J Intell Inf Syst (2013) 40:327–348
DOI 10.1007/s10844-012-0228-1

UniDis: a universal discretization technique

Yu Sang · Yingwei Jin · Keqiu Li · Heng Qi

Received: 2 November 2011 / Revised: 30 October 2012 / Accepted: 30 October 2012 /
Published online: 22 November 2012

Abstract Discretization techniques have played an important role in machine learning and data mining as most methods in such areas require that the training data set contains only discrete attributes. Data discretization unification (DDU), one of the state-of-the-art discretization techniques, trades off classification errors and the number of discretized intervals, and unifies existing discretization criteria. However, it suffers from two deficiencies. First, the efficiency of DDU is very low as it conducts a large number of parameters to search good results, which does not still guarantee to obtain an optimal solution. Second, DDU does not take into account the number of inconsistent records produced by discretization, which leads to unnecessary information loss. To overcome the above deficiencies, this paper presents a ***Uni****versal* ***Dis****cretization technique*, namely ***UniDis***. We first develop a non-parametric normalized discretization criteria which avoids the effect of relatively large difference between classification errors and the number of discretized intervals on discretization results. In addition, we define a new entropy-based measure of inconsistency for multi-dimensional variables to effectively control information loss while producing a concise summarization of continuous variables. Finally, we propose a heuristic algorithm to guarantee better discretization based on the non-parametric normalized discretization criteria and the entropy-based inconsistency. Besides theoretical analysis, experimental results demonstrate that our approach is statistically comparable to DDU evaluated by a popular statistical test and it yields a better discretization scheme which significantly improves the accuracy of

Y. Sang · K. Li (✉) · H. Qi
School of Computer Science and Technology, Dalian University of Technology,
No.2, Linggong Road, Dalian, 116024, China
e-mail: likeqiu@gmail.com

Y. Jin
School of Management, Dalian University of Technology, Dalian, 116024, China

classification than previously other known discretization methods except for DDU by running J4.8 decision tree and Naive Bayes classifier.

Keywords Discretization · Inconsistency · Entropy · J4.8 decision tree · Naive Bayes classifier

1 Introduction

Data mining is a broad area that integrates techniques from several fields including machine learning, statistics, pattern recognition, artificial intelligence, and database systems, for the analysis of large volumes of data. There have been a large number of data mining algorithms rooted in these fields to perform different data analysis tasks. For example, the novel constrained *k*NN query method (Mahady et al. 2010) is proposed to guarantee that the number of cells that are accessed to compute the constrained *k*NNs is minimal, and can be several times faster than the previous method. Jin and Qu (2009) propose a multi-dimension, multi-objective optimum dynamic programming method under circumstances of complicated information. It is relatively more universal than other ones, and the optimal result is more accurate. Wang and Zaniolo (2000) proposes a fast decision tree classifier using multivariate predictions called CMP, which achieves better decision tree rules.

Data discretization is one of the preprocessing techniques used frequently in data mining, machine learning and knowledge discovery (Hand et al. 2001; Cios and Kurgan 2007; Ling and Zhang 2002; Quinlan 1993, 1986). Many real-world data mining tasks involve continuous attributes. However, almost all of data mining techniques can not handle such attributes. Therefore, it is necessary and important to slice the value domain of each continuous attribute into a number of intervals to generate attributes with a small number of distinct values. Some modern classification systems such as ID3 (Quinlan 1986) and C4.5 decision trees (Quinlan 1993) have also implemented discretization methods as built-in functions. A good discretization algorithm not only produces a concise summarization of continuous attributes to help the experts and users understand the data but also makes learning efficient and effective (Liu et al. 2002).

Data discretization has been extensively studied (Dougherty et al. 1995; Kerber 1992; Liu and Setiono 1997; Tay and Shen 2002; Su and Hsu 2005; Fayyad and Irani 1993; Ching et al. 1995; Kurgan and Cios 2004; Tsai et al. 2008; Liu et al. 2004; Biba et al. 2007; Schmidberger and Frank 2005; Boulle 2006; Bondu et al. 2008). DDU proposed by Jin et al. (2007) is one of the most efficient and effective discretization techniques, which uses Minimum Description Length Principle (MDLP) (Fayyad and Irani 1993; Hansen and Yu 2001) to accomplish the discretization scheme by trading off classification errors and the number of discretized intervals generated by discretization. Clearly, the more the discretization intervals discretized, the fewer the number of classification errors is; and vice versa. In DDU, it is also proved that existing discretization methods based on information theory and statistical independence are approximately equivalent. They propose a parametrized discretization criteria to unify several existing discretization criteria and provide a flexible framework to access a potentially infinite of discretization criteria. They also design a dynamic programming algorithm to select the best cut points based on the proposed unified

discretization criterion and use an experimental validation approach to choose the optimal parameters.

However, DDU has two deficiencies. First, DDU is a parameter-based method and has to implement a large number of parameters to search a desirable result. Even so, it does not still guarantee to find an optimal solution as it is difficult to determine a good parameter value or scope. This can lead to two extreme cases that data may be discretized into one interval or not be discretized. Second, DDU does not consider the number of inconsistencies in multivariate discretized data sets as it only treats discretization of a single continuous variable as a 1-dimension classification problem, which may lead to unnecessary information loss after discretization.

The objective of discretization is to find an effective criterion to select a group of good cut-points on each continuous attribute with minimal information loss, aiming to produce a concise summarization of continuous variables and making learning efficient and effective. In this paper, we present a universal discretization technique (*UniDis*). The main **contributions** of this paper are summarized as follows:

1. We propose a non-parametric normalized discretization criteria which avoids the effect of relatively large difference between classification errors and the number of discretized intervals on discretization results.
2. We propose a new entropy-based measure of inconsistency for multiple attributes to effectively control information loss while producing a concise summarization of continuous variables.
3. We propose a heuristic algorithm to guarantee better discretization based on the non-parametric normalized discretization criteria and the entropy-based inconsistency.
4. We conduct a simulation experiment to evaluate our discretization method. The simulation results show that our approach is statistically comparable to DDU evaluated by a popular statistical test and it significantly improves the mean accuracy of classification than existing ones except for DDU.

The remainder of this paper is organized as follows. We introduce related work in Section 2. Section 3 presents our proposed method. Experiments and performance evaluation are introduced in Section 4. Finally, we summarize our work and conclude this paper in Section 5.

2 Related work

Existing discretization techniques can be divided into top-down vs. bottom-up, while top-down can be further classified into unsupervised vs. supervised (Dougherty et al. 1995). Top-down techniques start from the initial interval and recursively split it into smaller intervals, while bottom-up techniques begin with the set of single value intervals and iteratively merge adjacent intervals. Unsupervised methods provide no class information, such as EQW and EQF (Dougherty et al. 1995), KDE (Biba et al. 2007), TDE (Schmidberger and Frank 2005). In the unsupervised methods, continuous ranges are divided into subranges by the user specified width (range of values) or frequency (number of instances in each interval). The former two techniques (EQW and EQF) starting with naive methods can be implemented with a low computational cost. However, This may not give good results in cases

where the distribution of the continuous values is not uniform. Furthermore it is vulnerable to outliers as they affect the ranges significantly. Specifically, EQW method involves sorting the observed values of a continuous attribute and dividing the range of observed values for the variable into k equally sized bins, where k is a parameter supplied by the user. EQF method divides a continuous attribute into k bins, and each bin contains N/k (possibly duplicated) adjacent values, where N is the number of instances. The latter two techniques (KDE and TDE) are state-of-the-art unsupervised top-down techniques, which use density estimators to select the best cut points and automatically adapt subintervals to the data. They determine the discretized number of intervals by the cross-validated log-likelihood. While, supervised methods provide class information with each attribute value and they are much more sophisticate, such as the Chi2-based heuristic algorithms (Kerber 1992; Liu and Setiono 1997; Tay and Shen 2002; Su and Hsu 2005), class-attribute interdependency-based methods like CADD (Ching et al. 1995), CAIM (Kurgan and Cios 2004), CACC (Tsai et al. 2008), OCDD (Liu et al. 2004), and entropy-based discretization (Fayyad and Irani 1993). The Chi2-based methods are famous bottom-up supervised discretization techniques based on statistical independence. The chi-square statistic is used to determine whether the current point is to be moved or not. These algorithms trade off the number of intervals with the number of inconsistent instances and control the process of discretization by introducing inconsistency with the aim to control the degree of misclassification. Class-attribute interdependency-based methods are distinguished top-down supervised discretization techniques with the objective to maximize the interdependence between the class and the continuous-valued attribute and to generate a possibly minimal number of discrete intervals. Entropy-based method recursively selects the cut-points on each target attribute to minimize the overall entropy and determines the appropriate number of intervals by using Minimum Description Length Principle (MDLP) (Fayyad and Irani 1993).

