2009 Ninth IEEE International Conference on Data Mining

Naive Bayes Classification of Uncertain Data

Jiangtao Ren*, Sau Dan Lee†, Xianlu Chen*, Ben Kao†, Reynold Cheng† and David Cheung†
*Department of Computer Science, Sun Yat-sen University, Guangzhou, 510275, China
Email: issrjt@mail.sysu.edu.cn, zsuchxl@163.com
†Department of Computer Science, The University of Hong Kong, Hong Kong
Email: {sdlee,kao,ckcheng,dcheung}@cs.hku.hk

***Abstract*—Traditional machine learning algorithms assume that data are exact or precise. However, this assumption may not hold in some situations because of data uncertainty arising from measurement errors, data staleness, and repeated measurements, etc. With uncertainty, the value of each data item is represented by a probability distribution function (pdf). In this paper, we propose a novel naive Bayes classification algorithm for uncertain data with a pdf. Our key solution is to extend the class conditional probability estimation in the Bayes model to handle pdf's. Extensive experiments on UCI datasets show that the accuracy of naive Bayes model can be improved by taking into account the uncertainty information.**

***Keywords*-Uncertain data mining; naive Bayes model**

I. Introduction

Traditional machine learning algorithms often assume that the data values are exact or precise. In many emerging applications, however, the data is inherently uncertain. Sampling errors and instrument errors are both sources of uncertainty, and data are typically represented by probability distributions rather than by deterministic values. There are many learning algorithms used in the classification of deterministic data points, but few algorithms have been proposed for classification of distribution-based uncertain data objects.

Data uncertainty arises naturally in many applications due to various reasons. For example, data obtained from measurements by physical devices are often imprecise due to measurement errors. Another source of error is quantization errors introduced by the digitization process. In some applications, such as sensor networks, data values are continuously changing and recorded information is always stale. Uncertainty may also come from repeated measurements.

In this paper we study the problem of classifying objects with multi-dimensional uncertainty. In particular, an object is not a simple point in space, but is represented by an uncertainty region over which a pdf is defined. Formally, we consider a set of n objects in a d-dimensional space. The location of each object is represented by a pdf p that specifies the probability density of each possible location. We assume that the pdf of each tuple is independent of the others.

This research is supported by National Natural Science Foundation of China under Grant No. 60703110 and Hong Kong Research Grants Council GRF Grants HKU 713406 and HKU 513806.

Naive Bayes is a widely used classification method based on Bayes theory. Based on class conditional density estimation and class prior probability, the posterior class probability of a test data point can be derived and the test data will be assigned to the class with the maximum posterior class probability.

The key problem in naive Bayes method is the class conditional density estimation. Traditionally the class conditional density is estimated based on data points. For uncertain classification problems, however, we should learn the class conditional density from uncertain data objects represented by probability distributions. In order to extend the naive Bayes method to handle uncertain data, we propose three methods in this paper:

Averaging (AVG): We first obtain the average point of every uncertain data object. Then, these points are passed to naive Bayes.

Sample-based method (SBC): The kernel function, which is the key function used in naive Bayes, is redesigned to consider values sampled from the uncertain data as input. In this method, the probability distributions can be arbitrary.

Formula-based method (FBC): This is a special application of the sample-based method, where a closed-formula for the kernel function is derived. We have derived the formula for Gaussian distribution. (Lacking space, we omit the results for uniform distribution.)

As shown by the extensive experimental results on several widely-used benchmark datasets, all our newly-designed classifiers yield more accurate results than the naive Bayes method that does not consider uncertainty. While AVG is the simplest method among our proposals, it is not as good as SBC and FBC. FBC is more accurate than SBC, and can be performed in an efficient manner.

In the rest of this paper, we first give some related works in Section II. Then we introduce preliminaries in Section III and define the problem formally in Section IV. Our algorithmic framework is presented in Section V. The formula-based and sample-based approaches are described in Sections VI and VII respectively. Experimental studies on accuracy and performance are presented in Section VIII. The paper is discussed and concluded in Sections IX and X.

1550-4786/09 $26.00 © 2009 IEEE
DOI 10.1109/ICDM.2009.90

IEEE computer society

II. Related works

In this section, we will introduce some related works about uncertain data mining, uncertain data classification, naive Bayes model and kernel density estimation.

