A Discretization Algorithm for Uncertain Data

Jiaqi Ge[1,*], Yuni Xia[1], and Yicheng Tu [2]

[1] Department of Computer and Information Science,
Indiana University – Purdue University, Indianapolis, USA
{jiaqge,yxia}@cs.iupui.edu
[2] Computer Science and Engineering
University of South Florida
ytu@cse.usf.edu

Abstract. This paper proposes a new discretization algorithm for uncertain data. Uncertainty is widely spread in real-world data. Numerous factors lead to data uncertainty including data acquisition device error, approximate measurement, sampling fault, transmission latency, data integration error and so on. In many cases, estimating and modeling the uncertainty for underlying data is available and many classical data mining algorithms have been redesigned or extended to process uncertain data. It is extremely important to consider data uncertainty in the discretization methods as well. In this paper, we propose a new discretization algorithm called UCAIM (Uncertain Class-Attribute Interdependency Maximization). Uncertainty can be modeled as either a formula based or sample based probability distribution function (*pdf*). We use probability cardinality to build the quanta matrix of these uncertain attributes, which is then used to evaluate class-attribute interdependency by adopting the redesigned *ucaim* criterion. The algorithm selects the optimal discretization scheme with the highest *ucaim* value. Experiments show that the usage of uncertain information helps UCAIM perform well on uncertain data. It significantly outperforms the traditional CAIM algorithm, especially when the uncertainty is high.

Keywords: Discretization, Uncertain data.

1 Introduction

Data discretization is a commonly used technique in data mining. Data discretization reduces the number of values for a given continuous attribute by dividing the range of the attribute into intervals. Interval labels are then used to replace actual data values. Replacing numerous values of a continuous attribute by a small number of interval labels thereby simplifies the original data. This leads to a concise, easy-to-use, knowledge-level representation of mining results [32]. Discretization is often performed

[*] Please note that the LNCS Editorial assumes that all authors have used the western naming convention, with given names preceding surnames. This determines the structure of the names in the running heads and the author index.

P. García Bringas et al. (Eds.): DEXA 2010, Part II, LNCS 6262, pp. 485–499, 2010.

prior to the learning process and has played an important role in data mining and knowledge discovering. For example, many classification algorithms as AQ [1], CLIP [2], and CN2 [3] are only designed for category data, therefore, numerical data are usually first discretized before being processed by these classification algorithms. Assume A is one of the continuous attributes of a dataset, A can be discretized into n intervals as $D = \{[d_0, d_1), [d_1, d_2),\ldots, [d_{n-1}, d_n]\}$, where d_i is the value of the endpoints in each interval. Then D is called as a discretization scheme on attribute A. A good discretization algorithm not only produces a concise view of continuous attributes so that experts and users can have a better understanding of the data, but also helps machine learning and data mining applications to be more effective and efficient [4]. A number of discretization algorithms have been proposed in literature, most of them focus on certain data. However, data tends to be uncertain in many applications [9], [10], [11], [12], [13]. Uncertainty can originate from diverse sources such as data collection error, measurement precision limitation, data sampling error, obsolete source, and transmission error. The uncertainty can degrade the performance of various data mining algorithms if it is not well handled. In previous work, uncertainty in data is commonly treated as a random variable with probability distribution. Thus, uncertain attribute value is often represented as an interval with a probability distribution function over the interval [14], [15].

In this paper, we propose a data discretization technique called Uncertain Class-Attribute Interdependency Maximization (UCAIM) for uncertain data. It is based on the CAIM discretization algorithm and we extend it with a new mechanism to process uncertainty. Probability distribution function (*pdf*) is commonly used to model data uncertainty and *pdf* can be represented as either formulas or samples. We adopt the concept of probability cardinality to build the quanta matrix for uncertain data. Based on the quanta matrix, we define a new criterion value *ucaim* to measure the interdependency between uncertain attributes and uncertain class memberships. The optimal discretization scheme is determined by searching the one with the largest *ucaim* value. In the experiments, we applied the discretization algorithm as the preprocessing step of an uncertain naïve Bayesian classifier [16], and measured the discretization quality by its classification accuracy. Results illustrated that the application of the UCAIM algorithm as a front-end discretization algorithm significantly improve the classification performance.

The paper is organized as following. In section 2, we discuss related work. Section 3 introduces the model of uncertain data. In section 4, we present the ucaim algorithm in detail. The experiments results are shown in section 5, and section 6 concludes the paper.

2 Related Work

Discretization algorithms can be divided into top-down and bottom-up methods according to how the algorithms generate discrete schemes [6]. Both top-down and bottom-up discretization algorithms can be further subdivided into unsupervised and supervised methods [17]. Equal Width and Equal Frequency [5] are well-known unsupervised top-down algorithms, while the supervised top-down algorithms include MDLP [7], CADD (class-attribute dependent discretize algorithm) [18], Information

Entropy Maximization [19], CAIM (class-attribute interdependent maximization algorithm) [8] and FCAIM (fast class-attribute interdependent maximization algorithm) [20]. Since CAIM selects the optimal discretization algorithm that has the highest interdependence between target class and discretized attributes, it is proven to be superior to other top-down discretization algorithms in helping the classifiers to achieve high classification accuracy [8]. FCAIM extends CAIM by using a different strategy to select fewer boundary points during the initialization, which speeds up the process of finding the optimal discretization scheme.

