Naive Bayes Classifier for Positive Unlabeled Learning with Uncertainty*

Jiazhen He† Yang Zhang‡ Xue Li§ Yong Wang¶

Abstract

Existing algorithms for positive unlabeled learning (PU learning) only work with certain data. However, data uncertainty is prevalent in many real-world applications such as sensor network, market analysis and medical diagnosis. In this paper, based on positive naive Bayes (PNB), which is a PU learning algorithm for certain data, we propose an algorithm to handle uncertain data . However, it requires the prior probability of positive class and in real-life applications it is generally difficult for the users to provide this parameter, which is a drawback inherited from traditional PNB algorithm. We improve it by selecting the value of the prior probability of positive class automatically that can make the obtained classifier achieved optimal performance on the validation set. The conducted experiments show that the proposed algorithm yields good performance without user-specified the prior probability of positive class and has satisfactory performance even on highly uncertain data.

1 Introduction

A key problem in data mining, classification under PU learning scenario which builds a classifier using only labeled positive examples and unlabeled examples has been widely investigated recently. Current algorithms for PU learning are all dedicated to handle certain data [2, 3, 11, 12, 13, 14, 26]. However, data uncertainty is often found in real-world applications such as sensor network, market analysis and medical diagnosis where the precise values of data might be unknown due to imprecise measurement, outdated sources, or decision errors. Uncertainty may appear in numerical attributes. For example, vast amount of uncertain data are present in sensor network as a result of the imperfect hardware used for the collection process. Uncertainty can also arise in categorical attributes. For instance, in cancer diagnosis, it is difficult for the doctor to accurately decide a tumor to be benign or malignant due to the experiment precision limitation. Therefore it would be better to be represented by probabilities to be benign or malignant [1].

PNB is proposed by Denis [2] to deal with the PU learning problem in text classification domain. Nevertheless, it is only applicable in the case that the instances are represented as bag of words. Calvo *et al.* generalize it to handle general discrete attributes in [3]. In this paper, we address the problem of classification on uncertain data under PU learning scenario. Based on PNB [3], we propose an algorithm to cope with uncertain categorical attributes by means of the definition of probabilistic cardinality [1]. However, it requires the user to provide the prior probability of positive class, which is a shortcoming inherited from traditional PNB algorithm. In order to make up the defect, we select the value from 0.1 to 0.9 which could make the classification performance highest on the validation set as the prior probability of positive class.

This paper is organized as follows. In the next section, we discuss related work briefly. In Section 3 we define the problem of classification on uncertain data under PU learning scenario. Section 4 illustrates the proposed algorithms in detail. The experimental results are shown in Section 5. And finally, Section 6 concludes the paper.

2 Related Work

To the best of our knowledge, there is no work so far on classification of uncertain data under PU learning scenario. While there has been widely research in uncertain data management and mining.

A survey on uncertain data mining and management applications has been provided in [4]. In the case of uncertain data mining, studies include clustering [15, 16, 17], classification [1, 5, 6, 27, 7], frequent item mining [18, 19, 20, 21, 22] and outlier detection [23]. Here, we mainly focus on the studies on classification of uncertain data. Qin *et al.* proposed a rule-based algorithm to cope with uncertain data in [1] where the value of uncertain numeric attribute is represented as a range and the probability distribution function over this

*Supported by the National Natural Science Foundation of China (60873196) and Chinese Universities Scientific Fund (QN2009092).

†Northwest A&F University, China, hejiazhen@nwsuaf.edu.cn

‡Corresponding author, Northwest A&F University, China, zhangyang@nwsuaf.edu.cn

§The University of Queensland, Australia, xueli@itee.uq.edu.au

¶Northwest Polytechnical University, China, wangyong@nwpu.edu.cn

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

range, and the value of uncertain categorical attribute is represented as a set of values with associated probabilities. A new measure called probabilistic information gain was introduced for generating rules. Subsequently, in [5] Qin *et al.* presented a new decision tree for classifying and predicting both certain and uncertain data (DTU) by extending traditional measurement, such as information entropy and information gain. In [6], Tsang *et al.* extended classical decision tree algorithm to Uncertain Decision Tree (UDT) algorithm to handle uncertain data represented by probability density function (pdf) adopting the technique of fractional tuple for splitting tuples into subsets when the domain of its pdf spans across the split point. In [27], Ren *et al.* introduced a naive Bayes classification algorithm for uncertain data with pdf by extending the class conditional probability estimation in the naive Bayes model. Our work is different since our algorithm is devised for PU learning and we focus on uncertain categorical data. In [7], Bi *et al.* proposed Total Support Vector Classification (TSVC), a formulation of support vector classification to handle uncertain data.

The problem of PU learning can be addressed by discarding the unlabeled examples and learning from only labeled positive examples, e.g., One-Class SVM [8]. However, in this way, useful information in the unlabeled examples has not been considered. It is beneficial to construct a classifier using positive and unlabeled examples. A survey on the learning from positive and unlabeled examples can be found in [9] where PU learning algorithms, which focus on text classification tasks, were divided into three categories. In [3], Calvo *et al.* extended the PNB algorithm [2] which can only be applied to text classification, to cope with general discrete attributes. Furthermore, they extended it to more complex Bayesian classifier, positive tree augmented naive Bayes (PTAN), and the averaged version of these classifiers (APNB and APTAN) was introduced by stimulating the prior probability of positive class by means of Beta distribution.

To the best of our knowledge, all the current PU learning algorithms are devised for certain data. In this paper, we explore the classification of uncertain data under PU learning scenario. Here, we mainly focus on uncertain categorical data. Existing uncertain data classification algorithms require fully labeled training data. However, in many real-life applications, it is expensive and time-consuming to get fully labeled data. In this paper, only positive and unlabeled examples are utilized. We integrate data uncertainty into PNB algorithm. However, the traditional PNB algorithm [2, 3] requires the prior probability of positive (p), as a parameter, provided by the user. Although Calvo *et al.* proposed to model the uncertainty about p by means of Beta distribution, users need to provide the parameters for the Beta distribution, which is difficult for the users in real-life applications. Our algorithm is different from traditional PNB algorithm as it could yield good performance without user-specified parameter p.

3 Problem Definition

Here, we define the problem of classification on uncertain data under PU learning scenario formally.

Under PU learning with uncertainty scenario, the training dataset D consists of a set of positive examples P and a set of unlabeled examples U, $D = P \cup U$. Each example can be represented by $< X^{u_c}, C, S >$. Here, $X^{u_c} = \{X_1^{u_c}, \ldots, X_n^{u_c}\}$, representing the attribute vector with n uncertain categorical attributes; $C \in \{0, 1\}$, representing the class label of X^{u_c}, 0 for negative and 1 for positive; $S \in \{0, 1\}$, representing X^{u_c} is labeled ($S = 1$), or unlabeled ($S = 0$). Each instance in P is labeled and positive, represented by $< X^{u_c}, C = 1, S = 1 >$, and the class label of each instance in U is unlabeled, denoted as $< X^{u_c}, C =?, S = 0 >$. Here, $X_i^{u_c} (1 \leq i \leq n)$ is an uncertain categorical attribute (UCA), whose attribute value is uncertain [1], with $X_{ij}^{u_c}$ denoting the j-th instance of attribute $X_i^{u_c}$. The concept of UCA was introduced in [25]. We use $V(X_i^{u_c})$ to denote the domain from which the attribute $X_i^{u_c}$ may take values. Assuming $V(X_i^{u_c}) = \{x_{i1}, \ldots, x_{im}\}$, as described in [1], an uncertain categorical attribute $X_i^{u_c}$ can be represented by the probability distribution over $V(X_i^{u_c})$, and formalized as probability vector $P_i = \{p_{i1}, \ldots, p_{im}\}$, such that $P(X_i^{u_c} = x_{ik}) = p_{ik} (1 \leq k \leq m)$, which means that $X_i^{u_c}$ takes value of x_{ik} with probability p_{ik}. Here, we have $\sum_{k=1}^{m} p_{ik} = 1$. Certain attribute X_i whose value is x_{i1} can be viewed as a special case of uncertain attribute $X_i^{u_c}$, such that the value of X_i is a single value in $V(X_i^{u_c})$, $V(X_i^{u_c}) = \{x_{i1}\}$.