A. Uncertain data mining

There has been a growing interest in uncertain data mining [1], including clustering [2]–[5], classification [6]–[8], outlier detection [9], frequent pattern mining [10], [11], streams mining [12] and skyline analysis [13] on uncertain data, etc.

An important branch of mining uncertain data is to build classification models on uncertain data. While [6], [7] study the classification of uncertain data using the support vector model, [8] performs classification using decision trees. This paper unprecedentedly explores yet another classification model, naive Bayes classifiers, and extends them to handle uncertain data.

B. Naive Bayes classifiers

In probability theory, Bayes theorem relates the conditional and marginal probabilities of two random events. It is often used to compute posterior probabilities given observations. Let $x = (x^1, x^2, \ldots, x^d)$ be a d-dimensional instance which has no class label, and our goal is to build a classifier to predict its unknown class label based on Bayes theorem. Let $C = \{C_1, C_2, \ldots, C_K\}$ be the set of the class labels. $P(C_k)$ is the prior probability of C_k $(k = 1, 2, ..., K)$ that are inferred before new evidence; $P(x|C_k)$ be the conditional probability of seeing the evidence x if the hypothesis C_k is true. A technique for constructing such classifiers to employ Bayes' theorem to obtain:

$$P(C_k|x) = \frac{P(x|C_k)P(C_k)}{\sum_{k'} P(x|C_{k'})P(C_{k'})} \tag{1}$$

A naive Bayes classifier assumes that the value of a particular feature of a class is unrelated to the value of any other feature, so that[1]:

$$P(x|C_k) = \prod_{j=1}^{d} P(x^j|C_k) \tag{2}$$

C. Class conditional density estimation

Probability density estimation constitutes an unsupervised method that attempts to model the underlying density function from which a given set of unlabeled data have been generated. In this paper, we take the non-parametric approach to solve classification problems.

[1] In this paper, we use the superscript "j" on multi-dimensional quantities to represent their values in the j-th dimension.

III. Preliminaries

A. Kernel density estimation

Kernel density estimation is a non-parametric way of estimating the probability density function of a random variable. As an illustration, given a sample of a population, kernel density estimation makes it possible to extrapolate the sample to the entire population.

If $x_1, x_2, \ldots, x_N \sim f$ are independent and identically-distributed samples of a scalar random variable, then the kernel density approximation of its probability density function is:

$$\hat{f}_h(x) = \frac{1}{Nh} \sum_{n=1}^{N} K\left(\frac{x - x_n}{h}\right) \tag{3}$$

where K is some kernel and h is a smoothing parameter called the bandwidth. A typical choice of K is the standard Gaussian function with zero mean and unit variance, i.e.,

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \tag{4}$$

B. Naive Bayes classification based on kernel density estimation

Let $x_{n_k} = (x^1_{n_k}, x^2_{n_k}, \ldots, x^d_{n_k}), n_k = 1, 2, \ldots, N_k$ represent training data points of class C_k, and N_k is the number of instances in class C_k. To classify $x = (x^1, x^2, \ldots, x^d)$ using naive Bayes model with (2), we need to estimate the class condition density $P(x^j|C_k)$. We use $\hat{f}_{h^j_k}(x^j)$ as the estimation. From (3) and (2), we get:

$$P(x|C_k) = \prod_{j=1}^{d} \left\{ \frac{1}{N_k h^j_k} \sum_{n_k=1}^{N_k} K\left(\frac{x^j - x^j_{n_k}}{h^j_k}\right) \right\} \tag{5}$$

With this, we can compute $P(C_k|x)$ using (1) and predict the label of x as $y = \arg\max_{C_k \in C} P(C_k|x)$.

IV. Problem definition

Suppose that $D = \{D_1, D_2, \ldots, D_K\}$ is a labeled training dataset with K classes, and each $D_k = \{X_1, X_2, \ldots, X_{N_k}\}$ $(k = 1, 2, \ldots, K)$ represents the k-th class, which contains N_k uncertain data objects. $N = \sum_{k=1}^{K} N_k$ is the total number of training data objects. X_{n_k} $(n_k = 1, 2, \ldots, N_k)$ represents the n_k-th data object in the data set D_k with class label C_k. Let C be the set of the class labels, $C = \{C_1, C_2, \ldots, C_K\}$.