In the bottom-up category, there are widely used algorithms such as ChiMerge [21], Chi2 [22], Modified Chi2 [23], and Extended Chi2 [24]. Bottom-up method starts with the complete list of all continuous value of the attribute as cut-points, so its computational complexity is usually higher than the top-down method [29]. Algorithms like ChiMerge require users to provide some parameters such as significant level and minimal/ maximal interval numbers during the discretization process. [25] illustrates that all these different supervised discretization algorithms can be viewed as assigning different parameters to a unified goodness function, which can be used to evaluate the quality of discretization algorithms. There also exist some dynamic discretization algorithms [26] which are designed for particular machine learning algorithms such as decision tree and naïve Bayesian classifier.

All the algorithms mentioned above are based on certain datasets. To the best of our knowledge, no discretization algorithm has been proposed for uncertain data that are represented as probability distribution functions. In the recent years, there have been growing interests in uncertain data mining. For example, a number of classification algorithms have been extended to process uncertain datasets, as uncertain support vector machine [27], uncertain decision tree [28], uncertain naïve Bayesian classifier. It is extremely important that data preprocessing techniques like discretization properly handle this kind of uncertainty as well. In this paper, we propose a new discretization algorithm for uncertain data.

3 Data Uncertainty Model

When the value of a numerical type attribute A is uncertain, the attribute is called an uncertain attribute (UNA), denoted by A^{u_n} [29]. In uncertain dataset D, each tuple t_i is associated with a feature vector $Vi = (f_{i,1}, f_{i,2}, \ldots, f_{i,k})$ to model its uncertain attributes. $f_{i,j}$ is a probability distribution function (*pdf*) representing the uncertainty of attribute $A_{ij}^{u_n}$ in tuple t_i. Meanwhile, a probability distribution c_i is assigned to the t_i's uncertain class label C_i as class membership.

In practice, uncertainties are usually modeled in forms of Gaussian distributions, and parameters such as mean μ and standard variance σ are used to describe the Gaussian distributed uncertainty. In this case, uncertain attribute $A_{ij}^{u_n}$ has a formula based probability representation over the interval $[A_{ij}^{u_n}.l, A_{ij}^{u_n}.r]$ as

$$p_{ij} = \int_{A_{ij}^{u_n}.l}^{A_{ij}^{u_n}.r} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(x-\mu)^2}{\sigma^2}} dx .$$

Here p_{ij} is the probability distribution of uncertain attribute $A_{ij}^{u_n}$ which can be seen as a random variable.

In case that the uncertainty cannot be modeled by any mathematical formula expression, a sample based method is often used to model the probability distribution:

$$p_{ij} = \{A_{ij}^{u_n} | (x_1 : p_1), (x_2 : p_2), \ldots (x_i : p_i), \ldots, (x_n : p_n)\}.$$

Where, $X = \{x_1, x_2 \ldots x_{i \ldots} x_n\}$ is the set of all possible values for attribute $A_{ij}^{u_n}$, and p_i is the probability that $A_{ij}^{u_n} = x_i$.

Not only can the attributes be uncertain, class labels may also contain uncertainty. Instead of having the accurate class label, a class membership may be a probability distribution as following:

$$C_i = \{c | (c_1 : p_1), (c_2 : p_2), \ldots, (c_n : p_n)\}$$

Here, $\{c_1, c_2, \ldots, c_n\}$ is the set containing all possible class labels, and p_i is the probability that this instance t_i belongs to class c_i.

Table 1 shows an example of an uncertain database. Both attributes and class labels of the dataset are uncertain. Their precise values are unavailable and we only have knowledge of the probability distribution. For attribute 1, its uncertainty is represented as a Gaussian distribution with parameters (μ, σ) to model the *pdf*. For attribute 2, it lists all possible values with their corresponding probabilities for each instance. Note that the uncertainty of class label is always represented in the sample format as the values are discrete.

Table 1. An example of uncertain dataset

ID	Class Type	Attribute 1	Attribute 2
1	T: 0.3, F :0.7	(105, 5)	(100: 0.3, 104: 0.6, 110:0.1)
2	T: 0.4, F:0.6	(110,10)	(102:0.2, 109: 0.8)
3	T: 0.1, F:0.9	(70,10)	(66: 0.4, 72:0.4, 88:0.2)

4 UCAIM Discretization Algorithm

4.1 Cardinality Count for Uncertain Data

According to the uncertainty models, an uncertain attribute $A_{ij}^{u_n}$ is associated with a *pdf* either in a formula based or sample based format. The probability that the value of $A_{ij}^{u_n}$ falls in a partition [*left*, *right*] is:

For formula based *pdf*:

$$p_{A_{ij}^{u_n}} = \int_{left}^{right} A_{ij}^{u_n} . f(x)\, dx \tag{1}$$

Where, $A_{ij}^{u_n} . f(x)$ is the probability density distribution function of $A_{ij}^{u_n}$.

