illustrates a comparison network where each node corresponds to a discretization algorithm and a directed vertex between two nodes indicates that the algorithm of the start node has been compared with the algorithm of the end node. The direction of the arrows is always from the newest method to the oldest, but it does not influence the results. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

size of the node is correlated to the number of input and output vertices. We can see that most of the discretizers are represented by small nodes and that the graph is far from being complete, which has prompted the present paper. The most compared techniques are EqualWidth, EqualFrequency, MDLP [16], ID3 [10], ChiMerge [17], 1R [27], D2 [28] and Chi2 [18].

These reasons motivate the global purpose of this paper, which can be divided into three main objectives:

- To propose a complete taxonomy based on the main properties observed in the discretization methods. The taxonomy will allow us to characterize their advantages and drawbacks in order to choose a discretizer from a theoretical point of view.
- To make an empirical study analyzing the most representative and newest discretizers in terms of the number of intervals obtained and inconsistency level of the data.
- Finally, to relate the best discretizers for a set of representative DM models using two metrics to measure the predictive classification success.

The experimental study will include a statistical analysis based on nonparametric tests. We will conduct experiments involving a total of 30 discretizers; 6 classification methods belonging to lazy, rules, decision trees and bayesian learning families; and 40 data sets. The experimental evaluation does not correspond to an exhaustive search for the best parameters for each discretizer, given the data at hand. Then, its main focus is to properly relate a subset of best performing discretizers to each classic classifier using a general configuration for them.

This paper is organized as follows. The related and advanced work on discretization is provided in Section 2. Section 3 presents the discretizers reviewed, their properties and the taxonomy proposed. Section 4 describes the experimental framework, examines the results obtained in the empirical study and presents a discussion of them. Section 5 concludes the paper. Finally, we must point out that the paper has an associated web site http://sci2s.ugr.es/discretization which collects additional information regarding discretizers involved in the experiments such as implementations and detailed experimental results.

2 Related and Advanced Work

Research in improving and analyzing discretization is common and in high demand currently. Discretization is a promising technique to obtain the hoped results, depending on the DM task, which justifies its relationship to other methods and problems. This section provides a brief summary of topics closely related to discretization from a theoretical and practical point of view and describes other works and future trends which have been studied in the last few years.

- *Discretization Specific Analysis:* Susmaga proposed an analysis method for discretizers based on binarization of continuous attributes and rough sets measures [29]. He emphasized that his analysis method is useful for detecting redundancy in discretization and the set of cut points which can be removed without decreasing the performance. Also, it can be applied to improve existing discretization approaches.
- *Optimal Multisplitting:* Elomaa and Rousu characterized some fundamental properties for using some classic evaluation functions in supervised univariate discretization. They analyzed entropy, information gain, gain ratio, training set error, gini index and normalized distance measure, concluding that they are suitable for use in the optimal multisplitting of an attribute [30]. They also developed an optimal algorithm for performing this multisplitting process and devised two techniques [31], [32] to speed it up.
- *Discretization of Continuous Labels:* Two possible approaches have been used in the conversion of a continuous supervised learning (regression problem) into a nominal supervised learning (classification problem). The first one is simply to use regression tree algorithms, such as CART [33]. The second consists of applying discretization to the output attribute, either statically [34] or in a dynamic fashion [35].
- *Fuzzy Discretization:* Extensive research has been carried out around the definition of linguistic terms that divide the domain attribute into fuzzy regions [36]. Fuzzy discretization is characterized by membership value, group or interval number and affinity corresponding to an attribute value, unlike crisp discretization which only considers the interval number [37].
- *Cost-Sensitive Discretization:* The objective of cost-based discretization is to take into account the cost of making errors instead of just minimizing the total sum of errors [38]. It is related to problems of imbalanced or cost-sensitive classification [39], [40].
- *Semi-Supervised Discretization:* A first attempt to discretize data in semi-supervised classification problems has been devised in [41], showing that it is asymptotically equivalent to the supervised approach.

The research mentioned in this section is out of the scope of this survey. We point out that the main objective of this paper is to give a wide overview of the discretization methods found in the literature and to conduct an exhaustive experimental comparison of the most relevant discretizers without considering external and advanced factors such as those mentioned above or derived problems from classic supervised classification.

3 Discretization: Background and Techniques

This section presents a taxonomy of discretization methods and the criteria used for building it. First, in Subsection 3.1, the main characteristics which will define the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

categories of the taxonomy will be outlined. Then, in Subsection 3.2, we enumerate the discretization methods proposed in the literature we will consider by using their complete and abbreviated name together with the associated reference. Finally, we present the taxonomy.

3.1 Common Properties of Discretization Methods

This section provides a framework for the discussion of the discretizers presented in the next subsection. The issues discussed include several properties involved in the structure of the taxonomy, since they are exclusive to the operation of the discretizer. Other, less critical issues such as parametric properties or stopping conditions will be presented although they are not involved in the taxonomy. Finally, some criteria will also be pointed out in order to compare discretization methods.

3.1.1 Main Characteristics of a Discretizer

In [6], [7], [8], various axis have been described in order to make a categorization of discretization methods. We review and explain them in this section, emphasizing the main aspects and relations found among them and unifying the notation. The taxonomy proposed will be based on these characteristics:

- *Static vs. Dynamic:* This characteristic refers to the moment and independence which the discretizer operates in relation with the learner. A dynamic discretizer acts when the learner is building the model, thus they can only access partial information (local property, see later) embedded in the learner itself, yielding compact and accurate results in conjuntion with the associated learner. Otherwise, a static discretizer proceeds prior to the learning task and it is independent from the learning algorithm [6]. Almost all known discretizers are static, due to the fact that most of the dynamic discretizers are really subparts or stages of DM algorithms when dealing with numerical data [42]. Some examples of well-known dynamic techniques are ID3 discretizer [10] and ITFP [43].
- *Univariate vs. Multivariate:* Multivariate techniques, also known as 2D discretization [44], simultaneously consider all attributes to define the initial set of cut points or to decide the best cut point altogether. They can also discretize one attribute at a time when studying the interactions with other attributes, exploiting high order relationships. By contrast, univariate discretizers only work with a single attribute at a time, once an order among attributes has been established, and the resulting discretization scheme in each attribute remains unchanged in later stages. Interest has recently arisen in developing multivariate discretizers since they are very influential in deductive learning [45], [46] and in complex classification problems where high interactions among multiple attributes exist, which univariate discretizers might obviate [47], [48].
- *Supervised vs. Unsupervised:* Unsupervised discretizers do not consider the class label whereas supervised ones do. The manner in which the latter consider the class attribute depends on the interaction between input attributes and class labels, and the heuristic measures used to determine the best cut points (entropy, interdependence, etc.). Most discretizers proposed in the literature are supervised and theoretically, using class information, should automatically determine the best number of intervals for each attribute. If a discretizer is unsupervised, it does not mean that it cannot be applied over supervised tasks. However, a supervised discretizer can only be applied over supervised DM problems. Representative unsupervised discretizers are EqualWidth and EqualFrequency [49], PKID and FFD [12] and MVD [45].
- *Splitting vs. Merging:* This refers to the procedure used to create or define new intervals. Splitting methods establish a cut point among all the possible boundary points and divide the domain into two intervals. By contrast, merging methods start with a pre-defined partition and remove a candidate cut point to mix both adjacent intervals. These properties are highly related to *Top-Down* and *Bottom-up* respectively (explained in the next section). The idea behind them is very similar, except that top-down or bottom-up discretizers assume that the process is incremental (described later), according to a hierarchical discretization construction. In fact, there can be discretizers whose operation is based on splitting or merging more than one interval at a time [50], [51]. Also, some discretizers can be considered *hybrid* due to the fact that they can alternate splits with merges in running time [52], [53].
- *Global vs. Local:* To make a decision, a discretizer can either require all available data in the attribute or use only partial information.. A discretizer is said to be local when it only makes the partition decision based on local information. Examples of widely used local techniques are MDLP [16] and ID3 [10]. Few discretizers are local, except some based on top-down partition and all the dynamic techniques. In a top-down process, some algorithms follow the divide-and-conquer scheme and when a split is found, the data is recursively divided, restricting access to partial data. Regarding dynamic discretizers, they find the cut points in internal operations of a DM algorithm, so they never gain access to the full data set.
- *Direct vs. Incremental:* Direct discretizers divide the range into k intervals simultaneously, requiring an additional criterion to determine the value of k. They do not only include one-step discretization methods, but also discretizers which perform several stages in their operation, selecting more than a single cut point at every step. By contrast, incremen-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

tal methods begin with a simple discretization and pass through an improvement process, requiring an additional criterion to know when to stop it. At each step, they find the best candidate boundary to be used as a cut point and afterwards the rest of the decisions are made accordingly. Incremental discretizers are also known as hierarchical discretizers [23]. Both types of discretizers are widespread in the literature, although there is usually a more defined relationship between incremental and supervised ones.

- *Evaluation Measure:* This is the metric used by the discretizer to compare two candidate schemes and decide which is more suitable to be used. We consider five main families of evaluation measures:
  - *Information:* This family includes *entropy* as the most used evaluation measure in discretization (MDLP [16], ID3 [10], FUSINTER [54]) and other derived information theory measures such as the *Gini index* [55].
  - *Statistical:* Statistical evaluation involves the measurement of dependency/correlation among attributes (Zeta [56], ChiMerge [17], Chi2 [18]), probability and bayesian properties [19] (MODL [20]), interdependency [57], contingency coefficient [58], etc.
  - *Rough Sets:* This group is composed of methods that evaluate the discretization schemes by using rough set measures and properties [21], such as lower and upper approximations, class separability, etc.
  - *Wrapper:* This collection comprises methods that rely on the error provided by a classifier that is run for each evaluation. The classifier can be a very simple one, such as a majority class voting classifier (Valley [59]) or general classifiers such as Naive Bayes (NBIterative [60]).
  - *Binning:* This category refers to the absence of an evaluation measure. It is the simplest method to discretize an attribute by creating a specified number of bins. Each bin is defined a priori and allocates a specified number of values per attribute. Widely used binning methods are EqualWidth and EqualFrequency.

3.1.2 Other Properties

We can remark other properties related to discretization. They also influence the operation and results obtained by a discretizer, but to a lower degree than the characteristics explained above. Furthermore, some of them present a large variety of categorizations and may harm the interpretability of the taxonomy.

- *Parametric vs. NonParametric:* This property refers to the automatic determination of the number of intervals for each attribute by the discretizer. A nonparametric discretizer computes the appropriate number of intervals for each attribute considering a trade-off between the loss of information or consistency and obtaining the lowest number of them. A parametric discretizer requires a maximum number of intervals desired to be fixed by the user. Examples of non-parametric discretizers are MDLP [16] and CAIM [57]. Examples of parametric ones are ChiMerge [17] and CADD [52].
- *Top-Down vs. Bottom Up:* This property is only observed in incremental discretizers. Top-Down methods begin with an empty discretization. Its improvement process is simply to add a new cutpoint to the discretization. On the other hand, Bottom-Up methods begin with a discretization that contains all the possible cutpoints. Its improvement process consists of iteratively merging two intervals, removing a cut point. A classic Top-Down method is MDLP [16] and a well-known Bottom-Up method is ChiMerge [17].
- *Stopping Condition:* This is related to the mechanism used to stop the discretization process and must be specified in nonparametric approaches. Well-known stopping criteria are the Minimum Description Length measure [16], confidence thresholds [17], or inconsistency ratios [24].
- *Disjoint vs. Non-Disjoint:* Disjoint methods discretize the value range of the attribute into disassociated intervals, without overlapping, whereas non-disjoint methods dicsretize the value range into intervals that can overlap. The methods reviewed in this paper are disjoint, while fuzzy discretization is usually non-disjoint [36].
- *Ordinal vs. Nominal:* Ordinal discretization transforms quantitative data intro ordinal qualitative data whereas nominal discretization transforms it into nominal qualitative data, discarding the information about order. Ordinal discretizers are less common, not usually considered classic discretizers [113].

