Learning Naive Bayes Classifier from Noisy Data

Yirong Yang, Yi Xia, Yun Chi, and Richard R. Muntz

University of California, Los Angeles, CA 90095, USA
{yyr,xiayi,ychi,muntz}@cs.ucla.edu

Abstract. Classification is one of the major tasks in knowledge discovery and data mining. Naive Bayes classifier, in spite of its simplicity, has proven surprisingly effective in many practical applications. In real datasets, noise is inevitable, because of the imprecision of measurement or privacy preserving mechanisms. In this paper, we develop a new approach, LinEar-Equation-based noise-aWare bAYes classifier (*LEEWAY*), for learning the underlying naive Bayes classifier from noisy observations. Using linear system of equations and optimization methods, *LEEWAY* reconstructs the underlying probability distributions of the noise-free dataset based on the given noisy observations. By incorporating the noise model into the learning process, we improve the classification accuracy. Furthermore, as an estimate of the underlying naive Bayes classifier for the noise-free dataset, the reconstructed model can be easily combined with new observations that are corrupted at different noise levels to obtain a good predictive accuracy. Several experiments are presented to evaluate the performance of *LEEWAY*. The experimental results show that *LEEWAY* is an effective technique to handle noisy data and it provides higher classification accuracy than other traditional approaches.
keywords: *naive Bayes classifier, noisy data, classification, Bayesian network.*

1 Introduction

Classification is one of the major tasks in knowledge discovery and data mining. Naive Bayes classifier, in spite of its simplicity, has proven surprisingly effective in many practical applications, including natural language processing, pattern classification, medical diagnosis and information retrieval [12]. The input dataset for naive Bayes classifier is a set of structured tuples comprised of <feature vector, class value> pairs. The fundamental assumption of naive Bayes classifier is that the feature variables are conditionally independent given the class value. This classifier learns from the training dataset the conditional probability distribution of each feature variable X_i given the class value c. Given a new instance $< x_1, x_2, ..., x_n >$ of the feature vector $< X_1, X_2, ..., X_n >$, the goal of the classification then is to predict its class value c with the highest posterior probability $P(C = c|x_1, x_2, ..., x_n)$.

The classification accuracy depends not only on the learning algorithm, but also on the quality of the input dataset. In a real dataset, noise is inevitable,

because of the imprecision of measurement or privacy preserving mechanisms [1]. In this paper, we develop a new approach, LinEar-Equation-based noise-aWare bAYes classifier (*LEEWAY*), for learning the underlying naive Bayes classifier from noisy observations. Using linear system of equations and optimization methods, *LEEWAY* reconstructs the probability distributions as an estimate of the underlying real naive Bayes classifier. By incorporating the noise model into the learning process, we improve the classification accuracy. Furthermore, with new observations that are corrupted at different noise levels, we obtain an extended naive Bayes structure that combines the reconstructed probability distributions with the noise information. From this extended naive Bayes structure, we obtain a better classifier of the new observations. Since the noise model can be added either on the class variable or on feature variables or both, we will study each of the scenarios respectively in section 3.

1.1 Related Work

Noise identification and data cleaning have been studied in several different data mining tasks, including classification ([10], [5], [17]), pattern discovery ([8], [20]), privacy preserving ([1]), speech recognition ([19]), and Bayesian network learning ([16], [2]).

Different noise models are introduced for different purposes. In privacy preserving data mining, [1] studied the distortion method, in which data are randomly perturbed, and developed a way to reconstruct probability distribution from distorted data. [2] defined the noise model as the conditional probability distribution of the observed feature variable given the noise-free feature variable, and analyzed the effects of noise on learning Bayesian networks. [20] introduced *compatibility matrix* as a way to provide a probabilistic connection from the observed data to the underlying true data, and developed methods to discover long sequential patterns in a noisy environment. The notion of an *uncertain database* was first introduced in [13] for association rule mining. In an uncertain dataset, instead of giving explicit values, each tuple contains a tag value for each feature variable that gives the probability of the feature values appearing in the underlying noise-free dataset given this observation.

There are several ways to represent and compensate for noise in the observed dataset. One approach is to develop robust algorithms that allow for noise by avoiding over-fitting the model to the data ([15], [6]). Another approach is to pre-processing the input data before learning. [5] applied a set of learning algorithms to create classifiers as filters to identify and eliminate mislabelled training instances. [10] examined and extended $C4.5$ decision tree algorithm by pruning the outliers. There are two weaknesses in eliminating corrupted tuples. First, by eliminating the whole tuple, it also eliminates potentially useful information such as the uncorrupted feature values. Secondly, when there is a large amount of noise in the dataset, the amount of information in the remaining clean dataset may not be sufficient for building the classifier. Therefore, new pre-processing techniques have been developed to correct the corrupted data. [11] presented an approach for identifying corrupted fields and using the remaining non-corrupted

fields for subsequent modelling and analysis. [16] used Bayesian methods to clean corrupted data that have dependencies among features. However, this technique requires an expert to provide a model in the form of a Bayesian network for the basic structure of the data, and a small amount of cleaned data. [17] presented a method for correcting misclassified data to improve classification accuracy based on the other predicted feature values and the remaining feature values. However, in a noisy dataset where each value is corrupted, to correct the corrupted data is neither easy nor effective. Thus, further techniques are developed. Instead of correcting each value, [1] reconstructed the probability distributions of the underlying noise-free dataset with continuous feature variables.

1.2 Our Contributions

The main contributions of this paper are: (1) We focus on learning naive Bayes classifier from noisy dataset, where each value is corrupted and noise is added on either feature values or class values or both. For such noisy dataset, neither eliminating nor correction technique will be effective or easy for pre-processing. (2) We introduce extended naive Bayes structures, which combine the naive Bayes classifier for the noise-free dataset and the noisy observations. (3) Instead of pre-processing the noisy observations, we incorporate the noise model into the learning process and develop a new approach, LinEar-Equation-based noise-aWare bAYes classifier (*LEEWAY*), to reconstruct the conditional probability distribution of the feature variables given the class value for the underlying noise-free dataset. *LEEWAY* is a general method in the sense that it is suitable for any noise model. (4) We have performed several experiments to evaluate the performance of *LEEWAY* on different noisy datasets. The experimental results show that *LEEWAY* is an effective technique to handle noisy data and it provides higher classification accuracy than other traditional approaches.

The rest of the paper is organized as follows. In section 2, we give a brief description of the problem and introduce extended naive Bayes structures. In section 3, we develop *LEEWAY* approach for learning the naive Bayes classifier from noisy training data in three scenarios according to whether noise is added on feature variables or class variable or both. In section 4, we present experimental results. Finally, section 5 gives conclusions and future research work directions.

