

Associative Classifier for Uncertain Data*

View metadata, citation and similar papers at core.ac.uk

brought to you by  CORE

provided by University of Queensland eSpace

{xiangju, zhangyang}nwsuaf.edu.cn

² School of Information Technology and Electrical Engineering,
The University of Queensland, Australia
xueli@itee.uq.edu.au

³ School of Computer, Northwest Polytechnical University, P.R. China
wangyong@nwpu.edu.cn

Abstract. Associative classifiers are relatively easy for people to understand and often outperform decision tree learners on many classification problems. Existing associative classifiers only work with certain data. However, data uncertainty is prevalent in many real-world applications such as sensor network, market analysis and medical diagnosis. And uncertainty may render many conventional classifiers inapplicable to uncertain classification tasks. In this paper, based on U-Apriori algorithm and CBA algorithm, we propose an associative classifier for uncertain data, uCBA (uncertain Classification Based on Associative), which can classify both certain and uncertain data. The algorithm redefines the support, confidence, rule pruning and classification strategy of CBA. Experimental results on 21 datasets from UCI Repository demonstrate that the proposed algorithm yields good performance and has satisfactory performance even on highly uncertain data.

Keywords: Associative Classification, Uncertain Data, Multiple Rules Classification, Expected Support.

1 Introduction

In recent years, due to advances in technology and deep understanding of data acquisition and processing, uncertain data has attracted more and more attention in the literature. Uncertain data is ubiquitous in many real-world applications, such as environmental monitoring, sensor network, market analysis and medical diagnosis [1]. A number of factors contribute to the uncertainty. It may be caused by imprecision measurements, network latencies, data staling and decision errors [2,3]. Uncertainty can arise in categorical attributes and numeric attributes [1,2]. For example, in cancer diagnosis, it is often very difficult for the doctor to exactly classify a tumor to be benign or malignant due to the experiment precision limitation. Therefore it would be better to represent by probability to be benign or malignant [2].

* This work is supported by the National Natural Science Foundation of China (60873196) and Chinese Universities Scientific Fund (QN2009092).

** Corresponding author.

Associative classifiers are relatively easy for people to understand and often outperform decision tree learners on many classification problems [5,7,8]. However, data uncertainty may render many conventional classifiers inapplicable to uncertain classification tasks. Consequently, the following adaptations are required to ensure that the extension to CBA [5] can classify uncertain data: Firstly, due to uncertainty, we need to modify the initial definition of *support* and *confidence* of associative rules [11] to mine association rules from uncertain data. Secondly, CBA only utilizes the rule with the highest *confidence* for classification. For uncertain data, an instance may be partially covered by a rule. We define the weight of instance covered by a rule and introduce *multiple rules classification*, and this help to improve the performance of the proposed algorithm. To the best of our knowledge, this is the first work devoted to associative classification of uncertain data.

To sum up, in this paper, based on the *expected support* [4], we extend CBA algorithm [5] and propose an associative classifier, uCBA, for uncertain data. We perform experiments on real datasets with uncertainty, and the experimental results demonstrate that uCBA algorithm perform well even on highly uncertain data.

This paper is organized as follows. In the next section, we survey the related work. Section 3 gives problem statement. Section 4 illustrates the proposed algorithm in detail. The experimental results are shown in Section 5. And finally, we conclude the paper and give future work in Section 6.

2 Related Work

A detailed survey of uncertain data mining techniques may be found in [9]. In the case of uncertain data mining, studies include clustering [23,24,25], classification [2,3,10,22], frequent itemsets mining [4,12,13,16,17] and outlier detection [26]. Here, we mainly focus on associative classification of uncertain data.

At present, existing works about classification of uncertain data all fall into extensions to traditional classification algorithms. Qin *et al.* proposed a rule-based algorithm to cope with uncertain data [2]. Later, in [3], Qin *et al.* presented DTU, which based on decision tree algorithm, to deal with uncertain data by extending traditional measurements, such as information entropy and information gain. In [10], Tsang *et al.* extended classical decision tree algorithm and proposed UDT algorithm to handle uncertain data which is represented by probability density function (pdf). In [22], Bi *et al.* proposed Total Support Vector Classification (TSVC), a formulation of support vector classification to handle uncertain data. Associative classifiers for certain dataset have been widely studied [5,7,8]. However, the problem studied in this paper is different from works mentioned above, we consider classification of uncertain data from the perspective of association rule mining and propose an associative classifier for uncertain data, uCBA.