For sample based *pdf*:

$$p_{A_{ij}^{u_n}} = \sum_{A_{ij}^{u_n}.x_k \in [left,right]} A_{ij}^{u_n}.p_k \quad (2)$$

Where, $A_{ij}^{u_n}.x_k$ is the possible value of $A_{ij}^{u_n}$, and $A_{ij}^{u_n}.p_k$ is the probability that $A_{ij}^{u_n} = x_k$.

We assume that class uncertainty is independent to the probability distributions of attribute values. Thus, for a tuple t_i belonging to class C, the probability that its attribute value $A_{ij}^{u_n}$ falls in the interval [*left*, *right*] is:

$$P\left(A_{ij}^{u_n} \in [left, right], c_i = C\right) = p_{A_{ij}^{u_n}} * p(c_i = C) \quad (3)$$

$p_{A_{ij}^{u_n}}$ is defined in formula (1) and (2) and $p(c_i = C)$ is the probability that t_i belongs to class C.

For each class C, we compute the sum of the probabilities that an uncertain attribute A_{ij}^{u} falls in partition [*left*, *right*] for all the tuples in dataset D. This summation is called *probabilistic cardinality*. For example, the probability cardinality of partition P= [a,b) for class C is calculated as:

$$P_C(p) = \sum_{i=1}^{n} P\left(A_{ij}^{u_n} \in [a, b)\right) * P(c_i = C) \quad (4)$$

Probability cardinalities provide us valuable insight during the discretization process and it used to build the quanta matrix for uncertain data, as shown in the next section.

4.2 Quanta Matrix for Uncertain Data

The discretization algorithm aims to find the minimal number of discrete intervals while minimizing the loss of class-attribute interdependency. Suppose F is a continuous numeric attribute, and there exists a discretization scheme D on F, which divides the whole continuous domain of attribute F into n discrete intervals bounded by the endpoints as:

$$D: \{[d_0, d_1), [d_1, d_2), [d_2, d_3), \ldots, [d_{n-1}, d_n] \} \quad (5)$$

where d_0 is the minimal value and d_n is the maximal value of attribute F; $d_1, d_2, \ldots, d_{n-1}$ are cutting points arranged in ascending order.

For certain dataset, every value of attribute F is precise; therefore it will fall into only one of the n intervals defined in (5). However, the value of an uncertain attribute can be an interval or a series of values with associated probability distribution. Therefore, it could fall into multiple intervals. The class membership for a specific interval in (5) varies with different discretization scheme D.

The class variable and the discretization variable of attribute F are treated as two random variables defining a two-dimensional quanta matrix (also known as the contingency table). Table 2 is an example of quanta matrix.

In Table 2, q_{ir} is the probability cardinality of the uncertain attribute $A_F^{u_n}$ which belongs to the i^{th} class and has its value within the interval $[d_{r-1}, d_r]$. Thus, according to formula (4), q_{ir} can be calculated as:

$$q_{ir} = P_c(c = Ci, A_F^{u_n} \in [d_{r-1}, d_r]) \quad (6)$$

Table 2. Quanta matrix for uncertain attribute A_F^{un} and discretization scheme D

class	Interval					Class Total
	$[d_0,d_1)$	…	$[d_{r-1},d_r)$	…	$[d_{n-1},d_n]$	
C_1	q_{11}	…	q_{1r}	…	q_{1n}	M_{1+}
⋮	…	…	…	…	…	
C_i	q_{i1}	…	q_{ir}	…	q_{in}	M_{i+}
⋮	…	…	…	…	…	
C_s	q_{s1}	…	q_{sr}	…	q_{sn}	M_{s+}
Interval Total	M_{+1}		M_{+r}		M_{+n}	M

M_{i+} is the sum of the probability cardinality for objects belonging to the i^{th} class, and M_{+r} is the total probability cardinality of uncertain attribute $A_F^{u_n}$ that are within the interval $[d_{r-1}, d_r]$, for i = 1, 2… S, and r= 1, 2… n.

