Mining With Noise Knowledge: Error-Aware Data Mining

Xindong Wu and Xingquan Zhu

***Abstract*—Real-world data mining deals with noisy information sources where data collection inaccuracy, device limitations, data transmission and discretization errors, or man-made perturbations frequently result in imprecise or vague data. Two common practices are to adopt either data cleansing approaches to enhance the data consistency or simply take noisy data as quality sources and feed them into the data mining algorithms. Either way may substantially sacrifice the mining performance. In this paper, we consider an error-aware (EA) data mining design, which takes advantage of statistical error information (such as noise level and noise distribution) to improve data mining results. We assume that such noise knowledge is available in advance, and we propose a solution to incorporate it into the mining process. More specifically, we use noise knowledge to restore original data distributions, which are further used to rectify the model built from noise-corrupted data. We materialize this concept by the proposed EA naive Bayes classification algorithm. Experimental comparisons on real-world datasets will demonstrate the effectiveness of this design.**

***Index Terms*—Classification, data mining, naive Bayes (NB), noise handling, noise knowledge.**

I. Introduction

REAL-WORLD data are dirty, and therefore, noise handling is a defining characteristic for data mining research and applications. A typical data mining application consists of four major steps: data collection and preparation, data transformation and quality enhancement, pattern discovery, and interpretation and evaluation of patterns (or postmining processing) [22], [30]. In the Cross Industry Standard Process for Data Mining framework [31], [45], this process is decomposed into six major phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. It is expected that the whole process starts with raw data and finishes with the extracted knowledge. Because of its data-driven nature, previous research efforts have concluded that data mining results crucially rely on the quality of the underlying data, and for most of the data mining applications, the process of data collection, data preparation, and data enhancement cost the majority of the project budget and also the developing time circle [2], [31]. However, data imperfections, such as erroneous or inaccurate attribute values, still commonly exist in practice, where data often carry a significant amount of errors, which will have negative impact on the mining algorithms [1]. In addition, existing research on privacy-preserving data mining [3], [4] often uses intentionally injected errors, which are commonly referred to as data perturbations, for privacy-preserving purposes, such that sensitive information in data records can be protected, but knowledge in the dataset is still available for mining. As these systematic or man-made errors will eventually deteriorate the data quality, conducting effective mining from data imperfections becomes a challenging and real issue for the data mining community.

Take the problem of supervised learning as an example, where the task is to form decision theories that can be used to classify previously unlabeled (test) instances accurately. In order to do so, a learning set D which consists of a number of training instances, i.e., (x_n, y_n), $n = 1, 2, \ldots, N$, is given in advance, from which the learning algorithm can construct a decision theory. Here, each single instance (x_n, y_n) is characterized by a set of M attribute values $x_n = \langle a_1, a_2, \ldots, a_M \rangle$ and one class label $y_n, y_n \in \{c_1, c_2, \ldots, c_L\}$ (the notation of all the symbols is explained in Table I). The problems of data imperfections rise from the reality that attribute values x_n and class label y_n might be corrupted and contain incorrect values. Under such circumstances, incorrect attribute values and mislabeled class labels thus constitute attribute and class noises. Extensive research studies have shown that the existence of such data imperfections is mainly responsible for inferior decision theories [1], [23], [24], and eliminating highly suspicious data items often leads to an improved learner [5], [6], [25], [26] (compared with the one learned from the original noisy dataset), because of the enhanced data consistency and less confusion among the underlying data. Such elimination approaches are commonly referred to as data cleansing [7], [27], [28]. Data cleansing methods are effective in many scenarios, but some problems are still open.

1) **Data cleansing only takes effect on certain types of errors**, such as class noise. Although it has been demonstrated that cleansing class noise often results in better learners [5], [6], for datasets containing attribute noise or missing attribute values, no evidence suggests that data cleansing can lead to improved data mining results.

Manuscript received April 25, 2006; revised February 25, 2007. This work was supported by the National Science Foundation of China under Grant 60674109. An earlier version of this paper was published in the Proceedings of the 2006 IEEE International Conference on Granular Computing (GRC), Atlanta, GA, 2006. The new content added here, compared with the conference version of this paper, includes Sections 3.4, 4.4, and 5.2, Tables 1 and 2, and numerous revised paragraphs. This paper was recommended by Associate Editor J. Wu.