Each X_{n_k} contains d numerical (real valued) dimensions, $X_{n_k} = (X^1_{n_k}, X^2_{n_k}, \ldots, X^d_{n_k})$. The value of each dimension $X^j_{n_k}$ is uncertain. Being a scalar random variable, it is described not by a single value, but a probability distribution. The probability density function for $X^j_{n_k}$ is $p^j_{n_k}$. Since we have adopted the naive Bayes model, we assume that each $X^j_{n_k}$ $(j = 1, 2, \ldots, d)$ is independent of another.

Let X be an unlabeled uncertain data object used for testing. It has d dimension: $X = (X^1, X^2, \ldots, X^d)$, where each attribute is modelled by a pdf p^j $(j = 1, 2, \ldots, d)$.

The classification problem is to train a model that maps X to a posterior probability distribution $P(C_k|X)$. Then, we predict the label of X as $Y = \arg\max_{C_k \in C} P(C_k|X)$.

In the Bayes decision framework, Bayes' rule decomposes the computation of a posterior probability into the computation of a likelihood and a prior probability. The likelihood is measured by the class conditional density $P(X|C_k)$, which is estimated using the data subset of the corresponding class. Traditionally, this is estimated with (5), which is based on the deterministic data points in D_k. To handle uncertain data, however, we need to extend the kernel density estimation (3) to cope with the pdf's.

V. Proposed methods

In this section, we propose two approaches for handling uncertain data in naive Bayes classification problem. One is averaging and the other is distribution-based.

A. Averaging

A straight-forward method to deal with the uncertain information is to replace each pdf with its expected value, thus effectively converting the uncertain data objects to deterministic point-valued data. This reduces the problem back to the traditional classification problem and hence the traditional naive Bayes model and kernel density estimation can be reused.

B. Distribution-based

The key step here is the estimation of class conditional density on uncertain data. Following the approach described in Section III-B, we estimate $P(X^j|C_k)$ using $\hat{f}_{h_k^j}(X^j)$. However, we are now dealing with X^j, which is an uncertain value modelled by the pdf p^j. But the kernel function K is defined for scalar-valued parameters only. So, we need to extend (3) to create a kernel-density estimation for X^j.

Since X^j is a probability distribution, it is natural to replace K in (3) using its expected value. In other words, we replace (3) with:

$$\hat{f}_{h_k^j}(X^j) = \frac{1}{N_k h_k^j} \sum_{n_k=1}^{N_k} E\left[K\left(\frac{X^j - X_{n_k}^j}{h_k^j}\right)\right] = \frac{1}{N_k h_k^j} \sum_{n_k=1}^{N_k} \iint K\left(\frac{x^j - x_{n_k}^j}{h_k^j}\right) p^j(x^j) p_{n_k}^j(x_{n_k}^j)\, dx^j\, dx_{n_k}^j$$

Using this to estimate $P(X^j|C_k)$ in (2) gives:

$$P(X|C_k) = \prod_{j=1}^{d} \left\{ \frac{1}{N_k h_k^j} \sum_{n_k=1}^{N_k} \iint K\left(\frac{x^j - x_{n_k}^j}{h_k^j}\right) p^j(x^j)\, p_{n_k}^j(x_{n_k}^j)\, dx^j\, dx_{n_k}^j \right\} \quad (6)$$

The double integral in (6) can be computed through various ways. We give two possible methods in Sections VI and VII.

VI. Formula-based method

In the formula based approach, we first derive the formula for the kernel estimation for uncertain data objects. With this formula, we can then compute the kernel density and run the naive Bayes method to perform the classification. This method only works for some combinations of kernel functions and probability distributions, as closed-form formulas may not always be obtainable in the general case. We use a Gaussian kernel function and consider Gaussian distribution.

Suppose X and X_{n_k} are uncertain data objects with multivariate Gaussian distributions, i.e., $X \sim N(\mu, \Sigma)$ and $X_{n_k} \sim N(\mu_{n_k}, \Sigma_{n_k})$. Here, $\mu = (\mu^1, \mu^2, \ldots, \mu^d)$ and $\mu_{n_k} = (\mu_{n_k}^1, \mu_{n_k}^2, \ldots, \mu_{n_k}^d)$ are the means of X and X_{n_k} while Σ and Σ_{n_k} are their covariance matrixes, respectively. Because of the independence assumption, Σ and Σ_{n_k} are diagonal matrixes. Let σ^j and $\sigma_{n_k}^j$ be the standard deviations of the j-th dimension for X and X_{n_k} respectively. Then, $X^j \sim N(\mu^j, \sigma^j \cdot \sigma^j)$ and $X_{n_k}^j \sim N(\mu_{n_k}^j, \sigma_{n_k}^j \cdot \sigma_{n_k}^j)$. To classify X using naive Bayes model, we compute the all the class condition density $P(X|C_k)$ based on (6).