The PNB algorithm [2] assumes an underlying generative model where a class is selected according to class prior probabilities when solving the PU learning problem in text classification domain. It is assumed that unlabeled examples are generated according to the underlying probabilistic model, which means the prior probability of positive class is estimated on the unlabeled examples instead of the whole training set. Although it is not presented explicitly in [3], the assumption has been applied in the process of extending PNB [2] to work with general discrete attributes. Here, we use a simple model for PU learning: the labeled positive examples are selected completely randomly from all positive examples. This means that positive examples are randomly labeled positive with probability 1-a, and are left unlabeled with probability a. The model was intro-

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

duced in [10], utilized in [11, 12, 13], formalized in [14], and characterized as "selected completely at random" [14]. Under this assumption, any positive example has probability 1-a to be in the set P, and probability a to be in the set U .

The task of classification on uncertain data under PU learning scenario is to construct a classifier from only positive and unlabeled examples with uncertain data to predict unseen instances. In this paper, we only focus on uncertain categorical attributes. Based on the formalization above, we will construct a naive Bayes classifier from only positive and unlabeled examples with uncertain categorical attributes to predict the class labels for unseen instances.

4 Algorithms

4.1 Uncertain Positive Naive Bayes Based on PNB algorithm [3], here we present a PU learning algorithm to handle uncertain categorical attributes. According to the Bayes rule and assuming that the attributes are independent given the class label, for a given certain instance x, the posterior probability is formulated as

(4.1)
$$P(C=c|X=x) \propto P(C=c)\prod_{i=1}^{n} P(X_i=x_{ij}|C=c)$$

For certain data, we can predict the class label for an instance using Eq.(4.1). However, in the case of uncertainty, the value of $X_i^{u_c}$ is not single but the corresponding probability vector $P_i = \{p_{i1},\ldots,p_{im}\}$ over $V(X_i^{u_c}) = \{x_{i1},\ldots,x_{im}\}$. Therefore, we compute the conditional probability $P(X_i^{u_c} = x_{ik}|C=c)$ with probability p_{ik}, with which $X_i^{u_c}$ takes value of x_{ik} for all $k = 1,\ldots,m$, and accumulate them altogether. Thus, for uncertain instance x^{u_c}, the probability of instance x^{u_c} belonging to positive class and negative class can be estimated as

$$P(C=c|X^{u_c}=x^{u_c}) \propto$$
$$P(C=c)\prod_{i=1}^{n}\sum_{k=1}^{|V(X_i^{u_c})|} P(X_i^{u_c}=x_{ik}|C=c)p_{ik} \tag{4.2}$$

In §3, we mentioned that certain attribute can be viewed as a special case of uncertain attribute and in such circumstances Eq.(4.2) is equivalent to Eq.(4.1). Hence, Eq.(4.2) can also be used to predict the class label for certain data.

For Eq.(4.2), the parameters required to estimate are $P(C=1), P(X_i^{u_c}=x_{ij}|C=1), P(X_i^{u_c}=x_{ij}|C=0)$ for all $j=1,\ldots,|V(X_i^{u_c})|$, and $i=1,\ldots,n$. For the sake of brevity, the previous probabilities will be denoted by $p, P(x_{ij}|1), P(x_{ij}|0)$ respectively. Traditional naive Bayes algorithm falls into the supervised learning classification framework where the examples we have are all labeled, hence the parameters can be estimated from training data. While in the case of PU learning, we cannot directly estimate $P(x_{ij}|0)$ and p due to the absence of negative examples. In [3], p is regarded as the proportion of positive examples in the unlabeled examples. In our algorithm, p is viewed as the proportion of positive examples in the whole training set, where the positive examples include labeled positive examples P, and the hidden positive examples in U under the assumption that "selected completely at random".

For certain data, we can estimate $P(x_{ij}|0)$ based on p as

$$P(x_{ij}|0) = \frac{N(x_{ij},U) - P(x_{ij}|1)(|D|p - |P|)}{|D|(1-p)} \tag{4.3}$$

where $N(x_{ij},U)$ denotes the number of unlabeled instances where $X_i = x_{ij}$, $|D|$ denotes the cardinality of the training set and $|P|$ denotes the cardinality of the set of positive instances. However, when the dataset contains uncertain categorical attributes, $X_i^{u_c} = x_{ij}$ for an instance is uncertain and with certain probability. Inspired by the concept of probability cardinality (PC) [1], we can compute the probabilistic cardinality of unlabeled instances over $X_i^{u_c} = x_{ij}$, which is the sum of the probability of each instance in U whose corresponding $X_i^{u_c}$ takes value of x_{ij}. That is, $PC(x_{ij},U) = \sum_{k=1}^{|U|} P(X_{ik}^{u_c} = x_{ij})$. Therefore, in the case of uncertainty, we can estimate $P(x_{ij}|0)$ based on p as

$$P(x_{ij}|0) = \frac{PC(x_{ij},U) - P(x_{ij}|1)(|D|p - |P|)}{|D|(1-p)} \tag{4.4}$$

In order to avoid $P(x_{ij}|0)$ to be negative, we replace all negative estimations by 0, and then normalize all the probabilities following the method in [3], so that for each attribute $X_i^{u_c}$ $(1 \le i \le n)$, we have $\sum_{j=1}^{|V(X_i^{u_c})|} P(x_{ij}|0) = 1$. Then we can estimate $P(x_{ij}|0)$ as

$$P(x_{ij}|0) = \frac{1 + max(0, R_i(x_{ij}))\frac{1}{Z_i}}{|V(X_i^{u_c})| + |D|(1-p)} \tag{4.5}$$

Here, we have $R_i(x_{ij}) = PC(x_{ij},U) - P(x_{ij}|1)(|D|p - |P|)$, and the normalization factor $Z_i = \sum_{j=1}^{|V(X_i^{u_c})|} max(0, P(x_{ij}|0))$, where $P(x_{ij}|0)$ is estimated using Eq.(4.4). However, a problem to be noted here is that there may be a case that $P(x_{ij}|0) < 0$ for all $j = 1,\ldots,|V(X_i^{u_c})|$ when computing Eq.(4.4), which makes $Z_i = 0$ and there will be an exception when computing Eq.(4.5). This is a problem existing

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

in [3]. Here, we replace all negative estimations by 0.1 instead of 0, considering normalization and Laplace correction, $P(x_{ij}|0)$ can be estimated as

$$P(x_{ij}|0) = \frac{1 + max(0.1, R_i(x_{ij}))\frac{1}{Z_i}}{|V(X_i^{u_c})| + |D|(1-p)} \tag{4.6}$$

where $R_i(x_{ij}) = PC(x_{ij}, U) - P(x_{ij}|1)(|D|p - |P|)$ and $Z_i = \sum_{j=1}^{|V(X_i^{u_c})|} max(0.1, P(x_{ij}|0))$.

The parameter $P(x_{ij}|1)$ can be estimated on the labeled positive examples by means of maximum likelihood estimator. Using Laplace correction to avoid computing probability values of zero we can estimate $P(x_{ij}|1)$ as

$$P(x_{ij}|1) = \frac{1 + PC(x_{ij}, P)}{|V(X_i^{u_c})| + |P|} \tag{4.7}$$

where $PC(x_{ij}, P)$ stands for the probabilistic cardinality of positive instances where $X_i^{u_c} = x_{ij}$. Here, $PC(x_{ij}, P) = \sum_{k=1}^{|P|} P(X_{ik}^{u_c} = x_{ij})$.

The parameter p cannot be estimated from the training set and must be provided by the user.

Based on the equations above, we can build a naive Bayes classifier to cope with uncertain categorical attributes from positive and unlabeled examples, which is named as uncertain positive naive Bayes (UPNB). The time complexity of UPNB is $O(|D|nv)$ dominated by estimating $P(x_{ij}|1)$ and $P(x_{ij}|0)$, where n is the number of attributes, and v is the maximum number of values that an attribute may take.