X. Wu is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China, and also with the Department of Computer Science, University of Vermont, Burlington, VT 05405 USA (e-mail: xwu@cs.uvm.edu).

X. Zhu is with the Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: xqzhu@cse.fau.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2008.923034

TABLE I
SYMBOLS USED IN THIS PAPER

Symbol	*Description*
X	A vector of random variables denoting the observed attribute values of an instance
Y	A random variable for the observed class label of an instance
N	The number of instances in the training set
M	The number of attributes in the dataset
L	The number of class labels in the dataset
c_l, l=1, 2, .., L	The l^{th} class label in the dataset
a_i, i=1,…,M	The i^{th} attribute of the dataset
M_i, i=1,…,M	The number of attribute values of the i^{th} attribute a_i
$a_{i,j}$, j=1,..,M_i; i=1,..,M	One particular value of attribute a_i (here we consider nominal attributes; for numerical attributes, we apply discretization beforehand)
$I_n = (x_n, y_n)$	An instance I_n with attribute x_n and class label y_n
$x_n = <a_1, a_2, \ldots, a_M>$	The attribute values of instance I_n
$y_n \in \{c_1, c_2, \ldots, c_L\}$	The class label of instance I_n

2) **Data cleansing cannot result in perfect data.** As long as errors continuously exist in the data, they will most likely deteriorate the mining performance in some ways (although exceptions do exist). Consequently, the need for developing error-tolerant data mining algorithms has been a major concern in the area [2], [24], [29].
3) **Data cleansing cannot be unconditionally applied to any data sources.** For intentionally imposed errors, such as privacy-preserving data mining, data cleansing cannot be directly applied to cleanse the imputated (noisy) data records because privacy-preserving data mining intends to hide sensitive information by data randomization. Applying data cleansing to such data could lead to information loss and severely deteriorate the final results.
4) **Eliminating noisy data items may lead to information loss.** Just because a noisy instance contains erroneous attribute values or an incorrect class label, it does not necessarily mean that this instance is completely useless and therefore needs to be eliminated from the database. More specifically, it might be true that eliminating class noise from the training dataset is often beneficial for an accurate learner [1], but for erroneous attribute values, we may not simply eliminate a noisy instance from the dataset since other correct attribute values of the instance may still contribute to the learning process.
5) **The traditional data mining framework** [22], [30], [31] **(without error awareness) isolates data cleansing from the actual mining process.** Under a cleansing-based data mining framework, data cleansing and data mining are two isolated independent operations and have no intrinsic connections between them. Therefore, a data mining process has no awareness of the underlying data errors.

In addition to data cleaning, many other methods, such as data correction [7] and data editing [8], have also been used to correct suspicious data entries and enhance data quality. Data imputation [9], [32]–[36] is another body of work which fills in missing data entries for the benefit of the subsequent pattern discovery process. It is obvious that data cleansing, correction, or editing all try to polish the data before they are fed into the mining algorithms. The intuition behind such operations is straightforward. Enhancing data consistency will consequently improve the mining performance. Although this intuition has been empirically verified by numerous research efforts [6], [7], [28], [40], in reality, new errors may be introduced by data polishing, and correct data records may also be falsely cleansed, which lead to information loss (as demonstrated in [6, Table 18], where cleansed data lead to inferior decisions). As a result, for applications like medical or financial domains, users are reluctant to apply such tools to their data directly, unless the process of data cleansing/correction is under a direct supervision of domain experts, or a copy of the original data is kept separately, such that there is always a chance to turn back to the original data [39].

On the other hand, all previous efforts on data cleansing, editing, and correction have been primarily focused on enhancing the data quality for the benefit of the subsequent mining process, and little attention has been paid to address the challenge of unifying data quality and data mining to achieve an improved mining result. In other words, if we can make data mining algorithms aware of the underlying data errors, the mining process may adjust and rectify the model produced from the noisy data. This, however, raises two nontrivial concerns: 1) What kind of data quality or error information is available for data mining? and 2) how can such information be integrated into the mining process?

It is obvious that instance-based error information (i.e., information about which instance and/or which attribute values of the instance are incorrect) is difficult to get and unavailable with trivial endeavors, although a substantial amount of research has been trying to address this issue from different perspectives. However, there are many cases in reality that statistical error information of the whole database is known *a priori*.