The estimated joint probability that uncertain attribute values $A_F^{u_n}$ is within the interval D_r= $[d_{r-1}, d_r]$, and belong to class C_i can be calculated as:

$$p_{ir} = p\left(C_i, D_r \middle| A_F^{u_n}\right) = \frac{q_{ir}}{M} \tag{7}$$

4.3 Uncertain Class-Attribute Interdependent Discretization

We first introduce the Class-Attribute Interdependency Maximization (CAIM) discretization approach. CAIM is one of the classical discretization algorithms. It generates the optimal discretization scheme by quantifying the interdependence between classes and discretized attribute, and its criterion is defined as following:

$$CAIM\left(C, D \middle| A_F^{u_n}\right) = \frac{\sum_{r=1}^{n} \frac{max_r^2}{M_{+r}}}{n} \tag{8}$$

Where n is the number of intervals, r iterates through all intervals, i.e. r=1, 2,…, n, max_r is the maximum value among all q_{ir} values (maximum value within the r^{th} column of the quanta matrix), i = 1,2,…,S, M_{+r} is the total probability of continues values of attribute F that are within the interval D_r= $[d_{r-1}, d_r]$.

From the definition, we can see that caim value increases when the values of max_i grow, which corresponds to the increase of the interdependence between the class labels and the discrete intervals. Thus CAIM algorithm finds the optimal discretization scheme by searching the scheme with the highest *caim* value. Since the maximal value max_r is the most significant part in the definition of CAIM criterion, the class which max_r corresponds to is called main class and the larger max_r the more interdependent between this main class and the interval D_r= $[d_{r-1}, d_r]$.

Although CAIM performances well on certain datasets, it encounters new challenges in uncertain case. For each interval, CAIM algorithm only takes the main class into account, but does not consider the distribution over all other classes, which leads to problems when dealing with uncertain data. In an uncertain dataset, each instance no longer has a deterministic class label, but may have a probability distribution over

all possible classes and this reduces the interdependency between attribute and class. We use the probability cardinality to build the quanta matrix for uncertain attributes, and we observe that the original *caim* criterion causes problems when handling uncertain quanta matrix. Below we give one such example. Suppose a simple uncertain dataset containing 5 instances is shown in Table 3. Its corresponding quanta matrix is shown in Table 4.

Table 3. An example of uncertain dataset

Attribute (x: p_x)	*Class (label: probability)*
(0.1:0.3), (0.9: 0.7)	0: 0.9, 1: 0.1
(0.1:0.2), (0.9: 0.8)	0: 0.9, 1: 0.1
(0.9: 1.0)	0: 1.0, 1: 0.0
(0.2:0.7), (0.8: 0.3)	0: 0.1, 1: 0.9
(0.1:0.7), (0.8: 0.2), (0.9:0.1)	0: 0.1, 1: 0.9

From table 3, we can calculate the probability distribution of attribute values x each class as following:

$P(x=0.1, C=0) = 0.3*0.9 + 0.2*0.9 + 0.7*0.1 = 0.52$
$P(x=0.1, C=1) = 0.3*0.1 + 0.2*0.1 + 0.7*0.9 = 0.68$
$P(x=0.2, C=0) = 0.7*0.1 = 0.07$
$P(x=0.2, C=1) = 0.7*0.9 = 0.63$
$P(x=0.8, C=0) = 0.3*0.1 + 0.2*0.1 = 0.05$
$P(x=0.8, C=1) = 0.3*0.9 + 0.2*0.9 = 0.45$
$P(x=0.9, C=0) = 0.7*0.9 + 0.8*0.9 + 0.1*0.1 + 1.0*1.0 = 2.36$
$P(x=0.9, C=1) = 0.7*0.1 + 0.8*0.1 + 0.1*0.9 = 0.24$

Table 4. Quanta Matrix for the uncertain dataset

class	*Interval*
	[0, 1]
0	3
1	2

According to formula (8), the *caim* value for the quanta matrix in table 4 is: *caim* = $3^2/(3+2) = 1.8$. From the distribution of attribute values in each class, we can see the attribute values of instances in class 0 have a high probability around $x = 0.9$; and those for instances in class 1 are mainly located in the small end around x=0.1 and 0.2. Obviously, $x = 0.5$ is a reasonable cutting point to generate the discretization scheme {[0, 0.5) [0.5, 1]}. After the splitting, the quanta matrix is shown in table 5, whose corresponding *caim* value is:

$$caim = \left(\frac{\frac{1.31^2}{1.31+0.59} + \frac{2.41^2}{2.41+0.69}}{2} \right) = 1.38$$

Table 5. Quanta Matrix after splitting at $x = 0.5$

class	*Interval*	
	[0, 0.5)	[0.5, 1]
0	0.59	2.41
1	1.31	0.69

The goal of the CAM algorithm is to find the discretization scheme with highest caim value, so {[0, 0.5) [0.5, 1]} will not be accepted as a better discretization scheme, because the *caim* value decreases from 1.8 to 1.38 after splitting at $x = 0.5$.

Data uncertainty obscures the interdependence between classes and attribute values by flatting the probability distributions. Therefore, when the original CAIM criterion is applied to uncertain data, it results in two problems. First, it usually does not create enough intervals in the discretization scheme or it stops splitting too early, which causes the loss of much class-attribute interdependence. Second, in order to increase the caim value, it is possible that the algorithm generates intervals with very small probability cardinalities, which reduces the robustness of the algorithm.