4.2 Accurate Uncertain Positive Naive Bayes

In §4.1, we note that UPNB requires the prior probability of positive class p which is difficult for the user to provide in real-life applications. Here, we select a possible value which makes classifier yield best estimated performance on the validation set as p, so as to free the user from providing p. The parameter p is selected from 0.1 to 0.9, stepping by 0.1. The step can be more dense, but the experimental result shows that this doesn't always help to improve the performance significantly. Due to only positive and unlabeled instances on the validation set, we cannot estimate the performance of classifier directly. Therefore, it is important that how to estimate the performance on the positive and unlabeled instances.

In [11], Lee and Liu proposed a performance measure similar to F1, $precision \cdot recall/P(C = 1)$, which can help us to search a best value as p on the validation set. For uncertain data, $precision = P(C = 1|f(X^{u_c}) = 1)$, $recall = P(f(X^{u_c}) = 1|C = 1)$. As proved in [11], $precision \cdot recall/P(C = 1) = recall \cdot recall/P(f(X^{u_c}) = 1)$ where $recall$ can be estimated using the labeled positive examples in the validation set and $P(f(X^{u_c}) = 1)$ can be estimated using the entire validation set. Therefore, we can estimate the performance without labeled negative examples. The performance measure is similar to F1 in the sense that it is large when both *precision* and *recall* are large and is small if either *precision* or *recall* is small. Following the method in [11], the dataset was randomly split into three sets: the training set containing 50% of the instances, the validation set containing 20% of the instances and the test set containing 30% of the instances. After the best value for p is chosen, the naive Bayes classifier is retrained on the combined training and validation set and tested on the test set. This algorithm is named as accurate uncertain positive naive Bayes (aUPNB), which is illustrated in Algorithm 1.

Algorithm 1 accurate uncertain positive naive Bayes

Input:
Training set: T;
Validation set: V;

Output:
An accurate uncertain positive naive Bayes Classifier, B;

1: **for** i=1 to 9 **do**
2: $\hat{p} = i/10$;
3: $B_i = UPNB(T, \hat{p})$;
4: $f_i = evaluate(B_i, V)$;
5: **end for**;
6: $j = \underset{i}{\operatorname{argmax}}(f_i)$;
7: $p = j/10$;
8: $B = UPNB(T \cup V, p)$;
9: **return** B;

In this algorithm, $\hat{p}$ is an estimation of prior probability of positive class, B_i is the classifier trained on the training set T with current $\hat{p}$ using UPNB. The function $evaluate(B_i, V)$ evaluates the classifier B_i in terms of $recall \cdot recall/P(f(X^{u_c}) = 1)$ on the validation set V and output f_i, $f_i = recall \cdot recall/P(f(X^{u_c}) = 1)$. In step 6 and step 7, the value of $\hat{p}$ which maximizes f_i is assigned to p. Then in step 8, the final classifier B is obtained on the combined training and validation set with p using UPNB. Let's write l for the number of class labels. To obtain p, requires $9[O(|T|nv) + O(|V|nvl)]$ time, dominated by step 1 to 5 including training on T using UPNB and evaluating B_i on V, and building the final classifier B takes $O((|T| + |V|)nv)$ time, dominated by step 8. Here, l is always equal to 2 for PU learning. Hence, the time complexity of aUPNB is $O((|T| + |V|)nv)$.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

5 Experiment

In this section, we report our experimental results. Since our algorithm can only process uncertain categorical attributes, five datasets containing only categorical attributes from UCI repository [24] are used in the experiments. The class distribution of these datasets is appropriate to model PU learning problem. Information about these datasets is shown in Tab.1.

Table 1: Experimental datasets

Dataset	Size	#Attr	#Class	Pos/Neg
car evaluation	1728	6	4	70%/30%
kr-vs-kp	3196	36	2	52.2%/47.8%
mushroom	8124	22	2	51.8%/48.2%
nursery	12960	8	5	32.9%/67.1%
vote	435	16	2	61.4%/38.6%

In Tab.1, Pos/Neg presents the percentage of the number of positive examples against that of negative examples. As shown in Tab.1, *car evaluation* and *nursery* have more than two classes. For *car evaluation*, instances from the '*unacc*' class are regarded as positive and the rest as negative. For *nursery*, instances from the '*priority*' class are regarded as positive and the rest as negative.

We introduce synthetic uncertainty into the datasets for lack of real uncertain datasets adopting the technique in [1]. For an instance x^{u_c}, the values of attribute $X_i^{u_c}$ are converted into probability vector $P_i = \{p_{i1}, \ldots, p_{im}\}$. For instance, when we introduce 10% uncertainty, the attribute $X_i^{u_c}$ will take the original value with 0.9 probability, and 0.1 probability to take any of the other values. Suppose in the original accurate dataset $X_i = x_{i1}$, then we will assign $p_{i1} = 0.9$, and assign $p_{ij}(2 \leq j \leq m)$ to ensure $\sum_{j=2}^{m} p_{ij} = 0.1$.

In this section, we write a for the probability of positive examples being used as unlabeled examples, p for the prior probability of positive class which needs to be provided by the user in UPNB algorithm, and u for the extent of uncertainty in uncertain categorical attribute. We measure the performance on the test set in terms of F1 and Accuracy which are commonly used in PU learning [14, 3]. The experiments are conducted on a PC with Core 2 CPU and 2.0 GB main memory. For each of the experiments settings here, ten trails of experiments are conducted, and the averaged classification performance is reported.

5.1 aUPNB vs. UPNB In this group of experiment, we compare the classification performance between aUPNB and UPNB on certain and uncertain datasets with various parameter settings. We select half of the attributes as uncertain attribute for each dataset.

Firstly, we present the performance of aUPNB and UPNB on certain and uncertain datasets under a=40%. Fig.1 and Fig.2 show the results on certain data (u=0), and the results of uncertain data with u=30% are given in Fig.3 and Fig.4. In Fig.1 to Fig.4, the horizontal axis represents the parameter p provided by the user, and the vertical axis represents F1 or Accuracy.

It can be observed from Fig.1 to Fig.4 that the performance of UPNB is influenced by the value of p provided by the user, e.g., it is sensitive to p on *car evaluation*, *kr-vs-kp* and *nursery*. And it is generally difficult for the user to provide a value of p that can make classification performance excellent. For *mushroom* and *vote*, it is not very sensitive where the value of p provided by the user may make UPNB perform well. However, aUPNB could yield good performance without user-specified parameter p as it is obvious from Fig.1 to Fig.4 that the performance curves of aUPNB are at the top or very close to the top of the performance curve of UPNB in most cases.

As mentioned in §4.1, the performance measure we used to select a value on the validation set is similar to F1. However, we can see from the presented experimental results that Accuracy can also be satisfactory in most cases.

Furthermore, we experimented with a=20%, 40%, 60%, 80%; u=0, 10%, 20%, 30%, 40%, 50%; p ranges from 0.1 to 0.9 stepping by 0.1. Note that it may be too tedious to present all the figures (5 datasets $\times 4$ $a \times$ 6 $u \times$ 2 performance=240 figures). Here, we present the experimental results in the form of tables although not intuitive as figures. Due to space limitation, only the results of averaged F1 under various parameter settings are shown in Tab.2 to Tab.6 where column 1 and 2 gives the value of a and u respectively; columns 3 to 11 give the performance of UPNB under different values of p; the last column lists the performance of aUPNB.

As we can observe from Tab.2 to Tab.6, the performance of UPNB varies with p. However, aUPNB performs well compared with the most cases of UPNB under different settings of p. And even if the value of p provided by the user could achieve the best performance, aUPNB is very close to it, with the decrement within 1% in most cases. Overall, aUPNB does not require the user to estimate the prior probability of positive class and could achieve satisfactory performance on certain and uncertain datasets.