Recently, many researchers have focused on the production of new discretization techniques, i.e., DDU (Jin et al. 2007), MODL (Boulle 2006) and SSDM (Bondu et al. 2008). Ruoming Jin et al. present a latest unification technique of data discretization (DDU). They prove that discretization methods based on information theory and statistical independence are approximately equivalent. A parameterized goodness function is derived to unify six discretization criteria, providing a flexible framework to access a potentially infinite of goodness functions. MODL is another latest discretization method. It builds an optimal criterion based on a Bayesian model. Three algorithms are developed to find the optimal criteria. Beyond the supervised and unsupervised methods, Bondu et al. (2008) developed only one semi-supervised method lately. It is based on the MODL framework and discretizes the numerical domain of a continuous input variable, while keeping the information relative to the prediction of classes.

3 A universal discretization technique

In this section, we present a universal discretization technique. First, we analyze the motivation in detail in Section 3.1. Then, we describe our proposed method in Section 3.2. Finally, a heuristic algorithm is presented in Section 3.3.

3.1 Motivation

In this section, we analyze the motivation. First, we state the problem of discretization. A discretization task requires a training data consisting of N instances, where each instance belongs to only one of S classes. Next, there exists a discretization scheme D, which discretizes the continuous domain of attribute into I intervals bounded by the pairs of numbers:

$$D : \{[d_0, d_1], (d_1, d_2], \cdots, (d_{I-1}, d_I]\}$$

where d_0 is the minimal value and d_I is the maximal value of a continuous attribute. The values in D are arranged in ascending order. For the purpose of discretization, the entire dataset is projected onto the targeted continuous attribute. The result of such a projection is a two dimensional contingency table, see Table 1, with I rows and S columns. Each row corresponds to an initial data interval, and each column corresponds to a different class. N_{ij} represents the number of instances with jth class in the ith interval R_i. $N_{\cdot j}$ is the total number of instances belonging to the jth class. $N_{i\cdot}$ is the total number of instances that are within the interval R_i.

As stated in DDU, finding the best discretization is finding the best trade-off between classification errors and the number of discretized intervals generated by discretization; DDU uses MDLP to achieve the discretization scheme, and it unifies six discretization criteria by introducing two parameters. MDLP associates a *cost* with each discretization, for the detailed deviation for each discretization function see Jin et al. (2007). Formally, the unified parametrized *cost* function is defined as follows:

$$cost_{\alpha,\beta}(D) = \sum_{i=1}^{I} N_{i\cdot} H_\beta(R_i) + \alpha(I-1)(S-1) f(\beta) \tag{1}$$

where $H_\beta(R_i) = \sum_{j=1}^{S} \frac{N_{ij}}{N_{i\cdot}} \Big[1 - \Big(\frac{N_{ij}}{N_{i\cdot}}\Big)^\beta\Big] \Big/ \beta$ is called the generalized entropy (Mussard et al. 2003) of interval R_i, $f(\beta) = \Big[1 - \big(\frac{1}{N}\big)^\beta\Big] \Big/ \beta$, $\alpha > 0$, and $0 < \beta \leq 1$. We can see from (1) that $cost_{\alpha,\beta}(D)$ evaluates classification errors by $\sum_{i=1}^{I} N_{i\cdot} H_\beta(R_i)$ and the number of discretized intervals (penalty for discretization) by $\alpha(I-1)(S-1) f(\beta)$. Intuitively, when a classification error increases, the penalty decreases and vice versa.

However, DDU has two deficiencies. First, the efficiency is very low as it conducts a large number of parameters to search good results. In other words, it is difficult to

Table 1 Notations of contingency table

Intervals	Class label				Sum of row
	Class 1	Class 2	$\cdots$	Class S	
$R_1 : [d_0, d_1]$	N_{11}	N_{12}	$\cdots$	N_{1S}	$N_{1\cdot}$
$R_2 : (d_1, d_2]$	N_{21}	N_{22}	$\cdots$	N_{2S}	$N_{2\cdot}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$R_I : (d_{I-1}, d_I]$	N_{I1}	N_{I2}	$\cdots$	N_{IS}	$N_{I\cdot}$
Sum of column	$N_{\cdot 1}$	$N_{\cdot 2}$	$\cdots$	$N_{\cdot S}$	N (total)

determine a good parameter value to guarantee an optimal solution. Although DDU can find the best discretization theoretically, it is not operational in practice. In the following, we do simply analysis. To facilitate our discussion, we take the limit $\beta \to 0$; $\lim_{\beta\to 0} H_\beta(R_i) = H(R_i)$ and $\lim_{\beta\to 0} f(\beta) = \ln N$ (see Theorem 2 in Appendix).

The paper presents a dynamic programming strategy to find the best discretization to minimize the parameterized *cost* function. Let sub-scheme be $D[i : i+k] : \{[d_{i-1}, d_i], \cdots, (d_{i+k-1}, d_{i+k}]\}$ and $F\big(D[i : i+k]\big) = \Big(\sum_{r=i}^{i+k} N_{r\cdot}\Big) \times H_\beta\Big(\bigcup_{r=i}^{i+k} R_r\Big)$. Let $opt(i, i+k)$ be the optimum which corresponds to the best discretization and can be calculated recursively as follows:

$$opt(i, i+k) = \min\Big(F(D[i : i+k]), \min_{0\le l\le k-1}\big(opt(i, i+l) + opt(i+l+1, i+k) + \alpha(S-1)\ln N\big)\Big)$$

where $1 \le i \le I-1$ and $1 \le k \le I-1$. The algorithm ultimately returns $opt(1, I)$ by calculating $opt(i, i+k)$ recursively for threeply loops: $i = 1$ to $I-1$, $k = 1$ to $I-i$ and $l = 0$ to $k-1$. If

$$F\big(D[1 : I]\big) < \min_{0\le l\le I-2}\big(opt(1, 1+l) + opt(2+l, I) + \alpha(S-1)\ln N\big) \qquad (2)$$

then this means that data would be discretized into one interval, i.e., $\{d_0, d_I\}$, which leads to over-discretization and makes learning accuracy worse. Note that $F\big(D[1 : I]\big) = F(D)$. However, whether the inequality (2) is true or not would be determined by α. Whereas, kinds of complicated data sets have different information themselves, i.e., N, S and the entropy that can reflect the class distribution, so it is difficult to find a good α value or scope to achieve a good trade-off between classification errors and the number of discretized intervals. Obviously, it is more unreasonable that DDU apply an experimental validation approach to choose the optimal parameters in the same parameter domain for different data sets. In addition, since $H_\beta(R_i)$ and $f(\beta)$ are the monotonous decreasing functions related to β (see Theorem 1 in Appendix), similar conclusions can also be derived when β takes on real values from the interval $(0, 1)$.