2 Problem Description

Let $\mathcal{D}$ be the noise-free dataset that contains feature vector $< X_1, X_2, ..., X_n >$ and the class variable C. Let $\mathcal{X}_i$ be the domain of feature variable X_i ($i = 1, ..., n$) and $\mathcal{S}$ be the domain of C. Then each tuple in $\mathcal{D}$ has the structure $< x_{1,j_1}, x_{2,j_2}, ..., x_{n,j_n}, c_i >$, where $x_{i,j_i} (\in \mathcal{X}_i)$ is a value of X_i and $c_i (\in \mathcal{S})$ is a class value. Let $|\mathcal{X}_i|$ be the size of domain $\mathcal{X}_i$ and $|\mathcal{S}|$ be the size of domain $\mathcal{S}$. After adding noise on $\mathcal{D}$, the observed noisy dataset $\mathcal{D}'$ contains a set of tuples in the form of $< x'_{1,j_1}, x'_{2,j_2}, ..., x'_{n,j_n}, c'_i >$, where $x'_{i,j_i} (\in \mathcal{X}_i)$ is a value of the observed feature variable X'_i and $c'_i (\in \mathcal{S})$ is a value of observed class variable

C'. We assume that a noisy variable takes its values from the same domain as the corresponding noise-free variable. Given $\mathcal{D}'$, our goal is to constitute the naive Bayes classifier for $\mathcal{D}$, and apply the learned classifier to new observations for classification. Based on the fundamental assumption of naive Bayes classifier that the feature variables are independent given the class value,

$$P(X_1, X_2, ..., X_n|C = c) = \prod_{i=1}^{n} P(X_i|C = c) \tag{1}$$

the conditional probabilities $P(X_i|C = c)$ for each feature variable X_i can be estimated separately and independently. Therefore, in the following analysis, we will focus on one feature variable and all the analysis is applicable to other feature variables. Let X be a feature variable ranging over the feature vector in $\mathcal{D}$ and $\mathcal{X}$ be the domain of feature variable X. In this paper, we consider only discrete feature variables and discrete class variable.

In a noisy environment, the underlying naive Bayes classifier structure can be extended to include the observed noisy variables. We assume that noise is added on each value of each tuple independently, and each tuple in the noisy dataset is observed independently. Under this assumption, the extended structures for three scenarios, according to whether noise is added to feature variables or the class variable or both, are shown in Fig. 1. As we can see from these extended structures, the observed feature variables preserve the conditional independencies given the class value.

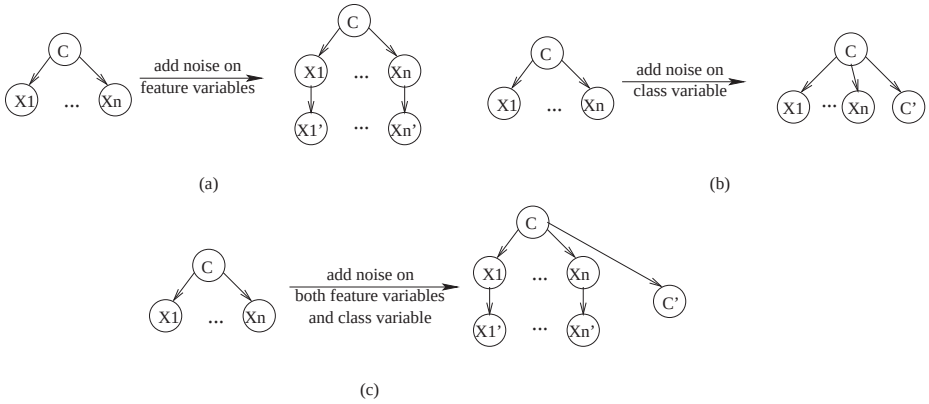

Fig. 1. Extended naive Bayes classifier structures: (a) case of noisy feature variables with noise-free class variable; (b) case of noise-free feature variables with noisy class variable; (c) case of noisy feature variables and noisy class variable.

3 Learning Naive Bayes Classifier from Noisy Data

In the following subsections, we study each of the three scenarios presented in Fig. 1 and describe our *LEEWAY* method for learning naive Bayes classifiers from noisy data.

3.1 Noisy Feature Variables with Noise-free Class Variable

In this scenario, noise is added on each feature value in each tuple independently, while the observed class value is noise free. A good example would be a dataset perturbed for privacy preserving purpose. As shown in Figure 1 (a), the observed feature variables preserve the conditional independencies given the class value. In this case, the major task of learning the naive Bayes classifier is to estimate the conditional probability $P(X|C)$ based on the noisy observation $< X', C >$ in $\mathcal{D}'$. According to probability theory, we have

$$\begin{aligned} P(X'|C) &= \sum_{x_i \in \mathcal{X}} P(X', X = x_i|C) \\ &= \sum_{x_i \in \mathcal{X}} P(X'|X = x_i, C) \cdot P(X = x_i|C) \\ &= \sum_{x_i \in \mathcal{X}} P(X'|X = x_i) \cdot P(X = x_i|C) \end{aligned} \tag{2}$$

The last equation holds due to the assumption that X' is independent of C given $X = x_i$. Specifically,

$$P(X' = x_j|C) = \sum_{x_i \in \mathcal{X}} P(X' = x_j|X = x_i) \cdot P(X = x_i|C), \qquad x_j \in \mathcal{X} \tag{3}$$

For each fixed class value, there are $|\mathcal{X}|$ equations, one for each $P(X' = x_j|C)$. In matrix form, these equations are:

$$\begin{pmatrix} P(X'=x_1|X=x_1) & P(X'=x_1|X=x_2) & \dots & P(X'=x_1|X=x_{|\mathcal{X}|}) \\ P(X'=x_2|X=x_1) & P(X'=x_2|X=x_2) & \dots & P(X'=x_2|X=x_{|\mathcal{X}|}) \\ \vdots & \vdots & \ddots & \vdots \\ P(X'=x_{|\mathcal{X}|}|X=x_1) & P(X'=x_{|\mathcal{X}|}|X=x_2) & \dots & P(X'=x_{|\mathcal{X}|}|X=x_{|\mathcal{X}|}) \end{pmatrix} \cdot \begin{pmatrix} P(X=x_1|C) \\ P(X=x_2|C) \\ \vdots \\ P(X=x_{|\mathcal{X}|}|C) \end{pmatrix} = \begin{pmatrix} P(X'=x_1|C) \\ P(X'=x_2|C) \\ \vdots \\ P(X'=x_{|\mathcal{X}|}|C) \end{pmatrix} \tag{4}$$

Let $P(X' = x_j|X = x_i) = p_{ji}$, $x_i, x_j \in \mathcal{X}$. Then the matrix $(p_{ji})_{|\mathcal{X}|\times|\mathcal{X}|}$ gives a clear representation of the likelihood of value distortion. Then Eq. 4 can be rewritten as

$$\begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1,|\mathcal{X}|} \\ p_{21} & p_{22} & \cdots & p_{2,|\mathcal{X}|} \\ \vdots & \vdots & \ddots & \vdots \\ p_{|\mathcal{X}|,1} & p_{|\mathcal{X}|,2} & \cdots & p_{|\mathcal{X}|,|\mathcal{X}|} \end{pmatrix} \cdot \begin{pmatrix} P(X = x_1|C) \\ P(X = x_2|C) \\ \vdots \\ P(X = x_{|\mathcal{X}|}|C) \end{pmatrix} = \begin{pmatrix} P(X' = x_1|C) \\ P(X' = x_2|C) \\ \vdots \\ P(X' = x_{|\mathcal{X}|}|C) \end{pmatrix} \tag{5}$$

Since $P(X' = x_j|C)$ $(x_j \in \mathcal{X})$ can be estimated from the observed dataset and p_{ji} can be obtained from the noise model, the solution to the above set of linear equations will give the values of $P(X = x_i|C)$, $(x_i \in \mathcal{X})$. If $0 \leq P(X = x_i|C) \leq 1$, then this set of values is a feasible solution.