5.2 Experiment on uncertain attributes In this group of experiment, we show the classification performance when different attributes are selected as uncer-

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

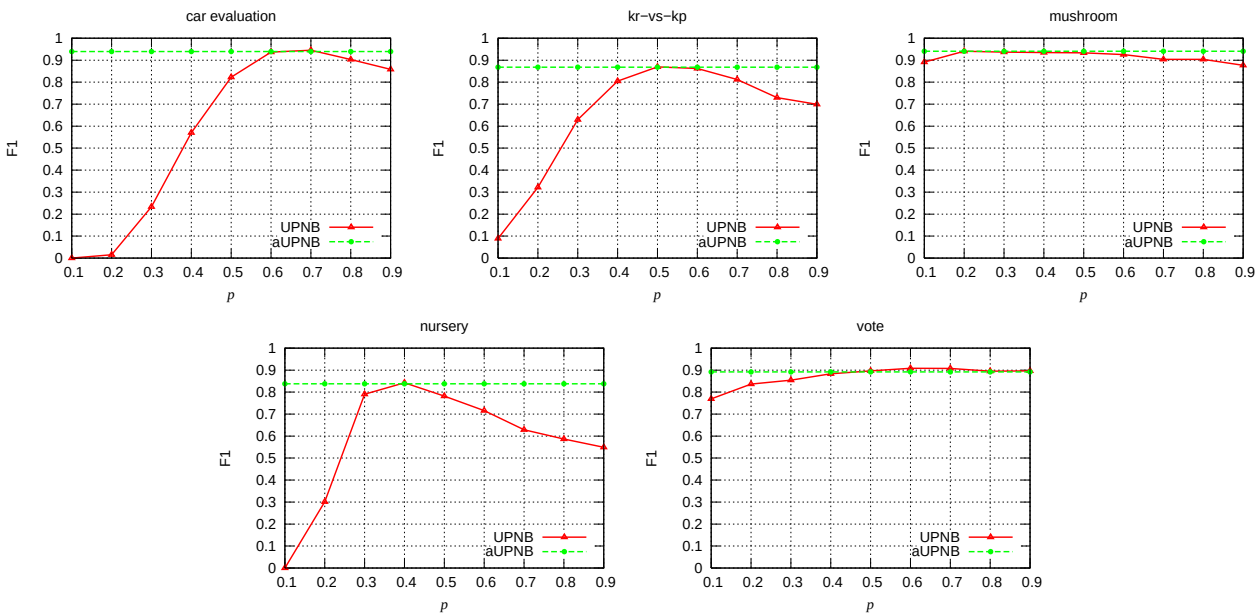

Figure 1: F1 for certainty with a=40%

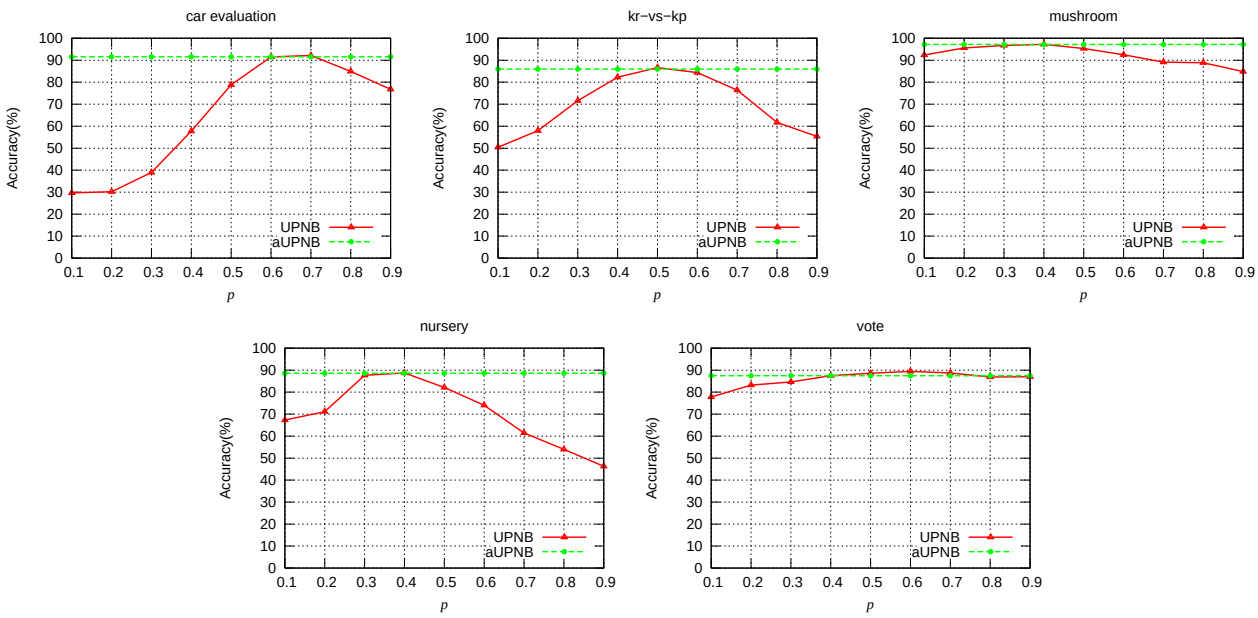

Figure 2: Accuracy for certainty with a=40%

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

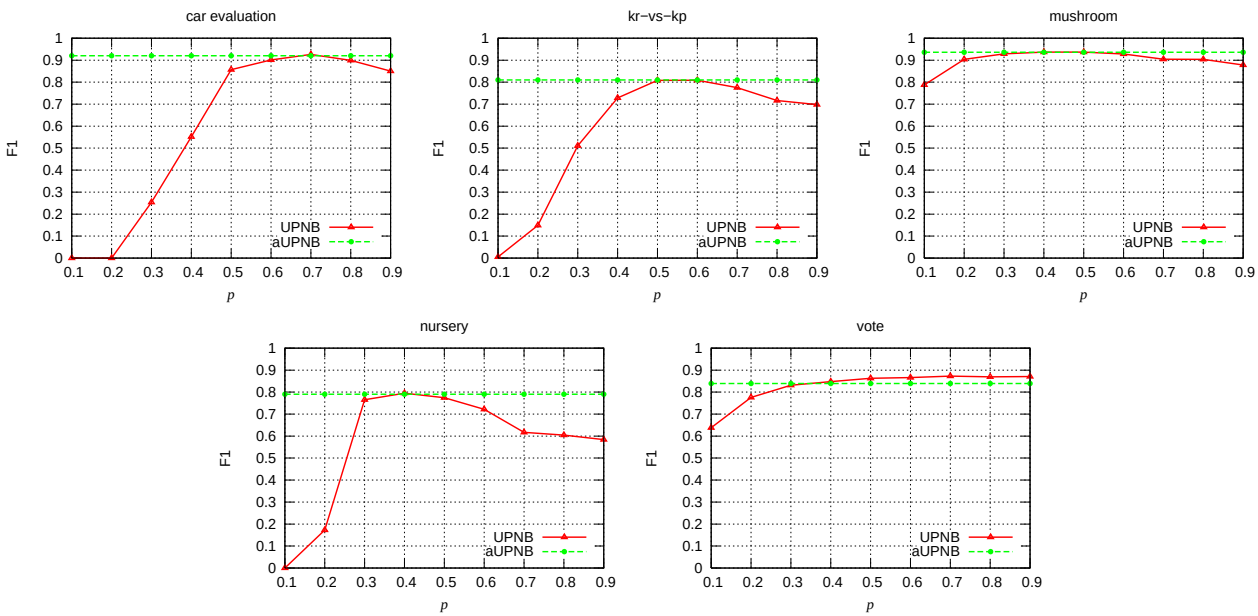

Figure 3: F1 for uncertainty with a=40%, u=30%

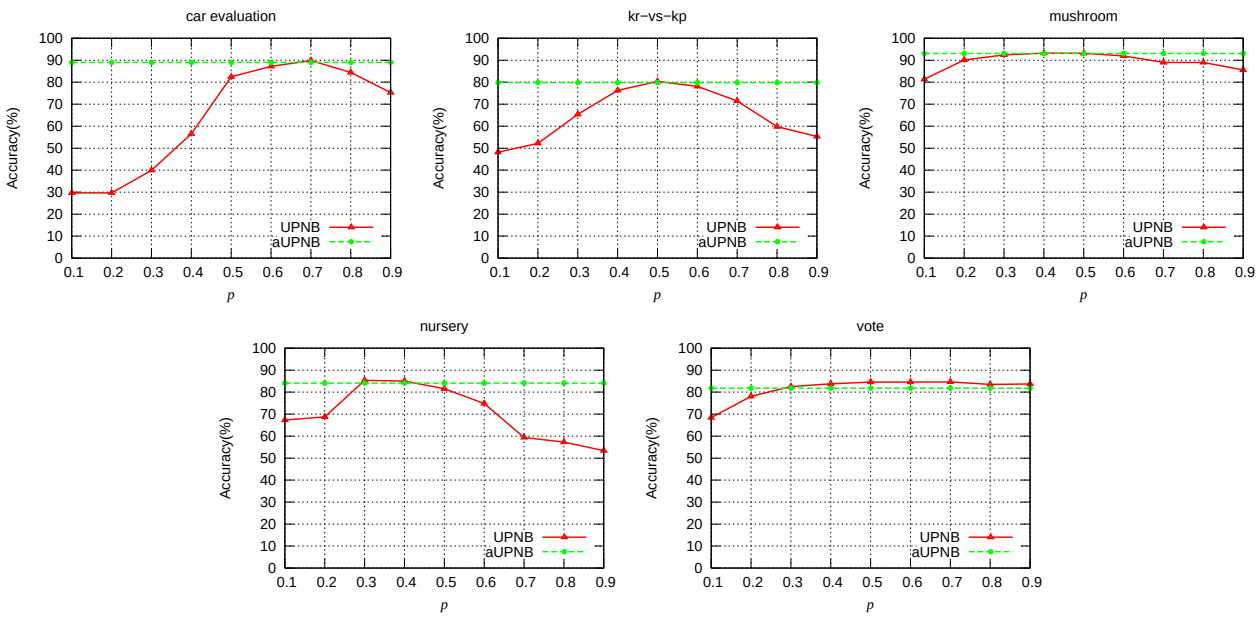

Figure 4: Accuracy for uncertainty with a=40%, u=30%

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Table 2: F1 for UPNB and aUPNB on *car evaluation*

a	u	UPNB									aUPNB
		p=0.1	p=0.2	p=0.3	p=0.4	p=0.5	p=0.6	p=0.7	p=0.8	p=0.9	
20%	0	0.0000	0.0138	0.2495	0.5709	0.8278	0.9375	**0.9574**	0.9044	0.8598	0.9568
	10%	0.0000	0.0044	0.2518	0.5570	0.8484	0.9174	**0.9539**	0.9062	0.8584	0.9513
	20%	0.0000	0.0011	0.2636	0.5389	0.8655	0.8992	**0.9432**	0.9064	0.8553	0.9377
	30%	0.0000	0.0000	0.2692	0.5217	0.8767	0.8883	**0.9242**	0.9039	0.8488	0.9228
	40%	0.0000	0.0000	0.2709	0.5016	0.8805	0.8827	**0.9094**	0.8939	0.8416	0.9025
	50%	0.0000	0.0000	0.2709	0.4917	0.8810	0.8810	**0.8957**	0.8814	0.8355	0.8887
40%	0	0.0000	0.0153	0.2335	0.5700	0.8229	0.9365	**0.9453**	0.9030	0.8585	0.9393
	10%	0.0000	0.0039	0.2262	0.5629	0.8357	0.9285	**0.9475**	0.9045	0.8549	0.9408
	20%	0.0000	0.0006	0.2408	0.5545	0.8512	0.9157	**0.9414**	0.9048	0.8532	0.9372
	30%	0.0000	0.0000	0.2540	0.5519	0.8569	0.9008	**0.9261**	0.8990	0.8505	0.9203
	40%	0.0000	0.0000	0.2587	0.5433	0.8605	0.8942	**0.9129**	0.8917	0.8458	0.9006
	50%	0.0000	0.0000	0.2631	0.5428	0.8617	0.8904	**0.9020**	0.8821	0.8396	0.8910
60%	0	0.0000	0.0252	0.2526	0.5740	0.8258	0.9115	**0.9270**	0.8982	0.8583	0.9193
	10%	0.0000	0.0173	0.2378	0.5667	0.8374	0.9128	**0.9230**	0.8986	0.8586	0.9203
	20%	0.0000	0.0049	0.2342	0.5594	0.8463	0.9051	**0.9185**	0.8995	0.8539	0.9137
	30%	0.0000	0.0000	0.2355	0.5640	0.8551	0.8978	**0.9101**	0.8918	0.8522	0.9053
	40%	0.0000	0.0000	0.2350	0.5634	0.8594	0.8937	**0.8996**	0.8839	0.8487	0.8927
	50%	0.0000	0.0000	0.2376	0.5674	0.8597	0.8888	**0.8908**	0.8786	0.8462	0.8832
80%	0	0.0000	0.0340	0.2632	0.5840	0.8267	0.9085	**0.9190**	0.8905	0.8538	0.9017
	10%	0.0000	0.0151	0.2592	0.5825	0.8320	0.9035	**0.9158**	0.8908	0.8538	0.8976