Recently, there have been some studies on frequent itemset mining from uncertain transaction databases. In [4], Chui *et al.* extended Apriori [11] algorithm and proposed U-Apriori algorithm to mine frequent itemsets from uncertain data. U-Apriori computes the *expected support* of itemsets by summing all itemset probabilities. Later, in [12], they additionally proposed a probabilistic filter in order to prune candidates early.

Leung *et al.* proposed the UF-growth algorithm in [13]. Like U-Apriori, UF-growth computes frequent itemsets based on the *expected support*, and it uses the FP-tree [14] approach in order to avoid expensive candidate generation. In [15], Aggarwal *et al.* extended several classical frequent itemset mining algorithms to study their performances when applied to uncertain data. In [16], Zhang *et al.* proposed exact and sampling-based algorithms to find likely frequent items in streaming probabilistic data. In [17], Bernecker *et al.* proposed to find frequent itemsets from uncertain transaction database in a probabilistic way. Different from studies in [4,12,13], work in [17] mined probabilistic frequent itemsets by means of the *probabilistic support*. All the works mentioned above belong to the framework of mining frequent itemsets from uncertain transaction databases, and do not consider mining association rules from uncertain data. In this paper, we apply the *expected support* [4] to mine associative rules from uncertain data, and then perform associative classification for uncertain data.

At present, there are few works about mining associative rules from uncertain data. In [18], Weng *et al.* developed an algorithm to mine fuzzy association rules from uncertain data which is represented by possibility distributions. In their study, there are relations between all the possible values of a categorical attribute, and they provide a similarity matrix to compute the similarity between values of this attribute. While recent studies in the literature about uncertain data management and mining generally base on *possible world model* [6]. In this paper, we integrate *possible world model* [6] into mining association rules from uncertain data.

3 Problem Statement

For simplicity, in this paper, we only consider uncertain categorical attributes, and following studies in [2,3], we also assume the class label is certain.

3.1 A Model for Uncertain Categorical Data

When dealing with uncertain categorical attribute, we utilize the same model as studies in [1,2,3] to represent uncertain categorical data. Under the uncertain categorical model, a dataset can have attributes that are allowed to take uncertain values [2]. And we call these attributes Uncertain Categorical Attributes, UCA. The concept of UCA was introduced in [1]. Let's write $A_i^{u_c}$ for the i^{th} uncertain categorical attribute, and $V_i = \{v_{i1}, v_{i2}, \dots, v_{i|V_i|}\}$ for its domain. As described in [2], for instance t_j , its attribute value of $A_i^{u_c}$ can be represented by the probability distribution over V_i , and formalized as $P_{ji} = \langle p_{i1}, p_{i2}, \dots, p_{i|V_i|} \rangle$, such that $P_{ji}(A_i^{u_c} = v_{ik}) = p_{ik} (1 \leq k \leq |V_i|)$, and $\sum_{k=1}^{|V_i|} p_{ik} = 1.0$, which means $A_i^{u_c}$ takes value of v_{ik} with probability p_{ik} . Certain attribute can be viewed as a special case of uncertain attribute. In this case, the attribute value of $A_i^{u_c}$ for instance t_j can only take one value, v_{ik} , from domain V_i , i.e. $P_{ji}(A_i^{u_c} = v_{ik}) = 1.0 (1 \leq k \leq |V_i|)$, $P_{ji}(A_i^{u_c} = v_{ih}) = 0.0 (1 \leq h \leq |V_i|, h \neq k)$.

3.2 Associative Rules for Uncertain Data

Let D be the uncertain dataset, Y be the set of class labels, and $y \in Y$ be a class label. Each uncertain instance $t \in D$ follows the scheme $(A_1^{u_c}, A_2^{u_c}, \dots, A_m^{u_c})$, where

$(A_1^{u_c}, A_2^{u_c}, \dots, A_m^{u_c})$ are m attributes. Following methods in [5,8], we also map all the possible values of each UCA to a set of attribute-value pairs. With these mappings, we can view an uncertain instance as a set of (attribute, attribute-value) pairs and a class label.