For uncertain data, the attribute-class interdependence is in form of a probability distribution. The original caim definition as in formula (8) ignores this distribution, and only considers the main class. Therefore, we need to revise the original definition to handle uncertain data. Now that uncertainty blurs the attribute-class interdependence and reduces the difference between the main class and the rest of the classes, we try to make the CAIM value more sensitive to change of values in quanta matrix. We propose the uncertain CAIM criterion UCAIM, which is defined as follows:

$$UCAIM\left\langle C,D\middle|A_F^{Un}\right\rangle = \frac{\sum_{r=1}^{n}\frac{\max_v^2 \times Offset_r}{M_{+r}}}{n} \tag{9}$$

Where

$$Offset_r = \frac{\sum_{i=1,q_{ir}\neq \max_r}^{s}(\max_r - q_{ir})}{s-1} \tag{10}$$

In formula (9), max_r is the maximum value among all q_{ir} values (maximum value within the r^{th} column of the quanta matrix), $i = 1, 2,\ldots, S$, M_{+r} is the total probability of continues values of attribute F that are within the interval $D_r = [d_{r-1}, d_r]$. $Offset_r$ defined in (10) is the average of the offsets or differences for all other q_{ir} values to max_r.

Because the larger the attribute-class interdependence, the larger the value max_r/M_{+r}, CAIM therefore uses it to identify splitting points in formula (8). In the UCAIM definition we proposed, $Offset_r$ shows how significant the main class is, compared to other classes. When $Offset_r$ is large, it means that within interval r, the probability an instance belongs to the main class is much higher than the other classes, so the interdependence between interval r and the main class becomes is also high. Therefore, we propose the ucaim definition in formula (9) for the following reasons:

1) Compared with max_r/M_{+r}, we multiply it with the factor $Offset_r$ to make the value $Offset_r*max_r/M_{+r}$ more sensitive to interdependence changes, which are usually less significant for uncertain data.

2) The value max_r/M_{+r} may be large merely because $M_{+r}$is small, which happens when there are not many instances falling into interval r. However, $Offset_r$ does not have such this problem, because it measures the relative relationship between main class and other classes.

Now we apply the new definition to the sample uncertain dataset in Table 3. For the original quanta matrix as in Table 4, the *ucaim* value is

$$ucaim = \frac{3^2 \times (3-2)}{5} = 1.8$$

For the quanta matrix after splitting as in Table 5, we have

$$S1 = 1.31\text{-}0.59 = 0.72;\ S2 = 2.41\text{-}0.69 = 1.72$$

$$ucaim = \left(\frac{\frac{1.31^2}{1.31+0.59} \times 0.72 + \frac{2.41^2}{2.41+0.69} \times 1.72}{2} \right) = 1.98$$

Since ucaim value increases after the splitting, the cutting point x = 0.5 will be accepted in the discretization scheme. We can see in this example that ucaim is more effective in finding the interdependence between attribute and class, compared to the original approach.

Table 6. UCAIM discretization algorithm

Algorithm
1. Find the maximal and minimal possible values of the uncertain attribute A_F^{un}, recorded as d_0, d_n.
2. Create a set B of all potential boundary endpoints. For uncertain attribute modelled in sample based *pdf*, simply sort all distinct possible values and use them as the set; for uncertain data modelled as formula based pdf, we use the mean of each distribution to build the set.
3. Set the initial discretization scheme as D:{[d_0, d_n]}, set GlobalUCAIM = 0
4. initialize k=1;
5. tentatively add an inner boundary, which is not already in D, from B and calculate corresponding UCAIM value
6. after all the tentative additions have been tested, accept the one with the highest value of UCAIM
7. if UCAIM > GlobalUCAIM or k<S, update D with the accepted boundary and set GlobalUCAIM = UCAIM, else terminate
8. set k=k+1 and go to 5
Output: D

4.4 Uncertain Discretization Algorithm

The Uncertain Discretization algorithm is shown in table 6. It consists of two steps: (1) initialization of the candidate interval boundaries and the initial discretization scheme; (2) iterative additions of new splitting points to achieve the highest value of the UCAIM criterion.

The time complexity of ucaim algorithm is similar to caim algorithm. For a single attribute, in the worst case, the running time of caim is O(Mlog(M)) [8], and M is the number of distinct values of the discretization attributes. In ucaim algorithm, the additional computation is to calculate S_r, whose time complexity is O(C*M). C is the number of classes, and usually very small comparing with M. Therefore, the addition in time complexity is O(M), and the final running cost of ucaim is still O(Mlog(M)). Please note that this algorithm works on certain data as well since certain data can be viewed as a special case of uncertain data.

5 Experiments

In this section, we present the experimental results of *ucaim* discretization algorithm on eight datasets. We compare our technique with the traditional CAIM discretization algorithm, to show the effectiveness of UCAIM algorithm on uncertain data.