	20%	0.0000	0.0031	0.2503	0.5765	0.8406	0.8885	**0.9124**	0.8861	0.8513	0.8823
	30%	0.0000	0.0016	0.2550	0.5762	0.8498	0.8814	**0.8976**	0.8786	0.8505	0.8845
	40%	0.0000	0.0000	0.2561	0.5806	0.8549	0.8785	**0.8871**	0.8685	0.8477	0.8783
	50%	0.0000	0.0000	0.2594	0.5814	0.8587	0.8761	**0.8775**	0.8571	0.8421	0.8681

Table 3: F1 for UPNB and aUPNB on *kr-vs-kp*

a	u	UPNB									aUPNB
		p=0.1	p=0.2	p=0.3	p=0.4	p=0.5	p=0.6	p=0.7	p=0.8	p=0.9	
20%	0	0.0845	0.3288	0.6406	0.8174	**0.8788**	0.8622	0.8127	0.7301	0.7002	0.8751
	10%	0.0223	0.2308	0.5643	0.7864	**0.8516**	0.8426	0.7976	0.7216	0.6993	0.8444
	20%	0.0055	0.1692	0.5307	0.7581	**0.8335**	0.8310	0.7876	0.7176	0.6991	0.8315
	30%	0.0039	0.1397	0.5118	0.7433	**0.8179**	0.8125	0.7784	0.7142	0.6989	0.8147
	40%	0.0028	0.1208	0.5060	0.7378	**0.8087**	0.7992	0.7703	0.7121	0.6988	0.8064
	50%	0.0028	0.1165	0.5049	0.7401	**0.8065**	0.7952	0.7670	0.7117	0.6985	0.8052
40%	0	0.0892	0.3214	0.6298	0.8051	**0.8692**	0.8619	0.8122	0.7300	0.6996	0.8680
	10%	0.0289	0.2311	0.5457	0.7622	0.8432	0.8459	0.7981	0.7220	0.6982	**0.8460**
	20%	0.0113	0.1789	0.5207	0.7348	0.8259	0.8273	0.7884	0.7191	0.6983	**0.8298**
	30%	0.0053	0.1487	0.5107	0.7287	0.8084	0.8089	0.7752	0.7163	0.6984	**0.8102**
	40%	0.0053	0.1353	0.5059	0.7292	0.7967	**0.7983**	0.7630	0.7131	0.6981	0.7981
	50%	0.0053	0.1294	0.5041	0.7303	**0.7943**	0.7925	0.7584	0.7120	0.6975	0.7937
60%	0	0.0859	0.3365	0.6215	0.8038	**0.8574**	0.8518	0.8162	0.7324	0.7024	0.8542
	10%	0.0234	0.2315	0.5397	0.7512	0.8341	0.8445	0.8086	0.7252	0.7024	**0.8459**
	20%	0.0087	0.1807	0.5017	0.7338	0.8189	**0.8253**	0.7980	0.7220	0.7028	0.8240
	30%	0.0067	0.1513	0.4930	0.7225	**0.8139**	0.8087	0.7842	0.7174	0.7023	0.8123
	40%	0.0067	0.1338	0.4865	0.7227	**0.8054**	0.7975	0.7737	0.7146	0.7021	0.8007
	50%	0.0063	0.1266	0.4916	0.7239	**0.8015**	0.7934	0.7690	0.7138	0.7019	0.7997
80%	0	0.0697	0.3238	0.6299	0.7896	**0.8513**	0.8373	0.8053	0.7360	0.6986	0.8480
	10%	0.0147	0.2057	0.5474	0.7469	0.8228	**0.8293**	0.7991	0.7291	0.6994	0.8287
	20%	0.0011	0.1508	0.5182	0.7223	0.8040	**0.8145**	0.7875	0.7254	0.6995	0.8019
	30%	0.0011	0.1251	0.4941	0.7125	0.7925	**0.7980**	0.7735	0.7201	0.6982	0.7857
	40%	0.0011	0.1015	0.4837	0.7136	0.7860	**0.7903**	0.7648	0.7171	0.6975	0.7803
	50%	0.0011	0.0957	0.4890	0.7084	0.7838	**0.7878**	0.7574	0.7147	0.6971	0.7770