We can illustrate this through the following two examples. We take the age dataset1 in Table 2 as the training data consisting of 18 instances and 2 classes. Obviously, the best discretization scheme is that the age dataset1 is divided into six intervals (denoted by 1, 2, 3, 4, 5 and 6, respectively): [15.50, 19.65], (19.65, 23.05], (23.05, 24.60], (24.60, 27.55], (27.55, 37.40] and (37.40, 48.00]. Interval 1 contains instances 1–3, interval 2 has instances 4–6, etc.

If DDU is applied to discretize the age dataset1 (let $\beta \to 0$), $opt(1, 6)$ can be calculated recursively according to the dynamic programming algorithm as follows:

$$opt(1, 6) = \min\Big(F\big(D[1 : 6]\big), \min_{0\le l\le 4}\big(opt(1, 1+l) + opt(2+l, 6) + \alpha\cdot(2-1)\cdot\ln 18\big)\Big)$$

Table 2 Age dataset1

Person ID	Age	Occupation (target class)
1	15.5	Education
2	16.2	Education
3	19.0	Education
4	20.3	Work
5	22.1	Work
6	23.0	Work
7	23.1	Education
8	23.5	Education
9	24.2	Education
10	25.0	Work
11	26.4	Work
12	27.1	Work
13	28.0	Education
14	30.0	Education
15	35.2	Education
16	39.6	Work
17	43.8	Work
18	48.0	Work

Specifically calculated results as follows:

$$
\begin{aligned}
&opt(1,2) = \min(6, 2.89\alpha)\\
&opt(1,3) = \min\big(8.4, opt(1,2) + 2.89\alpha\big)\\
&opt(1,4) = \min\big(\min(12, opt(1,3) + 2.89\alpha), \min(12, opt(1,2) + 2.89\alpha)\big)\\
&opt(1,5) = \min\big(\min(14.6, opt(1,4) + 2.89\alpha), \min(14.6, opt(1,2) + opt(1,3) + 2.89\alpha)\big)\\
&opt(1,6) = \min\big(\min(18, opt(1,5) + 2.89\alpha), \min(18, opt(1,2) + opt(1,4) + 2.89\alpha),\\
&\min(18, 2opt(1,3) + 2.89\alpha)\big)
\end{aligned}
$$

where $opt(1,2) = opt(2,3) = opt(3,4) = opt(4,5) = opt(5,6)$, $opt(1,3) = opt(2,4) = opt(4,6)$, $opt(1,4) = opt(2,5) = opt(3,6)$ and $opt(1,5) = opt(2,6)$.

By computing, we can derive that if $\alpha > 1.384$, then

$$F\big(D[1:6]\big) < \min_{0 \le l \le 4} \big(opt(1, 1+l) + opt(2+l, 6) + 2.89\alpha\big)$$

Therefore, the age dataset1 would be discretized into one interval ([15.5,48.0]) in this case and this goes to one extreme. In this example, we let $\beta \to 0$. For other values of β, we can receive the similar results when β is given, since $H_\beta(R_i)$ and $f(\beta)$ are the monotonous decreasing functions related to β in the interval $(0, 1)$ (see Theorem 1 in Appendix).

Next, we take another example to illustrate that DDU may go to another extreme. The age dataset2 in Table 3 is as the training data consisting of 12 instances and 3 classes. Good discretization need to trade off classification errors and the number of discretized intervals. Therefore, the best discretization scheme is that the age dataset2 would be divided into two intervals: [8.0, 17.0] and (17.0, 23.6] because it has smaller interval number and fewer classification errors, intuitively. However, if $\alpha < 0.127$ obtained by calculating recursively according to the dynamic programming strategy of DDU, the age dataset2 would be divided into four intervals: [8.0, 11.85], (11.85, 12.05], [12.05, 17.0] and (17.0, 23.6], which only considers classification errors and generates the most intervals.

Table 3 Age dataset2

Person ID	Age	Other	Education (target class)
1	8.0	···	Elementary school
2	11.0	···	Elementary school
3	11.5	···	Elementary school
4	11.6	···	Elementary school
5	12.1	···	Junior high school
6	12.2	···	Elementary school
7	12.3	···	Elementary school
8	15.0	···	Elementary school
9	19.0	···	Bachelor
10	19.5	···	Bachelor
11	20.7	···	Bachelor
12	23.6	···	Bachelor

Through analyzing the above two examples, DDU may lead to the two extremes. The reason is that it only considers either classification errors or the discretized interval number when selecting the bad parameter scope. This means that there is a relatively large difference between classification errors and the number of discretized intervals. If a classifier is learning with such a discretized data set, the accuracy would be worse.

The second deficiency of DDU is that it does not take into account the number of inconsistencies in multivariate discretized data sets, which may lead to unnecessary information loss after discretization. As it is well known, discretization is accompanied by information loss. However, most of current efforts only consider the number of classification errors aiming at discretizing one single continuous attribute, not taking into account the number of inconsistencies in multivariate discretized data sets.

3.2 Universal discretization method

In this section, we present a universal discretization method to overcome the above deficiencies. First, a non-parametric normalized discretization criteria is proposed to avoid parameter search of low efficiency and the effect of relatively large difference between classification errors and the number of discretized intervals on discretization scheme as large difference means that it selects bad choices of parameter values, which leads to poor discretization results.

Non-parametric normalized discretization criteria First, we analyze the range of the two components in (1). According to the extremum property of Entropy (Cover and Thomas 2006), we have $H_\beta(R_i) \leq \log S$. So, the range of $\sum_{i=1}^{I} N_{i\cdot} H_\beta(R_i)$ is $[0, N \times \log S]$. Besides, according to Theorem 1 in Appendix, we have

$$\max_\beta \left\{ f(\beta) \right\} = \max_\beta \left\{ \frac{1 - (\frac{1}{N})^\beta}{\beta} \right\} = \lim_{\beta \to 0} \frac{1 - (\frac{1}{N})^\beta}{\beta} = \ln N$$

So, the range of the other component takes $[0, \alpha(I-1)(S-1)\ln N]$. As I and S are smaller, it is easy to see that the difference between the two ranges is large. To avoid

parameter-dependent and reduce the effect of difference between classification errors and the number of discretized intervals on discretization results, we aims to drop the parameters and normalize the both by mapping their value domain into [0,1].

The entropy $H(R_i)$ can evaluate classification errors and I measures the number of discretized intervals. Thus, we can define the new non-parametric normalized discretization criteria (termed UDT) referring to the two ranges, as follows:

$$UDT = \sum_{i=1}^{I} H(R_i) \Big/ I \log S + (I-1) \Big/ \left[(S-1)\ln N\right] \tag{3}$$

Note that if I is relatively larger, it may happen that $\Phi = (I-1) \Big/ \left[(S-1)\ln N\right] > 1$. Thus, we have to require a maximal number of the discretized intervals to enable $\Phi \in [0, 1]$. In order to be reliable, it requires that every cell of the contingency table has an expected value of at least 5. This reliability constraint is equivalent to a minimum frequency constraint for each interval of the discretization. The purpose is to approximate the true class attribute distribution from the observed distribution of the training sample on the basis of intervals. This process can be considered as an inductive algorithm, therefore subject to overfitting (Tsai et al. 2008; Boulle 2004). In order to prevent overfitting, a solution is to increase the minimum frequency by constraining the intervals to have a frequency greater than the square root of the sample size $\sqrt{N}$. A detailed description can be found in Boulle (2004). So, we can rewrite (3) as follows:

$$UDT = \sum_{i=1}^{I} H(R_i) \Big/ I \log S + (I-1) \Big/ \left[(S-1)\sqrt{N}\right] \tag{4}$$

We can see from (4) that this new criteria is parameterless and normalize the both into domain [0, 1], which can reduce the difference of both and avoid extreme situations of the examples appeared in Section 3.1. If our proposed algorithm presented in Section 3.3 based on the new discretization criteria is applied to discretize the age dataset1 and dataset2 in Section 3.1, the best discretization claimed would be achieved. Actually, the main components in desired discretization criterion is the measure of classification errors and the measure of the discretized interval number. Indeed, (4) is equivalent to trade off these two terms with normalization.