3.1.3 Criteria to Compare Discretization Methods

When comparing discretization methods, there are a number of criteria that can be used to evaluate the relative strengths and weaknesses of each algorithm. These include the number of intervals, inconsistency, predictive classification rate and time requirements

- *Number of Intervals:* A desirable feature for practical discretization is that discretized attributes have as few values as possible, since a large number of intervals may make the learning slow and ineffective. [28].
- *Inconsistency:* A supervision-based measure used to compute the number of unavoidable errors produced in the data set. An unavoidable error is one associated to two examples with the same values for input attributes and different class labels. In general, data sets with continuous attributes are consistent, but when a discretization scheme is applied over the data, an inconsistent data set may be obtained. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

TABLE 1: Discretizers

Complete name	Abbr. name	Reference	Complete name	Abbr. name	Reference
Equal Width Discretizer	**EqualWidth**	[61]	Self Organizing Map Discretizer	SOM-Disc	[62]
Equal Frequency Discretizer	**EqualFrequency**	[61]	Optimal Class-Dependent Discretizer	OCDD	[26]
No name specified	Chou91	[63]	*No name specified*	Butterworth04	[64]
Adaptive Quantizer	AQ	[65]	*No name specified*	Zhang04	[22]
Discretizer 2	D2	[28]	**Khiops**	**Khiops**	[66]
ChiMerge	**ChiMerge**	[17]	**Class-Attribute Interdependence Maximization**	**CAIM**	[57]
One-Rule Discretizer	**1R**	[27]	**Extended Chi2**	**Extended Chi2**	[67]
Iterative Dichotomizer 3 Discretizer	**ID3**	[10]	**Heterogeneity Discretizer**	**Heter-Disc**	[68]
Minimum Description Length Principle	**MDLP**	[16]	**Unsupervised Correlation Preserving Discretizer**	**UCPD**	[44]
Valley	Valley	[59], [69]	*No name specified*	Multi-MDL	[47]
Class-Attribute Dependent Discretizer	**CADD**	[52]	Difference Similitude Set Theory Discretizer	DSST	[70]
ReliefF Discretizer	ReliefF	[71]	Multivariate Interdependent Discretizer	MIDCA	[72]
Class-driven Statistical Discretizer	StatDisc	[14]	**MODL**	**MODL**	[20]
No name specified	NBIterative	[60]	Information Theoretic Fuzzy Partitioning	ITFP	[43]
Boolean Reasoning Discretizer	BRDisc	[21]	*No name specified*	Wu06	[73]
Minimum Description Length Discretizer	MDL-Disc	[74]	Fast Independent Component Analysis	FastICA	[75]
Bayesian Discretizer	**Bayesian**	[19]	Linear Program Relaxation	LP-Relaxation	[76]
No name specified	Friedman96	[77]	**Hellinger-Based Discretizer**	**HellingerBD**	[50]
Cluster Analysis Discretizer	**ClusterAnalysis**	[24]	**Distribution Index-Based Discretizer**	**DIBD**	[78]
Zeta	**Zeta**	[56]	Wrapper Estimation of Distribution Algorithm	WEDA	[53]
Distance-based Discretizer	**Distance**	[79]	Clustering + Rought Sets Discretizer	Cluster-RS-Disc	[25]
Finite Mixture Model Discretizer	FMM	[80]	**Interval Distance Discretizer**	**IDD**	[51]
Chi2	**Chi2**	[18]	**Class-Attribute Contingency Coefficient**	**CACC**	[58]
No name specified	FischerExt	[81]	Rectified Chi2	Rectified Chi2	[82]
Contextual Merit Numerical Feature Discretizer	CM-NFD	[83]	**Ameva**	**Ameva**	[84]
Concurrent Merger	ConMerge	[85]	Unification	Unification	[55]
Knowledge EXplorer Discretizer	KEX-Disc	[86]	Multiple Scanning Discretizer	MultipleScan	[87]
LVQ-based Discretization	LVQ-Disc	[88]	Optimal Flexible Frequency Discretizer	OFFD	[89]
No name specified	Multi-Bayesian	[90]	**Proportional Discretizer**	**PKID**	[12]
No name specified	A*	[91]	**Fixed Frequency Discretizer**	**FFD**	[12]
FUSINTER	**FUSINTER**	[54]	Discretization Class intervals Reduce	DCR	[92]
Cluster-based Discretizer	Cluster-Disc	[93]	MVD-CG	MVD-CG	[94]
Entropy-based Discretization According to Distribution of Boundary points	EDA-DB	[95]	Approximate Equal Frequency Discretizer	AEFD	[96]
No name specified	Clarke00	[97]	*No name specified*	Jiang09	[96]
Relative Unsupervised Discretizer	RUDE	[98]	Random Forest Discretizer	RFDisc	[99]
Multivariate Discretization	**MVD**	[45]	Supervised Multivariate Discretizer	SMD	[100]
Modified Learning from Examples Module	MODLEM	[101]	Clustering Based Discretization	CBD	[46]
Modified Chi2	**Modified Chi2**	[102]	Improved MDLP	Improved MDLP	[103]
HyperCluster Finder	HCF	[104]	Imfor-Disc	Imfor-Disc	[105]
Entropy-based Discretization with Inconsistency Checking	EDIC	[49]	Clustering ME-MDL	Cluster ME-MDL	[106]
Unparametrized Supervised Discretizer	**USD**	[107]	Effective Bottom-up Discretizer	EBDA	[108]
Rough Set Discretizer	RS-Disc	[109]	Contextual Discretizer	Contextual-Disc	[110]
Rough Set Genetic Algorithm Discretizer	RS-GA-Disc	[111]	**Hypercube Division Discretizer**	**HDD**	[48]
Genetic Algorithm Discretizer	GA-Disc	[112]			

desired inconsistency level that a discretizer should obtain is 0.0.

- *Predictive Classification Rate:* A successful algorithm will often be able to discretize the training set without significantly reducing the prediction capability of learners in test data which are prepared to treat numerical data.
- *Time requirements:* A static discretization process is carried out just once on a training set, so it does not seem to be a very important evaluation method. However, if the discretization phase takes too long it can become impractical for real applications. In dynamic discretization, the operation is repeated many times as the learner requires, so it should be performed efficiently.

3.2 Discretization Methods and Taxonomy

At the time of writting, more than 80 discretization methods have been proposed in the literature. This section is devoted to enumerating and designating them according to a standard followed in this paper. We have used 30 discretizers in the experimental study, those that we have identified as the most relevant ones. For more details on their descriptions, the reader can visit the URL associated to the KEEL project[1]. Additionaly, implementations of these algorithms in Java can be found in KEEL software [114], [115].

Table 1 presents an enumeration of discretizers reviewed in this paper. The complete name, abbreviation and reference are provided for each one. This paper does

1. http://www.keel.es

not collect the descriptions of the discretizers due to space restrictions. Instead, we recommend that readers consult the original references to understand the complete operation of the discretizers of interest. Discretizers used in the experimental study are depicted in bold. The ID3 discretizer used in the study is a static version of the well-known discretizer embedded in C4.5.

The properties studied above can be used to categorize the discretizers proposed in the literature. The seven characteristics studied allows us to present the taxonomy of discretization methods following an established order. All techniques enumerated in Table 1 are collected in the taxonomy drawn in Figure 2. It illustrates the categorization following a hierarchy based on this order: static/dynamic, univariate/multivariate, supervised/unsupervised, splitting/merging/hybrid, global/local, direct/incremental and evaluation measure. The rationale behind the choice of this order is to achieve a clear representation of the taxonomy.

The proposed taxonomy assists us in the organization of many discretization methods so that we can classify them into categories and analyze their behavior. Also, we can highlight other aspects in which the taxonomy can be useful. For example, it provides a snapshot of existing methods and relations or similarities among them. It also depicts the size of the families, the work done in each one and what currently is missing. Finally, it provides a general overview on the state-of-the-art in discretization for researchers/practitioners who are starting in this topic or need to discretize data in real applications.

4 Experimental Framework, Empirical Study and Analysis of Results

This section presents the experimental framework followed in this paper, together with the results collected and discussions on them. Subsection 4.1 will describe the complete experimental set up. Then, we offer the study and analysis of the results obtained over the data sets used in Subsection 4.2.

4.1 Experimental Set Up

The goal of this section is to show all the properties and issues related to the experimental study. We specify the data sets, validation procedure, classifiers used, parameters of the classifiers and discretizers, and performance metrics. The statistical tests used to contrast the results are also briefly commented at the end of this section.

The performance of discretization algorithms is analyzed by using 40 data sets taken from the UCI Machine Learning Database Repository [116] and KEEL data set repository [115] [2]. The main characteristics of these data sets are summarized in Table 2. For each data set, the name, number of examples, number of attributes (numeric and nominal) and number of classes are defined.

2. http://www.keel.es/datasets.php

TABLE 2: Summary description for classification data sets

Data Set	#Ex.	#Atts.	#Num.	#Nom.	#Cl.
abalone	4,174	8	7	1	28
appendicitis	106	7	7	0	2
australian	690	14	8	6	2
autos	205	25	15	10	6
balance	625	4	4	0	3
banana	5,300	2	2	0	2
bands	539	19	19	0	2
bupa	345	6	6	0	2
cleveland	303	13	13	0	5
contraceptive	1,473	9	9	0	3
crx	690	15	6	9	2
dermatology	366	34	34	0	6
ecoli	336	7	7	0	8
flare-solar	1066	9	9	0	2
glass	214	9	9	0	7
haberman	306	3	3	0	2
hayes	160	4	4	0	3
heart	270	13	13	0	2
hepatitis	155	19	19	0	2
iris	150	4	4	0	3
mammographic	961	5	5	0	2
movement	360	90	90	0	15
newthyroid	215	5	5	0	3
pageblocks	5,472	10	10	0	5
penbased	10,992	16	16	0	10
phoneme	5,404	5	5	0	2
pima	768	8	8	0	2
saheart	462	9	8	1	2
satimage	6,435	36	36	0	7
segment	2,310	19	19	0	7
sonar	208	60	60	0	2
spambase	4,597	57	57	0	2
specfheart	267	44	44	0	2
tae	151	5	5	0	3
titanic	2,201	3	3	0	2
vehicle	846	18	18	0	4
vowel	990	13	13	0	11
wine	178	13	13	0	3
wisconsin	699	9	9	0	2
yeast	1484	8	8	0	10

In this study, six classifiers have been used in order to find differences in performance among the discretizers. The classifiers are:

- *C4.5* [10]: A well-known decision tree, considered one of the top 10 DM algorithms [9].
- *DataSqueezer* [117]: This learner belongs to the family of inductive rule extraction. In spite of its relative simplicity, DataSqueezer is a very effective learner. The rules generated by the algorithm are compact and comprehensible, but accuracy is to some extent degraded in order to achieve this goal.
- *KNN*: One of the simplest and most effective methods based on similarities among a set of objects. It is also considered one of the top 10 DM algorithms [9] and it can handle nominal attributes using proper distance functions such as HVDM [118]. It belongs to the lazy learning family [119], [120].
- *Naive Bayes*: This is another of the the top 10 DM al-

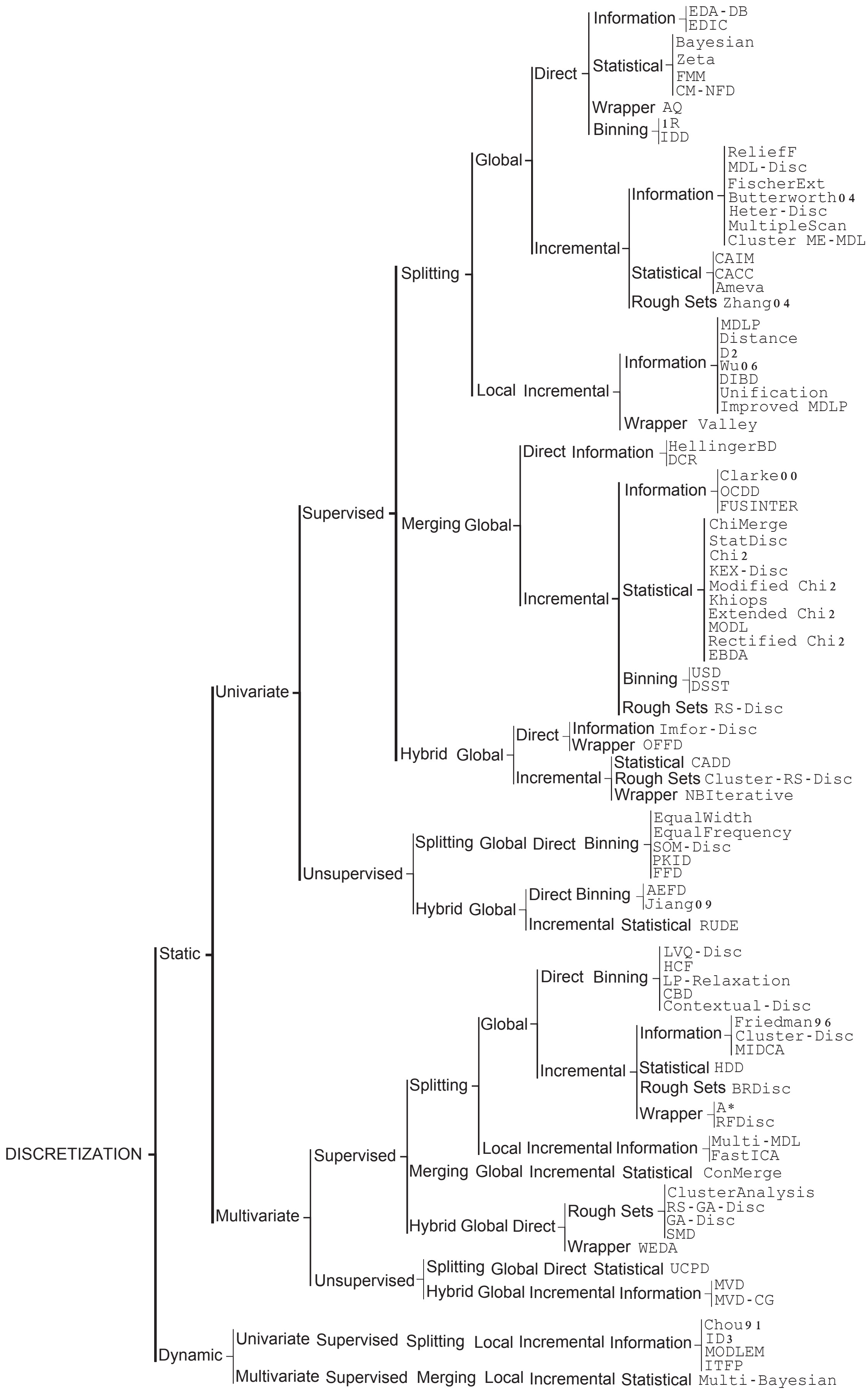


Fig. 2: Discretization Taxonomy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

TABLE 3: Parameters of the discretizers and classifiers

Method	Parameters
C4.5	pruned tree, confidence = 0.25, 2 examples per leaf
DataSqueezer	pruning and generalization threshold = 0.05
KNN	K=3, HVDM distance
PUBLIC	25 nodes between prune
Ripper	k=2, grow set = 0.66
1R	6 examples of the same class per interval
CADD	confidence threshold = 0.01
Chi2	inconsistency threshold = 0.02
ChiMerge	confidence threshold = 0.05
FDD	frequency size = 30
FUSINTER	$\alpha = 0.975$, $\lambda = 1$
HDD	coefficient = 0.8
IDD	neighborhood = 3, windows size = 3, nominal distance
MODL	optimized process type
UCPD	intervals = [3, 6], KNN map type, neighborhood = 6, minimum support = 25, merged threshold = 0.5, scaling factor = 0.5, use discrete

gorithms [9]. Its aim is to construct a rule which will allow us to assign future objects to a class, assuming independence of attributes when probabilities are established.

- *PUBLIC* [121]: It is an advanced decision tree that integrates the pruning phase with the building stage of the tree in order to avoid the expansion of branches that would be pruned afterwards.
- *Ripper* [122]: This is a widely used rule induction method based on a *separate and conquer* strategy. It incorporates diverse mechanisms to avoid overfitting and to handle numeric and nominal attributes simultaneously. The models obtained are in the form of decision lists.

The data sets considered are partitioned using the ten fold cross-validation (10-fcv) procedure. The parameters of the discretizers and classifiers are those recommended by their respective authors. They are specified in Table 3 for those methods which require them. We assume that the choice of the values of parameters is optimally chosen by their own authors. Nevertheless, in discretizers that require the input of the number of intervals as a parameter, we use a rule of thumb which is dependent on the number of instances in the data set. It consists in dividing the number of instances by 100 and taking the maximum value between this result and the number of classes. All discretizers and classifiers are run one time in each partition because they are non-stochastic.

Two performance measures are widely used because of their simplicity and successful application when multi-class classification problems are dealt. We refer to accuracy and Cohen's kappa [123] measures, which will be adopted to measure the efficacy discretizers in terms of the generalization classification rate.

- *Accuracy*: is the number of successful hits relative to the total number of classifications. It has been by far the most commonly used metric for assessing the performance of classifiers for years [2], [124].
- *Cohen's kappa*: is an alternative to *accuracy*, a method, known for decades, which compensates for random hits [123]. Its original purpose was to measure the degree of agreement or disagreement between two people observing the same phenomenon. Cohen's kappa can be adapted to classification tasks and its use is recommended because it takes random successes into consideration as a standard, in the same way as the AUC measure [125].

  An easy way of computing Cohen's kappa is to make use of the resulting confusion matrix in a classification task. Specifically, the Cohen's kappa measure can be obtained using the following expression:

$$kappa = \frac{N \sum_{i=1}^{C} y_{ii} - \sum_{i=1}^{C} y_{i.} y_{.i}}{N^2 - \sum_{i=1}^{C} y_{i.} y_{.i}},$$

  where y_{ii} is the cell count in the main diagonal of the resulting confusion matrix, N is the number of examples, C is the number of class values, and $y_{.i}$,$y_{i.}$ are the columns' and rows' total counts of the confusion matrix, respectively. Cohen's kappa ranges from -1 (total disagreement) through 0 (random classification) to 1 (perfect agreement). Being a scalar, it is less expressive than ROC curves when applied to binary-classification. However, for multi-class problems, kappa is a very useful, yet simple, meter for measuring the accuracy of the classifier while compensating for random successes.

TABLE 4: Average results collected from intrinsic properties of the discretizers: number of intervals obtained and inconsistency rate in training and test data

Number Int.		*Incons. Train*		*Incons. Tst*	
Heter-Disc	8.3125	ID3	0.0504	ID3	0.0349
MVD	18.4575	PKID	0.0581	PKID	0.0358
Distance	23.2125	Modified Chi2	0.0693	FFD	0.0377
UCPD	35.0225	FFD	0.0693	HDD	0.0405
MDLP	36.6600	HDD	0.0755	Modified Chi2	0.0409
Chi2	46.6350	USD	0.0874	USD	0.0512
FUSINTER	59.9850	ClusterAnalysis	0.0958	Khiops	0.0599
DIBD	64.4025	Khiops	0.1157	ClusterAnalysis	0.0623
CADD	67.7100	EqualWidth	0.1222	EqualWidth	0.0627
ChiMerge	69.5625	EqualFrequency	0.1355	EqualFrequency	0.0652
CAIM	72.5125	Chi2	0.1360	Chi2	0.0653
Zeta	75.9325	Bayesian	0.1642	FUSINTER	0.0854
Ameva	78.8425	MODL	0.1716	MODL	0.0970
Khiops	130.3000	FUSINTER	0.1735	HellingerBD	0.1054
1R	162.1925	HellingerBD	0.1975	Bayesian	0.1139
EqualWidth	171.7200	IDD	0.2061	UCPD	0.1383
Extended Chi2	205.2650	ChiMerge	0.2504	ChiMerge	0.1432
HellingerBD	244.6925	UCPD	0.2605	IDD	0.1570
EqualFrequency	267.7250	CAIM	0.2810	CAIM	0.1589
PKID	295.9550	Extended Chi2	0.3048	Extended Chi2	0.1762
MODL	335.8700	Ameva	0.3050	Ameva	0.1932
FFD	342.6050	1R	0.3112	CACC	0.2047
IDD	349.1250	CACC	0.3118	1R	0.2441
Modified Chi2	353.6000	MDLP	0.3783	Zeta	0.2454
CACC	505.5775	Zeta	0.3913	MDLP	0.2501
ClusterAnalysis	1116.1800	MVD	0.4237	DIBD	0.2757
USD	1276.1775	Distance	0.4274	Distance	0.2987
Bayesian	1336.0175	DIBD	0.4367	MVD	0.3171
ID3	1858.3000	CADD	0.6532	CADD	0.5688
HDD	2202.5275	Heter-Disc	0.6749	Heter-Disc	0.5708

The empirical study involves 30 discretization meth-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

ods from those listed in Table 1. We want to outline that the implementations are only based on the descriptions and specifications given by the respective authors in their papers.

Statistical analysis will be carried out by means of nonparametric statistical tests. In [126], [127], [128], authors recommend a set of simple, safe and robust nonparametric tests for statistical comparisons of classifiers. The Wilcoxon test [129] will be used in order to conduct pairwise comparisons among all discretizers considered in the study. More information about these statistical procedures specifically designed for use in the field of Machine Learning can be found at the SCI2S thematic public website on *Statistical Inference in Computational Intelligence and Data Mining* [3].