Definition 1. Let an item x be the (attribute, attribute-value) pair, denoted as $x = (A_i^{u_c}, v_{ik})$, where v_{ik} is a value of attribute $A_i^{u_c}$. Let I be the set of all items in D . An instance t_j satisfies an item $x = (A_i^{u_c}, v_{ik})$ if and only if $P_{ji}(A_i^{u_c} = v_{ik}) > 0.0$, where v_{ik} is the value of i^{th} attribute of t_j .

Following the definition of rule in [8], we can define a rule for uncertain data:

Definition 2. An associative rule R for uncertain data, is defined as $R : x_1 \wedge x_2 \wedge \dots \wedge x_l \rightarrow y$. Here $X = x_1 \wedge x_2 \wedge \dots \wedge x_l$ is the antecedent of R , and y is the class label of R .

An uncertain instance t satisfies R if and only if it satisfies every item in R . If t satisfies R , R predicts the class label of t to be y . If a rule contains zero item, then its antecedent is satisfied by any instance.

In U-Apriori [4] algorithm, to handle uncertain data, instead of incrementing the support counts of candidate itemsets by their actual support, the algorithm increments the support counts of candidate itemsets by their expected support under the possible world model.

Definition 3. The expected support of antecedent, X , of rule R on uncertain dataset D can be defined as [4]:

$$\text{expSup}(X) = \sum_{j=1}^{|D|} \prod_{x \in X} p_{t_j}(x) \quad (1)$$

where $\prod_{x \in X} p_{t_j}(x)$ is the joint probability of antecedent X in instance t_j [4], and $p_{t_j}(x)$ is the existence probability of item x in t_j , $p_{t_j}(x) > 0.0$.

Accordingly, we can compute the expected support of R , $\text{expSup}(R)$, as following:

$$\text{expSup}(R) = \sum_{j=1}^{|D|} \prod_{\substack{x \in X, t_j.\text{class}=y}} p_{t_j}(x) \quad (2)$$

Rule R is considered to be *frequent* if its expected support exceeds $\rho_s \cdot |D|$, where ρ_s is a user-specified expected support threshold.

Definition 4. For association rule $R : X \rightarrow y$ on uncertain data, with expected support $\text{expSup}(X)$ and $\text{expSup}(R)$, its confidence can be formalized as:

$$\text{confidence}(R) = \text{expSup}(R)/\text{expSup}(X) \quad (3)$$

The intuition behind $\text{confidence}(R)$ is to show the expected accuracy of rule R under the possible world model. Rule R is considered to be accurate if and only if its confidence exceeds ρ_c , where ρ_c is a user-specified confidence threshold.

Weight of Uncertain Instance Covered by a Rule. Due to data uncertainty, an instance may be partially covered by a rule. The intuition to define this weight is twofold. On

one hand, inspired by pruning rules based on database coverage [7], in classifier builder algorithm, instead of removing one instance from the training data set immediately after it is covered by some selected rule like CBA does, we let it stay there until its covered weight reached 1.0, which ensures that each training instance is covered by at least one rule. This allows us to select more rules. When classifying a new instance, it may have more rules to consult and better chance to be accurately predicted. On the other hand, in *multiple rules classification* algorithm, we utilize this weight to further control the number of matched rules using to predict a test instance. In this paper, we follow the method proposed in [2] to compute the weight of instance covered by rule.

Definition 5. We define the weight, $w(t_j, R_l)$, of instance t_j covered by the l^{th} rule R_l in the rule sequence as following:

$$\begin{aligned} w(t_j, R_1) &= 1.0 * P(t_j, R_1) \\ w(t_j, R_2) &= (1.0 - w(t_j, R_1)) * P(t_j, R_2) \\ &\dots \\ w(t_j, R_l) &= (1.0 - \sum_{k=1}^{l-1} w(t_j, R_k)) * P(t_j, R_l) \end{aligned} \quad (4)$$

In formula (4), $P(t_j, R_l)$ is the probability of instance t_j that covered by rule R_l , which can be formalized as following:

$$P(t_j, R_l) = \prod_{x \in R_l.\text{Antecedent}} p_{t_j}(x) \quad (5)$$

Multiple Rules Classification. When classifying an uncertain instance, a natural approach is directly following CBA [5], which only utilizes the rule with the highest confidence for classification, i.e. *single rule classification*. However, this direct approach ignores uncertain information in the instance, and hence may decrease the prediction accuracy.