Since $X_{n_k}^j$ follows Gaussian distribution, we have:

$$p_{n_k}^j(x_{n_k}^j) = \frac{1}{\sigma_{n_k}^j \sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x_{n_k}^j - \mu_{n_k}^j}{\sigma_{n_k}^j}\right)^2\right) \quad (7)$$

and similarly for X^j (by omitting all subscripts in (7)).

Based on formulas (4), (6), (7), we get, after simplifications:

$$P(X|C_k) = \prod_{j=1}^{d} \left\{ \sum_{n_k=1}^{N_k} \frac{\exp\left(-\frac{1}{2}\left(\frac{\mu^j - \mu_{n_k}^j}{\nu_{k,n_k}^j}\right)^2\right)}{N_k \nu_{k,n_k}^j \sqrt{2\pi}} \right\} \quad (8)$$

where $\nu_{k,n_k}^j = \sqrt{h_k^j \cdot h_k^j + \sigma^j \cdot \sigma^j + \sigma_{n_k}^j \cdot \sigma_{n_k}^j}$. This gives the class condition density estimate. We need to repeat this calculation for every class C_k. Based on formula (8), it is clear that the time complexity is $\sum_{k=1}^{K} O(N_k d) = O(Nd)$. Recall that N is the total number of the training data objects, and d is the dimension of the data objects.

VII. Sample-based method

In sample based approach, every training and testing uncertain data object is represented by sample points based on their own distributions. When using kernel density estimation for a data object, every sample point contributes to the density estimation. The integral of density can be transformed into the summation of the data points' contribution with their probability as weights. Equation (6) is thus

Table I
SELECTED DATASETS

Dataset	Tuples	Features	Classes
glass	214	10	6
ionosphere	351	32	2
wine	178	13	3
segment	2310	14	7
waveform	400	40	3
Optdigits	569	64	10
Diabetes	768	8	2
Hear-Statlog	270	13	2
Blood Transfusion	748	4	2
Vowel	990	10	11

Table II
ACCURACY

Dataset	AVG	SBC	FBC	w
glass	0.513	0.544	0.553	3
ionosphere	0.915	0.915	0.920	5
wine	0.966	0.984	0.978	1
segment	0.808	0.865	0.872	4
waveform	0.777	0.789	0.787	3
Optdigits	0.808	0.905	0.919	7
Diabetes	0.750	0.767	0.771	4
Hear-Statlog	0.833	0.844	0.852	17
Blood Transfusion	0.754	0.767	0.773	10
Vowel	0.573	0.599	0.595	1

replaced by:

$$P(X|C_k) = \prod_{j=1}^{d} \frac{1}{N_k h_k^j} \sum_{n_k=1}^{N_k} \sum_{c=1}^{s} \sum_{d=1}^{s} K\left(\frac{x_c^j - x_{n_k,d}^j}{h_k^j}\right) P(x_c^j) P(x_{n_k,d}^j) \quad (9)$$

Here x_c^j represents the c-th sample point of uncertain test data object X along the j-th dimension. $x_{n_k,d}^j$ represents the d-th sample point of uncertain training data object X_{n_k} along the j-th dimension. $P(x_c^j)$ and $P(x_{n_k,d}^j)$ are probabilities according to X and X_{n_k}'s distribution respectively. And s is the number of samples used for each of X^j and $X_{k_n}^j$ along the j-th dimension. For the computation of (9), we will sample s random points on object X and every X_{n_k}'s each dimension, and get the corresponding probability of each sample point. After computing the $P(X|C_k)$ for X with each class C_k, we can compute the posterior probability $P(C_k|X)$ based on (1) and X can be assigned to the class with the maximum $P(C_k|X)$. Since we need to evaluate (9) for each class C_k, the time complexity of this method is $\sum_{k=1}^{K} O(N_k d s^2) = O(N d s^2)$.