4.2 Analysis and Empirical Results

Table 4 presents the average results corresponding to the number of intervals and inconsistency rate in training and test data by all the discretizers over the 40 data sets. Similarly, Tables 5 and 6 collect the average results associated to accuracy and kappa measures for each classifier considered. For each metric, the discretizers are ordered from the best to the worst. In Tables 5 and 6, we highlight those discretizers whose performance is within 5% of the range between the best and the worst method in each measure, that is, $value_{best} - (0.05 \cdot (value_{best} - value_{worst}))$. They should be considered as outstanding methods in each category, regardless of their specific position in the table.

All detailed results for each data set, discretizer and classifier (including average and standard deviations), can be found at the URL http://sci2s.ugr.es/discretization. In the interest of compactness, we will include and analyze summarized results in the paper.

The Wilcoxon test [129], [126], [127] is adopted in this study considering a level of significance equal to $\alpha = 0.05$. Tables 7, 8 and 9 show a summary of all possible comparisons involved in the Wilcoxon test among all discretizers and measures, for number of intervals and inconsistency rate, accuracy and kappa respectively. Again, the individual comparisons between all possible discretizers are exhibited in the aforementioned URL mentioned above, where a detailed report of statistical results can be found for each measure and classifier. The tables in this paper (7, 8 and 9) summarize, for each method in the rows, the number of discretizers outperformed by using the Wilcoxon test under the column represented by the '+' symbol. The column with the '±' symbol indicates the number of wins and ties obtained by the method in the row. The maximum value for each column is highlighted by a shaded cell.

Finally, to illustrate the magnitude of the differences in average results and the relationship between the number of intervals yielded by each discretizer and the accuracy obtained for each classifier, Figure 3 depicts a confrontation between the average number of intervals and accuracy reflected by an X-Y axis graphic, for each classifier. It also helps us to see the differences in the behavior of discretization when it is used over distinct classifiers.

TABLE 7: Wilcoxon test results in number of intervals and inconsistencies

	N. Intervals		Incons. Tra		Incons. Tst	
	+	±	+	±	+	±
1R	10	21	3	17	2	20
Ameva	13	21	6	16	4	21
Bayesian	2	4	10	29	7	29
CACC	7	22	4	17	4	21
CADD	21	28	0	1	0	1
CAIM	14	23	6	19	6	20
Chi2	15	26	9	20	9	20
ChiMerge	15	23	6	20	6	23
ClusterAnalysis	1	4	15	29	9	29
DIBD	21	27	2	7	2	8
Distance	26	28	2	6	2	6
EqualFrequency	7	12	12	26	11	29
EqualWidth	11	18	16	26	13	29
Extended Chi2	14	27	2	14	2	18
FFD	5	8	21	29	16	29
FUSINTER	14	22	11	23	8	29
HDD	0	2	18	29	14	29
HellingerBD	9	15	8	21	7	26
Heter-Disc	29	29	0	1	0	1
ID3	0	1	23	29	16	29
IDD	5	11	8	28	6	29
Khiops	9	15	15	27	12	29
MDLP	22	27	3	9	3	11
Modified Chi2	7	13	17	26	15	29
MODL	5	14	12	24	7	29
MVD	23	28	2	13	2	13
PKID	5	8	22	29	16	29
UCPD	17	25	6	17	5	20
USD	2	4	18	29	15	29
Zeta	12	23	3	9	3	13

Once the results are presented in the mentioned tables and graphics, we can stress some interesting properties observed from them, and we can point out the best performing discretizers:

- Regarding the number of intervals, the discretizers which divide the numerical attributes in fewer intervals are *Heter-Disc*, *MVD* and *Distance*, whereas discretizers which require a large number of cut points are *HDD*, *ID3* and *Bayesian*. The Wilcoxon test confirms that *Heter-Disc* is the discretizer that obtains the least intervals outperforming the rest.
- The inconsistency rate both in training data and test data follows a similar trend for all discretizers,

3. http://sci2s.ugr.es/sicidm/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

TABLE 5: Average results of accuracy considering the six classifiers

C4.5		*DataSqueezer*		*KNN*		*Naive Bayes*		*PUBLIC*		*Ripper*	
FUSINTER	0.7588	Distance	0.5666	PKID	0.7699	PKID	0.7587	FUSINTER	0.7448	Modified Chi2	0.7241
ChiMerge	0.7494	CAIM	0.5547	FFD	0.7594	Modified Chi2	0.7578	CAIM	0.7420	Chi2	0.7196
Zeta	0.7488	Ameva	0.5518	Modified Chi2	0.7573	FUSINTER	0.7576	ChiMerge	0.7390	PKID	0.7097
CAIM	0.7484	MDLP	0.5475	EqualFrequency	0.7557	ChiMerge	0.7543	MDLP	0.7334	MODL	0.7089
UCPD	0.7447	Zeta	0.5475	Khiops	0.7512	FFD	0.7535	Distance	0.7305	FUSINTER	0.7078
Distance	0.7446	ChiMerge	0.5472	EqualWidth	0.7472	CAIM	0.7535	Zeta	0.7301	Khiops	0.6999
MDLP	0.7444	CACC	0.5430	FUSINTER	0.7440	EqualWidth	0.7517	Chi2	0.7278	FFD	0.6970
Chi2	0.7442	Heter-Disc	0.5374	ChiMerge	0.7389	Zeta	0.7507	UCPD	0.7254	EqualWidth	0.6899
Modified Chi2	0.7396	DIBD	0.5322	CAIM	0.7381	EqualFrequency	0.7491	Modified Chi2	0.7250	EqualFrequency	0.6890
Ameva	0.7351	UCPD	0.5172	MODL	0.7372	MODL	0.7479	Khiops	0.7200	CAIM	0.6870
Khiops	0.7312	MVD	0.5147	HellingerBD	0.7327	Chi2	0.7476	Ameva	0.7168	HellingerBD	0.6816
MODL	0.7310	FUSINTER	0.5126	Chi2	0.7267	Khiops	0.7455	HellingerBD	0.7119	USD	0.6807
EqualFrequency	0.7304	Bayesian	0.4915	USD	0.7228	USD	0.7428	EqualFrequency	0.7110	ChiMerge	0.6804
EqualWidth	0.7252	Extended Chi2	0.4913	Ameva	0.7220	ID3	0.7381	MODL	0.7103	ID3	0.6787
HellingerBD	0.7240	Chi2	0.4874	ID3	0.7172	Ameva	0.7375	CACC	0.7069	Zeta	0.6786
CACC	0.7203	HellingerBD	0.4868	ClusterAnalysis	0.7132	Distance	0.7372	DIBD	0.7002	HDD	0.6700
Extended Chi2	0.7172	MODL	0.4812	Zeta	0.7126	MDLP	0.7369	EqualWidth	0.6998	Ameva	0.6665
DIBD	0.7141	CADD	0.4780	HDD	0.7104	ClusterAnalysis	0.7363	Extended Chi2	0.6974	UCPD	0.6651
FFD	0.7091	EqualFrequency	0.4711	UCPD	0.7090	HellingerBD	0.7363	HDD	0.6789	CACC	0.6562
PKID	0.7079	1R	0.4702	MDLP	0.7002	HDD	0.7360	FFD	0.6770	Extended Chi2	0.6545
HDD	0.6941	EqualWidth	0.4680	Distance	0.6888	UCPD	0.7227	PKID	0.6758	Bayesian	0.6521
USD	0.6835	IDD	0.4679	IDD	0.6860	Extended Chi2	0.7180	USD	0.6698	ClusterAnalysis	0.6464
ClusterAnalysis	0.6813	USD	0.4651	Bayesian	0.6844	CACC	0.7176	Bayesian	0.6551	MDLP	0.6439
ID3	0.6720	Khiops	0.4567	CACC	0.6813	Bayesian	0.7167	ClusterAnalysis	0.6477	Distance	0.6402
1R	0.6695	Modified Chi2	0.4526	DIBD	0.6731	DIBD	0.7036	ID3	0.6406	IDD	0.6219
Bayesian	0.6675	HDD	0.4308	1R	0.6721	IDD	0.6966	MVD	0.6401	Heter-Disc	0.6084
IDD	0.6606	ClusterAnalysis	0.4282	Extended Chi2	0.6695	1R	0.6774	IDD	0.6352	1R	0.6058
MVD	0.6499	PKID	0.3942	MVD	0.6062	MVD	0.6501	1R	0.6332	DIBD	0.5953
Heter-Disc	0.6443	ID3	0.3896	Heter-Disc	0.5524	Heter-Disc	0.6307	Heter-Disc	0.6317	MVD	0.5921
CADD	0.5689	FFD	0.3848	CADD	0.5064	CADD	0.5669	CADD	0.5584	CADD	0.4130

TABLE 6: Average results of kappa considering the six classifiers

C4.5		*DataSqueezer*		*KNN*		*Naive Bayes*		*PUBLIC*		*Ripper*	
FUSINTER	0.5550	CACC	0.2719	PKID	0.5784	PKID	0.5762	CAIM	0.5279	Modified Chi2	0.5180
ChiMerge	0.5433	Ameva	0.2712	FFD	0.5617	Modified Chi2	0.5742	FUSINTER	0.5204	Chi2	0.5163
CAIM	0.5427	CAIM	0.2618	Modified Chi2	0.5492	FUSINTER	0.5737	ChiMerge	0.5158	MODL	0.5123
Zeta	0.5379	ChiMerge	0.2501	Khiops	0.5457	FFD	0.5710	MDLP	0.5118	FUSINTER	0.5073
MDLP	0.5305	FUSINTER	0.2421	EqualFrequency	0.5438	ChiMerge	0.5650	Distance	0.5074	Khiops	0.4939
UCPD	0.5299	UCPD	0.2324	EqualWidth	0.5338	Chi2	0.5620	Zeta	0.5010	PKID	0.4915
Ameva	0.5297	Zeta	0.2189	CAIM	0.5260	CAIM	0.5616	Ameva	0.4986	EqualFrequency	0.4892
Chi2	0.5290	USD	0.2174	FUSINTER	0.5242	EqualWidth	0.5593	Chi2	0.4899	ChiMerge	0.4878
Distance	0.5288	Distance	0.2099	ChiMerge	0.5232	Khiops	0.5570	UCPD	0.4888	EqualWidth	0.4875
Modified Chi2	0.5163	Khiops	0.2038	MODL	0.5205	EqualFrequency	0.5564	Khiops	0.4846	CAIM	0.4870
MODL	0.5131	HDD	0.2030	HellingerBD	0.5111	MODL	0.5564	CACC	0.4746	Ameva	0.4810
EqualFrequency	0.5108	EqualFrequency	0.2016	Chi2	0.5100	USD	0.5458	HellingerBD	0.4736	FFD	0.4809
Khiops	0.5078	HellingerBD	0.1965	Ameva	0.5041	Zeta	0.5457	Modified Chi2	0.4697	Zeta	0.4769
HellingerBD	0.4984	Bayesian	0.1941	USD	0.4943	Ameva	0.5456	MODL	0.4620	HellingerBD	0.4729
CACC	0.4961	MODL	0.1918	HDD	0.4878	ID3	0.5403	EqualFrequency	0.4535	USD	0.4560
EqualWidth	0.4909	MDLP	0.1875	ClusterAnalysis	0.4863	HDD	0.5394	DIBD	0.4431	UCPD	0.4552
Extended Chi2	0.4766	PKID	0.1846	Zeta	0.4831	MDLP	0.5389	EqualWidth	0.4386	CACC	0.4504
DIBD	0.4759	ID3	0.1818	ID3	0.4769	Distance	0.5368	Extended Chi2	0.4358	MDLP	0.4449
FFD	0.4605	EqualWidth	0.1801	UCPD	0.4763	HellingerBD	0.5353	HDD	0.4048	Distance	0.4429
PKID	0.4526	Modified Chi2	0.1788	MDLP	0.4656	ClusterAnalysis	0.5252	FFD	0.3969	HDD	0.4403
HDD	0.4287	DIBD	0.1778	Distance	0.4470	UCPD	0.5194	PKID	0.3883	ID3	0.4359
USD	0.4282	Chi2	0.1743	CACC	0.4367	CACC	0.5128	USD	0.3845	Extended Chi2	0.4290
ClusterAnalysis	0.4044	IDD	0.1648	IDD	0.4329	Extended Chi2	0.4910	MVD	0.3461	ClusterAnalysis	0.4252
ID3	0.3803	FFD	0.1635	Extended Chi2	0.4226	Bayesian	0.4757	ClusterAnalysis	0.3453	Bayesian	0.3987
IDD	0.3803	ClusterAnalysis	0.1613	Bayesian	0.4201	DIBD	0.4731	Bayesian	0.3419	DIBD	0.3759
MVD	0.3759	Extended Chi2	0.1465	DIBD	0.4167	IDD	0.4618	ID3	0.3241	IDD	0.3650
Bayesian	0.3716	MVD	0.1312	1R	0.3940	1R	0.3980	IDD	0.3066	MVD	0.3446
1R	0.3574	1R	0.1147	MVD	0.3429	MVD	0.3977	1R	0.3004	1R	0.3371
Heter-Disc	0.2709	Heter-Disc	0.1024	Heter-Disc	0.2172	Heter-Disc	0.2583	Heter-Disc	0.2570	Heter-Disc	0.2402
CADD	0.1524	CADD	0.0260	CADD	0.1669	CADD	0.1729	CADD	0.1489	CADD	0.1602