As an illustration, we assume a simple noise model on the feature variable X as follows[1]:

$$P(X' = x_j|X = x_i) = \begin{cases} 1-t, & \text{if } i = j \\ \frac{t}{|\mathcal{X}|-1}, & \text{if } i \neq j \end{cases} \qquad \text{for all } x_i, x_j \in \mathcal{X} \tag{6}$$

That is, for noise level t, the observed value of a feature variable is the same as its real value with probability $1-t$ and other values with equal probability $\frac{t}{|\mathcal{X}|-1}$. Then, Eq. 4 can be rewritten as:

$$\begin{pmatrix} (1-t) & \frac{t}{|\mathcal{X}|-1} & \cdots & \frac{t}{|\mathcal{X}|-1} \\ \frac{t}{|\mathcal{X}|-1} & (1-t) & \cdots & \frac{t}{|\mathcal{X}|-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{t}{|\mathcal{X}|-1} & \frac{t}{|\mathcal{X}|-1} & \cdots & (1-t) \end{pmatrix} \cdot \begin{pmatrix} P(X = x_1|C) \\ P(X = x_2|C) \\ \vdots \\ P(X = x_{|\mathcal{X}|}|C) \end{pmatrix} = \begin{pmatrix} P(X' = x_1|C) \\ P(X' = x_2|C) \\ \vdots \\ P(X' = x_{|\mathcal{X}|}|C) \end{pmatrix} \tag{7}$$

After a series of linear transformation, we have

$$\begin{pmatrix} (1-t)-\frac{t}{|\mathcal{X}|-1} & 0 & \cdots & 0 \\ 0 & (1-t)-\frac{t}{|\mathcal{X}|-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (1-t)-\frac{t}{|\mathcal{X}|-1} \end{pmatrix} \cdot \begin{pmatrix} P(X=x_1|C) \\ P(X=x_2|C) \\ \vdots \\ P(X=x_{|\mathcal{X}|}|C) \end{pmatrix} = \begin{pmatrix} P(X'=x_1|C)-\frac{t}{|\mathcal{X}|-1} \\ P(X'=x_2|C)-\frac{t}{|\mathcal{X}|-1} \\ \vdots \\ P(X'=x_{|\mathcal{X}|}|C)-\frac{t}{|\mathcal{X}|-1} \end{pmatrix} \tag{8}$$

Therefore,

$$\begin{aligned} P(X = x_i|C) &= \frac{P(X' = x_i|C) - \frac{t}{|\mathcal{X}|-1}}{(1-t) - \frac{t}{|\mathcal{X}|-1}} \\ &= \frac{1}{1 - \frac{|\mathcal{X}|}{|\mathcal{X}|-1}t} \cdot P(X' = x_i|C) - \frac{\frac{t}{|\mathcal{X}|-1}}{1 - \frac{|\mathcal{X}|}{|\mathcal{X}|-1}t} \end{aligned} \tag{9}$$

for all $x_i \in \mathcal{X}$.

First, it is obvious that Eq. 9 satisfies the normalization condition that $\sum_{x_i \in \mathcal{X}} P(X = x_i|C) = 1$.

[1] The following analysis can be applied to other noise models.

Secondly, in order to be a feasible solution, the values in Eq. 9 must satisfy the condition that $0 \leq P(X = x_i|C) \leq 1$, for any $x_i \in \mathcal{X}$. That is,

$$\begin{cases} P(X' = x_i|C) \geq \frac{t}{|\mathcal{X}|-1} & (a) \\ P(X' = x_i|C) \leq 1 - t & (b) \\ 0 \leq t < 1 - \frac{1}{|\mathcal{X}|} & (c) \end{cases}$$
$$\text{or}$$
$$\begin{cases} P(X' = x_i|C) \leq \frac{t}{|\mathcal{X}|-1} & (d) \\ P(X' = x_i|C) \geq 1 - t & (e) \\ 1 - \frac{1}{|\mathcal{X}|} < t \leq 1 & (f) \end{cases} \tag{10}$$

- For condition (c) and (f): If $t = 1 - \frac{1}{|\mathcal{X}|}$, then $P(X' = x_j|X = x_i) = \frac{1}{|\mathcal{X}|}$, for any $x_i, x_j \in \mathcal{X}$. It means, given the real value of X, its observation X' takes any value in $\mathcal{X}$ with the same probability $\frac{1}{|\mathcal{X}|}$. Thus, X' is independent of X, and the matrix of coefficients of Eq. 7 is singular. Since $(1 - \frac{1}{|\mathcal{X}|})$ increases as $|\mathcal{X}|$ increases, $t < 1 - \frac{1}{|\mathcal{X}|}$ holds as long as $t < 0.5$ (0.5 is the threshold when $|\mathcal{X}| = 2$). To have noise level $t < 0.5$ is also reasonable in practice. Actually, when $t < 0.5$, the matrix of coefficients of Eq. 7 is a *strictly diagonally dominant matrix*, and Eq. 7 has a unique solution, not necessary a feasible one though. When condition (f) is satisfied for higher noise level, similarly, Eq. 7 has a unique solution, not necessary a feasible one though.
- Condition (a) is satisfied if

$$t \leq (|\mathcal{X}| - 1) \cdot \min_{x_i \in \mathcal{X}} P(X' = x_i|C) \tag{11}$$

  Condition (d) is satisfied if

$$t \geq (|\mathcal{X}| - 1) \cdot \max_{x_i \in \mathcal{X}} P(X' = x_i|C) \tag{12}$$

- Condition (b) is satisfied if

$$t \leq 1 - \max_{x_i \in \mathcal{X}} P(X' = x_i|C) \tag{13}$$

  Condition (e) is satisfied if

$$t \geq 1 - \min_{x_i \in \mathcal{X}} P(X' = x_i|C) \tag{14}$$

From the above analysis, we can see that the unique solution given in Eq. 9 is a feasible solution if t satisfies the following bounds:

$$\begin{cases} 0 \leq t < \min\left\{(|\mathcal{X}| - 1) \cdot \min_{x_i \in \mathcal{X}} P(X' = x_i|C), 1 - \max_{x_i \in \mathcal{X}} P(X' = x_i|C)\right\} \\ \text{or} \\ \max\left\{(|\mathcal{X}| - 1) \cdot \max_{x_i \in \mathcal{X}} P(X' = x_i|C), 1 - \min_{x_i \in \mathcal{X}} P(X' = x_i|C)\right\} < t \leq 1 \end{cases} \tag{15}$$