1) **Information transformation errors.** Information transformation, particularly wireless networking, often raises a certain amount of errors in communicated data. For error control purposes, the statistical errors of the signal transmission channel should be investigated in advance and can be used to estimate the error rate in the transformed information.
2) **Device errors.** When collecting information from different devices, the inaccuracy level of each device is often

available, as it is part of the system features. For example, fluorescent labeling for gene chips in microarray experiments usually contains inaccuracy caused by sources such as the influence of background intensity [10]. The values of collected gene chip data are often associated with a probability to indicate the reliability of the current value.

3) **Data discretization errors**. Data discretization is a general procedure of discretizing the domain of a continuous variable into a finite number of intervals [46], [47]. Because this process uses a certain number of discrete values to estimate infinite continuous values, the difference between the discrete value and the actual value of the continuous variable thus leads to a possible error. Such discretization errors can be measured in advance and, therefore, are available for a data mining procedure [41].
4) **Data perturbation errors.** As a representative example of artificial errors, privacy-preserving data mining intentionally perturbs the data; thus, private information in data records can be protected, but knowledge conveyed in the datasets is still minable. In such cases, the level of errors introduced is certainly known for data mining algorithms [3], [4], [42].

The availability of the aforementioned statistical error information directly leads to the question of how to integrate such information into the mining process. Most data mining methods, however, do not accommodate such error information in their algorithm design. They either take noisy data as quality sources or adopt data cleansing beforehand to eliminate and/or correct the errors. Either way may considerably deteriorate the performance of the succeeding data mining algorithms because of the negative impact of data errors and the limitations and practical issues of data cleansing. The aforementioned observations raise an interesting and important concern on error-aware (EA) data mining, where **previously known error information (or noise knowledge) can be incorporated into the mining process for improved mining results**.

In this paper, we report our recent research efforts toward this goal. We will propose an EA data mining framework which accommodates noise knowledge to enhance data classification accuracy. By using naive Bayes (NB) classification to materialize our idea, our experimental results on real-world datasets from University of California, Irvine (UCI) machine learning database repository [20] will demonstrate that such an EA data mining procedure is superior to cleansing-based data mining and can significantly improve data mining results in noisy environments when the statistical error information is provided beforehand.

II. Related Work

Mining with noisy data has always been an active research topic for data mining [5]–[9], [13], [14], [23], [24], [26], [28]. As most data mining algorithms crucially depend on the quality of their input data to produce reliable models, a general consensus among data mining practitioners is that low-quality data often lead to wrong decisions or even ruin the projects ("garbage in, garbage out") [2]. In supervised learning, noise usually takes two forms, namely, class noise and attribute noise, depending on whether the errors (inconsistency, contradiction, or missing values) are introduced to the class label or the attributes [1]. Existing endeavors from data preprocessing [5]–[7] and data quality [11] perspectives have come up with many solutions such as class noise identification [5], [6], erroneous attribute value location [12] and correction [7], missing attribute value imputation [8], [9] and acquisition [13], and editing training instances for instance-based learning [14]. An essential goal of all these efforts is to enhance the quality of the training data so that it can possibly benefit the mining process. Although data cleansing is a very useful tool, in practice, it has to be carefully applied to the data, as careless data cleansing may deteriorate the mining performance, due to reasons such as information loss incurred by incorrect data elimination, correction, or editing. For example, in the class noise elimination approach developed by Brodley and Friedl [6], it is possible that a cleansed dataset may lead to an inferior learner (in [6, Table 18], a learner built from the cleansed dataset is inferior to the one from the original noisy dataset). Similarly, in data warehousing applications, one popular data cleansing technique is data merge and purge [40] which identifies and removes duplicated data records from databases such that they may not bias an analysis. However, a recent report [39] suggests that "in many cases the records in the two databases may include some information that is unique to each, so just deleting one of the duplicates is not always a good option as it can lead to valuable data loss." Therefore, "it is best to add corrections to the database while retaining the original data in a separate field or fields so that there is always the chance of going back to the original information" [39].