Note that CMAR [7] performs classification based on a weighted χ^2 analysis by using multiple strong associative rules, we can follow the idea of CMAR and modify the formula of weighted χ^2 by using the expected support of rules. In this paper, we refer to methods in [8,10,2] to predict class label, which classifies the test instances by combining multiple rules. Similar to works in [8,10,2], which use the expected accuracy of rules on training dataset [8,10] or the class probability distribution of training instances that follow at the tree leaf to classify the test instance [2], we utilize the confidence of rules to compute the class probability distribution of test instance t to predict the class label, which can be formalized as:

$$y = \arg \max_{y \in Y} \left\{ \sum_{R_l.\text{class}=y} \text{confidence}(R_l) * w(t, R_l) | y \in Y, R_l \in rs \right\} \quad (6)$$

where rs is the set of rules that t matches. This method is also similar to the idea of CMAR.

Pruning Rules based on Pessimistic Error Rate. Since there are strong associations in some datasets, the number of association rules can be huge, and there may be a large

number of insignificant association rules, which make no contributions to classification and may even do harm to classification by introducing noise. Consequently, rule pruning is helpful. Following CBA [5], we also utilize pessimistic error rate (PER) based pruning method presented in C4.5 [19] to prune rules, which is formalized as:

$$e = \frac{r + \frac{z^2}{2n} + z\sqrt{\frac{r}{n} - \frac{r^2}{n} + \frac{z^2}{4n}}}{1 + \frac{z^2}{n}} \quad (7)$$

Here under the uncertain data scenario, r is the observed error rate of rule R_l , $r = 1.0 - confidence(R_l)$; n is the expected support of R_l , $n = expSup(R_l)$; $z = \Phi^{-1}(c)$, c is confidence level given by the user.

4 uCBA Algorithm

Based on CBA [5], we propose uCBA algorithm for associative classification of uncertain data. It consists of three parts, a *rule generator* (uCBA-RG), a *classifier builder* (uCBA-CB), and *multiple rules classification* (uCBA-MRC).

4.1 Algorithm for uCBA-RG

Conventional CBA mines associative rules based on Apriori [5] algorithm, while uCBA generates uncertain associative rules based on U-Apriori [4] algorithm. And the general framework of uCBA-RG is the same with that of CBA-RG [5], the differences lie in their ways to accumulate the support counts: if instance t matches R_l , then for CBA-RG, *support* of antecedent X increase incrementally by 1, i.e. $X.condSupCount++$. Furthermore, if R_l and t have the same class label, then *support* of rule also increase incrementally by 1, i.e. $X.ruleSupCount++$ [5]. While for uCBA-RG, if t matches R_l , then the *expected support* of X increase incrementally by probability of t covered by R_l , i.e. $X.expSup += P(t, R_l)$, $P(t, R_l)$ can be computed following formula (5). *Expected support* of R_l updates similarly. The algorithm is omitted here for lack of space.

4.2 Algorithm for uCBA-CB

Here we give our uCBA-CB algorithm, which is illustrated in Algorithm 1. It has three steps:

Step 1 (Line 1): According to relation ' \succ ' for rules in CBA [5], we sort the set of generated rules R based on *expected support* and *confidence*, which guarantees that we will choose the highest precedence rules for our classifier.

Step 2 (line 2-23): Selecting rules for the classifier from R following the sorted sequence. For each rule R_l , we traverse D to find those instances covered by R_l (line 7) and compute *probability* (line 8) and *weight* (line 9) of instances covered by R_l . If weight of instance covered by R_l is greater than 0, then records the weight and marks R_l if it correctly classifies t_j (line 11). If R_l is marked, then it will be a potential rule in our classifier (line 15,16). Meanwhile, we need to update the weight of instances covered by R_l (line 17,18,19). For uncertain data, we should ensure that the total weight of each

Algorithm 1. Classifier Builder for uCBA**Input:**

D : Training Dataset ;
 R : A rule set generated by Algorithm uCBA-RG;

Output:

C : The final uCBA Classifier;
1: $R = \text{sort}(R)$;
2: $C = \emptyset$;
3: Initialize $\text{totalWeight}[j] = 0, j \in [1, |D|]$;
4: **for** each rule $R_l \in R$ in sequence **do**
5: Initialize $\text{curCoverWeight}[j] = 0, j \in [1, |D|]$;
6: **for** each instance $t_j \in D$ **do**
7: **if** $\text{totalWeight}[j] \leq 1.0$ **&&** t_j satisfies the antecedent of R_l **then**
8: Compute $P(t_j, R_l)$ following formula (5);
9: Compute $w(t_j, R_l)$ following formula (4);
10: **if** $w(t_j, R_l) > 0$ **then**
11: $\text{curCoverWeight}[j] = w(t_j, R_l)$ and mark R_l if it correctly classifies t_j ;
12: **end if**
13: **end if**
14: **end for**
15: **if** R_l is marked **then**
16: $C = C \cup \{R_l\}$;
17: **for** $k = 1$ to $|D|$ **do**
18: $\text{totalWeight}[k] += \text{curCoverWeight}[k]$;
19: **end for**
20: Select a default class for the current C ;
21: Compute the total errors of C ;
22: **end if**
23: **end for**
24: Find the rule $R_k \in C$ with the lowest total errors and drop all the rules after $R_k \in C$;
25: Add the default class associated with R_k to the end of C ;
26: **return** C ;

instance covered by all the matching rules is not greater than 1.0 (line 7). Similar to CBA-CB [5], we also select a default class for each potential rule in the classifier (line 20). We then compute and record the total errors that are made by the current C and the default class (line 21). When there is no rule or no training instance left, the rule selection procedure terminates.

Step 3 (line 24-26): Selecting the set of the most predictable rules on training dataset as the final classifier. Same with CBA [5], for our uCBA-CB, the first rule at which there is the least total errors recorded on D is the cutoff rule.

4.3 Algorithm for uCBA-MRC

Here we introduce the algorithm of *multiple rules classification* in uCBA, which is given in Algorithm 2, it has two steps:

Step 1 (line 1-13) : For the test instance t_j , we traverse classifier C to find the matched rules (line 3). Note that, if we use all the matched rules to predict the class

label, and do not filter rules with low precedence, it may introduce noise and decrease the prediction accuracy. And we will validate this idea in the experimental study. Therefore, in our uCBA-MRC, under the circumstances of ensuring that the *weight* of each test instance covered by rules is less than 1.0, we further constrain the number of multiple rules not to exceed a user-specified threshold (line 4). When t_j satisfies R_l , we compute the *probability* (line 5) and *weight* (line 6) of instance covered by R_l . If the covered *weight* is greater than 0, then we update the *weight* of instance covered by the matched rules (line 8), and insert R_l into the *multiple rule set* (line 10).

Step 2 (line 14-15): Predict the class label of instance according to formula (6).

Algorithm 2. Multiple-Rule Classification for uCBA

Input:

C : The final uCBA Classifier generated by Algorithm 1;
 t_j : A testing instance;
 $covThreshold$: The coverage threshold;

Output:

y : The class label predicted for t_j ;
1: $totalWeight = 0$;
2: Initialize the multiple-rule set: $rs = \phi$;
3: **for** each rule $R_l \in C$ in sorted order **do**
4: **if** $totalWeight \leq 1.0 \ \&\& |rs| < covThreshold \ \&\& t_j$ satisfies R_l **then**
5: Compute $P(t_j, R_l)$ following formula (5);
6: Compute $w(t_j, R_l)$ following formula (4);
7: **if** $w(t_j, R_l) > 0$ **then**
8: $totalWeight += w(t_j, R_l)$;
9: Record $w(t_j, R_l)$ for prediction;
10: $rs = rs \cup \{R_l\}$;
11: **end if**
12: **end if**
13: **end for**
14: Predict the class label, y , of t_j using rs and recorded weights, following formula (6);
15: **return** y ;

5 Experimental Study

In order to evaluate the classification performance of our uCBA algorithm, we perform experiments on 21 datasets from UCI [20]Repository. At present, for simplicity, our algorithm only consider uncertain categorical attribute. For datasets with numeric attributes, each numeric attribute is first discretized into a categorical one using method as in [5]. At present, there are no standard and public uncertain datasets in the literature. Experiments on uncertain data in the literature all perform on synthetic datasets, which means researchers obtain uncertain dataset via introducing uncertain information into certain dataset [2,3,10].

For all of the experiments in this paper, we utilize the model introduced in Section 3.1 to represent uncertain dataset. For example, as described in [2], when we introduce 20% uncertainty, this attribute will take the original value with 80% probability, and take other values with 20% probability. Meanwhile, we utilize Information Gain, IG,

to select the top K attributes with maximum IG values, and transform these top K attributes into uncertain ones. And the uncertain dataset is denoted by TopKuA (Top K uncertain Attribute).