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Table 4: F1 for UPNB and aUPNB on *mushroom*

a	u	UPNB									aUPNB
		p=0.1	p=0.2	p=0.3	p=0.4	p=0.5	p=0.6	p=0.7	p=0.8	p=0.9	
20%	0	0.9227	0.9562	0.9692	**0.9767**	0.9616	0.9351	0.9062	0.9045	0.8806	0.9751
	10%	0.9002	**0.9407**	0.9369	0.9366	0.9379	0.9258	0.9045	0.9045	0.8872	0.9386
	20%	0.8503	0.9236	0.9330	0.9375	**0.9396**	0.9273	0.9045	0.9045	0.8900	0.9366
	30%	0.7932	0.9084	0.9303	0.9376	**0.9410**	0.9284	0.9046	0.9045	0.8922	0.9400
	40%	0.7219	0.8905	0.9245	0.9361	**0.9412**	0.9287	0.9047	0.9045	0.8908	0.9405
	50%	0.6523	0.8657	0.9163	0.9331	**0.9399**	0.9282	0.9046	0.9045	0.8877	0.9383
40%	0	0.9201	0.9555	0.9677	**0.9728**	0.9561	0.9322	0.9049	0.9031	0.8722	0.9721
	10%	0.8914	**0.9410**	0.9365	0.9346	0.9334	0.9257	0.9039	0.9039	0.8769	0.9404
	20%	0.8447	0.9219	0.9318	**0.9364**	0.9352	0.9268	0.9039	0.9039	0.8782	0.9344
	30%	0.7877	0.9036	0.9288	**0.9370**	0.9369	0.9280	0.9041	0.9039	0.8778	0.9360
	40%	0.7175	0.8851	0.9209	0.9354	**0.9370**	0.9287	0.9041	0.9039	0.8743	0.9371
	50%	0.6461	0.8592	0.9114	0.9314	**0.9365**	0.9281	0.9041	0.9039	0.8672	0.9355
60%	0	0.9150	0.9552	0.9676	**0.9715**	0.9524	0.9273	0.9045	0.9027	0.8687	0.9686
	10%	0.8844	**0.9389**	0.9351	0.9341	0.9327	0.9234	0.9042	0.9042	0.8755	0.9351
	20%	0.8336	0.9200	0.9304	**0.9355**	0.9341	0.9242	0.9042	0.9042	0.8751	0.9320
	30%	0.7759	0.9009	0.9259	**0.9357**	**0.9357**	0.9253	0.9044	0.9042	0.8737	0.9348
	40%	0.7054	0.8799	0.9194	0.9336	**0.9355**	0.9260	0.9045	0.9042	0.8694	0.9329
	50%	0.6293	0.8563	0.9103	0.9296	**0.9342**	0.9261	0.9044	0.9040	0.8613	0.9312
80%	0	0.9150	0.9552	0.9676	**0.9715**	0.9524	0.9273	0.9045	0.9027	0.8687	0.9686
	10%	0.8844	**0.9389**	0.9351	0.9341	0.9327	0.9234	0.9042	0.9042	0.8755	0.9351
	20%	0.8336	0.9200	0.9304	**0.9355**	0.9341	0.9242	0.9042	0.9042	0.8751	0.9320
	30%	0.7759	0.9009	0.9259	**0.9357**	**0.9357**	0.9253	0.9044	0.9042	0.8737	0.9348
	40%	0.7054	0.8799	0.9194	0.9336	**0.9355**	0.9260	0.9045	0.9042	0.8694	0.9329
	50%	0.6293	0.8563	0.9103	0.9296	**0.9342**	0.9261	0.9044	0.9040	0.8613	0.9312

Table 5: F1 for UPNB and aUPNB on *nursery*

a	u	UPNB									aUPNB
		p=0.1	p=0.2	p=0.3	p=0.4	p=0.5	p=0.6	p=0.7	p=0.8	p=0.9	
20%	0	0.0000	0.3016	0.7925	**0.8433**	0.7876	0.7229	0.6385	0.5970	0.5548	**0.8433**
	10%	0.0000	0.2468	0.7976	**0.8204**	0.7840	0.7255	0.6364	0.6135	0.5773	0.8140
	20%	0.0000	0.2038	0.7868	**0.8077**	0.7806	0.7281	0.6360	0.6147	0.5858	0.7973
	30%	0.0000	0.1744	0.7692	**0.8002**	0.7793	0.7293	0.6382	0.6176	0.5925	0.7942
	40%	0.0000	0.1720	0.7530	**0.7903**	0.7780	0.7285	0.6413	0.6224	0.5978	0.7850
	50%	0.0000	0.1734	0.7418	**0.7799**	0.7752	0.7288	0.6424	0.6249	0.6003	0.7771
40%	0	0.0000	0.3010	0.7901	**0.8426**	0.7817	0.7158	0.6289	0.5870	0.5489	0.8377
	10%	0.0000	0.2481	0.7953	**0.8202**	0.7782	0.7187	0.6241	0.6047	0.5712	0.8148
	20%	0.0000	0.2017	0.7847	**0.8057**	0.7760	0.7209	0.6196	0.6044	0.5785	0.8029
	30%	0.0000	0.1725	0.7655	**0.7947**	0.7747	0.7215	0.6170	0.6048	0.5839	0.7903
	40%	0.0000	0.1640	0.7485	**0.7855**	0.7726	0.7216	0.6173	0.6070	0.5882	0.7818
	50%	0.0000	0.1683	0.7359	**0.7764**	0.7708	0.7218	0.6180	0.6084	0.5903	0.7733
60%	0	0.0000	0.3089	0.7830	**0.8416**	0.7800	0.7123	0.6232	0.5800	0.5475	**0.8416**
	10%	0.0000	0.2534	0.7898	**0.8223**	0.7769	0.7141	0.6154	0.5985	0.5685	0.8111
	20%	0.0000	0.2035	0.7806	**0.8083**	0.7737	0.7158	0.6085	0.5971	0.5734	0.7983
	30%	0.0000	0.1702	0.7669	**0.7985**	0.7720	0.7161	0.6027	0.5943	0.5754	0.7884
	40%	0.0000	0.1633	0.7477	**0.7884**	0.7715	0.7158	0.5984	0.5917	0.5779	0.7786
	50%	0.0000	0.1563	0.7344	**0.7794**	0.7712	0.7151	0.5959	0.5907	0.5809	0.7724
80%	0	0.0000	0.3003	0.7759	**0.8371**	0.7816	0.7074	0.6160	0.5756	0.5407	**0.8371**
	10%	0.0000	0.2470	0.7791	**0.8220**	0.7798	0.7073	0.6123	0.5967	0.5626	0.8187
	20%	0.0000	0.2066	0.7686	**0.8076**	0.7775	0.7082	0.6066	0.5945	0.5666	0.8033
	30%	0.0000	0.1893	0.7518	**0.7947**	0.7743	0.7081	0.6032	0.5917	0.5681	0.7881
	40%	0.0000	0.1845	0.7327	**0.7837**	0.7693	0.7082	0.6022	0.5897	0.5683	0.7782
	50%	0.0000	0.1882	0.7220	**0.7768**	0.7667	0.7077	0.6020	0.5891	0.5678	0.7724