5.1 Experiment Setup

The datasets selected to test the ucaim algorithm are: Iris Plants dataset (iris), Johns Hopkins University Ionosphere dataset (ionosphere), Pima Indians Diabetes dataset (pima), Glass Identification dataset (glass), Wine dataset (wine), Breast Cancer Wisconsin Original dataset (breast), Vehicle Silhouettes dataset (vehicle), Statlog Heart dataset (heart). All these datasets were obtained from UCI ML repository [30], and their detailed information is shown in table 7.

Table 7. Properties experimental datasets

Datasets	*# of class*	*# of instance*	*# of attribute*	*# of continues attribute*
iris	3	1	150	4
ionosphere	2	351	34	34
pima	2	768	8	8
glass	7	214	10	10
wine	3	178	13	13
breast	2	699	10	10
vehicle	4	846	18	18

These datasets are made uncertain by adding a Gaussian distributed noise as in [31][9][14]. For each attribute, we add a Gaussian noise with a zero mean, and a standard variance drawn from the unification distribution [0, 2**f**Sigma]. Here, Sigma is the standard variance of the attribute values, and *f* is an integer parameter used to define different uncertain level. The value of *f* is selected from the set {1, 2, 3}. For class

label uncertainty, we assume the original class for each instance is the main class, and assign it a probability p_{mc}, and there is a uniform distribution over all other classes. As a comparison, assume the real data does not center in the original position, but sit in the noised value, and the noises are in the same distribution as those described above.

We use the accuracy of uncertain naïve Bayesian classifier to evaluate the quality of discretization algorithms. As the purpose of our experiment is to compare discretization algorithms, when we build the classifier, we ignore nominal attributes. In the experiments, we first compare our UCAIM algorithm for uncertain data with the original algorithm CAIM-O which does not take the uncertainty into account. We also compare the UCAIM with a discretization algorithm named CAIM-M which simply applies CAIM-O algorithm on uncertain quanta matrix (without using the Offset).

5.2 Experiment Results

The accuracy of uncertain naïve Bayesian classifier on these 8 dataset is shown in table 8. Table 9 shows the average classification accuracy under different uncertain level for all three discretization algorithms. Figure 1 shows detailed performance comparison of these algorithms at each uncertain level.

Table 8. Accuracies of the uncertain Naïve Bayesian classifier with different discretization algorithms

dataset	uncertain level	UCAIM	CAIM-M	CAIM-O
iris	f=1, p_{mc}=0.9	88.67%	81.67%	80.58%
	f=2, p_{mc}=0.8	76.67%	73.33%	69.56%
	f=3, p_{mc}=0.7	72.66%	71.33%	63.85%
wine	f=1, p_{mc}=0.9	96.07%	94.38%	85.39%
	f=2, p_{mc}=0.8	93.09%	89.32%	85.39%
	f=3, p_{mc}=0.7	88.44%	73.59%	77.53%
glass	f=1, p_{mc}=0.9	61.07%	57.94%	47.66%
	f=2, p_{mc}=0.8	57.94%	53.27%	37.07%
	f=3, p_{mc}=0.7	50.93%	43.92%	35.98%
ionosphere	f=1, p_{mc}=0.9	74.09%#	81.26%	76.31%
	f=2, p_{mc}=0.8	78.34%	77.13%	72.17%
	f=3, p_{mc}=0.7	77.20%	75.88%	69.66%
pima	f=1, p_{mc}=0.9	77.13%	75.74%	71.35%
	f=2, p_{mc}=0.8	72.32%	70.89%	63.97%
	f=3, p_{mc}=0.7	70.45%	68.66%	62.33%
breast	f=1, p_{mc}=0.9	95.42%	94.27%	93.36%
	f=2, p_{mc}=0.8	90.70%	87.83%	87.14%
	f=3, p_{mc}=0.7	87.83%	83.12%	80.68%
vehicle	f=1, p_{mc}=0.9	61.22%	55.39%	50.13%
	f=2, p_{mc}=0.8	57.44%	52.12%	44.72%
	f=3, p_{mc}=0.7	53.19%	43.61%	37.87%
Heart	f=1, p_{mc}=0.9	82.59%	78.88%	75.33%
	f=2, p_{mc}=0.8	78.19%	72.16%	70.15%
	f=3, p_{mc}=0.7	73.63%	69.95%	67.76%