Our algorithms are implemented in Java based on the WEKA¹ software packages, and the experiments are conducted on a PC with Core 2 CPU, 2.0GB memory and Windows XP OS. We set *expected support* threshold to 1% and *confidence* threshold to 50%. Following CBA [5], we also set a limit of 80,000 on the total number of candidate rules in memory. As in [2,3,10], we measure the classification performance of the proposed classifier by accuracy. All the experimental results reported here are the average accuracy of 10-fold cross validation.

5.1 Performance of uCBA on Uncertain Datasets

In this group of experiment, we evaluate the performance of uCBA algorithm on different level of uncertain dataset. In the following, UT represents dataset with $T\%$ uncertainty, and we denote certain dataset as $U0$. Note that, uCBA algorithm is equivalent to the traditional CBA when applying to certain dataset, that is to say, when we set $T = 0$, our uCBA algorithm performs the same as CBA does.

Table 1. The Comparison of uCBA for Top2uA on Accuracy

Dataset	CBA	U10		U20		U30		U40	
		Single	Multiple	Single	Multiple	Single	Multiple	Single	Multiple
balance-scale	0.693	0.685	0.747	0.685	0.752	0.685	0.722	0.685	0.685
breast-w	0.959	0.930	0.961	0.944	0.953	0.956	0.961	0.951	0.960
credit-a	0.855	0.725	0.762	0.761	0.768	0.772	0.775	0.752	0.788
diabetes	0.770	0.699	0.704	0.698	0.699	0.699	0.714	0.699	0.701
heart-c	0.792	0.769	0.795	0.785	0.785	0.799	0.782	0.832	0.828
heart-h	0.837	0.827	0.813	0.813	0.823	0.827	0.827	0.823	0.816
hepatitis	0.813	0.826	0.794	0.819	0.826	0.819	0.839	0.806	0.819
hypothyroid	0.982	0.923	0.923	0.923	0.923	0.923	0.923	0.923	0.923
ionosphere	0.929	0.923	0.929	0.915	0.929	0.915	0.920	0.926	0.909
labor	0.912	0.860	0.930	0.877	0.912	0.877	0.912	0.877	0.912
lymph	0.824	0.797	0.824	0.791	0.818	0.804	0.797	0.824	0.797
segment	0.946	0.900	0.925	0.904	0.924	0.907	0.926	0.917	0.930
sick	0.976	0.939	0.939	0.939	0.939	0.939	0.939	0.939	0.939
sonar	0.817	0.822	0.841	0.822	0.837	0.846	0.856	0.841	0.841
soybean	0.893	0.748	0.792	0.808	0.881	0.865	0.899	0.883	0.895
breast-cancer	0.654	0.664	0.692	0.696	0.661	0.675	0.668	0.689	0.661
car	0.974	0.700	0.864	0.700	0.853	0.700	0.825	0.700	0.782
kr-vs-kp	0.975	0.912	0.909	0.859	0.854	0.819	0.823	0.804	0.802
mushroom	0.9995	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998
nursery	0.935	0.671	0.675	0.671	0.674	0.671	0.673	0.671	0.667
vote	0.940	0.883	0.897	0.890	0.887	0.890	0.910	0.897	0.890
AveAccuracy	0.880	0.819	0.844	0.824	0.843	0.828	0.842	0.830	0.835

¹ <http://www.cs.waikato.ac.nz/ml/weka/>

Table 1 gives the performance for uCBA on Top2uA and different level of uncertainty (U0-U40) dataset. In Table 1, column *CBA* represents accuracy of CBA on certain datasets; column *Single* represents accuracy of uCBA's *single rule classification*; and column *Multiple* represents accuracy of uCBA's *multiple rules classification*.

As shown in Table 1, with increasing of uncertain level, the accuracy of uCBA degrades to some extent. It can be observed from Table 1 that, in most cases, the accuracy of *Multiple* exceeds that of *Single*; and on all the uncertainty datasets, the averaged accuracy of *Multiple* is higher than that of *Single*. It can be observed from Table 1 that, uCBA performs differently for different datasets. For some datasets, for example, balance-scale, labor and sonar, when introducing uncertainty into the datasets, uCBA-MRC can improve the prediction accuracy comparing with the accuracy of CBA. For most datasets, the performance decrement are within 7%, even when data uncertainty reaches 30%. The worst performance decrement is for the nursery dataset, the classifier has over 94% accuracy on certain data, reduces to around 67.5% when the uncertainty is 10%, to 67.4% when the uncertainty is 20%, and to 67.3% when the uncertainty reaches 30%.