To sum up, the new non-parametric normalized discretization criteria has the following properties:

- It is parameterless and only implements discretization one time to find better scheme, unlike DDU that searches a large number of parameters to find desired results and does not still guarantee to obtain an optimal solution.
- It conducts with the effect of information themselves of data, i.e., sample points N and the number of classes S. Intuitively, the larger sample points N and the number of classes S, the more the number of classification errors, and so need slightly more discretized interval number, and vice versa.

For bottom-up discretization, we find the adjacent two intervals in all the candidate interval pairs, which have minimal UDT value after merging them, and then we first merge them; for top-down discretization, we find the cut point in all the candidate cut points, which have minimal UDT value after splitting an interval, and then we first add it.

In the following, we propose an entropy-based measure of inconsistency that considers the number of inconsistencies generated by discretization in multivariate discretized data sets.

Entropy-based measure of inconsistency Authors Kerber (1992), Liu and Setiono (1997), Tay and Shen (2002), Su and Hsu (2005) have proposed the measures of inconsistencies as stopping rules to control information loss and automate the discretization process. Nevertheless, these measures of inconsistency either has low fidelity of the original data set or discretize data in such a conservative way that it does not allow any loss of information. The detailed shortcomings can be listed as follows:

- In Chi2 (Liu and Setiono 1997), some input variables are removed according to the larger inconsistency count. However, these results are obtained on the basis of decreasing the fidelity of the original data because the calculation of inconsistency rate in Chi2 is the total number of the instances minus the largest number of the instances of class label, considering only the largest number, not the difference among all the number of the instances of class label.
- In Mod-Chi2 (Tay and Shen 2002), it replaces the measure of inconsistency in Chi2 with the level of consistency in RST (Pawlak 1982), which guarantees that the fidelity of the training data can be maintained to be the same after discretization. However, this measure is defined too strictly. Suppose that there are 100 inconsistent instances for a consistent data after discretization, among which 99 instances and 1 instance belong to two different classes, respectively. If the level of consistency is applied as the stopping criterion, these two adjacent intervals can not be merged. But, merging the interval pair does not generate misclassification of the training data basically.
- Ext-Chi2 (Su and Hsu 2005) introduces the measure of the relative degree of misclassification between two sets in variable precision rough sets ($VPRS$) model (Ziarko 1993) to determine the inconsistency rate. The termination criterion is defined as the point at which the discretized inconsistency rate exceeds the predefined rate determined by the original data. However, this stopping criterion may lead to more misclassification since it generates more inconsistent instances after discretization. In the process of discretization, Ext-Chi2 proceeds to discretize data by decreasing the significance level α. We find that the number of the inconsistent instances is more and more with the reduction in the significance level α from 0.5 to 0.0005. However, the discretized inconsistency rate still does not exceed 0.4 even if α has been decreased to 0.0005. Thus, more misclassifications are generated when data continues to be discretized.

In this paper, we propose a novel entropy-based measure of inconsistency for multi-dimensional variables. Entropy is a measure of the uncertainty (inconsistency)

associated with random variables and can commendably measure information loss. We first formally introduce a notion of entropy as follows:

Definition 1 The entropy of a discrete random variable X is a measure of the amount of uncertainty associated with the value of X, is defined in (5)

$$H(X) = -\sum_{x \in \Omega_x} p(x) \log p(x) \tag{5}$$

where Ω_x is the set of all messages $\{x_1, x_2, \ldots, x_n\}$ that X could be, and $p(x)$ is the probability of X given some $x \in \Omega_x$.

Let $A = \{a_1, a_2, \cdots, a_m\}$. m is the number of attribute variables and a_i denotes the ith continuous variable, $1 \leq i \leq m$. Based on Definition 1, the entropy of multi-dimensional variables is defined as follows:

Definition 2 The extended entropy of multi-dimensional variables is defined

$$H(A) = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_m=1}^{I_m} \sum_{j=1}^{S} \frac{N_{i_1 i_2 \cdots i_m j}}{N} \log \frac{N}{N_{i_1 i_2 \cdots i_m j}} \tag{6}$$

where I_i is the discretized interval number of the ith attribute and a_i denotes the ith attribute, $N_{i_1 i_2 \cdots i_m \cdot}$ is the number of instances in the input variables cell $(a_1, a_2, \cdots, a_m)$, and $N_{i_1 i_2 \cdots i_m j}$ is the number of instances of class j in the input variables cell $(a_1, a_2, \cdots, a_m)$. This entropy-based inconsistency criteria can exactly measure the uncertainty of entire data in the process of discretization. The heuristic algorithm presented in Section 3.3 just uses the entropy-based measure of inconsistency to control information loss and decide when the algorithm stops discretization.

3.3 A heuristic algorithm

In this section, we design a heuristic algorithm (*UniDis*). The objective of this algorithm is to adopt our proposed UDT criteria and entropy-based measure of inconsistency to find best discretization scheme with minimal information loss. The algorithm includes two phases. The first phase is the initialization. The second phase is the optimization stage of the discretization. We explain the rationale behind each step as follows:

3.3.1 Initialization

A good initialization can lead to fewer updates. So, we define a reasonable heuristic initial number of the discretized intervals for each input variable as follows:

$$I_{\text{initial}(i)} = \frac{\sqrt{S \times N}}{m} + MI(a_i) \tag{7}$$

where $MI(a_i) = \sum_{j=1}^{I_i'} \sum_{r=1}^{S} N_{jr} \log_2 \frac{N_{jr}}{N_{j\cdot} N_{\cdot r}}$, I_i' denotes the original number of intervals. It is called Mutual Information (Ching et al. 1995) between input variables and

output variable. It reflects the interdependence. In fact, it is known that the Mutual Information between input and output variables takes maximum value if input and output are totally dependent, and the mutual information takes minimum value if they are totally independent. Note that, originally each distinct value of an attribute is considered to be one interval.

Generally, the fewer the number of classification errors for univariate, the more discretization intervals created. Besides, the number of intervals is also close to some prior information. Intuitively, the fewer the sample points and the number of classes, and the larger the number of attributes, then there should be fewer the number of the discretized intervals, and vice versa. Therefore, we use $\sqrt{S \times N}/m$ as one component of $I_{\text{initial}(i)}$, and the reason for square root is that $S \times N$ is much larger than m in most cases and it need to be reduced the order of magnitude. In addition, each input variable has a certain important degree with regard to output variable. In other words, there is an interdependence between them. The smaller the Mutual Information value, the lower the interdependence between the class labels and the continuous attribute, and thus the number of the discretized intervals can be fewer, and vice versa. So, (7) can synthetically reflects the initial number of intervals as a heuristic function. After creating the initial number of intervals, the algorithm finds $I_{\text{initial}(i)}$ cut points which minimize UDT value for each continuous variable.

3.3.2 Optimization

Based on initialization, we further optimize the discretization process. In this algorithm, we use the entropy-based measure of inconsistency $H(A)$ (Definition 1) to control information loss and as a stopping rule of discretization process. After initialization, the algorithm optimizes the discretization process, mainly in the following two forms:

- If $H(A) > \xi$, it shows that there is an intolerable information loss. Thus, the algorithm has to discover an optimal split from all the continuous variables with the aim to reduce information loss. ξ is a predefined tolerable amount of information loss supplied by the user.
- If $H(A) \leq \xi$, it shows that there is no information loss or a tolerable information loss. Thus, the algorithm can discover an optimal split and an optimal merge from all the continuous variables.

The heuristic algorithm *UniDis* can be described as shown in Algorithm 1.