However, in practical applications, only the frequencies of $(X' = x_i|C)$ can be counted from the sampled dataset as an estimate of the probabilities $P(X' = x_i|C)$. Thus, the equations in Eq. 4 do not always hold exactly. In this case and

the situation when noise level t is beyond the bound in Eq. 15, we can get a feasible approximation by minimizing the deviation between both sides of Eq. 4. We use square-error to measure the distance, and take the constraints defined by probability theory as the constraints of the objective function. Putting in a mathematical way, this becomes an optimization problem described as follows:

$$\begin{cases} \min_{\substack{P(X=x_i|C) \\ x_i \in \mathcal{X}}} \sum_{x_j \in \mathcal{X}} \left(\sum_{x_i \in \mathcal{X}} P(X' = x_j | X = x_i) \cdot P(X = x_i | C) - P(X' = x_j | C) \right)^2 \\ \text{subj. to } \ 0 \leq P(X = x_i | C) \leq 1, \ \text{ for any } x_i \in \mathcal{X} \\ \qquad \sum_{x_i \in \mathcal{X}} P(X = x_i | C) = 1 \end{cases}$$

Given a new observation $< X'_1 = x_1, X'_2 = x_2, ..., X'_n = x_n >$, the classification task is to decide which class this new instance belongs to. If the new observation has noise different from the training dataset, which is also common in practice such as a series of real-time noisy observations, simply applying the naive Bayes classifier of the noisy training dataset can not fit in all the noise levels and will decrease the predictive accuracy. However, if we consider the feature values in the new observation as instances of some noisy variables, we can combine the naive Bayes classifier structure of the noise-free dataset with the newly-observed variables. The extended structure with parameters is shown in Fig. 2.

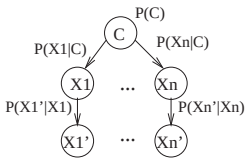

Fig. 2. Naive Bayes classifier structure combined with new noisy observations.

The class value with the highest posterior probability $P(C = c | X'_1 = x_1, X'_2 = x_2, ..., X'_n = x_n)$ gives the classification of the new observation. Based on the extended structure shown in Fig. 2, the maximal posterior probability $P(C = c | X'_1 =' x_1, X'_2 = x_2, ..., X'_n = x_n)$ can be estimated as follows:

$$\begin{aligned} & \max_{c_i \in \mathcal{S}} P(C = c_i | X'_1 = x_1, X'_2 = x_2, ..., X'_n = x_n) \\ = & \max_{c_i \in \mathcal{S}} \frac{P(X'_1 = x_1, X'_2 = x_2, ..., X'_n = x_n | C = c_i) \times P(C = c_i)}{P(X'_1 = x_1, X'_2 = x_2, ..., X'_n = x_n)} \\ = & \max_{c_i \in \mathcal{S}} P(X'_1 = x_1, X'_2 = x_2, ..., X'_n = x_n | C = c_i) \times P(C = c_i) \\ = & \max_{c_i \in \mathcal{S}} \prod_{j=1}^{n} P(X'_j = x_j | C = c_i) \times P(C = c_i) \\ = & \max_{c_i \in \mathcal{S}} \prod_{j=1}^{n} \left(\sum_{k=1}^{|\mathcal{X}|} P(X'_j = x_j | X_j = x_k) P(X_j = x_k | C = c_i) \right) \times P(C = c_i) \end{aligned} \tag{16}$$

Therefore, from the learned naive Bayes classifier for the noise-free dataset, we can get a good estimate of the posterior probability $P(C = c | X'_1 = x_1, X'_2 = x_2, ..., X'_n = x_n)$, and thus give a good prediction.

3.2 Noise-free Feature Variables with Noisy Class Variable

In this scenario, noise is added on the class variable, while the feature variables are noise free. A good example is the dataset generated from one or more machine learning algorithms. The tuples in such dataset are also known as "mislabelled training instances" [5]. An empirical dataset is the medical records which are classified by the patient's symptom. Some cases are likely to be confused, because they have similar symptoms. The extended naive Bayes classifier structure in this case is shown in Figure 1 (b). As we can see from the figure, the observed class variable C' depends directly on its real value c_i and the noise model. In this case, the major task of learning the naive Bayes classifier is to estimate the conditional probability $P(X|C)$ based on the noisy observation $< X, C' >$ in $\mathcal{D}'$. Similar to the analysis in section 3.1, we have

$$P(X|C') = \sum_{c_j \in \mathcal{S}} P(X|C = c_j) \cdot P(C = c_j|C') \tag{17}$$

Specifically,

$$P(X|C' = c_i) = \sum_{c_j \in \mathcal{S}} P(X|C = c_j) \cdot P(C = c_j|C' = c_i), \qquad c_i \in \mathcal{S} \tag{18}$$

For each fixed value of a feature variable, there are $|\mathcal{S}|$ equations, one for each $P(X|C' = c_i), i = 1, 2, ..., |\mathcal{S}|$. In matrix form, these equations are

$$\begin{pmatrix} P(C=c_1|C'=c_1) & P(C=c_2|C'=c_1) & \dots & P(C=c_{|\mathcal{S}|}|C'=c_1) \\ P(C=c_1|C'=c_2) & P(C=c_2|C'=c_2) & \dots & P(C=c_{|\mathcal{S}|}|C'=c_2) \\ \vdots & \vdots & \ddots & \vdots \\ P(C=c_1|C'=c_{|\mathcal{S}|}) & P(C=c_2|C'=c_{|\mathcal{S}|}) & \dots & P(C=c_{|\mathcal{S}|}|C'=c_{|\mathcal{S}|}) \end{pmatrix} \cdot \begin{pmatrix} P(X|C=c_1) \\ P(X|C=c_2) \\ \vdots \\ P(X|C=c_{|\mathcal{S}|}) \end{pmatrix} = \begin{pmatrix} P(X|C'=c_1) \\ P(X|C'=c_2) \\ \vdots \\ P(X|C'=c_{|\mathcal{S}|}) \end{pmatrix} \tag{19}$$

Let $P(C = c_j|C' = c_i) = p_{ij}$, $c_i, c_j \in \mathcal{S}$. Then the matrix $(p_{ij})_{|\mathcal{S}|\times|\mathcal{S}|}$, also known as the *compatibility matrix* in [20], gives a clear representation of the likelihood of value substitutions. Then, Eq. 19 can be rewritten as

$$\begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1,|\mathcal{S}|} \\ p_{21} & p_{22} & \cdots & p_{2,|\mathcal{S}|} \\ \vdots & \vdots & \ddots & \vdots \\ p_{|\mathcal{S}|,1} & p_{|\mathcal{S}|,2} & \cdots & p_{|\mathcal{S}|,|\mathcal{S}|} \end{pmatrix} \cdot \begin{pmatrix} P(X|C = c_1) \\ P(X|C = c_2) \\ \vdots \\ P(X|C = c_{|\mathcal{S}|}) \end{pmatrix} = \begin{pmatrix} P(X|C' = c_1) \\ P(X|C' = c_2) \\ \vdots \\ P(X|C' = c_{|\mathcal{S}|}) \end{pmatrix} \tag{20}$$

Since $P(X|C' = c_i)$, $(c_i \in \mathcal{S})$ can be estimated from the observed dataset and p_{ij} can be obtained from the noise model, the solution to the above set of linear equations will give the values of $P(X|C = c_j)$, $(c_j \in \mathcal{S})$. If $0 \leq P(X|C = c_j) \leq 1$, then this set of values is a feasible solution.