The similar experiment results could be observed when we set K to 1, 3 and other values, and are omitted here for limited space. Overall speaking, the accuracy of uCBA classifier remains relatively stable. Even when the uncertainty level reaches 40% (U40), the average accuracy of uCBA-MRC (83.5%) on 21 datasets is still quite comparable to CBA (88.0%), and only decreases by 4.5% on accuracy. The experiments shows uCBA is quite robust against data uncertainty. Meanwhile, the difference of accuracy of the two methods among different experiment settings is significant on paired-sample t-Test [21], which means *Multiple* is more robust than *Single*.

5.2 Parameter Analysis on *coverThreshold*

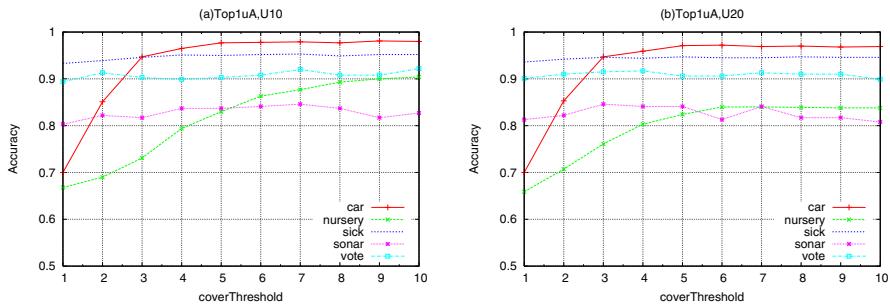
In this group of experiment, we analyze the effect of parameter, *coverThreshold*, on accuracy. As discussed earlier, this parameter controls the number of rules for classification. Generally speaking, for uncertain data classification, if we use very few rules for classification, the instance may not be fully covered by rules, and this may lead to bad classification performance; on the other hand, if we use too many rules for classification, we may introduce noise and it may also lead to bad classification performance.

As an example, we analyze the effect of *coverThreshold* on accuracy over 5 uncertain datasets with different level of uncertainty. From Fig.1, we can see that when the number of rules for classification exceeds 5, the performance of uCBA over the 5 datasets tend to be stable. Therefore, we set this parameter to 5 in all of the experiments in this paper.

5.3 Time and Space Analysis of uCBA

Here we analyze the number of association rules generated and the time token to generate these rules. We select 5 datasets from UCI Repository to perform this experiment, and analyze time and space consumption over Top1uA, Top2uA with U20 uncertain dataset. In Table 2, column *w/o pru* represents without rule pruning, and column *pru* represents rule pruning with PER following formula (7).

We can see from Table 2 that PER can greatly reduce the number of rules, and prune insignificant rules. Meanwhile, we can also see that the number of association rules on

**Fig. 1.** Experiment with parameter *coverThreshold***Table 2.** Analysis of Time and the Number of Rules

Dataset	CBA		U20, No. of Rules				U20, Run time(s)		U20	
	No. of Rules		Top1uA		Top2uA		uCBA-RG, pru		No. of Rules in C	
	w/o pru	pru	w/o pru	pru	w/o pru	pru	Top1uA	Top2uA	Top1uA	Top2uA
car	1072	118	1063	108	1090	98	0.5	0.6	14	19
nursery	2976	415	2919	402	2842	346	12.9	20.3	178	132
sick	15037	6279	15360	6073	16165	5862	23.4	25.6	242	317
sonar	3307	2795	3303	2783	3306	2796	1.0	1.0	28	27
vote	25590	2033	26620	2197	27525	2221	3.8	4.4	88	243
Average	9596	2328	9853	2312	10186	2265	8	10	110	148

uncertain data is larger than that on certain data, this is because there are many uncertain information in uncertain data. It is shown that under the same level of uncertainty, the more the number of uncertain attributes, the longer it takes to mine association rules.

6 Conclusion and Future Work

Data uncertainty is prevalent in many real-world applications. In this paper, based on the *expected support*, we extend CBA algorithm and propose an associative classifier, uCBA, for uncertain data classification task. We redefine the support, confidence, rule pruning and classification strategy of CBA to build uncertain associative classifier. Experimental results on 21 datasets from UCI Repository demonstrate that the proposed algorithm yields good performance and has satisfactory performance even on highly uncertain data.