The proposed UDT criterion plays an important role in Algorithm 1. The best split when $H(A) > \xi$ is that the algorithm searches such a cut point in all the candidate cut points, which has minimal UDT value and can reduce information loss after adding the searched cut point to data. The best merge when $H(A) \leq \xi$ is that the algorithm searches such an interval pair in all the candidate interval pairs, which has minimal UDT value and can reduce information loss after merging the searched intervals. If the searched target cut point with minimal UDT value can not reduce information loss, we have to search the next one with sub-optimal value, and so on. And, if the searched target pair of intervals with minimal UDT value increase information loss, we have to search the next one with sub-optimal value, and so on.

UDT merely reflects discretization trade-off of a single continuous attribute as a 1-dimension classification problem. While the entropy-based measure of

Algorithm 1: The heuristic algorithm (*UniDis*)

Input : Data set with m attributes, N instances and S target classes
Output: Discretization scheme with I_i intervals for each continuous attribute a_i

Initialization:
Create an initial number of intervals $I_{\text{initial}(i)}$ for each continuous attribute according to (7);
for each continuous attribute **do**
    Sort variable values in ascending order;
    Find $I_{\text{initial}(i)}$ cut points which minimize UDT value;
Optimization of the discretization:
Calculate initial $H(A)$ value of information system;
Set the predefined tolerable amount of information loss ξ;
while $H(A) > \xi$ **do**
    Search for the best split in all the continuous attributes;
    Calculate current $H(A)$ value;
while $H(A) \leq \xi$ **do**
    Search for the best merge in all the continuous attributes;
    Calculate current $H(A)$ value;

inconsistency $H(A)$ reflects the number of inconsistencies generated by discretization in multivariate discretized data sets. So, we find the best target interval pairs or cut points by UDT criterion and control information loss by $H(A)$. To sum up, the goal of discretization is to create minimal number of discretization intervals with minimal information loss for multi-dimension data sets.

The computational complexity of Algorithm 1 is analyzed as follows. For initialization, the time complexity of computing $I_{\text{initial}(i)}$ and UDT is $O(mN)$, but each sort needs $O(N \log N)$. So for m continuous attributes, the complexity of initialization is $O(mN \log N)$. For, the optimization of the discretization process, the time complexity of calculating $H(a_1, a_2, \cdots, a_m)$ is $O(mN^2)$, and the complexity of searching best split or merge is also $O(mN^2)$. Hence, the computational complexity of Algorithm 1 is $O(mN \log N + mN^2)$, i.e., $O(mN^2)$. Since m is often smaller, the complexity can be $O(N^2)$.

4 Experiments and performance evaluation

In order to evaluate *UniDis* in a real-world situation, eleven data sets are selected from the UC Irvine machine learning data repository (Hettich and Bay 1999) with numeric features and varying data sizes, including one large data sets. The data are fully consistent or correct (inconsistency rate is zero) except for artificial data sets, and contain real-life information from the medical and scientific fields which have been used widely in testing pattern recognition and machine learning methods. A summary of data sets can be found in Table 4.

Table 4 The summary of data sets

Data sets	Number of continuous attributes	Number of discrete attributes	Number of classes	Number of instances
Artificial	6	1	10	5109
Breast	9	0	2	683
Bupa	6	0	2	345
Chapman	6	0	2	200
Glass	9	0	7	214
Heart	6	7	2	296
Iris	4	0	3	150
Ionosphere	32	2	2	351
Pima	8	0	2	768
Vehicle	18	0	4	846
Wine	13	0	3	178

We compare our proposed method *UniDis* with the following techniques for performance evaluation.

1. Con: classification performance evaluation on the continuous data using data mining tools;
2. Ext-Chi2: the newest bottom-up algorithm (Su and Hsu 2005);
3. CACC: the latest top-down method (Tsai et al. 2008);
4. DDU: unified discretization technique (Jin et al. 2007);
5. MODL: one of the latest discretization techniques (Boulle 2006);
6. MDLP: entropy-based method using the minimum description length principle (Fayyad and Irani 1993);
7. EQF: a typical unsupervised top-down method (Dougherty et al. 1995).

Among the discretization methods, EQF, DDU and Ext-Chi2 require the user to specify in advance some parameters of discretization. For EQF, the number of intervals is set to 10. For Ext-Chi2, we set the level of significance to 0.9995. The parameters set of DDU is similar to Jin et al. (2007). CACC, MODL and MDLP have their respective automatic stopping rule and do not require any parameter setting.

Many machine learning methods, i.e. J4.8 decision trees (Witten and Frank 2000) and Naive Bayes classifier (Hand et al. 2001), require that the training data contains only discrete values. J4.8 is an open source Java implementation of the C4.5 algorithm in the weka data mining tool (Weka 3 Data mining software in Java 2007). Although C4.5 has implemented discretization as built-in function, it is only a simple discretization process. So, we need to investigate more effective discretization for continuous data to improve the performance of these machine learning methods. In the following experiments, each data set is discretized respectively by the eight methods mentioned above. The 10-fold cross-validation test method is applied to all data sets. Each data set is divided into ten parts, among which nine parts are used as the training sets and one as the testing set. The experiments are repeated ten times. The final predictive accuracy is taken as the average predictive accuracy. As suggested by Demsar (2006), we use the Friedman test and the Holm's post-hoc tests with significance level $\alpha = 0.05$ to statistically verify the hypothesis of performance improvement on the classification accuracy of J4.8 decision trees and Naive Bayes classifier.

Table 5 The predictive accuracy (percent) using J48 with different discretization methods

Data sets	J4.8 (mean accuracy %)							
	Con	Ext-Chi2	CACC	*UniDis*	DDU	MODL	MDLP	EQF
Artificial	57.6	56.9	60.5	62.4	60.9	61.2	59.7	56.1
Breast	94.4	96.3	94.8	97.6	95.9	95.3	95.0	93.8
Bupa	63.7	67.5	62.3	65.2	69.4	65.9	61.2	57.8
Chapman	75.5	81.0	87.0	86.0	86.0	85.0	86.0	81.0
Glass	66.7	70.6	78.6	74.1	74.8	73.7	73.2	57.9
Heart	79.7	85.6	81.3	87.9	83.4	84.6	83.4	77.9
Iris	94.4	94.7	92.6	94.7	96.9	95.2	93.3	92.6
Ionosphere	90.4	92.9	90.0	94.7	92.9	91.6	86.8	82.6
Pima	73.9	75.8	67.9	80.6	77.6	75.2	70.3	66.2
Vehicle	70.2	70.9	67.8	75.6	72.0	71.4	67.7	65.4
Wine	94.0	96.4	92.3	97.1	96.0	95.3	83.3	90.7
Mean rank	5.91	3.68	5.14	1.86	2.45	3.45	5.77	7.73

The predictive accuracy of these eight methods are presented in Tables 5 and 6. The comparison results show that on the average, *UniDis* achieves the highest classification accuracy, which demonstrates that *UniDis* can produce a high quality discretization scheme. Quick comparisons of the eight methods can be obtained by checking the mean rank in the last row in Tables 5 and 6. Each value of this row is acquired by average ranking of each discretization method for all the eleven data sets. We rank the algorithms for each data set separately, the algorithm with the best performance gets the rank of 1, the second best gets the rank of 2, and so on.