considering that the inconsistency obtained in test data is always lower than in training data. *ID3* is the discretizer that obtains the lowest average inconsistency rate in training and test data, albeit the Wilcoxon test cannot find significant differences between it and the other two discretizers: *FFD* and *PKID*. We can observe a close relationship between the number of intervals produced and the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

TABLE 8: Wilcoxon test results in accuracy

	C4.5		Data Squeezer		KNN		Naive Bayes		PUBLIC		Ripper	
	+	±	+	±	+	±	+	±	+	±	+	±
1R	1	12	3	23	2	19	1	9	1	12	1	11
Ameva	14	29	17	29	8	26	9	29	13	29	9	29
Bayesian	1	9	5	26	2	12	2	11	0	11	2	17
CACC	9	28	16	29	2	18	5	28	9	29	4	26
CADD	0	1	1	22	0	1	0	1	0	6	0	0
CAIM	16	29	16	29	11	28	10	29	16	29	11	28
Chi2	13	29	4	26	6	27	9	29	11	29	19	29
ChiMerge	17	29	18	29	13	28	10	29	17	29	9	28
ClusterAnalysis	1	10	0	12	5	24	6	27	1	11	2	20
DIBD	6	21	8	29	2	9	2	8	9	23	1	5
Distance	13	29	16	29	2	17	7	26	13	28	2	13
EqualFrequency	10	27	3	21	18	29	9	29	10	26	11	27
EqualWidth	7	20	2	18	11	28	8	29	6	20	9	27
Extended Chi2	9	27	4	26	3	19	3	17	6	25	2	25
FFD	5	15	0	5	20	28	8	29	1	13	10	27
FUSINTER	21	29	9	29	12	28	15	29	20	29	11	29
HDD	1	18	0	14	4	23	5	28	0	24	7	26
HellingerBD	10	27	4	22	7	26	7	28	10	26	6	26
Heter-Disc	0	9	9	29	0	2	0	3	0	11	1	10
ID3	1	10	0	5	5	22	4	28	0	11	5	26
IDD	1	10	3	23	4	21	2	14	0	12	1	16
Khiops	12	27	3	18	18	29	9	29	9	27	11	29
MDLP	14	29	14	29	3	22	8	29	15	29	2	16
Modified Chi2	11	27	3	21	17	28	10	29	9	29	23	29
MODL	12	28	5	23	14	28	9	29	10	28	17	29
MVD	1	15	5	29	1	8	1	7	0	19	1	13
PKID	5	15	0	6	27	29	9	29	1	13	15	29
UCPD	14	29	7	26	4	17	2	15	14	28	3	19
USD	1	13	3	19	6	23	6	29	1	19	7	25
Zeta	14	29	17	29	4	20	9	29	14	29	7	27

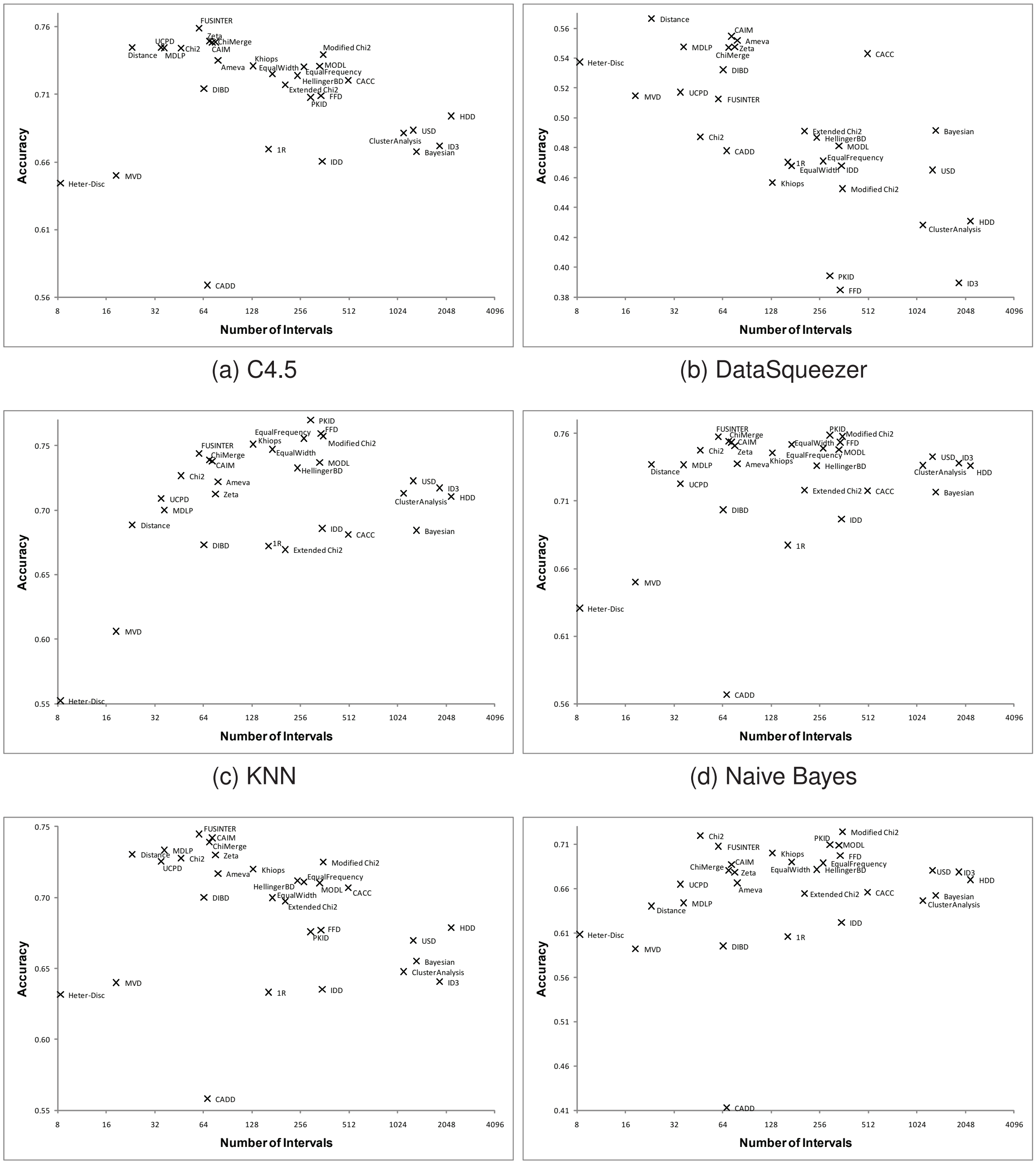
TABLE 9: Wilcoxon test results in kappa

	C4.5		Data Squeezer		KNN		Naive Bayes		PUBLIC		Ripper	
	+	±	+	±	+	±	+	±	+	±	+	±
1R	1	11	0	15	2	16	2	8	1	13	1	11
Ameva	15	29	24	29	11	26	11	29	16	29	9	29
Bayesian	1	8	1	24	2	10	2	8	1	11	2	17
CACC	11	28	25	29	3	16	7	25	13	29	4	26
CADD	0	1	0	3	0	1	0	1	0	2	0	0
CAIM	17	29	22	29	13	28	11	29	21	29	11	28
Chi2	14	29	2	24	11	27	10	29	13	29	19	29
ChiMerge	19	29	22	29	13	28	11	29	18	29	9	28
ClusterAnalysis	2	10	2	21	5	23	6	22	1	11	2	20
DIBD	8	20	1	24	2	10	2	7	7	18	1	5
Distance	16	29	1	26	2	16	7	28	16	29	2	13
EqualFrequency	11	25	3	25	18	29	10	29	10	23	11	27
EqualWidth	7	20	2	23	14	27	8	28	6	18	9	27
Extended Chi2	10	27	1	20	2	17	3	16	6	23	2	25
FFD	6	14	1	19	23	28	12	29	2	14	10	27
FUSINTER	21	29	16	29	14	28	18	29	19	29	11	29
HDD	2	17	5	25	5	22	6	25	1	22	7	26
HellingerBD	11	23	4	23	9	26	7	21	11	24	6	26
Heter-Disc	0	6	0	12	0	2	0	2	0	8	1	10
ID3	1	8	2	22	4	20	6	26	0	10	5	26
IDD	1	9	1	23	2	18	2	15	1	11	1	16
Khiops	11	24	5	24	18	29	10	29	13	25	11	29
MDLP	16	29	1	24	6	22	8	29	19	29	2	16
Modified Chi2	12	27	1	21	17	27	14	29	9	28	23	29
MODL	12	28	4	24	14	27	12	29	11	28	17	29
MVD	1	12	0	19	1	10	1	6	1	16	1	13
PKID	5	14	2	23	27	29	14	29	2	14	15	29
UCPD	14	29	15	28	4	16	4	16	13	25	3	19
USD	4	13	9	25	6	23	6	25	3	15	7	25
Zeta	15	29	9	27	3	18	6	27	16	29	7	27

inconsistency rate, where discretizers that compute fewer cut points are usually those which have a high inconsistency rate. They risk the consistency of the data in order to simplify the result, although the consistency is not usually correlated with the accuracy, as we will see below.