As an illustration, we assume a simple noise model on the class variable C as follows[2]:

$$P(C = c_j|C' = c_i) = \begin{cases} 1 - t, & \text{if } i = j \\ \frac{t}{|\mathcal{S}|-1}, & \text{if } i \neq j \end{cases} \qquad \text{for all } c_i, c_j \in \mathcal{S} \tag{21}$$

[2] The following analysis can be applied to other noise models.

According to this compatibility matrix, the probability of C taking the same value as C' based on the observations is $1-t$, and other values different from C' with the same probability $\frac{t}{|\mathcal{S}|-1}$. Then, Eq. 19 can be rewritten as:

$$\begin{pmatrix} (1-t) & \frac{t}{|\mathcal{S}|-1} & \cdots & \frac{t}{|\mathcal{S}|-1} \\ \frac{t}{|\mathcal{S}|-1} & (1-t) & \dots & \frac{t}{|\mathcal{S}|-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{t}{|\mathcal{S}|-1} & \frac{t}{|\mathcal{S}|-1} & \dots & (1-t) \end{pmatrix} \cdot \begin{pmatrix} P(X|C=c_1) \\ P(X|C=c_2) \\ \vdots \\ P(X|C=c_{|\mathcal{S}|}) \end{pmatrix} = \begin{pmatrix} P(X|C'=c_1) \\ P(X|C'=c_2) \\ \vdots \\ P(X|C'=c_{|\mathcal{S}|}) \end{pmatrix} \tag{22}$$

After a series of linear transformation, we have

$$\begin{pmatrix} (1-t)-\frac{t}{|\mathcal{S}|-1} & 0 & \dots & 0 \\ 0 & (1-t)-\frac{t}{|\mathcal{S}|-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & (1-t)-\frac{t}{|\mathcal{S}|-1} \end{pmatrix} \cdot \begin{pmatrix} P(X|C=c_1) \\ P(X|C=c_2) \\ \vdots \\ P(X|C=c_{|\mathcal{S}|}) \end{pmatrix}$$
$$= \begin{pmatrix} P(X|C'=c_1)-\frac{t}{|\mathcal{S}|-1}\cdot\sum_{c_i\in\mathcal{S}} P(X|C'=c_i) \\ P(X|C'=c_2)-\frac{t}{|\mathcal{S}|-1}\cdot\sum_{c_i\in\mathcal{S}} P(X|C'=c_i) \\ \vdots \\ P(X|C'=c_{|\mathcal{S}|})-\frac{t}{|\mathcal{S}|-1}\cdot\sum_{c_i\in\mathcal{S}} P(X|C'=c_i) \end{pmatrix} \tag{23}$$

Therefore,

$$\begin{aligned} P(X|C=c_j) &= \frac{P(X|C'=c_j) - \frac{t}{|\mathcal{S}|-1}\cdot\sum_{c_i\in\mathcal{S}} P(X|C'=c_i)}{(1-t)-\frac{t}{|\mathcal{S}|-1}} \\ &= \frac{1}{1-\frac{|\mathcal{S}|}{|\mathcal{S}|-1}t}\cdot P(X|C'=c_j) - \frac{\frac{t}{|\mathcal{S}|-1}}{1-\frac{|\mathcal{S}|}{|\mathcal{S}|-1}t}\cdot\sum_{c_i\in\mathcal{S}} P(X|C'=c_i) \end{aligned} \tag{24}$$

for all $c_j \in \mathcal{S}$.

First, it is obvious that solutions in Eq. 24 satisfy the normalization condition that $\sum_{x\in\mathcal{X}} P(X=x|C=c_j)=1$, for any $c_j\in\mathcal{S}$.

Secondly, in order to be a feasible solution, the values in Eq. 24 must satisfy the condition that $0 \le P(X|C=c_j) \le 1$, for any $c_j \in \mathcal{S}$. That is,

$$\begin{cases} P(X|C'=c_i) \ge \frac{t}{|\mathcal{S}|-1}\cdot\sum_{c_k\in\mathcal{S}} P(X|C'=c_k) & (a) \\ P(X|C'=c_i) \le 1+\frac{\sum_{c_k\in\mathcal{S}} P(X|C'=c_k)-|\mathcal{S}|}{|\mathcal{S}|-1}t & (b) \\ 0 \le t < 1-\frac{1}{|\mathcal{S}|} & (c) \end{cases}$$
or
$$\begin{cases} P(X|C'=c_i) \le \frac{t}{|\mathcal{S}|-1}\cdot\sum_{c_k\in\mathcal{S}} P(X|C'=c_k) & (d) \\ P(X|C'=c_i) \ge 1+\frac{\sum_{c_k\in\mathcal{S}} P(X|C'=c_k)-|\mathcal{S}|}{|\mathcal{S}|-1}t & (e) \\ 1-\frac{1}{|\mathcal{S}|} < t \le 1 & (f) \end{cases} \tag{25}$$

– For condition (c) and (f): If $t = 1 - \frac{1}{|\mathcal{S}|}$, then $P(C = c_j|C' = c_i) = \frac{1}{|\mathcal{S}|}$, for any $c_i, c_j \in \mathcal{S}$. It means, given the observed value of C', its noise-free variable C takes any value in $\mathcal{S}$ with the same probability $\frac{1}{|\mathcal{S}|}$. Thus, C is independent of C', and the matrix of coefficients of Eq. 22 is singular. Since $(1 - \frac{1}{|\mathcal{S}|})$ increases as $|\mathcal{S}|$ increases, $t < 1 - \frac{1}{|\mathcal{S}|}$ holds as long as $t < 0.5$ (0.5 is the threshold when $|\mathcal{S}| = 2$). To have noise level $t < 0.5$ is also reasonable in practice. Actually, when $t < 0.5$, the matrix of coefficients of Eq. 22 is a *strictly diagonally dominant matrix*, and Eq. 22 has a unique solution, not necessary a feasible one though. When condition (f) is satisfied for higher noise level, similarly, Eq. 22 has a unique solution, not necessary a feasible one though.
– Condition (a) is satisfied if

$$t \leq \frac{(|\mathcal{S}| - 1) \cdot \min_{c_i \in \mathcal{S}} P(X|C' = c_i)}{\sum_{c_k \in \mathcal{S}} P(X|C' = c_k)} \tag{26}$$