It is true that no data processing effort can result in perfect data, and in reality, most algorithms would still have to conduct knowledge discovery from noisy sources, regardless of whether they have noise-handling mechanisms or not. The problem of learning in noisy environments has been the focus of much attention in data mining, and most inductive learning algorithms have a mechanism for noise handling. For example, pruning in decision trees is designed to reduce the chance that the trees are overfitting to noise [15]. A common practice in reducing the noise impact is to adopt some thresholding measures to remove poor knowledge drawn from noisy data. Although simple, this mechanism has generated very impressive results. For example, Integrative Windowing [17] adopts good rule selection criteria to reduce noise impact, and instance-based learning algorithms [18] select representative prototypes to remove poor training samples. It is clear that in these algorithm designs, the existence of noise has been taken into consideration, but they still follow the same direction as data cleansing and have not realized that noise knowledge, if carefully utilized, can be beneficial for the mining process. Different from the existing research efforts, our objective is to let a data mining process be aware of the underlying data errors and make use of this information instead of simply removing or correcting data errors.

Recent research in privacy-preserving data mining has raised an issue of perturbing data entries to protect privacy and maintain data mining performance, where randomization is a popular mechanism for this purpose. The intuition behind it is to intentionally introduce errors (often in the form of randomness)

into sensitive data entries and to reveal randomized information about each record in exchange for not having to reveal the original records to anyone [3], [4]. Although the data records were modified, the imposed randomness was controlled so that knowledge in the dataset is still minable, with a little sacrifice in performance. Such a randomization procedure requires a compromise between the levels of privacy that the system tries to protect and the mining performance from the perturbed data.

1) As randomization can eventually ruin useful knowledge in the dataset, the level of perturbations should be well controlled to avoid making a perturbed dataset totally useless.
2) The distribution of the errors is available for both database managers and data mining practitioners, as without this information, the mining results would deteriorate significantly. For example, normally distributed data perturbations are often adopted for numerical attributes.

For categorical attributes, Du and Zhan [4] adopted randomized response techniques to scramble the original data entries for privacy-preserving data mining. This method assumes that attributes contain binary values only, and the values are collected in such a way that the information providers tell the truth about all their answers to sensitive questions with the probability θ and that they tell the lie about all their answers with the probability $1-\theta$. For example, if the original attribute values were $a_1 = 1$, $a_2 = 1$, and $a_3 = 0$, then there are θ chances that all these values remain unchanged (users telling the truth), and there are $1-\theta$ chances that all values were flipped to $a_1 = 0$, $a_2 = 0$, and $a_3 = 1$ (users lying). The assumption of this approach, however, is too strong as it assumes that once users decided to lie, they will lie on all questions, which is hardly the case in practice. In fact, users may randomly tell the truth or lie on each single question, i.e., lying on attribute a_1 but telling the truth on attribute a_2 or vice versa. Thus, the perturbation introduced to each attribute (question) may be totally independent. In our system, we consider realistic cases where perturbations are randomly and independently introduced to each attribute.

III. EA Data Mining for NB

A. NB Classification

In supervised learning, each instance is described by a vector of attribute values. A set of instances with their classes are provided as the training data, where each instance in the training data is denoted by a vector of attribute values x_k and a class label y_k, i.e., $I_k = (x_k, y_k)$. Given a test instance I_n with an unknown class label y_n, i.e., $I_n = (x_n, ?)$, the learner is asked to predict I_n's class label according to the evidence provided by the training data.

By assuming that $P(Y = c_l|I_n)$ denotes the probability that example I_n belongs to class c_l, the Bayes theorem can be used to optimally predict the class label of a previously unseen example I_n, given a set of training examples in advance. According to Bayes theorem, the expected classification error can be minimized by choosing the maximal posterior probability, i.e., $\arg\max_l\{P(Y = c_l|I_n)\}$. Given an example I_n, the Bayes theorem provides a method to compute $P(Y = c_l|I_n)$ with

$$P(Y = c_l|I_n) = \frac{P(Y = c_l) \cdot P(X = x_n|Y = c_l)}{P(X = x_n)}. \quad (1)$$

By assuming that attributes $x_n = \langle a_1, a_2, \ldots, a_M\rangle$ are conditionally independent given the class label, the conditional probability in (1), i.e., $P(X = x_n|Y = c_l)$, can be decomposed into the product $P(a_1|c_l) \times P(a_2|c_l) \times \cdots \times P(a_M|c_l)$, where M is the number of attributes of the dataset. Then, the probability that an example I_n belongs to class c_l is given by

$$P(Y = c_l|I_n) = \frac{P(Y = c_l) \cdot \prod_{i=1}^{M} P(X = a_i|Y = c_l)}{P(X = x_n)} \quad (2)$$

which can be rewritten as (3) to avoid the product of the conditional probability $\prod_{i=1}^{M} P(X = a_i|Y = c_l)$ quickly vanishing to zero.