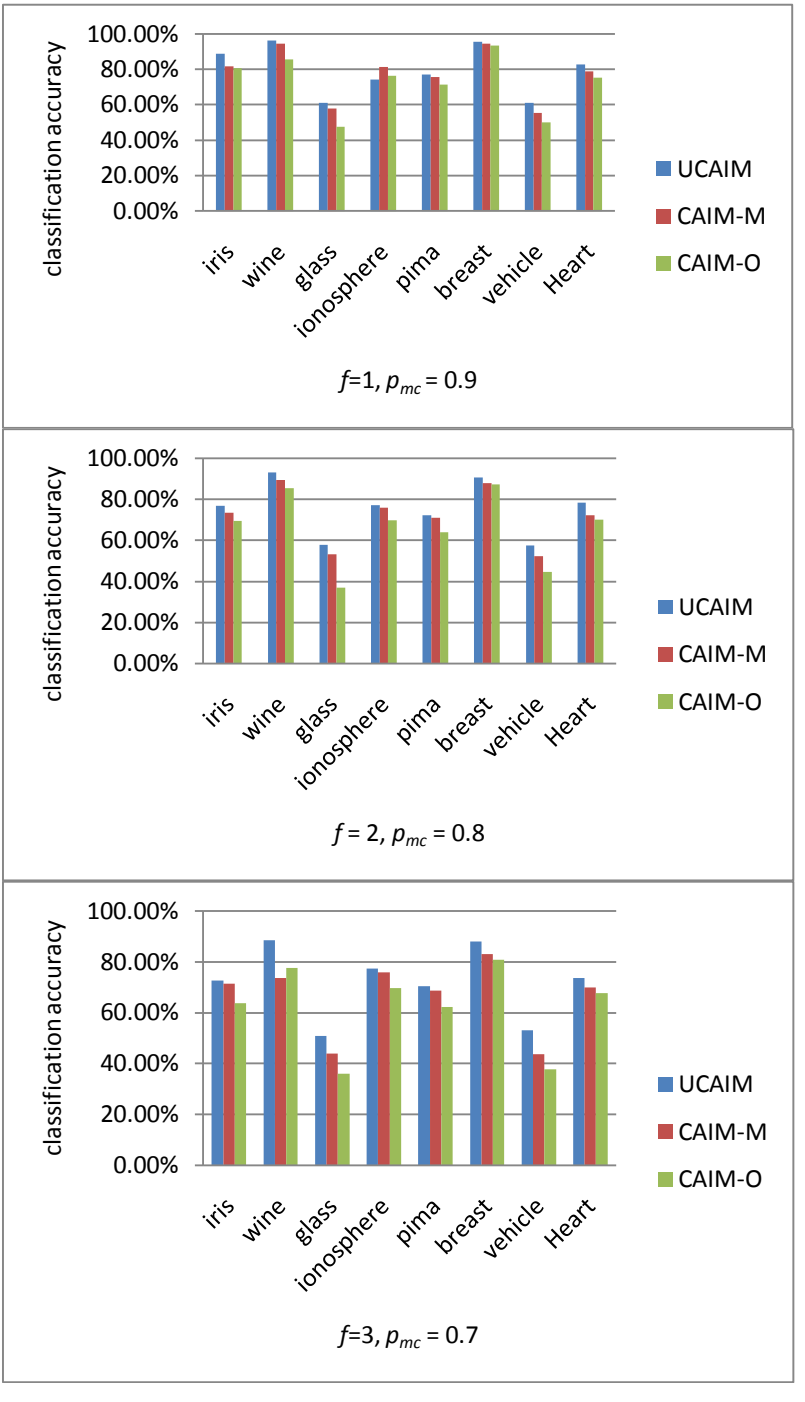

Fig. 1. Classification accuracies with different discretization methods under different uncertain level

Table 9. Average classification accuracies with different discretization methods under different uncertain level

Uncertain level	UCAIM	CAIM-M	CAIM-O
f=1, p_{mc}=0.9	79.53%	77.44%	72.51%
f=2, p_{mc}=0.8	78.19%	72.16%	70.15%
f=3, p_{mc}=0.7	71.79%	66.26%	61.96%

From table 8, table 9 and figure 1, we can see that UCAIM outperforms the other two algorithms in most cases. Particularly, UCAIM has a more significant performance improvement for datasets with higher uncertainty. That is because UCAIM utilizes extra information such as probability distribution of uncertain data, and employs the new criterion to retrieve the class-attribute interdependency which is not obvious when data is uncertain. Therefore, the discretization process of UCAIM is more sophisticated and comprehensive, and the discretized data can help data mining algorithms such as Naïve Bayesian classifier to gain a higher accuracy.

6 Conclusion

In this paper, we propose a new discretization algorithm for uncertain data. We employ both the formula based and sample based probability distribution function to model data uncertainty. We use probability cardinality to build the uncertain quanta matrix, which is then used to calculate *ucaim* to find the optimal discretization scheme with highest class-attribute interdependency. Experiments show that our algorithm can help the naïve Bayesian classifier to reach higher classification accuracy. We also observe that the proper use of data uncertainty information can significantly improve the quality of data miming results and we plan to explore more data mining approaches for various uncertain models in the future.