In order to obtain the statistical support, we then use the Friedman test to check if the measured mean ranks statistically reach significant differences. If the Friedman test shows that there is a significant difference, the Bonferroni-Dunn test in the

Table 6 The predictive accuracy (percent) using Naive Bayes with different discretization methods

Data sets	Naive Bayes (mean accuracy %)							
	Con	Ext-Chi2	CACC	*UniDis*	DDU	MODL	MDLP	EQF
Artificial	57.6	59.9	59.5	63.5	62.7	60.4	60.1	56.7
Breast	94.4	96.9	97.0	97.6	97.6	97.6	95.0	95.0
Bupa	55.2	62.3	64.7	78.6	79.1	75.3	63.8	60.6
Chapman	78.0	83.5	85.0	88.5	87.0	87.5	86.5	81.5
Glass	46.3	81.6	75.8	85.3	86.4	83.2	72.3	68.2
Heart	79.5	87.7	84.3	88.2	87.9	86.3	83.2	81.4
Iris	95.7	97.1	96.7	96.7	97.1	96.5	94.2	92.6
Ionosphere	91.4	91.2	93.4	95.7	94.6	92.3	93.5	93.4
Pima	73.5	83.4	85.3	87.2	86.5	86.1	76.4	72.6
Vehicle	44.9	66.2	66.9	67.9	64.7	66.7	65.8	62.7
Wine	90.6	94.7	94.5	96.2	96.9	96.5	89.5	92.4
Mean rank	7.09	4.68	4.36	1.68	2.14	3.27	5.5	6.91

Holm's post-hoc test is used to further analyze the comparisons of all the methods against *UniDis*. The Friedman statistic is described as follows:

$$\chi_F^2 = \frac{12Q}{P(P+1)}\left[\sum_j L_j^2 - \frac{P(P+1)^2}{4}\right] \tag{8}$$

where P is the number of discretization algorithms, Q is the number of data sets, $L_j = \frac{1}{Q}\sum_i u_i^j$, and u_i^j is the rank of the jth of P algorithms on the ith of Q data sets. The Friedman statistic is distributed according to χ_F^2 with $v-1$ degrees of freedom, when P and Q are big enough (as a rule of a thumb, $Q > 10$ and $P > 5$). For a smaller number of algorithms and data sets, exact critical values have been computed (Zar 1998).

For the measured mean ranks in Table 5, the corresponding value of the Friedman test is

$$\chi_F^2 = \frac{12 \cdot 11}{8 \times 9}\Big[(5.91^2 + 3.68^2 + 5.14^2 + 1.86^2 + 2.45^2 + 3.45^2 + 5.77^2 + 7.73^2) - \frac{8 \times 9^2}{4}\Big] = 71.8722$$

which is larger than the threshold 14.1. So, the visualization of the Bonferroni-Dunn test in the Holm's post-hoc test can be illustrated in Fig. 1 according to Demsar (2006). We can see that the top line in the figure is the axis on which we plot the average ranks of all the methods while a method on the left side means that it performs better. A method with rank outside the marked interval in Fig. 1A means that it is significantly different from *UniDis*. We can see from Fig. 1A that

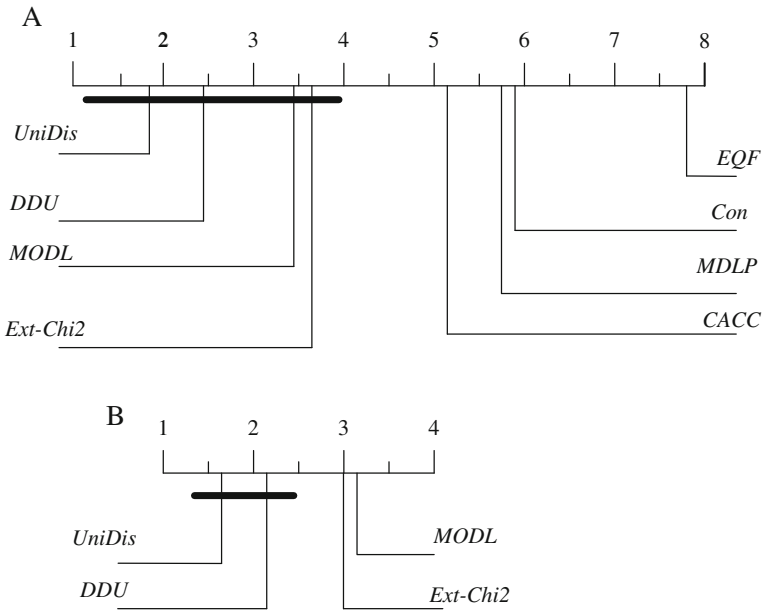

Fig. 1 Comparison of J4.8 performance with Holm's post-hoc tests ($\alpha = 0.05$)

the mean predictive accuracy of *UniDis* is statistically comparable to that of DDU, MODL and Ext-Chi2, and it performs significantly better than that of all the other seven methods. The comparison of the measured mean ranks among *UniDis*, DDU, MODL and Ext-Chi2 does not achieve statistically significant difference since we compare all eight algorithms. If we remove Con, CACC, MDLP and EQF, we can obtain Fig. 1B in which *UniDis* and DDU perform significantly better than MODL and Ext-Chi2. Note that the mean predictive accuracy of *UniDis* is statistically comparable to that of DDU from the statistical point of view though mean rank of *UniDis* is higher than that of DDU. Similarly, for Table 6 we can see from Fig. 2A that the mean predictive accuracy of *UniDis* is statistically comparable to that of DDU, MODL and CACC. If we remove Con, Ext-Chi2, MDLP and EQF, we can obtain Fig. 2B in which the mean predictive accuracy of *UniDis* is statistically comparable to that of DDU, and they perform significantly better than MODL and CACC. For MODL, it is one of the latest discretization techniques based on entropy and information theory properties, and the criterion measures classification errors and the number of discretized intervals. It can be viewed as a specific form derived by the parameter-based criterion in DDU. As there is a relatively large difference between classification errors and the number of discretized intervals, DDU may lead to the two extremes analyzed in Section 3.1. So, MODL may have the same defects. Our proposed normalized discretization criteria can avoid such a situation and achieve higher calssification accuracy.

In the following, we validate information loss generated by discretization by using the level of consistency (Tay and Shen 2002; Pawlak 1982) that is an important knowledge in Rough Set Theory. The level of consistency ($0 \leq L_c(A) \leq 1$) can accurately reflect information validity of the processed data. Specifically, let U

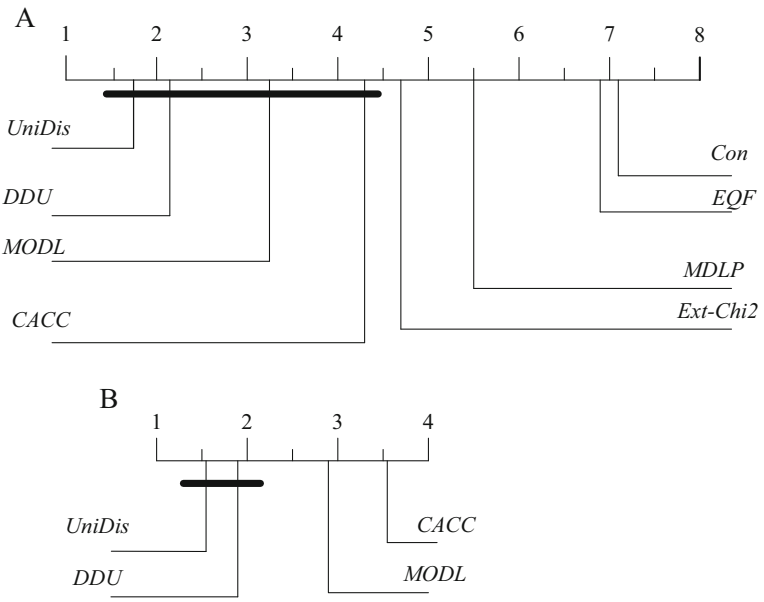

Fig. 2 Comparison of Naive Bayes performance with Holm's post-hoc tests ($\alpha = 0.05$)

Table 7 The predictive accuracy (percent) using J48 with different discretization methods