- In decision trees (*C4.5* and *PUBLIC*), a subset of discretizers can be stressed as the best performing ones. Considering average accuracy, *FUSINTER*, *ChiMerge* and *CAIM* stand out from the rest. Considering average kappa, *Zeta* and *MDLP* are also added to this subset. The Wilcoxon test confirms this result and adds another discretizer, *Distance*, which outperforms 16 of the 29 methods. All methods emphasized are supervised, incremental (except *Zeta*) and use statistical and information measures as evaluators. Splitting/Merging and Local/Global properties have no effect on decision trees.
- Considering rule induction (*DataSqueezer* and *Ripper*), the best performing discretizers are *Distance*, *Modified Chi2*, *Chi2*, *PKID* and *MODL* in average accuracy and *CACC*, *Ameva*, *CAIM* and *FUSINTER* in average kappa. In this case, the results are very irregular due to the fact that the Wilcoxon test emphasizes the *ChiMerge* as the best performing discretizer for *DataSqueezer* instead of *Distance* and incorporates *Zeta* in the subset. With *Ripper*, the Wilcoxon test confirms the results obtained by averaging accuracy and kappa. It is difficult to discern a common set of properties that define the best performing discretizers due to the fact that rule induction methods differ in their operation to a greater extent than decision trees. However, we can remark that, in the subset of best methods, incremental and supervised discretizers predominate in the statistical evaluation.
- Lazy and bayesian learning can be analyzed together, due to the fact that the HVDM distance used in KNN is highly related to the computation of bayesian probabilities considering attribute independence [118]. With respect to lazy and bayesian learning, *KNN* and *Naive Bayes*, the subset of remarkable discretizers is formed by *PKID*, *FFD*, *Modified Chi2*, *FUSINTER*, *ChiMerge*, *CAIM*, *EqualWidth* and *Zeta*, when average accuracy is used; and *Chi2*, *Khiops*, *EqualFrequency* and *MODL* must be added when average kappa is considered. The statistcal report by Wilcoxon informs us of the existence of two outstanding methods: *PKID* for *KNN*, which outperforms 27/29 and *FUSINTER* for *Naive Bayes*. Here, supervised and unsupervised, direct and incremental, binning and statistical/information evaluation are characteristics present in the best perfoming methods. However, we can see that all of them are global, thus identifying a trend towards binning methods.
- In general, accuracy and kappa performance registered by discretizers do not differ too much. The behavior in both evaluation metrics are quite similar, taking into account that the differences in kappa are usually lower due to the compensation of random success offered by it. Surprisingly, in *DataSqueezer*, accuracy and kappa offer the greatest differences in behavior, but they are motivated by the fact that this method focuses on obtaining simple rule sets,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



(a) C4.5 (b) DataSqueezer

(c) KNN (d) Naive Bayes

(e) PUBLIC (f) Ripper

Fig. 3: Accuracy vs. Number of Intervals

leaving precision in the background.

- It is obvious that there is a direct dependence between discretization and the classifier used. We have pointed out that a similar behavior in decision trees and lazy/bayesian learning can be detected, whereas in rule induction learning, the operation of the algorithm conditions the effectiveness of the discretizer. Knowing a subset of suitable discretizers for each type of discretizer is a good starting point to understand and propose improvements in the area.
- Another interesting remark can be made about the relationship between accuracy and the number of intervals yielded by a discretizer. Figure 3 supports the hypothesis that there is no direct correlation between them. A discretizer that computes few cut points does not have to obtain poor results in accuracy and vice versa. Figures 3a, 3c, 3d and 3e point out that there is a minimum limit in the number of intervals to guarantee accurate models, given by the cut points computed by *Distance*. Figure 3b shows how *DataSqueezer* is worse as the number of intervals increases, but this is an inherent behavior of the classifier.
- Finally, we can stress a subset of global best dis-

cretizers considering a trade-off between the number of intervals and accuracy obtained. In this subset, we can include *FUSINTER*, *Distance*, *Chi2*, *MDLP* and *UCPD*.

On the other hand, an analysis centered on the 30 discretizers studied is given as follows:

- Many classic discretizers are usually the best performing ones. This is the case of *ChiMerge*, *MDLP*, *Zeta*, *Distance* and *Chi2*.
- Other classic discretizers are not as good as they should be, considering that they have been improved over the years: *EqualWidth*, *EqualFrequency*, *1R*, *ID3* (the static version is much worse than the dynamic inserted in C4.5 operation), *CADD*, *Bayesian* and *ClusterAnalysis*.
- Slight modifications of classic methods have greatly enhanced their results, such as, for example, *FUSINTER*, *Modified Chi2*, *PKID* and *FFD*; but in other cases, the extensions have diminished their performance: *USD*, *Extended Chi2*.
- Promising techniques that have been evaluated under unfavorable circumstances are *MVD* and *UCP*, which are unsupervised methods useful for application to other DM problems apart from classification.
- Recent proposed methods that have been demonstrated to be competitive compared with classic methods and even outperforming them in some scenarios are *Khiops*, *CAIM*, *MODL*, *Ameva* and *CACC*. However, recent proposals that have reported bad results in general are *Heter-Disc*, *HellingerBD*, *DIBD*, *IDD* and *HDD*.
- Finally, this study involves a higher number of data sets than the quantity considered in previous works and the conclusions achieved are impartial towards an specific discretizer. However, we have to stress some coincidences with the conclusions of these previous works. For example in [102], the authors propose an improved version of *Chi2* in terms of accuracy, removing the user parameter choice. We check and measure the actual improvement. In [12], the authors develop an intense theoretical and analytical study concerning *Naive Bayes* and propose *PKID* and *FFD* according to their conclusions. In this paper we corroborate that *PKID* is the best suitable method for *Naive Bayes* and even for *KNN*. Finally, we may note that *CAIM* is one of the simplest discretizers and its effectiveness has also been shown in this study.

5 CONCLUDING REMARKS AND GLOBAL GUIDELINES

The present paper offers an exhaustive survey of the discretization methods proposed in the literature. Basic and advanced properties, existing work and related fields have been studied. Based on the main characteristics studied, we have designed a taxonomy of discretization methods. Furthermore, the most important discretizers (classic and recent) have been empirically analyzed over a vast number of classification data sets. In order to strengthen the study, statistical analysis based on nonparametric tests has been added supporting the conclusions drawn. Several remarks and guidelines can be suggested:

- A researcher/practitioner interested in applying a discretization method should be aware of the properties that define them in order to choose the most appropriate in each case. The taxonomy developed and the empirical study can help to make this decision.
- In the proposal of a new discretizer, the best approaches and those which fit with the basic properties of the new proposal should be used in the comparison study. In order to do this, the taxonomy and the analysis of results can guide a future proposal in the correct way.
- This paper assists non-experts in discretization to differentiate among methods, making an appropriate decision about their application and understanding their behavior.
- It is important to know the main advantages of each discretizer. In this paper, many discretizers have been empirically analyzed but we cannot give a single conclusion about which is the best performing one. This depends upon the problem tackled and the data mining method used, but the results offered here could help to limit the set of candidates.
- The empirical study allows us to stress several methods among the whole set:
  - *FUSINTER*, *ChiMerge*, *CAIM* and *Modified Chi2* offer excellent performances considering all types of classifiers.
  - *PKID*, *FFD* are suitable methods for lazy and bayesian learning and *CACC*, *Distance* and *MODL* are good choices in rule induction learning.
  - *FUSINTER*, *Distance*, *Chi2*, *MDLP* and *UCPD* obtain a satisfactory trade-off between the number of intervals produced and accuracy.

It would be desirable that a researcher/practitioner who wants to decide which discretization scheme to apply to his/her data needs to know how the experiments of this paper or data will benefit and guide him/her. As future work, we propose the analysis of each property studied in the taxonomy with respect to some data characteristics, such as number of labels, dimensions or dynamic range of original attributes. Following this trend, we expect to find the most suitable discretizer taking into consideration some basic characteristic of the data sets.

ACKNOWLEDGMENTS

This work was supported by the Research Projects TIN2011-28488 and TIC-6858. J.A. Sáez and V. Lopez hold an FPU scholarship from the Spanish Ministry of

Education. The authors would like to thank the reviewers for their suggestions.

REFERENCES

[1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, 2006.

[2] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical machine learning tools and techniques. 3rd Edition.* Morgan Kaufmann, 2011.

[3] I. Kononenko and M. Kukar, *Machine Learning and Data Mining: Introduction to Principles and Algorithms*. Horwood Publishing Limited, 2007.

[4] K. J. Cios, W. Pedrycz, R. W. Swiniarski, and L. A. Kurgan, *Data Mining: A Knowledge Discovery Approach*. Springer, 2007.

[5] D. Pyle, *Data preparation for data mining*. Morgan Kaufmann Publishers Inc., 1999.

[6] H. Liu, F. Hussain, C. L. Tan, and M. Dash, "Discretization: An enabling technique," *Data Mining and Knowledge Discovery*, vol. 6, no. 4, pp. 393–423, 2002.

[7] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proceedings of the Twelfth International Conference onMachine Learning (ICML)*, 1995, pp. 194–202.

[8] Y. Yang, G. I. Webb, and X. Wu, "Discretization methods," in *Data Mining and Knowledge Discovery Handbook*, 2010, pp. 101–116.

[9] X. Wu and V. Kumar, Eds., *The Top Ten Algorithms in Data Mining*. Chapman & Hall/CRC Data Mining and Knowledge Discovery, 2009.

[10] J. R. Quinlan, *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.

[11] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the 20th Very Large Data Bases conference (VLDB)*, 1994, pp. 487–499.

[12] Y. Yang and G. I. Webb, "Discretization for naive-bayes learning: managing discretization bias and variance," *Machine Learning*, vol. 74, no. 1, pp. 39–74, 2009.

[13] M. J. Flores, J. A. Gámez, A. M. Martínez, and J. M. Puerta, "Handling numeric attributes when comparing bayesian network classifiers: does the discretization method matter?" *Applied Intelligence, in press DOI: 10.1007/s10489-011-0286-z*, 2011.

[14] M. Richeldi and M. Rossotto, "Class-driven statistical discretization of continuous attributes," in *Proceedings of the 8th European Conference on Machine Learning (ECML)*, ser. ECML '95, 1995, pp. 335–338.

[15] B. Chlebus and S. H. Nguyen, "On finding optimal discretizations for two attributes," in *Lecture Notes in Artificial Intelligence*, vol. 1424, 1998, pp. 537–544.

[16] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI)*, 1993, pp. 1022–1029.

[17] R. Kerber, "Chimerge: Discretization of numeric attributes," in *National Conference on Artifical Intelligence American Association for Artificial Intelligence (AAAI)*, 1992, pp. 123–128.

[18] H. Liu and R. Setiono, "Feature selection via discretization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 9, pp. 642–645, 1997.

[19] X. Wu, "A bayesian discretizer for real-valued attributes," *The Computer Journal*, vol. 39, pp. 688–691, 1996.

[20] M. Boullé, "MODL: A bayes optimal discretization method for continuous attributes," *Machine Learning*, vol. 65, no. 1, pp. 131–165, 2006.

[21] S. H. Nguyen and A. Skowron, "Quantization of real value attributes - rough set and boolean reasoning approach," in *Proceedings of the Second Joint Annual Conference on Information Sciences (JCIS)*, 1995, pp. 34–37.