References

1. Kaufman, K.A., Michalski, R.S.: Learning from inconsistent and noisy data: the AQ18 approach. In: Proceeding of 11th International Symposium on Methodologies for Intelligent Systems (1999)
2. Cios, K.J., et al.: Hybrid inductive machine learning: an overview of clip algorithm. In: Jain, L.C., Kacprzyk, J. (eds.) New Learning Paradigms in Soft Computing, pp. 276–322. Springer, Heidelberg (2001)
3. Clark, P., Niblett, T.: The CN2 Algorithm. Machine Learning 3(4), 261–283 (1989)
4. Catlett, J.: On Changing Continues Attributes into Ordered Discrete Attributes. In: Kodratoff, Y. (ed.) EWSL 1991. LNCS, vol. 482, pp. 164–178. Springer, Heidelberg (1991)
5. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: An Enable Technique. Data Mining and Knowledge Discovery 6, 393–423 (2002)

6. Fayyad, U.M., Irani, K.B.: Multi-Interval Discretization of Continues- Valued Attributes for Classification Learning. In: Proceedings of the 13th Joint Conference on Artificial Intelligence, pp. 1022–1029 (1993)
7. Hanse, M.H., Yu, B.: Model Selection and the Principle of Minimum Description Length. Journal of the American Statistical Association (2001)
8. Kurgan, L.A.: CAIM Discretization Algorithm. In: IEEE Transactions on Knowledge and Data Engineering, p. 145 (2004)
9. Aggarwal, C.C., Yu, P.: A framework for clustering uncertain data streams. In: IEEE International Conference on Data Engineering, ICDE (2008)
10. Cormode, G., McGregor, A.: Approximation algorithms for clustering uncertain data. In: Principle of Data base System, PODS (2008)
11. Kriegel, H., Pfeifle, M.: Density-based clustering of uncertain data. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp. 672–677 (2005)
12. Singh, S., Mayfield, C., Prabhakar, S., Shah, R., Hambrusch, S.: Indexing categorical data with uncertainty. In: IEEE International Conference on Data Engineering (ICDE), pp. 616–625 (2007)
13. Kriegel, H., Pfeifle, M.: Hierarchical density-based clustering of uncertain data. In: IEEE International Conference on Data Mining (ICDM), pp. 689–692 (2005)
14. Aggarwal, C.C.: On Density Based Transforms for uncertain Data Mining. In: IEEE International Conference on Data Engineering, ICDE (2007)
15. Aggarwal, C.C.: A Survey of Uncertain Data Algorithms and Applications. IEEE Transactions on Knowledge and Data Engineering 21(5) (2009)
16. Ren, J., et al.: Naïve Bayes Classification of Uncertain Data. In: IEEE International Conference on Data Mining (2009)
17. Dougherty, J., Kohavi, R., Sahavi, M.: Supervised and Unsupervised Discretization of Continues Attributes. In: Proceedings of the 12th International Conference on Machine Learning, pp. 194–202 (1995)
18. Linde, Y., Buzo, A., Gray, R.M.: An Algorithm for Vector Quantizer Design. IEEE Transactions on Communications 28, 84–95 (1980)
19. Wong, A.K.C., Chiu, D.K.Y.: Synthesizing Statistical Knowledge from Incomplete Mixed-Mode Data. IEEE Transactions on Pattern Analysis and Machine Intelligence 9, 796–805 (1987)
20. Kurgan, L., Cios, K.J.: Fast Class-Attribute Interdependence Maximization (CAIM) Discretization Algorithm. In: Proceeding of International Conference on Machine Learning and Applications, pp. 30–36 (2003)
21. Kerber, R.: ChiMerge: discretization of numeric attributes. In: Proceeding of 9th International Conference on Artificial Intelligence, pp. 123–128 (1992)
22. Liu, H., Setiono, R.: Feature Selection via discretization. IEEE Transactions on knowledge and Data Engineering 9(4), 642–645 (1997)
23. Tray, F., Shen, L.: A modified Chi2 algorithm for discretization. IEEE Transactions on Knowledge and Data Engineering 14(3), 666–670 (2002)
24. Su, C.T., Hsu, J.H.: An extended Chi2 algorithm for discretization of real value attributes. IEEE Transactions on Knowledge and Data Engineering 17(3), 437–441 (2005)
25. Jing, R., Breitbart, Y.: Data Discretization Unification. In: IEEE International Conference on Data Mining, p. 183 (2007)
26. Berzal, F., et al.: Building Multi-way decision Trees with Numerical Attributes. Information Sciences 165, 73–90 (2004)
27. Bi, J., Zhang, T.: Support Vector Machines with Input Data Uncertainty. In: Proc. Advances in Neural Information Processing Systems (2004)

28. Qin, B., Xia, Y., Li, F.: DTU: A Decision Tree for Classifying Uncertain Data. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 4–15. Springer, Heidelberg (2009)
29. Cheng, R., Kalashnikov, D., Prabhakar, S.: Evaluating Probabilistic Queries over Imprecise Data. In: Proceedings of the ACM SIGMOD, pp. 551–562 (2003)
30. Asuncion, A., Newman, D.: UCI machine learning repository (2007), http://www.ics.uci.edu/mlearn/MLRepository.html
31. Aggarwal, C.C., Yu, P.S.: Outlier Detection with Uncertain Data. In: SIAM International Conference on Data Mining (2009)
32. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2006)