Data sets	Inconsistent rate						
	Ext-Chi2	CACC	*UniDis*	DDU	MODL	MDLP	EQF
Artificial	0.013	0.435	0.013	0.125	0.104	0.238	0.018
Breast	0.00	0.126	0.01	0.105	0.086	0.134	0.013
Bupa	0.00	0.933	0.01	0.496	0.461	0.525	0.293
Chapman	0.00	0.785	0.01	0.365	0.269	0.455	0.111
Glass	0.00	0.224	0.01	0.156	0.073	0.009	0.084
Heart	0.00	0.064	0.01	0.014	0.021	0.057	0.00
Iris	0.00	0.127	0.01	0.00	0.014	0.122	0.153
Ionosphere	0.00	0.413	0.01	0.00	0.259	0.00	0.00
Pima	0.00	0.676	0.01	0.007	0.164	0.302	0.12
Vehicle	0.00	0.085	0.01	0.458	0.326	0.566	0.151
Wine	0.00	0.00	0.01	0.00	0.00	0.00	0.00

denote the set of instances. We say two instances x_i and x_j are indiscernible if their projection in space A are the same. For any instance $x_i \in U$, we use $[x_i]_A$ to denote its equivalence class with regard to the indiscernible relationship. Let Y_i be an equivalent class in the subspace defined by the class attribute. We define

$$s(A, Y_i) = \{x \mid [x]_A \subseteq Y_i\} \tag{9}$$

that is, $s(A, Y_i)$ is the set of instances whose equivalent classes in the space defined by A are entirely contained within a single equivalent class Y_i in the subspace defined by the class attribute. The level of consistency of A with respect to the class label is defined as follows:

$$L_c(A) = \frac{\sum_i \mid s(A, Y_i) \mid}{\mid U \mid} \tag{10}$$

where Y_i is the ith equivalence class in the subspace defined by the class attribute.

Thus, the inconsistent rate can be indicated by $1 - L_c(A)$. Table 7 presents the inconsistent rate with different discretization methods. The results show that CACC,

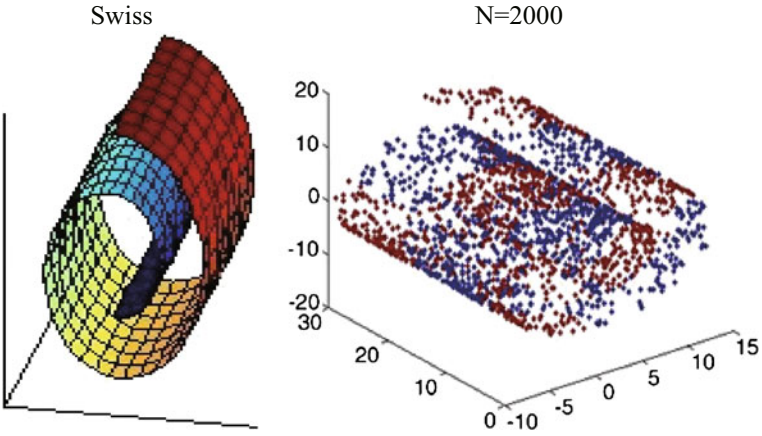

Fig. 3 Swiss dataset and sampled points with $N = 2000$

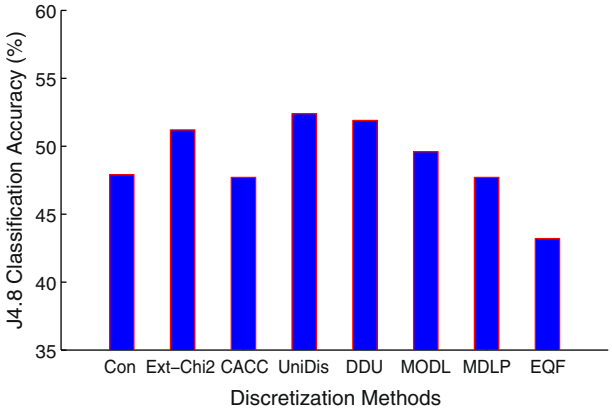

Fig. 4 J4.8 classification accuracy (%)

MODL and MDLP generate higher mean inconsistent rate as these algorithms do not consider information loss when discretizing continuous data. DDU and EQF generate relatively lower inconsistent rate. While the mean accuracy of EQF is low as it is an unsupervised discretization method although EQF generates a relatively lower inconsistent rate. For Ext-Chi2, it allows no information loss for consistent data. For our proposed method, ξ value in Algorithm 1 is similar to that of the bottom-up discretization methods (Kerber 1992; Liu and Setiono 1997; Tay and Shen 2002; Su and Hsu 2005). However, the ξ choice of these bottom-up algorithms either leads to low fidelity of the original data set or discretize data in such a conservative way that it does not allow any loss of information. So, how to determine the best ξ which can result in the maximal classification accuracy is still an open question and beyond the scope of this paper. Here, we set ξ to a smaller value ($\xi = 0.01$) for all of the data sets except 'Artificial' data set. This aims to achieve a less information loss after discretizing continuous data. We will open a new way of tackling the choice problems of the threshold ξ in further versions.

Besides the above real-world data from UCI datasets, we also evaluate the classification performance on the *Swiss* dataset (Roweis and Saul 2000) that is the artificially generated dataset in $\mathbb{R}^3$ as shown in Fig. 3. The aim is to explore the classification performance of the proposed method in different types of data. As the *Swiss* dataset is not real-world data, so we give an individual test. Figure 3 shows the scatter plot of the *Swiss* dataset. As we can see, the *Swiss* data are 2000 points generated randomly, and each point belongs to one of two classes. In this experiment, we use J48 to evaluate the classification performance of different discretization methods on the *Swiss* dataset. Figure 4 shows that our proposed *UniDis* method achieves the highest J4.8 classification accuracy.

5 Conclusions

Discretization algorithms have played an important role in data mining and knowledge discovery. They not only produce a concise summarization of continuous attributes to help the experts understand the data more easily, but also make learning more accurate and faster. In this paper, we propose a universal discretization tech-

nique *UniDis*, which develops a non-parametric normalized discretization criteria to avoid the effect of relatively large difference between classification errors and the number of discretized intervals on discretization results. In addition, we seamlessly define a new entropy-based measure of inconsistency for multi-dimensional variables with the aim to effectively control information loss while producing a concise summarization of continuous variables. A heuristic discretization algorithm is designed to search the best discretization based on the new non-parametric criteria and the entropy-based inconsistency. We conduct experiments on 11 real data sets demonstrate that our approach is statistically comparable to DDU and it generates a good discretization scheme which significantly improves the mean accuracy of classification than existing discretization methods except for DDU by running J4.8 decision tree.

Acknowledgements This work is supported by NSFC under Grant nos. of 60973115, 60973117, 61173160, 61173162 and 61173165, and New Century Excellent Talents in University (NCET) of Ministry of Education of China.

Appendix

In this section, we show the monotonicity of $f(\beta)$ and $H_\beta(R_i)$ with regard to β.

Theorem 1 *$f(\beta)$ and $H_\beta(R_i)$ are the monotonous decreasing functions of β in the interval (0,1].*

Proof Let β_1 and β_2 are two values in the interval (0,1], and $\beta_1 < \beta_2$. For $f(\beta)$ in (1), we have

$$f(\beta_1) - f(\beta_2) = \frac{1 - (\frac{1}{N})^{\beta_1}}{\beta_1} - \frac{1 - (\frac{1}{N})^{\beta_2}}{\beta_2} = \frac{\beta_2 - \beta_1 + \beta_1\left(\frac{1}{N}\right)^{\beta_2} - \beta_2\left(\frac{1}{N}\right)^{\beta_1}}{\beta_1\beta_2}$$

$\because\ 0 < \beta_1 < \beta_2 \leq 1$
$\therefore\ \beta_2 - \beta_1 > 0, \quad \beta_1\beta_2 > 0, \quad \beta_1\left(\frac{1}{N}\right)^{\beta_2} - \beta_2\left(\frac{1}{N}\right)^{\beta_1} < 0$
$\because\ N \gg 1$
$\therefore\ \beta_2 - \beta_1 > \mid \beta_1\left(\frac{1}{N}\right)^{\beta_2} - \beta_2\left(\frac{1}{N}\right)^{\beta_1} \mid$
$\therefore\ f(\beta_1) > f(\beta_2)$

Therefore, $f(\beta)$ increases with the reduction of β in the interval (0,1]. Similarly, $H_{\beta_1}(R_i) > H_{\beta_2}(R_i)$. Therefore, $H_\beta(R_i)$ is also monotone decreasing with regard to β in the interval (0,1]. □

Theorem 2 $f(\beta) \in \left[1 - \frac{1}{N}, \ln N\right]$, *and* $H_1(R_i) \leq H_\beta(R_i) \leq \log S$.