[22] G. Zhang, L. Hu, and W. Jin, "Discretization of continuous attributes in rough set theory and its application," in *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems (CIS)*, 2004, pp. 1020–1026.

[23] A. A. Bakar, Z. A. Othman, and N. L. M. Shuib, "Building a new taxonomy for data discretization techniques," in *Proceedings on Conference on Data Mining and Optimization (DMO)*, 2009, pp. 132–140.

[24] M. R. Chmielewski and J. W. Grzymala-Busse, "Global discretization of continuous attributes as preprocessing for machine learning," *International Journal of Approximate Reasoning*, vol. 15, no. 4, pp. 319–331, 1996.

[25] G. K. Singh and S. Minz, "Discretization using clustering and rough set theory," in *Proceedings of the 17th International Conference on Computer Theory and Applications (ICCTA)*, 2007, pp. 330–336.

[26] L. Liu, A. K. C. Wong, and Y. Wang, "A global optimal algorithm for class-dependent discretization of continuous data," *Intelligent Data Analysis*, vol. 8, pp. 151–170, 2004.

[27] R. C. Holte, "Very simple classification rules perform well on most commonly used datasets," *Machine Learning*, vol. 11, pp. 63–90, 1993.

[28] J. Catlett, "On changing continuous attributes into ordered discrete attributes," in *European Working Session on Learning (EWSL)*, ser. Lecture Notes on Computer Science, vol. 482. Springer-Verlag, 1991, pp. 164–178.

[29] R. Susmaga, "Analyzing discretizations of continuous attributes given a monotonic discrimination function," *Intelligent Data Analysis*, vol. 1, no. 1-4, pp. 157–179, 1997.

[30] T. Elomaa and J. Rousu, "General and efficient multisplitting of numerical attributes," *Machine Learning*, vol. 36, pp. 201–244, 1999.

[31] ——, "Necessary and sufficient pre-processing in numerical range discretization," *Knowledge and Information Systems*, vol. 5, pp. 162–182, 2003.

[32] ——, "Efficient multisplitting revisited: Optima-preserving elimination of partition candidates," *Data Mining and Knowledge Discovery*, vol. 8, pp. 97–126, 2004.

[33] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

[34] S. R. Gaddam, V. V. Phoha, and K. S. Balagani, "K-Means+ID3: A novel method for supervised anomaly detection by cascading k-means clustering and ID3 decision tree learning methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, pp. 345–354, 2007.

[35] H.-W. Hu, Y.-L. Chen, and K. Tang, "A dynamic discretization approach for constructing decision trees with a continuous label," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 11, pp. 1505–1514, 2009.

[36] H. Ishibuchi, T. Yamamoto, and T. Nakashima, "Fuzzy data mining: Effect of fuzzy discretization," in *IEEE International Conference on Data Mining (ICDM)*, 2001, pp. 241–248.

[37] A. Roy and S. K. Pal, "Fuzzy discretization of feature space for a rough set classifier," *Pattern Recognition Letters*, vol. 24, pp. 895–902, 2003.

[38] D. Janssens, T. Brijs, K. Vanhoof, and G. Wets, "Evaluating the performance of cost-based discretization versus entropy- and error-based discretization," *Computers & Operations Research*, vol. 33, no. 11, pp. 3107–3123, 2006.

[39] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[40] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 4, pp. 687–719, 2009.

[41] A. Bondu, M. Boulle, and V. Lemaire, "A non-parametric semi-supervised discretization method," *Knowledge and Information Systems*, vol. 24, pp. 35–57, 2010.

[42] F. Berzal, J.-C. Cubero, N. Marín, and D. Sánchez, "Building multi-way decision trees with numerical attributes," *Information Sciences*, vol. 165, pp. 73–90, 2004.

[43] W.-H. Au, K. C. C. Chan, and A. K. C. Wong, "A fuzzy approach to partitioning continuous attributes for classification," *IEEE Transactions on Knowledge Data Engineering*, vol. 18, no. 5, pp. 715–719, 2006.

[44] S. Mehta, S. Parthasarathy, and H. Yang, "Toward unsupervised correlation preserving discretization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 1174–1185, 2005.

[45] S. D. Bay, "Multivariate discretization for set mining," *Knowledge Information Systems*, vol. 3, pp. 491–512, 2001.

[46] M. N. M. García, J. P. Lucas, V. F. L. Batista, and M. J. P. Martín, "Multivariate discretization for associative classification in a sparse data application domain," in *Proceedings of the 5th International Conference on Hybrid Artificial Intelligent Systems (HAIS)*, 2010, pp. 104–111.

[47] S. Ferrandiz and M. Boullé, "Multivariate discretization by recursive supervised bipartition of graph," in *Proceedings of the 4th Conference on Machine Learning and Data Mining (MLDM)*, 2005, pp. 253–264.
[48] P. Yang, J.-S. Li, and Y.-X. Huang, "HDD: a hypercube division-based algorithm for discretisation," *International Journal of Systems Science*, vol. 42, no. 4, pp. 557–566, 2011.
[49] R.-P. Li and Z.-O. Wang, "An entropy-based discretization method for classification rules with inconsistency checking," in *Proceedings of the First International Conference on Machine Learning and Cybernetics (ICMLC)*, 2002, pp. 243–246.
[50] C.-H. Lee, "A hellinger-based discretization method for numeric attributes in classification learning," *Knowledge-Based Systems*, vol. 20, pp. 419–425, 2007.
[51] F. J. Ruiz, C. Angulo, and N. Agell, "IDD: A supervised interval Distance-Based method for discretization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 9, pp. 1230–1238, 2008.
[52] J. Y. Ching, A. K. C. Wong, and K. C. C. Chan, "Class-dependent discretization for inductive learning from continuous and mixed-mode data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 641–651, 1995.
[53] J. L. Flores, I. Inza, and Larra, "Wrapper discretization by means of estimation of distribution algorithms," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 525–545, 2007.
[54] D. A. Zighed, S. Rabaséda, and R. Rakotomalala, "FUSINTER: a method for discretization of continuous attributes," *International Journal of Uncertainty, Fuzziness Knowledge-Based Systems*, vol. 6, pp. 307–326, 1998.
[55] R. Jin, Y. Breitbart, and C. Muoh, "Data discretization unification," *Knowledge and Information Systems*, vol. 19, pp. 1–29, 2009.
[56] K. M. Ho and P. D. Scott, "Zeta: A global method for discretization of continuous variables," in *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD)*, 1997, pp. 191–194.
[57] L. A. Kurgan and K. J. Cios, "CAIM discretization algorithm," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 2, pp. 145–153, 2004.
[58] C.-J. Tsai, C.-I. Lee, and W.-P. Yang, "A discretization algorithm based on class-attribute contingency coefficient," *Information Sciences*, vol. 178, pp. 714–731, 2008.
[59] D. Ventura and T. R. Martinez, "BRACE: A paradigm for the discretization of continuously valued data,," in *Proceedings of the Seventh Annual Florida AI Research Symposium (FLAIRS)*, 1994, pp. 117–121.
[60] M. J. Pazzani, "An iterative improvement approach for the discretization of numeric attributes in bayesian classifiers," in *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD)*, 1995, pp. 228–233.
[61] A. K. C. Wong and D. K. Y. Chiu, "Synthesizing statistical knowledge from incomplete mixed-mode data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 796–805, 1987.
[62] M. Vannucci and V. Colla, "Meaningful discretization of continuous features for association rules mining by means of a SOM," in *Proocedings of the 12th European Symposium on Artificial Neural Networks (ESANN)*, 2004, pp. 489–494.
[63] P. A. Chou, "Optimal partitioning for classification and regression trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 340–354, 1991.
[64] R. Butterworth, D. A. Simovici, G. S. Santos, and L. Ohno-Machado, "A greedy algorithm for supervised discretization," *Journal of Biomedical Informatics*, vol. 37, pp. 285–292, 2004.
[65] C. Chan, C. Batur, and A. Srinivasan, "Determination of quantization intervals in rule based model for dynamic systems," in *Proceedings of the Conference on Systems and Man and and Cybernetics*, 1991, pp. 1719–1723.
[66] M. Boulle, "Khiops: A statistical discretization method of continuous attributes," *Machine Learning*, vol. 55, pp. 53–69, 2004.
[67] C.-T. Su and J.-H. Hsu, "An extended chi2 algorithm for discretization of real value attributes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 437–441, 2005.
[68] X. Liu and H. Wang, "A discretization algorithm based on a heterogeneity criterion," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 1166–1173, 2005.
[69] D. Ventura and T. R. Martinez, "An empirical comparison of discretization methods," in *Proceedings of the 10th International Symposium on Computer and Information Sciences (ISCIS)*, 1995, pp. 443–450.
[70] M. Wu, X.-C. Huang, X. Luo, and P.-L. Yan, "Discretization algorithm based on difference-similitude set theory," in *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics (ICMLC)*, 2005, pp. 1752–1755.
[71] I. Kononenko and M. R. Sikonja, "Discretization of continuous attributes using relieff," in *Proceedings of Elektrotehnika in Racunalnika Konferenca (ERK)*, 1995.
[72] S. Chao and Y. Li, "Multivariate interdependent discretization for continuous attribute," in *Proceedings of the Third International Conference on Information Technology and Applications (ICITA) Volume 2*, 2005, pp. 167–172.
[73] Q. Wu, J. Cai, G. Prasad, T. M. McGinnity, D. A. Bell, and J. Guan, "A novel discretizer for knowledge discovery approaches based on rough sets," in *Proceedings of the First International Conference on Rough Sets and Knowledge Technology (RSKT)*, 2006, pp. 241–246.
[74] B. Pfahringer, "Compression-based discretization of continuous attributes," in *Proceedings of the 12th International Conference on Machine Learning (ICML)*, 1995, pp. 456–463.
[75] Y. Kang, S. Wang, X. Liu, H. Lai, H. Wang, and B. Miao, "An ICA-based multivariate discretization algorithm," in *Proceedings of the First International Conference on Knowledge Science, Engineering and Management (KSEM)*, 2006, pp. 556–562.
[76] T. Elomaa, J. Kujala, and J. Rousu, "Practical approximation of optimal multivariate discretization," in *Proceedings of the 16th International Symposium on Methodologies for Intelligent Systems (ISMIS)*, 2006, pp. 612–621.
[77] N. Friedman and M. Goldszmidt, "Discretizing continuous attributes while learning bayesian networks," in *Proceedings of the 13th International Conference on Machine Learning (ICML)*, 1996, pp. 157–165.
[78] Q. Wu, D. A. Bell, G. Prasad, and T. M. McGinnity, "A distribution-index-based discretizer for decision-making with symbolic ai approaches," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, pp. 17–28, 2007.
[79] J. Cerquides and R. L. D. Mantaras, "Proposal and empirical comparison of a parallelizable distance-based discretization method," in *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD)*, 1997, pp. 139–142.
[80] R. Subramonian, R. Venkata, and J. Chen, "A visual interactive framework for attribute discretization," in *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD)*, 1997, pp. 82–88.
[81] D. A. Zighed, R. Rakotomalala, and F. Feschet, "Optimal multiple intervals discretization of continuous attributes for supervised learning," in *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD)*, 1997, pp. 295–298.
[82] W. Qu, D. Yan, Y. Sang, H. Liang, M. Kitsuregawa, and K. Li, "A novel chi2 algorithm for discretization of continuous attributes," in *Proceedings of the 10th Asia-Pacific web conference on Progress in WWW research and development*, ser. APWeb, 2008, pp. 560–571.
[83] S. J. Hong, "Use of contextual information for feature ranking and discretization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 9, pp. 718–730, 1997.
[84] L. González-Abril, F. J. Cuberos, F. Velasco, and J. A. Ortega, "Ameva: An autonomous discretization algorithm," *Expert Systems with Applications*, vol. 36, pp. 5327–5332, 2009.
[85] K. Wang and B. Liu, "Concurrent discretization of multiple attributes," in *Proceedings of the Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, 1998, pp. 250–259.
[86] P. Berka and I. Bruha, "Empirical comparison of various discretization procedures," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 12, no. 7, pp. 1017–1032, 1998.
[87] J. W. Grzymala-Busse, "A multiple scanning strategy for entropy based discretization," in *Proceedings of the 18th International Symposium on Foundations of Intelligent Systems*, ser. ISMIS, 2009, pp. 25–34.
[88] P. Perner and S. Trautzsch, "Multi-interval discretization methods for decision tree learning," in *Advances in Pattern Recognition, Joint IAPR International Workshops SSPR 98 and SPR 98*, 1998, pp. 475–482.
[89] S. Wang, F. Min, Z. Wang, and T. Cao, "OFFD: Optimal flexible frequency discretization for naive bayes classification," in

Proceedings of the 5th International Conference on Advanced Data Mining and Applications, ser. ADMA, 2009, pp. 704–712.