Similarly, condition (d) is satisfied if

$$t \geq \frac{(|\mathcal{S}| - 1) \cdot \max_{c_i \in \mathcal{S}} P(X|C' = c_i)}{\sum_{c_k \in \mathcal{S}} P(X|C' = c_k)} \tag{27}$$

– For condition (b), since

$$1 \geq 1 + \frac{\sum_{c_k \in \mathcal{S}} P(X|C' = c_k) - |\mathcal{S}|}{|\mathcal{S}| - 1} t > 1 + \frac{\sum_{c_k \in \mathcal{S}} P(X|C' = c_k) - |\mathcal{S}|}{|\mathcal{S}|} \geq 0 \tag{28}$$

and

$$\begin{aligned} & 1 + \frac{\sum_{c_k \in \mathcal{S}} P(X|C' = c_k) - |\mathcal{S}|}{|\mathcal{S}| - 1} t \\ = & \frac{t}{|\mathcal{S}| - 1} \cdot \sum_{c_k \in \mathcal{S}} P(X|C' = c_k) + 1 - \frac{|\mathcal{S}|}{|\mathcal{S}| - 1} t \\ > & \frac{t}{|\mathcal{S}| - 1} \cdot \sum_{c_k \in \mathcal{S}} P(X|C' = c_k) \end{aligned} \tag{29}$$

thus, condition (b) is satisfied if

$$t \leq \frac{(|\mathcal{S}| - 1) \cdot \Big(1 - \max_{c_i \in \mathcal{S}} P(X|C' = c_i)\Big)}{|\mathcal{S}| - \sum_{c_k \in \mathcal{S}} P(X|C' = c_k)} \tag{30}$$

Similarly, condition (e) is satisfied if

$$t \geq \frac{(|\mathcal{S}| - 1) \cdot \Big(1 - \min_{c_i \in \mathcal{S}} P(X|C' = c_i)\Big)}{|\mathcal{S}| - \sum_{c_k \in \mathcal{S}} P(X|C' = c_k)} \tag{31}$$

Because

$$|\mathcal{S}| \cdot \min_{c_i \in \mathcal{S}} P(X|C' = c_i) \leq \sum_{c_k \in \mathcal{S}} P(X|C' = c_k) \leq |\mathcal{S}| \cdot \max_{c_i \in \mathcal{S}} P(X|C' = c_i) \tag{32}$$

from the above analysis, we can see that the unique solution given in Eq. 24 is a feasible solution if t satisfies the following bounds:

$$\begin{cases} 0 \leq t < \min \left\{ \frac{(|\mathcal{S}|-1)\cdot \min_{c_i \in \mathcal{S}} P(X|C'=c_i)}{\sum_{c_k \in \mathcal{S}} P(X|C'=c_k)}, \frac{(|\mathcal{S}|-1)\cdot \left(1-\max_{c_i \in \mathcal{S}} P(X|C'=c_i)\right)}{|\mathcal{S}|-\sum_{c_k \in \mathcal{S}} P(X|C'=c_k)} \right\} \\ \text{or} \\ \max \left\{ \frac{(|\mathcal{S}|-1)\cdot \max_{c_i \in \mathcal{S}} P(X|C'=c_i)}{\sum_{c_k \in \mathcal{S}} P(X|C'=c_k)}, \frac{(|\mathcal{S}|-1)\cdot \left(1-\min_{c_i \in \mathcal{S}} P(X|C'=c_i)\right)}{|\mathcal{S}|-\sum_{c_k \in \mathcal{S}} P(X|C'=c_k)} \right\} < t \leq 1 \end{cases} \tag{33}$$

However, in practical applications, only the frequencies of $(X|C' = c_i)$ can be counted from the sampled dataset as an estimate of the probabilities $P(X|C' = c_i)$. Thus, the equations in Eq. 19 do not always hold exactly. In this case and the situation when noise level t is beyond the bound in Eq. 33, we can get a feasible approximation with the optimization method in the similar way as in section 3.1:

$$\begin{cases} \min_{\substack{P(X=x_k|C=c_j) \\ c_j \in \mathcal{S}, x_k \in \mathcal{X}}} \sum_{\substack{c_i \in \mathcal{S} \\ x_k \in \mathcal{X}}} \left(\sum_{c_j \in \mathcal{S}} P(X=x_k|C=c_j) \cdot P(C=c_j|C'=c_i) - P(X=x_k|C'=c_i) \right)^2 \\ \text{subj. to} \quad 0 \leq P(X|C=c_j) \leq 1, \quad \text{for any } c_j \in \mathcal{S} \\ \qquad \sum_{x_k \in \mathcal{X}} P(X=x_k|C=c_j) = 1, \quad \text{for any } c_j \in \mathcal{S} \end{cases}$$

3.3 Noisy Feature Variables with Noisy Class Variable

In this scenario, noise is added on each feature value and class value in each tuple simultaneously and independently. As shown in Figure 1 (c), the observed feature variables preserve the conditional independencies given the class value. Besides, the observed class variable C' depends directly on its real value c_i and the noise model. In this case, the major task of learning the naive Bayes classifier is to estimate the conditional probability $P(X|C)$ based on the noisy observation $< X', C' >$ in $\mathcal{D}'$.

This can be done in two steps. First, we correct the noise in the feature values. Given $C' = c_k (c_k \in \mathcal{S})$, we have

$$\begin{aligned} P(X' = x_j|C' = c_k) &= \sum_{x_i \in \mathcal{X}} P(X' = x_j|X = x_i, C' = x_k) \cdot P(X = x_i|C' = c_k) \\ &= \sum_{x_i \in \mathcal{X}} P(X' = x_j|X = x_i) \cdot P(X = x_i|C' = c_k) \end{aligned} \tag{34}$$

If $P(X'|X)$ is known, we can get an estimate of the probability distribution $P(X|C' = c_k)$ for each $c_k \in \mathcal{S}$ with the same method in section 3.1.

Secondly, we correct the noise in the class values.

$$P(X = x_i|C' = c_k) = \sum_{c_j \in \mathcal{S}} P(X = x_i|C = c_j, C' = x_k) \cdot P(C = c_j|C' = c_k)$$
$$= \sum_{c_j \in \mathcal{S}} P(X = x_i|C = c_j) \cdot P(C = c_j|C' = c_k) \quad (35)$$

If $P(C|C')$ is known, we can get an estimate of the probability distribution $P(X|C)$ with the same method in section 3.2.

4 Experiments

In this section, we describe the experiments to evaluate *LEEWAY* in terms of its classification accuracy according to different noise levels. We focus on the scenario of noisy feature variables with the noise-free class variable. The other two scenarios can be studied in the similar way. First we describe the dataset used in the experiments and noise introduction mechanism. Then we explain how the experiments are carried out and discuss the experimental results.