Proof According to Theorem 1, $f(\beta)$ achieves a minimum value when $\beta = 1$ and achieves a maximum value when $\beta \to 0$. Then, we have

$$1 - \frac{1}{N} \leq f(\beta) < \lim_{\beta \to 0} \frac{1 - (\frac{1}{N})^{\beta}}{\beta} = \ln N$$

Similarly,

$$\begin{aligned} H_1(R_i) \leq H_\beta(R_i) &< \lim_{\beta \to 0} H_\beta(R_i) \\ &= \lim_{\beta \to 0} \sum_{j=1}^{S} \frac{N_{ij}}{N_{i\cdot}} \Big[1 - \left(\frac{N_{ij}}{N_{i\cdot}} \right)^{\beta} \Big] \Big/ \beta \\ &= \sum_{j=1}^{S} \frac{N_{ij}}{N_{i\cdot}} \log \frac{N_{i\cdot}}{N_{ij}} \\ &= H(R_i) \end{aligned}$$

where $H(R_i)$ is Shannon's entropy (Cover and Thomas 2006) of interval R_i. According to the extremum property of entropy, $H(R_i) \leq \log S$. Therefore, the theorem is proven. □

References

Biba, M., Esposito, F., Ferilli, S., Mauro, N.D., Basile, T. (2007). Unsupervised discretization using kernel density estimation. In: *Proceedings of Twentieth International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 696–701).

Bondu, A., Boulle, M., Lemaire, V., Loiseau, S., Duval, B. (2008). A Non-parametric semi-supervised discretization method. In: *Proceedings of Eighth IEEE International Conference on Data Mining (ICDM)* (pp. 53–62).

Boulle, M. (2004). Khiops: a statistical discretization method of continuous attributes. *Machine Learning, 55*, 53–69.

Boulle, M. (2006). MODL: a bayes optimal discretization method for continuous attributes. *Machine Learning, 65*, 131–165.

Ching, J.Y., Wong, A.K.C., Chan, K.C.C. (1995). Class-dependent discretization for inductive learning from continuous and mixed-mode data. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 17*(7), 641–651.

Cios, K.J., & Kurgan, L.A. (2007). CLIP4: hybrid inductive machine learning algorithm that generates inequality rules. *Information Sciences, 177*(17), 3592–3612.

Cover, T.M., & Thomas, J.A. (2006). *Elements of information thoery* (2nd ed.). New York: Wiley.

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

Dougherty, J., Kohavi, R., Sahami M. (1995). Supervised and unsupervised discretization of continuous feature. In: *Proceedings of 12th International conference of Machine learning* (pp. 194–202).

Fayyad, U., & Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In: *Proceedings of thirteenth international joint conference on artificial intelligence* (pp. 1022–1027). San Mateo, CA: Morgan Kaufmann.

Hand, D., Mannila, H., Smyth, P. (2001). *Principles of data mining*. MIT Press.

Hansen, M.H., & Yu, B. (2001). Model selection and the principle of minimum description length. *Journal of the American Statistical Association, 96*(545), 746–774.

Hettich, S., & Bay, S.D. (1999). The UCI KDD Archive [DB/OL]. http://kdd.ics.uci.edu/. Accessed 12 Aug 2010.

Jin, R.M., Breitbart, Y., Muoh, C. (2007). Data discretization unification. In: *Proceedings of seventh IEEE International Conference on Data Mining (ICDM Best Paper)* (pp. 183–192).
Jin, Y.W., & Qu, W.Y. (2009). Multi-dimension multi-objective fuzzy optimum dynamic programming method with complicated information based on a maximal-sum-rule of decision sequence priority. In: *Eighth IEEE international conference on embedded computing; IEEE international conference on scalable computing and communications* (pp. 656–660). Dalian, China.
Kerber, R. (1992). ChiMerge: discretization of numeric attributes. In: *Proceedings of ninth national conference on artificial intelligence* (pp. 123–128). AAAI Press.
Kurgan, L.A., & Cios, K.J. (2004). CAIM discretization algorithm. *IEEE Transactions on Knowledge and Data Engineering, 16*(2), 145–153.
Ling, C.X., & Zhang, H.J. (2002). The representational power of discrete bayesian networks. *Journal of Machine Learning Research, 3*, 709–721.
Liu, L.L., Wong, A.K.C., Wang, Y. (2004). A global optimal algorithm for class-dependent discretization of continuous data. *Intelligent Data Analysis, 8*(2), 151–170.
Liu, H., Hussain, F., Tan, C.L., Dash, M. (2002). Discretization: an enabling technique. *Journal of Data Mining and Knowledge Discovery, 6*(4), 393–423.
Liu, H., & Setiono, R. (1997). Feature selection via discretization. *IEEE Transactions on Knowledge and Data Engineering, 9*(4), 642–645.
Mahady, H., Muhammad, A.C. , Qu, W.Y., Lin, X.M. (2010). Efficient algorithms to monitor continuous constrained *k* nearest neighbor queries. In: *Data base systems for advanced applications* (pp. 233–249). Tsukuba, Japan.
Mussard, S., Seyte, F., Terraza, M. (2003). Decomposition of Gini and the generalized entropy inequality measures. *Economic Bulletin, 4*(7), 1–6.
Pawlak, Z. (1982). Rough sets. *International Journal of Computer and Information Sciences, 11*(5), 341–356.
Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning, 1*, 81–106.
Quinlan, J.R. (1993). *C4.5: Programs for machine learning*. San Mateo, California: Morgan Kaufmann.
Roweis, S.T., & Saul, L.K. (2000). Science. *Nonlinear Dimensionality Reduction by Locally Linear Embedding, 290*(5500), 2323–2326.
Schmidberger, G., & Frank, E. (2005). Unsupervised discretization using tree-based density estimation. In: *Proceedings of The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)* (pp. 240–251).
Su, C.T., & Hsu, J.H. (2005). An extended Chi2 algorithm for discretization of real value attributes. *IEEE Transactions on Knowledge and Data Engineering, 17*(3), 437–441.
Tay, E.H., & Shen, L. (2002). A modified Chi2 algorithm for discretization. *IEEE Transactions on Knowledge and Data Engineering, 14*(3), 666–670.
Tsai, C.J., Lee, C.I., Yang, W.P. (2008). A discretization algorithm based on class-attribute contingency coefficient. *Information Sciences, 178*, pp. 714–731.
Wang, H.X., & Zaniolo, C. (2000). CMP: a fast decision tree classifier using multivariate predictions. In: *16th International Conference on Data Engineering (ICDE00)* (pp. 449–460).
Weka 3 Data mining software in Java (2007). http://www.cs.waikato.ac.nz/ml/weka. Accessed 26 Nov 2010.
Witten, I.H., & Frank, E. (2000). *Data mining: Practical machine learning tools and techniques with java implementations*. San Francisco, CA: Morgan Kaufmann.
Zar, J.H. (1998). *Biostatistical analysis* (4th ed.). Englewood Clifs, New Jersey: Prentice Hall.
Ziarko, W. (1993). Variable precision rough set model. *Journal of Computer and System Science, 46*, 39–59.