At present, our proposed algorithm only considers uncertain categorical attributes. We will consider uncertain numeric attributes in our future work.

References

1. Singh, S., Mayfield, C., Prabhakar, S., Shah, R., Hambrusch, S.: Indexing Uncertain Categorical Data. In: Proc. of ICDE 2007, pp. 616–625 (2007)
2. Qin, B., Xia, Y., Prabhakar, S., Tu, Y.: A Rule-based Classification Algorithm for Uncertain Data. In: The Workshop on Management and Mining of Uncertain Data, MOUND (2009)

3. Qin, B., Xia, Y., Li, F.: DTU: A Decision Tree for Uncertain Data. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 4–15. Springer, Heidelberg (2009)
4. Chui, C.K., Kao, B., Hung, E.: Mining frequent itemsets from uncertain data. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 47–58. Springer, Heidelberg (2007)
5. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: KDD, pp. 80–86 (1998)
6. Zimányi, E., Pirotte, A.: Imperfect information in relational databases. In: Uncertainty Management in Information Systems, pp. 35–88 (1996)
7. Li, W., Han, J., Pei, J.: CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. In: Proc. of ICDM 2001, pp. 369–380 (2001)
8. Yin, X., Han, J.: CPAR: Classification based on Predictive Association Rules. In: Proc. of SDM 2003, pp. 331–335 (2003)
9. Aggarwal, C.C., Yu, P.S.: A survey of Uncertain Data Algorithms and Applications. *IEEE Transactions on Knowledge and Data Engineering* 21(5), 609–623 (2009)
10. Tsang, S., Kao, B., Yip, K.Y., Ho, W.-S., Lee, S.D.: Decision Trees for Uncertain Data. In: Proc. of ICDE 2009, pp. 441–444 (2009)
11. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proc. of 20th VLDB, pp. 487–499. Morgan Kaufmann, San Francisco (1994)
12. Chui, C., Kao, B.: A decremental approach for mining frequent itemsets from uncertain data. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 64–75. Springer, Heidelberg (2008)
13. Leung, C.K.-S., Carmichael, C.L., Hao, B.: Efficient mining of frequent patterns from uncertain data. In: Proc. of ICDM Workshops, pp. 489–494 (2007)
14. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD Record, pp. 1–12 (2000)
15. Aggarwal, C.C., Li, Y., Wang, J., Wang, J.: Frequent pattern mining with uncertain data. In: Proc. of KDD 2009, pp. 29–38 (2009)
16. Zhang, Q., Li, F., Yi, K.: Finding Frequent Items in Probabilistic Data. In: Proc. of SIGMOD 2008, pp. 819–832 (2008)
17. Bernecker, T., Kriegel, H.P., Renz, M., Verhein, F., Zuefle, A.: Probabilistic frequent itemset mining in uncertain databases. In: Proc. of SIGKDD 2009, pp. 119–128 (2009)
18. Weng, C.-H., Chen, Y.-L.: Mining fuzzy association rules from uncertain data. *Knowledge and Information Systems* (2009)
19. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufman Publishers, San Francisco (1993)
20. <http://archive.ics.uci.edu/ml/datasets.html>
21. Dietterich, T.: Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation* 10(7), 1895–1923 (1998)
22. Bi, J., Zhang, T.: Support Vector Classification with Input Data Uncertainty. In: NIPS, pp. 161–168 (2004)
23. Ngai, W.K., Kao, B., Chui, C.K., Cheng, R., Chau, M., Yip, K.Y.: Efficient clustering of uncertain data. In: Perner, P. (ed.) ICDM 2006. LNCS (LNAI), vol. 4065, pp. 436–445. Springer, Heidelberg (2006)
24. Lee, S.D., Kao, B., Cheng, R.: Reducing UK-means to K-means. In: Proc. of ICDM Workshops, pp. 483–488 (2007)
25. Cormode, G., McGregor, A.: Approximation Algorithms for Clustering Uncertain Data. In: PODS 2008, pp. 191–200 (2008)
26. Aggarwal, C.C., Yu, P.S.: Outlier Detection with Uncertain Data. In: Jonker, W., Petković, M. (eds.) SDM 2008. LNCS, vol. 5159, pp. 483–493. Springer, Heidelberg (2008)