VIII. EXPERIMENTS

To study the performance of our algorithms, we have performed experiments on some UCI datasets [14] listed in Table I. These datasets are chosen because they contain all numerical attributes obtained from measurements. For the purpose of our experiments, classification models are learned on the numerical attributes and their "class label" attributes. All experiments are conducted on a computer with an Intel Core 2 Duo E6750 2.66GHz processor and 4GB of RAM.

Because the original data tuples contain point values without uncertainty, we have inserted the uncertainty information for the datasets, following [8]. For the AVG method, the original point value data are used as the expected value, and the experiments are performed on the original datasets. For the formula and sample based methods, we use Gaussian distribution as the uncertainty model. Suppose x_{min}^j and x_{max}^j are the minimum and maximum values for feature A_j, then the range of values for A_j is $(x_{max}^j - x_{min}^j)$. The model parameters are determined as follows: 1) for each uncertain object, use the original point value of the data as the mean value μ^j of the uncertain object; 2) set the standard deviation $\sigma^j = 0.25 \cdot (x_{max}^j - x_{min}^j) \cdot w\%$. Here w is a percentage parameter that allows us to control the uncertainty level of the objects. The greater the value of w, the higher the uncertain level (see also Section IX). After setting the distribution parameters, we can learn the naive Bayes model based on the related formulas in Sections VI and VII. For the sampling-based approach, s data points are sampled along each dimension for each object according to the distribution with the previously determined parameters.

For the bandwidth parameter h, we apply the widely used bandwidth estimation rule called the Silverman approximation rule [15], which suggests setting $h_k^j = 1.06.\sigma_{n_k}^j.N_k^{-\frac{1}{5}}$. Here h_k^j is the kernel bandwidth for the k-th class of dataset D_k along the j-th dimension, $\sigma_{n_k}^j$ is the standard deviation of uncertain object X_{n_k} in the j-th dimension, and N_k is the number of data objects of D_k. Thus, as the naive Bayes model observes more training data, its density estimation becomes increasingly local.

A. Accuracy

We have run the experiments on the selected datasets using naive Bayes model (AVG), sample-based method (SBC) and formula-based method (FBC). For each dataset, we use 10-fold cross validation to measure the accuracy. We have repeated the experiments using various values of $w\%$. For each dataset, Table II reports the best accuracy achieved over different w settings. The number of sample points used for SBC is $s = 100$.

From the table, we can see that our distribution-based methods (SBC and FBC) can consistently achieve higher accuracy than averaging (naive Bayes). This confirms our hypothesis that by considering the information of the whole pdf's rather than just the mean values, more accurate classifiers can be learnt. FBC generally gives higher accuracies than SBC, because SBC is essentially a numerical way of evaluating the double integral in (6). As such, calculation errors are incurred due to the finite number of sample points used.

Note that we have reported in Table II only the best accuracy values over a wide range of values of w we have tried. The reason is that we intend to present here the potential improvement on accuracy that can be achieved by considering the complete pdf information of the uncertain data. How to find out the suitable values of w is a subject of further research, and will be discussed further in Section IX.

In Figure 1, we plot the accuracy of FBC and naive Bayes against w for three of the datasets. We can see that the accuracy first rises and then drops as w increases. We hypothesize that the UCI datasets are not noise-free. They already contain measurement errors. The way we generate the uncertainty information is an attempt to model such errors. We conjecture that when the uncertainty information so generated can model the measurement errors accurately, then FBC can give a very high accuracy, significantly higher than a naive Bayes classifier. The observations that the accuracy attains a peak at particular values of w provides evidence for our conjecture: At these values of w, the injected uncertainty information most closely models the real measurement errors in the original UCI datasets. Therefore, the highly accurate classifiers are obtained. When w deviates from these values, the generated uncertainty information no longer models the measurement errors accurately. So, the resulting accuracies of FBC drop. Therefore, it is important to have a good model of the measurement errors. It should be noted that the best values of w given in Table II appear to agree with those given in [8],[2] even though [8] uses decision trees—a very different classification model from the naive Bayes model used in this paper. This agreement on the best values of w cannot be a mere coincidence. Rather, it suggests that the best value of w for a given distribution model is an intrinsic property of those datasets, independent of the learning algorithms employed. This is an evidence for our hypothesis above, that the datasets contains errors. When such errors are modelled properly, better classifiers can be learnt.