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Table 6: F1 for UPNB and aUPNB on *vote*

a	u	UPNB									aUPNB
		p=0.1	p=0.2	p=0.3	p=0.4	p=0.5	p=0.6	p=0.7	p=0.8	p=0.9	
20%	0	0.7668	0.8376	0.8630	0.8880	0.9012	0.9099	**0.9126**	0.9052	0.9035	0.8973
	10%	0.7440	0.8211	0.8616	0.8801	0.8903	0.8950	**0.8963**	0.8915	0.8898	0.8857
	20%	0.6849	0.8032	0.8473	0.8747	0.8797	0.8827	**0.8905**	0.8846	0.8824	0.8786
	30%	0.6019	0.7719	0.8361	0.8567	0.8695	0.8712	**0.8851**	0.8780	0.8773	0.8739
	40%	0.5135	0.7429	0.8286	0.8453	0.8559	0.8650	**0.8837**	0.8693	0.8631	0.8625
	50%	0.4920	0.7293	0.8235	0.8420	0.8506	0.8631	**0.8826**	0.8654	0.8479	0.8596
40%	0	0.7695	0.8367	0.8543	0.8836	0.8967	**0.9087**	0.9078	0.8953	0.8967	0.8919
	10%	0.7478	0.8235	0.8542	0.8796	0.8842	0.8888	**0.8924**	0.8831	0.8817	0.8646
	20%	0.7095	0.8113	0.8378	0.8664	0.8719	0.8791	**0.8811**	0.8800	0.8766	0.8490
	30%	0.6384	0.7756	0.8311	0.8479	0.8626	0.8661	**0.8725**	0.8692	0.8711	0.8392
	40%	0.5810	0.7490	0.8204	0.8330	0.8498	0.8589	**0.8680**	0.8578	0.8627	0.8239
	50%	0.5542	0.7373	0.8181	0.8326	0.8447	0.8589	**0.8674**	0.8568	0.8562	0.8394
60%	0	0.7552	0.8306	0.8543	0.8860	0.8991	0.9050	**0.9060**	0.8974	0.8952	0.8952
	10%	0.7297	0.8138	0.8499	0.8785	**0.8880**	0.8877	0.8855	0.8821	0.8805	0.8602
	20%	0.6695	0.7942	0.8320	0.8693	0.8754	**0.8810**	0.8809	0.8757	0.8748	0.8635
	30%	0.6128	0.7582	0.8188	0.8452	0.8656	0.8741	**0.8782**	0.8709	0.8703	0.8518
	40%	0.5526	0.7278	0.8089	0.8375	0.8547	0.8701	**0.8752**	0.8617	0.8679	0.8482
	50%	0.5105	0.7177	0.8055	0.8326	0.8506	0.8657	**0.8735**	0.8586	0.8525	0.8464
80%	0	0.7441	0.8028	0.8414	0.8715	0.8853	0.8853	**0.8991**	0.8956	0.8911	0.8624
	10%	0.6990	0.8035	0.8318	0.8644	**0.8791**	0.8749	0.8667	0.8741	0.8764	0.8311
	20%	0.6280	0.7909	0.8277	0.8499	0.8610	0.8689	0.8666	0.8637	**0.8727**	0.8401
	30%	0.5223	0.7558	0.8119	0.8288	0.8443	**0.8648**	0.8615	0.8585	0.8616	0.8414
	40%	0.4389	0.7166	0.8007	0.8349	0.8368	**0.8650**	0.8583	0.8455	0.8520	0.8250
	50%	0.4052	0.6951	0.7959	0.8291	0.8321	**0.8602**	0.8553	0.8457	0.8459	0.8163

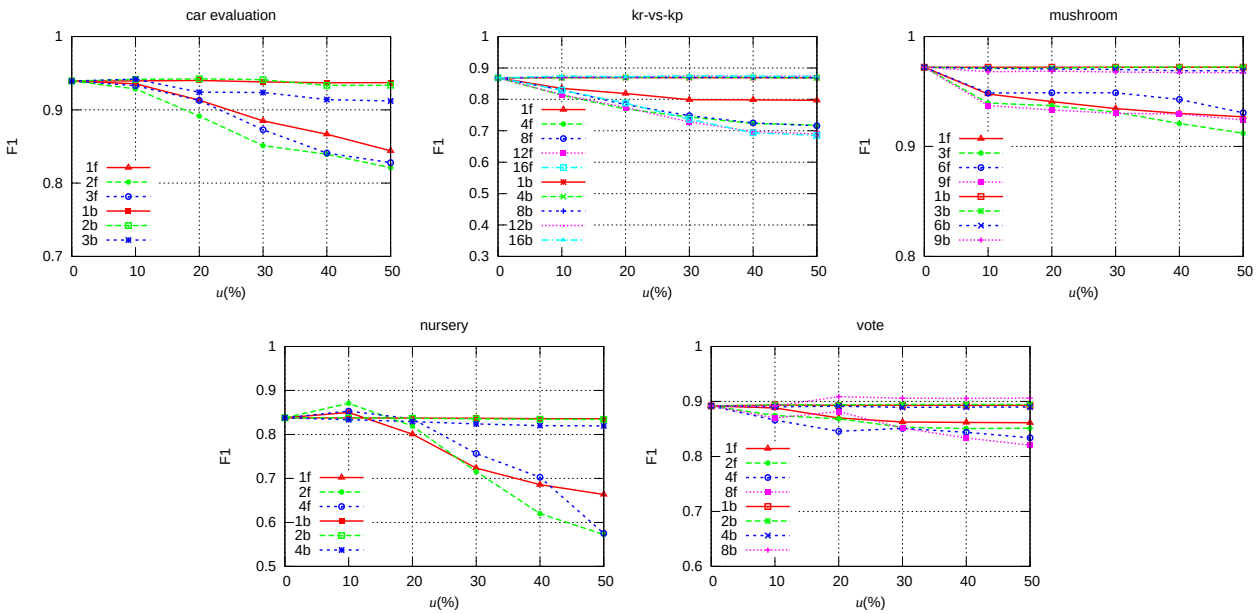

Figure 5: F1 for different uncertain attributes

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

tain attributes. We experimented with a=40%; u=0, 10%, 20%, 30%, 40%, 50%. The attributes are sorted in descending order according to their information gain index. The top h representative/unrepresentative attributes are selected as uncertain attributes to introduce synthetic uncertainty. As the averaged Accuracy has similar trend with F1, for lacking of space, we only give the results of averaged F1 in Fig.5. The horizontal axis represents the degree of uncertainty and the vertical axis represents averaged F1. In Fig.5, *1f* denotes the experiment with the top 1 representative attribute used as uncertain attribute; and *1b* for the top 1 unrepresentative one. The same for the others.

It can be observed from Fig.5 that when some of the most unrepresentative attributes are used as uncertain attribute, F1 remains relatively stable compared with certain data when uncertainty ranges from 10% to 50%. However, when we introduce synthetic uncertainty for some of the most representative attributes, F1 will decline as uncertainty increases. The decrement differs from dataset to dataset, e.g. sharply on *car evaluation*, *kr-vs-kp* and *nursery*, slowly on *mushroom* and *vote*. Overall, different uncertain attributes in the dataset have different impact on classification performance. The more representative the uncertain attributes, the more significant the impact on the classification performance and vice versa.

5.3 Performance of aUPNB on uncertain data

In this group of experiment, we present the performance of aUPNB on different extent of uncertain data. We experimented with a=40%; u=0, 10%, 20%, 30%, 40%, 50%. As demonstrated in §5.2, different uncertain attributes have different impact on classification performance. Here, we select attributes to introduce synthetic uncertainty according to the following method. For each dataset, we sort the attributes in descending order according to their information gain index and divided them into two approximately equal parts. Middle attributes of each part will be chosen to introduce synthetic uncertainty. For instance, *car evaluation* has six attributes. After sorted, the attributes in descending order are *safety, persons, buying, maint, lug_boot*, and *doors*. We split them into two equal parts, *safety, persons, buying* and *maint, lug_boot, doors*. When two attributes are chosen to be introduced uncertainty, *persons* and *lug_boot* will be selected. Due to space limitation and averaged accuracy has approximate tendency, we only give the averaged F1 on different datasets. The experimental results are shown in Fig.6, where k denotes the number of uncertain attributes.

As shown in Fig.6, F1 declines slowly when uncertainty ranges from 0 to 40% with different number of uncertain attributes. The overall decrease in F1 is within 10% compared with the one over certain data even when the uncertainty reaches 40%, which demonstrates aUPNB is quite robust against data uncertainty.