4.1 Dataset Description

We choose the Nursery dataset from UCI machine learning repository [4] as the underlying noise-free dataset. The Nursery dataset was derived from a hierarchical decision model originally developed to rank applications for nursery schools. It has 12960 complete instances, 5 classes, and 8 nominal attributes each with $3-5$ possible values. We take $2/3$ of the dataset as the training dataset and the remaining $1/3$ as the testing dataset.

A probability t is introduced to control the noise level. We apply noise levels of $0-1$ in increments of 0.05. In the following experiments, we use two different noise models:

[**Model U**] This noise model is defined as follows:

$$P(X' = x_j|X = x_i) = p_{ji} = \begin{cases} 1-t, & \text{if } i = j \\ \frac{t}{|\mathcal{X}|-1}, & \text{if } i \neq j \end{cases} \quad \text{for all } x_i, x_j \in \mathcal{X} \quad (36)$$

Similar to that defined in [2], the noisy dataset $\mathcal{D}'$ at noise level t is generated as follows: for each value x_i of the noise-free feature variable X in the noise-free dataset $\mathcal{D}$, the observed feature value in $\mathcal{D}'$ will remain as x_i with probability $1-t$, and will be replaced by other values in $\mathcal{X}$, each with probability $\frac{t}{|\mathcal{X}|-1}$.

[**Model R**] The noise model is defined as follows:

$$P(X' = x_j|X = x_i) = p_{ji} = \begin{cases} 1-t, & \text{if } i = j \\ r_{ji}, & \text{if } i \neq j \end{cases} \quad \text{for all } x_i, x_j \in \mathcal{X} \quad (37)$$

where r_{ji} is a random number and $\sum_{j,j\neq i} r_{ji} = t$.

As a "random" version of Model U, the noisy dataset $\mathcal{D}'$ at noise level t is generated as follows: for each value x_i of the noise-free feature variable X in the noise-free dataset $\mathcal{D}$, the observed feature value in $\mathcal{D}'$ will remain as x_i with probability $1-t$, and will be replaced by another value in $\mathcal{X}$ with probability r_{ji}.

4.2 Experimental Results

For each noise level, we generate 10 random samples of the noisy training datasets or noisy testing datasets, and the results are averaged over all 10 samples. We perform the following three sets of experiments.

Learning naive Bayes classifier from noisy training datasets In this case, the noisy training datasets are generated from the two noise models described above. In both models, different levels of noise are artificially introduced into the feature values of the training dataset. After learning the underlying naive Bayes classifier, we apply it to the clean testing dataset and evaluate its classification accuracy. We compare the performance of *LEEWAY* with that of the traditional naive Bayes classifier which takes the noisy observations as noise-free datasets. The experimental results for noise Model U and Model R are shown in Fig. 3 and Fig. 4, respectively.

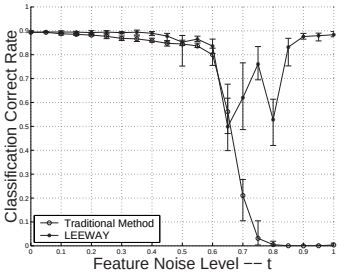

Fig. 3. Classification accuracy of naive Bayes classifier learned from noisy training dataset with noise Model U vs. noise level.

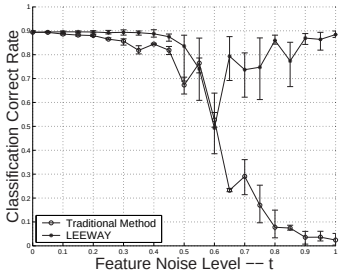

Fig. 4. Classification accuracy of naive Bayes classifier learned from noisy training dataset with noise Model R vs. noise level.

Both Fig. 3 and 4 demonstrate the following common properties:

1. The traditional naive Bayes classifier is quite tolerant to noise at lower levels. For noise level $t \leq 0.5$, the chance of the the observed variable X' keeping its correct value x_i remains high. Thus, the conditional probability $P(X' = x_i|C)$ is close to $P(X = x_i|C)$. Furthermore, under the conditional independence assumption, the posterior probability distribution of the class

variable given a new observation is:

$$\begin{aligned} & P(C = c_i | X_1 = x_1, X_2 = x_2, ..., X_n = x_n) \\ = & \frac{P(X_1=x_1, X_2=x_2, ..., X_n=x_n | C=c_i) \times P(C=c_i)}{P(X_1=x_1, X_2=x_2, ..., X_n=x_n)} \\ = & \frac{\prod_{j=1}^{n} P(X_j=x_j | C=c_i) \times P(C=c_i)}{P(X_1=x_1, X_2=x_2, ..., X_n=x_n)}, \quad c_i \in \mathcal{S} \end{aligned} \tag{38}$$

Thus, the order of the posterior probability $P(C = c_i | X_1 = x_1, X_2 = x_2, ..., X_n = x_n)$ for class variable does not change much. So, when tested on a clean dataset, it will give a correct classification with high probability. However, *LEEWAY* performs at least as well as the traditional method for lower noise levels.

2. For noise level $t = 0.6 \sim 0.8$, *LEEWAY* reaches its worst performance and the results have a larger variance over all the random dataset samples. In the Nursery dataset, each feature variable has $3 \sim 5$ possible values. When $t = 0.6 \sim 0.8$, $1 - t \approx \frac{t}{|\mathcal{X}|-1}$, and the matrix of coefficients in Eq. 4 is close to singular. Thus, neither equation system nor optimization method will give a good estimate of $P(X|C)$.
3. For noise level $t \geq 0.6$, the classification accuracy of the traditional naive Bayes classifier drops quickly to 0, because the chance of the observed variable X' keeping its real value x_i is no longer dominant. Thus, the probability distribution of the noisy dataset is far from that of the noise-free dataset and the classification accuracy greatly decreases. However, *LEEWAY* combines the noise information with the learning procedure and recovers the underlying probability distribution. Thus, its classification accuracy still stays high.

Testing the learned naive Bayes classifier on noisy testing datasets In this case, different levels of noise are artificially introduced into the feature values of the testing dataset. We use noise Model U to generate the noisy testing datasets, and perform two experiments. In one experiment, we first learn the underlying naive Bayes classifier from different noisy training datasets with noise level varying from 0 to 1.0, and then apply these classifiers to a testing dataset with noise level 0.2 to evaluate their classification accuracy. In the other experiment, we first learn the underlying naive Bayes classifier from a noisy training dataset with noise level 0.2 and then apply it to different noisy testing datasets with noise level varying from 0 to 1.0 to evaluate their classification accuracy. We compare the performance of *LEEWAY* with that of the traditional naive Bayes classifier which takes the noisy observations as noise-free datasets. The experimental results of testing the naive Bayes classifiers learned from different noisy training datasets on a fixed noisy testing dataset is shown in Fig. 5. The results of testing the naive Bayes classifier learned from a fixed noisy training dataset on different noisy testing datasets is shown in Fig. 6.