B. Performance

The execution times consumed by the algorithms on the various datasets are shown in Table III. All the time values are given in number of seconds. From the tables, it can be observed that SBC is 10000 times slower than naive Bayes. This is expected: To evaluate $P(X|C_k)$, SBC uses (9), which contains two summations over s sample points each, while naive Bayes uses a single value in the place of this double summation. This means that SBC needs to perform s^2 times more calculations than naive Bayes. Since we used $s = 100$ in our experiments, SBC is $100^2 = 10000$ times slower.

So, although SBC can build more accurate classifiers than naive Bayes (as shown in Section VIII-A), the bad performance makes it impractical. Nevertheless, FBC comes to the rescue. Examining the last column of Table III, we find that FBC gives impressive speedup ratios of the order 10^3 over SBC.

[2]when comparing the datasets that are common in both experiments, giving consideration to the granularity of w tested [8]

Table III
EXECUTION TIME

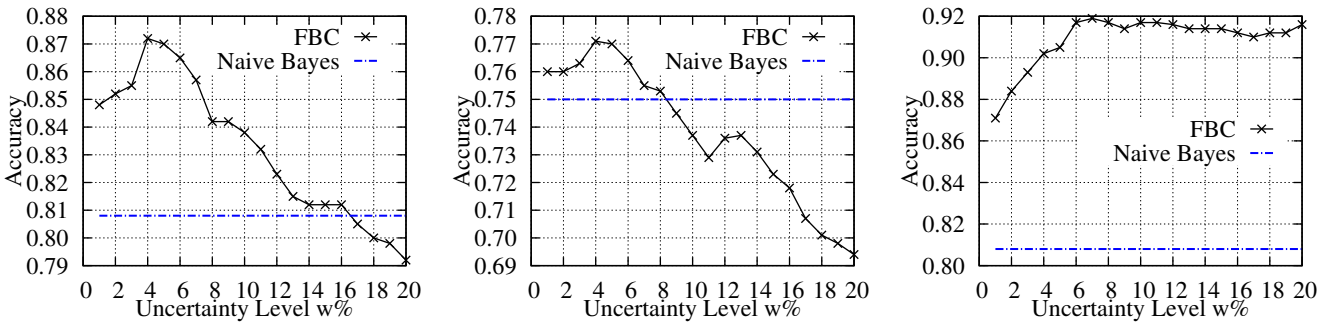
Dataset	AVG (sec.)	SBC (sec.)	FBC (sec.)	speedup (SBC/FBC)
glass	0.312	0.0170×10^4	2.55	0.667×10^3
ionosphere	2.20	2.02×10^4	11.4	1.77×10^3
wine	0.266	0.196×10^4	3.11	0.631×10^3
segment	1.52	1.46×10^4	7.83	1.87×10^3
waveform	3.30	3.38×10^4	17.3	1.95×10^3
Optdigits	9.88	10.8×10^4	37.3	2.90×10^3
Diabetes	2.31	2.44×10^4	7.84	3.12×10^3
Hear-Statlog	0.484	0.479×10^4	3.63	1.32×10^3
Blood Trans.	1.06	1.18×10^4	3.81	3.10×10^3
Vowel	4.70	5.29×10^4	14.2	3.72×10^3

Although highly efficient and scalable, FBC is still 3–15 times slower than naive Bayes. This is a trade-off between execution time and accuracy of the resulting classifiers. We remind readers that FBC and SBC, by considering the complete information of pdf's of the uncertain objects, can build more accurate classifiers than naive Bayes classifier, which uses only the mean of the pdf's.

IX. DISCUSSIONS

In the experiments, we have used uncertain datasets that are generated from real datasets from the UCI repository [14]. This was necessary as the UCI datasets do not provide uncertainty information. Each data tuple is represented by point-values, and no information about the distribution of the attribute values is provided. Nevertheless, we believe that the data values are not perfect and are subject to errors such as measurement errors, rounding errors, etc. Moreover, central to this paper is the conjecture that the better the uncertainty information can model the errors, the higher the accuracy of the classifiers that we can build using our novel algorithms, which can exploit the uncertainty information. Therefore, for experiment purposes, we had to generate the uncertainty information (in the form of pdf's) by guessing the error models. We have tried to model the errors using Gaussian distribution, over a wide range of parameter w, which controls the standard deviation of the pdf's. As reported in Section VIII-A, we have found that the accuracy of the classifiers built by FBC does attain a maximum at certain values of parameter w. This acts as an evidence for our conjecture.