For each dataset, F1 for various number of uncertain attributes are presented. It can be observed that F1 is not always decline as the number of uncertain attributes increases, e.g., the curve k=12 on *kr-vs-kp* and the curve k=12 on *vote*. Also, when all the attributes in the dataset are uncertain, F1 is not always the worst, e.g., the curve k=6 on *car evaluation* and the curve k=8 on *nursery*. This can be explained by the conclusion in §5.2 that different uncertain attributes in the dataset have different impact on classification performance.

6 Conclusion and Future work

The main contribution of this paper can be summarized as that we firstly tackled the problem of classification on uncertain data under PU learning scenario. Adopting the concept of probabilistic cardinality, we extend PNB, a PU learning algorithm which can only handle certain attributes, to the case of uncertain categorical attributes. Furthermore, since it is difficult for the users to provide the prior probability of positive class in real-life problems, in this paper, it is chosen on the validation set using a performance measure that can be estimated from positive and unlabeled examples. The experiments demonstrate that our proposed algorithm performs well without user-specified the prior probability of positive class and has satisfactory performance even when data is highly uncertain.

At present, we can only cope with classification on uncertain categorical attributes under PU learning scenario. We plan to study PU learning algorithm to deal with uncertain numeric attributes. On the other hand, we would like to estimate the prior probability of positive class directly instead of selecting on the validation set.

References

[1] B. Qin, Y. Xia, S. Prabhakar and Y. Tu, *A Rule-based Classification Algorithm for Uncertain Data*, ICDE, pp. 1633–1640, 2009.

[2] F. Denis, F. Denis, R. Gilleron, and M. Tommasi, *Text classification from positive and unlabeled examples*, IPMU, pp. 1927–1934, 2002.

[3] B. Calvo, P. Larranaga, and J. A. Lozano, *Learning Bayesian classifiers from positive and unlabeled examples*, Pattern Recognition Letters, 28(16), pp. 2375–2384, 2007.

[4] C. C. Aggarwal and P. S. Yu, *A Survey of Uncertain Data Algorithms and Applications*, IEEE TKDE, 21(5), pp. 609–623, 2009.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

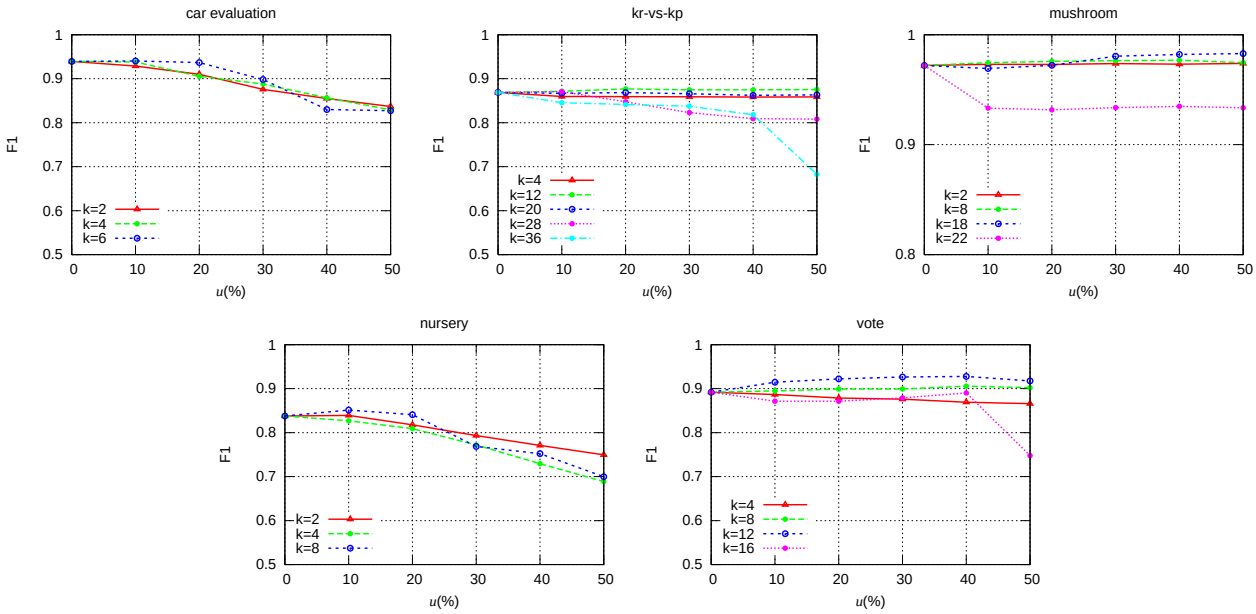

Figure 6: F1 for aUPNB on uncertain data

[5] B. Qin, Y. Xia and F. Li, *DTU: A Decision Tree for Uncertain Data*, PAKDD, pp. 4–15, 2009.

[6] S. Tsang, B. Kao, K. Y. Yip, W. S. Ho and S. D. Lee, *Decision Trees for Uncertain Data*, ICDE, 99(1), 2009.

[7] J. Bi and T. Zhang, *Support Vector Machines with Input Data Uncertainty*, NIPS, 2004.

[8] B. Scholkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola and R. C. Williamson, *Estimating the Support of a High-Dimensional Distribution*, Neural Computation, 13(7), pp. 1443–1471, 2001.

[9] B. Z. Zhang and W. L. Zuo, *Learning from positive and unlabeled examples: A survey*, ISIP, pp. 650–654, 2008.

[10] F. Denis, *PAC learning from positive statistical queries*, ALT, pp. 112–126, 1998.

[11] W. S. Lee and B. Liu, *Learning with positive and unlabeled examples using weighted logistic regression*, ICML, pp. 448–455, 2003.

[12] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu, *Building text classifiers using positive and unlabeled examples*, ICDM, pp. 179–188, 2003.

[13] D. Zhang and W. S. Lee, *A simple probabilistic approach to learning from positive and unlabeled examples*, UKCI, pp. 83–87, 2005.

[14] C. Elkan and K. Noto, *Learning classifiers from only positive and unlabeled data*, SIGKDD, pp. 213–220, 2008.

[15] W. K. Ngai, B. Kao, C. K. Chui, R. Cheng, M. Chau, and K. Y. Yip, *Efficient clustering of uncertain data*, ICDM, pp. 436–445, 2006.

[16] S. D. Lee, B. Kao, and R. Cheng, *Reducing UK-means to K-means*, ICDMW, pp. 483–488, 2007.

[17] G. Cormode and A. McGregor, *Approximation Algorithms for Clustering Uncertain Data*, PODS, pp. 191–200, 2008.

[18] C. K. Chui, B. Kao, and E. Hung, *Mining frequent itemsets from uncertain data*, PAKDD, pp. 47–58, 2007.

[19] C. K. Chui and B. Kao, *A Decremental Approach for Mining Frequent Itemsets from Uncertain Data*, PAKDD, pp. 64–75, 2008.

[20] Q. Zhang, F. Li, and K. Yi, *Finding Frequent Items in Probabilistic Data*, ACM SIGMOD, pp. 819–832, 2008.

[21] C. C. Aggarwal, Y. Li, J. Y Wang and J. Wang, *Frequent Pattern Mining with Uncertain Data*, SIGKDD, pp. 29–38, 2009.

[22] T. Bernecker, H. P. Kriegel, M. Renz, F. Verhein and A. Zuefle, *Probabilistic Frequent Itemset Mining in Uncertain Databases*, SIGKDD, pp. 119–128, 2009.

[23] C. C. Aggarwal and P. S. Yu, *Outlier Detection with Uncertain Data*, SIAM SDM, pp. 483–493, 2008.

[24] C. L. Blake and C. J. Merz, *UCI repository of machine learning databases*, Dept. of Information and Computer Sciences, University of California, Irvine, 1998, http://www.ics.uci.edu/*sim*mlearn/MLRepository.html.

[25] S. Singh, C. Mayfield, S. Prabhakar, R. Shah, and S. Hambrusch, *Indexing categorical data with uncertainty*, ICDE, pp. 616–625, 2007.

[26] X. L. Li, P. S. Yu, B. Liu and S. K. Ng, *Positive Unlabeled Learning for Data Stream Classification*, SIAM SDM, pp. 257–268, 2009.

[27] J. T. Ren, S. D. Lee, X. L. Chen, B. Kao, R. Cheng and D. Cheung, *Naive Bayes Classification of Uncertain Data*, ICDM, 2009.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.