Both Fig. 5 and 6 indicate that *LEEWAY* is applicable to the scenario when the testing dataset is corrupted with noise level different from the training dataset. Fig. 5 has similar properties as Fig. 3 and 4, except that its optimal classification accuracy which occurs at $t = 0$ is degraded from 0.9 to 0.7 since

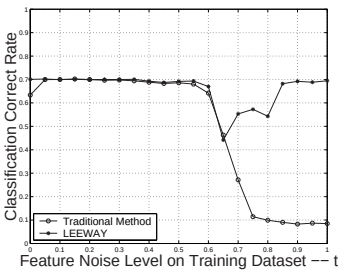

Fig. 5. Testing the naive Bayes classifiers learned from different noisy training datasets on testing dataset with 0.2 noise level.

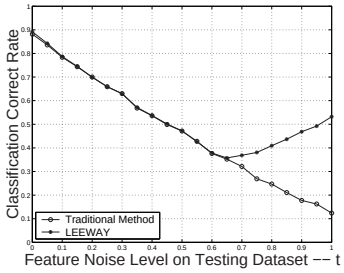

Fig. 6. Testing the naive Bayes classifier learned from a training dataset with 0.2 noise level on different noisy testing datasets.

the testing dataset has 0.2 noise. Fig. 6 indicates that *LEEWAY* has similar performance as the traditional naive Bayes classifier method for noise at lower levels, and better performance for noise at higher levels.

Bayesian Method We also compare our performance with that of the Bayesian method. If we view the extended naive Bayes structures in Fig. 1 as special Bayesian networks and take the unknown noise-free variables as hidden variables, the task here is to learn the posterior conditional probability distribution with this known network structure and the hidden variables. Another technique is to extend the training dataset by adding the unknown noise-free variables and take the extended dataset as an incomplete dataset that does not have values for the added unknown variables. Learning parameters of Bayesian networks from incomplete data can be done by gradient methods discussed in [3] and [18], or EM introduced in [7], or Gibbs sampling discussed in [9]. We use the Bayesian network toolbox developed by Kevin Murphy at MIT [14]. The result of the Bayesian method is shown in Fig. 7.

As shown in Fig. 7, the performance of Bayesian method for this problem turns out to be worse. The two techniques described above are not suitable to handle the noise data in this case. The extended naive Bayes structures in Fig. 1 are not exactly Bayesian networks with hidden variables, because the structure and the space of the unobserved variables are known but the learning procedure does not take this prior information. The extended training dataset is not exactly incomplete dataset, either, because there is not a single instance of these unobserved variables. So, we expect its performance to be no better than our approach. Besides, the complexity of learning Bayesian network parameters with hidden variables or incomplete data is much higher.

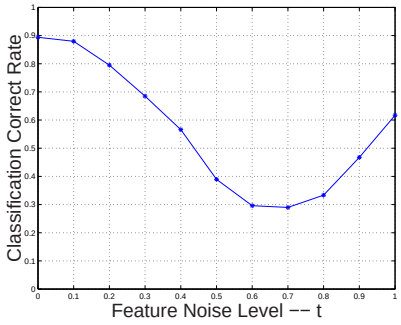

Fig. 7. Learning naive Bayes classifier from noisy data with Bayesian method.

5 Conclusion and Future Directions

In this paper, we addressed the problem of learning naive Bayes classifiers from data where noise is added either on the class variable or on feature variables or both. We proposed a new approach of reconstructing the underlying conditional probability distributions from the observed noisy dataset. We experimentally evaluated our approach on the Nursery dataset whose feature variables were artificially corrupted with different levels of noise, and compared its performance with the traditional naive Bayes classifier method and the Bayesian method. The experimental results demonstrate that our approach improves the classification accuracy for feature-corrupted data.

Several issues remain to be studied. One is to study the sample complexity of learning naive Bayes classifier from noisy data. The probability distribution of the observed dataset is approximated by counting the frequencies. Thus, its estimation accuracy depends on the size of the training dataset. The more data are sampled, the more accurate the estimates are. However, in real applications, it is hard to get a large size dataset which also covers the whole feature variables. So, a problem of interest is the sample size for a relatively reliable and stable classifier. Another possible direction is to study an uncertain dataset introduced in [13] and develop an approach of learning a naive Bayes classifier from the uncertain data.

Naive Bayes classifier is a simple Bayesian classifier. In classification problems, the training dataset contains a distinguished class variable and conditional independencies among feature variables. Making use of the independency relationship can simplify the learning process. However, dependencies among feature variables do exist in real application datasets. As the issue of learning Bayesian network from data becomes more and more popular, another extension of our work is to learn general Bayesian network from noisy data which embeds the dependencies among variables.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Nos. 0086116 and 0085773. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. of the ACM SIGMOD Conference on Management of Data 2000 (SIGMOD'00)*, pages 439–450. ACM Press, May 2000.
2. M. Bendou and P. Munteanu. Learning bayesian networks from noisy data. In *Proc. of the 5th International Conference on Enterprise Information Systems (ICEIS 2003)*, pages 26–33, April 22-26 2003.
3. J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29:213–244, 1997.
4. C. L. Blake and C. J. Merz. UCI repository of machine learning databases. http://www.ics.uci.edu/˜mlearn/MLRepository.html, 1998.
5. C. E. Brodley and M. A. Friedl. Identifying and eliminating mislabeled training instances. *AAAI/IAAI*, 1, 1996.
6. P. Clark and T. Niblett. The cn2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.
7. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, B 39:1–39, 1977.
8. A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *Proc. of 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, July 2002.
9. W. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo Methods in Practice*. CRC Press, 1996.
10. G. H. John. Robust decision trees: Removing outliers from databases. In *Proc. of the First International Conference on Knowledge Discovery and Data Mining (KDD'95)*, pages 174–179. AIII Press, 1995.
11. J. Kubica and A. Moore. Probabilistic noise identification and data cleaning. Technical Report CMU-RI-TR-02-26, CMU, 2002.
12. T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
13. R. R. Muntz and Y. Xia. Mining frequent itemsets in uncertain datasets. In *ICDM'03 Workshop on Fundations and New Directions in Data Mining*, November 2003.
14. K. P. Murphy. The bayes net toolbox for matlab. *Computing Science and Statistics*, 2001, http://www.ai.mit.edu/˜murphyk/Software/BNT/bnt.html.
15. J. R. Quinlan. Simplifying decision trees. *International Journal of Man-Machine Studies*, 27(3):221–234, 1987.
16. S. Schwarm and S. Wolfman. Cleaning data with bayesian methods, 2000.
17. C. M. Teng. *Correcting Noisy Data*. Machine Learning, 1999.
18. B. Thiesson. Accelerated quantification of bayesian networks with incomplete data. In *Proc. of the First International Conference on Knowledge Discovery and Data Mining (KDD'95)*, pages 306–311. AIII Press, 1995.

19. A. Wendemuth. Modeling uncertainty of data observation. In *Proc. of Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2001)*, pages 296–299, 2001.
20. J. Yang, W. Wang, P. S. Yu, and J. Han. Mining long sequential patterns in a noisy environment. In *Proc. of the ACM SIGMOD Conference on Management of Data 2002 (SIGMOD'02)*, June 2002.