[90] S. Monti and G. F. Cooper, "A multivariate discretization method for learning bayesian networks from mixed data," in *Proceedings on Uncertainty in Artificial Intelligence (UAI)*, 1998, pp. 404–413.

[91] J. Gama, L. Torgo, and C. Soares, "Dynamic discretization of continuous attributes," in *Proceedings of the 6th Ibero-American Conference on AI: Progress in Artificial Intelligence*, ser. IBERAMIA, 1998, pp. 160–169.

[92] P. Pongaksorn, T. Rakthanmanon, and K. Waiyamai, "DCR: Discretization using class information to reduce number of intervals," in *Proceedings of the International Conference on Quality issues, measures of interestingness and evaluation of data mining model (QIMIE)*, 2009, pp. 17–28.

[93] S. Monti and G. Cooper, "A latent variable model for multivariate discretization," in *Proceedings of the Seventh International Workshop on AI & Statistics (Uncertainty)*, 1999.

[94] H. Wei, "A novel multivariate discretization method for mining association rules," in *2009 Asia-Pacific Conference on Information Processing (APCIP)*, 2009, pp. 378–381.

[95] A. An and N. Cercone, "Discretization of Continuous Attributes for Learning Classification Rules," ser. Lecture Notes in Artificial Intelligence, vol. 1574, 1999, pp. 509–514.

[96] S. Jiang and W. Yu, "A local density approach for unsupervised feature discretization," in *Proceedings of the 5th International Conference on Advanced Data Mining and Applications*, ser. ADMA, 2009, pp. 512–519.

[97] E. J. Clarke and B. A. Barton, "Entropy and MDL discretization of continuous variables for bayesian belief networks," *International Journal of Intelligent Systems*, vol. 15, pp. 61–92, 2000.

[98] M.-C. Ludl and G. Widmer, "Relative unsupervised discretization for association rule mining," in *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, ser. The Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), 2000, pp. 148–158.

[99] A. Berrado and G. C. Runger, "Supervised multivariate discretization in mixed data with random forests," in *ACS/IEEE International Conference on Computer Systems and Applications (ICCSA)*, 2009, pp. 211–217.

[100] F. Jiang, Z. Zhao, and Y. Ge, "A supervised and multivariate discretization algorithm for rough sets," in *Proceedings of the 5th international conference on Rough set and knowledge technology*, ser. RSKT, 2010, pp. 596–603.

[101] J. W. Grzymala-Busse and J. Stefanowski, "Three discretization methods for rule induction," *International Journal of Intelligent Systems*, vol. 16, no. 1, pp. 29–38, 2001.

[102] F. E. H. Tay and L. Shen, "A modified chi2 algorithm for discretization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, pp. 666–670, 2002.

[103] W.-L. Li, R.-H. Yu, and X.-Z. Wang, "Discretization of continuous-valued attributes in decision tree generation," in *Proocedings of the Second International Conference on Machine Learning and Cybernetics (ICMLC)*, 2010, pp. 194–198.

[104] F. Muhlenbach and R. Rakotomalala, "Multivariate supervised discretization, a neighborhood graph approach," in *Proceedings of the 2002 IEEE International Conference on Data Mining*, ser. ICDM, 2002, pp. 314–320.

[105] W. Zhu, J. Wang, Y. Zhang, and L. Jia, "A discretization algorithm based on information distance criterion and ant colony optimization algorithm for knowledge extracting on industrial database," in *Proceedings of the 2010 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2010, pp. 1477–1482.

[106] A. Gupta, K. G. Mehrotra, and C. Mohan, "A clustering-based discretization for supervised learning," *Statistics & Probability Letters*, vol. 80, no. 9-10, pp. 816–824, 2010.

[107] R. Giráldez, J. Aguilar-Ruiz, J. Riquelme, F. Ferrer-Troyano, and D. Rodríguez-Baena, "Discretization oriented to decision rules generation," in *Frontiers in Artificial Intelligence and Applications 82*, 2002, pp. 275–279.

[108] Y. Sang, K. Li, and Y. Shen, "EBDA: An effective bottom-up discretization algorithm for continuous attributes," in *Proceedings of the 10th IEEE International Conference on Computer and Information Technology (CIT)*, 2010, pp. 2455–2462.

[109] J.-H. Dai and Y.-X. Li, "Study on discretization based on rough set theory," in *Proceedings of the First International Conference on Machine Learning and Cybernetics (ICMLC)*, 2002, pp. 1371–1373.

[110] L. Nemmiche-Alachaher, "Contextual approach to data discretization," in *Proceedings of the International Multi-Conference on Computing in the Global Information Technology (ICCGI)*, 2010, pp. 35–40.

[111] C.-W. Chen, Z.-G. Li, S.-Y. Qiao, and S.-P. Wen, "Study on discretization in rough set based on genetic algorithm," in *Proceedings of the Second International Conference on Machine Learning and Cybernetics (ICMLC)*, 2003, pp. 1430–1434.

[112] J.-H. Dai, "A genetic algorithm for discretization of decision systems," in *Proceedings of the Third International Conference on Machine Learning and Cybernetics (ICMLC)*, 2004, pp. 1319–1323.

[113] S. A. Macskassy, H. Hirsh, A. Banerjee, and A. A. Dayanik, "Using text classifiers for numerical classification," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2 (IJCAI)*, 2001, pp. 885–890.

[114] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, "KEEL: a software tool to assess evolutionary algorithms for data mining problems," *Soft Computing*, vol. 13, no. 3, pp. 307–318, 2009.

[115] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2011.

[116] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[117] K. J. Cios, L. A. Kurgan, and S. Dick, "Highly scalable and robust rule learner: performance evaluation and comparison," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 36, pp. 32–53, 2006.

[118] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," *Machine Learning*, vol. 38, no. 3, pp. 257–286, 2000.

[119] D. W. Aha, Ed., *Lazy Learning*. Springer, 2010.

[120] E. K. Garcia, S. Feldman, M. R. Gupta, and S. Srivastava, "Completely lazy learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1274–1285, 2010.

[121] R. Rastogi and K. Shim, "Public: A decision tree classifier that integrates building and pruning," *Data Mining and Knowledge Discovery*, vol. 4, pp. 315–344, 2000.

[122] W. W. Cohen, "Fast Effective Rule Induction," in *Proceedings of the Twelfth International Conference on Machine Learning (ICML)*, 1995, pp. 115–123.

[123] J. A. Cohen, "Coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, pp. 37–46, 1960.

[124] R. C. Prati, G. E. A. P. A. Batista, and M. C. Monard, "A survey on graphical methods for classification predictive performance evaluation," *IEEE Transaction on Knowledge and Data Engineering, in press DOI: 10.1109/TKDE.2011.59*, 2011.

[125] A. Ben-David, "A lot of randomness is hiding in accuracy," *Engineering Applications of Artificial Intelligence*, vol. 20, pp. 875–885, 2007.

[126] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[127] S. García and F. Herrera, "An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons," *Journal of Machine Learning Research*, vol. 9, pp. 2677–2694, 2008.

[128] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.

[129] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, pp. 80–83, 1945.

Salvador García received his M.Sc. and Ph.D. in computer science from the University of Granada, Granada, Spain, in 2004 and 2008, respectively.

He is currently an Assistant Professor in the Department of Computer Science, University of Jaén, Jaén, Spain. He has had more than 25 papers published in international journals. He has co-edited two special issues of international journals on different Data Mining topics. His research interests include data mining, data reduction, data complexity, imbalanced learning, semi-supervised learning, statistical inference and evolutionary algorithms.

Julián Luengo received his M.Sc. in computer science and Ph.D. from the University of Granada, Granada, Spain, in 2006 and 2011 respectively. He is currently an Assistant Professor in the Department of Civil Engineering, University of Burgos, Burgos, Spain. His research interests include machine learning and data mining, data preparation in knowledge discovery and data mining, missing values, data complexity and fuzzy systems.

José Antonio Sáez

received his M.Sc. in Computer Science from the University of Granada, Granada, Spain, in 2009. He is currently a Ph.D. student in the Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain. His research interests include data mining, data preprocessing, fuzzy rule based systems and imbalanced learning.

Victoria López received his M.Sc. in Computer Science from the University of Granada, Granada, Spain, in 2009. She is currently a Ph.D. student in the Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain. Her research interests include data mining, classification in imbalanced domains, fuzzy rule learning and evolutionary algorithms.

Francisco Herrera received his M.Sc. in Mathematics in 1988 and Ph.D. in Mathematics in 1991, both from the University of Granada, Spain.

He is currently a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. He has had more than 200 papers published in international journals. He is coauthor of the book “Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases” (World Scientific, 2001).

He currently acts as Editor in Chief of the international journal “Progress in Artificial Intelligence” (Springer) and serves as area editor of the Journal Soft Computing (area of evolutionary and bioinspired algorithms) and International Journal of Computational Intelligence Systems (area of information systems). He acts as associated editor of the journals: IEEE Transactions on Fuzzy Systems, Information Sciences, Advances in Fuzzy Systems, and International Journal of Applied Metaheuristics Computing; and he serves as member of several journal editorial boards, among others: Fuzzy Sets and Systems, Applied Intelligence, Knowledge and Information Systems, Information Fusion, Evolutionary Intelligence, International Journal of Hybrid Intelligent Systems, Memetic Computation, Swarm and Evolutionary Computation.

He received the following honors and awards: ECCAI Fellow 2009, 2010 Spanish National Award on Computer Science ARITMEL to the “Spanish Engineer on Computer Science”, and International Cajastur “Mamdani” Prize for Soft Computing (Fourth Edition, 2010).

His current research interests include computing with words and decision making, data mining, data preparation, instance selection, fuzzy rule based systems, genetic fuzzy systems, knowledge extraction based on evolutionary algorithms, memetic algorithms and genetic algorithms.