Because of this conjecture, we recommend that practitioners gather and keep uncertainty information when collecting their datasets. By gathering and storing the pdf's using external means, it is often possible to model the errors much more accurately than what we have done in the experiments. For examples, for physical measurements, high-end measuring equipments usually come with technical specifications that



(a) segment (b) Diabetes (c) Optdigits

Figure 1. Accuracy vs. uncertainty level for various datasets

state the error model. (e.g., the user manual may state that the readings are correct up to $\pm 5\%$, meaning that we may model the error by a uniform distribution that spans $\pm 5\%$ of the reading.) Such specifications are usually provided by the manufacturers of the measurement devices by carefully designed experiments and calibrations.

As our experiments have shown, exploiting uncertainty information, it is possible to use SBC/FBC to find classifiers that are more accurate than naive Bayes classifiers.

X. Conclusions

We address the problem of extending traditional naive Bayes model to the classification of uncertain data. The key problem in naive Bayes model is class conditional probability estimation, and kernel density estimation is a common way for that. We have extended the kernel density estimation method to handle uncertain data. This reduces the problem to the evaluation of double-integrals. For particular kernel functions and probability distributions, the double integral can be analytically evaluated to give a closed-form formula, allowing an efficient formula-based algorithm. In general, however, the double integral cannot be simplified in closed forms. In this case, a sample-based approach is proposed. Extensive experiments on several UCI datasets show that the uncertain naive Bayes model considering the full pdf information of uncertain data can produce classifiers with higher accuracy than the traditional model using the mean as the representative value of uncertain data. Time complexity analysis and performance analysis based on experiments show that the formula-based approach has great advantages over the sample-based approach.

References

[1] C. C. Aggarwal and P. S. Yu, "A survey of uncertain data algorithms and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 5, pp. 609–623, May 2009.

[2] H.-P. Kriegel and M. Pfeifle, "Hierarchical density-based clustering of uncertain data," in *ICDM*. IEEE Computer Society, 2005, pp. 689–692.

[3] B. Kao, S. D. Lee, D. W. Cheung, W.-S. Ho, and K. F. Chan, "Clustering uncertain data using Voronoi diagrams," in *ICDM*. IEEE Computer Society, 2008, pp. 333–342.

[4] G. Cormode and A. McGregor, "Approximation algorithms for clustering uncertain data," in *PODS*, M. Lenzerini and D. Lembo, Eds. ACM, 2008, pp. 191–200.

[5] S. D. Lee, B. Kao, and R. Cheng, "Reducing UK-means to k-means," in *ICDM Workshops*. IEEE Computer Society, 2007, pp. 483–488.

[6] J. Bi and T. Zhang, "Support vector classification with input data uncertainty," in *NIPS*, 2004.

[7] C. C. Aggarwal, "On density based transforms for uncertain data mining," in *ICDE*. IEEE, 2007, pp. 866–875.

[8] S. Tsang, B. Kao, K. Y. Yip, W.-S. Ho, and S. D. Lee, "Decision trees for uncertain data," in *ICDE*, IEEE. IEEE, 2009, pp. 441–444.

[9] C. C. Aggarwal and P. S. Yu, "Outlier detection with uncertain data," in *SDM*. SIAM, 2008, pp. 483–493.

[10] C. K. Chui and B. Kao, "A decremental approach for mining frequent itemsets from uncertain data," in *PAKDD*, vol. 5012. Springer, 2008, pp. 64–75.

[11] Q. Zhang, F. Li, and K. Yi, "Finding frequent items in probabilistic data," in *SIGMOD Conference*, J. T.-L. Wang, Ed. ACM, 2008, pp. 819–832.

[12] C. C. Aggarwal, "On high dimensional projected clustering of uncertain data streams," in *ICDE*, IEEE. IEEE, 2009, pp. 1152–1154.

[13] J. Pei, B. Jiang, X. Lin, and Y. Yuan, "Probabilistic skylines on uncertain data," in *VLDB*. ACM, 2007, pp. 15–26.

[14] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html

[15] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[16] *Proceedings of the 25th International Conference on Data Engineering, ICDE 2009, March 29 2009 - April 2 2009, Shanghai, China*, IEEE. IEEE, 2009.