$$P(Y = c_l|I_n) \propto \log\left(P(Y = c_l)\right) + \sum_{i=1}^{M} \log\left(P(X = a_i|Y = c_l)\right). \quad (3)$$

The (naive and strong) conditional independence assumption renders a classifier, i.e., an NB classifier, obtained by using the discriminant function in (2). The conditional independence assumption embodied in (2) makes the NB classifiers very efficient for large datasets because an NB classifier does not use attribute combinations as a predictor and can be constructed by only one scan of the dataset with a linear time complexity. Although the assumption of conditional independence among attributes is often violated in reality, the classification performance of NB is surprisingly good compared with other more complex classifiers [19], particularly when dealing with noisy datasets [37] (in Section IV-B, we will also demonstrate that, on average, NB is indeed more robust to data errors compared with C4.5 decision trees [15]). Because of these nice features, NB has been popularly applied to solve many real-world problems, including text classification [43] and mining housekeeping genes from biological data [38].

In noisy environments, erroneous attribute values will change conditional probabilities $P(X|Y = c_l)$, $l = 1, \ldots, L$, and then deteriorate NB's performance. The objective of EA data mining for NB classification is to let the NB be aware of the overall characteristics of the underlying data errors and then attempt to restore the original conditional probabilities and improve an NB classifier. In the case that errors exist in the class label as well, the same approach should be adopted to restore *priori* probability $P(Y = c_l)$, $l = 1, \ldots, L$.

Throughout this paper, error and noise are two equivalent terms that we use to denote erroneous (incorrect) values of the training examples, and errors or noise that we define here do not include missing values.

B. Data Distribution Restoration for NB

Assume that the previously known noise level in attribute a_i is denoted by p_i and that noise in each attribute is uniformly

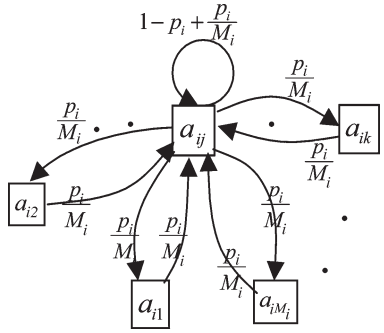

Fig. 1. Random value transformation for the attribute value a_{ij}.

distributed. A noise level p_i indicates that for any particular attribute value, for example, a_{ij}, it has a p_i probability of being randomly corrupted to any other values $a_{i1}, \ldots, a_{iM_i}$, including itself. Thus, for any two values a_{ij} and a_{ik}, a_{ij} has a p_i/M_i probability of being changed to a_{ik} and vice versa. Such a random transformation model for attribute value a_{ij} is shown in Fig. 1.

Given a dataset D with $|D|$ instances, assume that it was corrupted from an error-free dataset E (which does not exist). Let $|D_{ij}|$ and $|E_{ij}|$ denote the numbers of instances in D and E, respectively, which contain the attribute value a_{ij}. When noise is uniformly distributed, as shown in Fig. 1, for any attribute a_i, the relationship between $|D_{ij}|$ and $|E_{ij}|$, $j = 1, 2, \ldots, M_i$, can be expressed as

$$\begin{aligned}
|D_{i1}| &= |E_{i1}| \cdot \left(1 - p_i + \frac{p_i}{M_i}\right) + \cdots + |E_{ij}| \cdot \frac{p_i}{M_i} + \cdots \\
&\quad + |E_{i_{M_i}}| \cdot \frac{p_i}{M_i} \\
&\cdots \\
|D_{ij}| &= |E_{i1}| \cdot \frac{p_i}{M_i} + \cdots + |E_{ij}| \cdot \left(1 - p_i + \frac{p_i}{M_i}\right) + \ldots \\
&\quad + |E_{i_{M_i}}| \cdot \frac{p_i}{M_i} \\
&\cdots \\
|D_{i_{M_i}}| &= |E_{i1}| \cdot \frac{p_i}{M_i} + \cdots + |E_{ij}| \cdot \frac{p_i}{M_i} + \cdots \\
&\quad + |E_{i_{M_i}}| \cdot \left(1 - p_i + \frac{p_i}{M_i}\right). \qquad (4)
\end{aligned}$$

Equation (4) can be written in a matrix form as

$$A \cdot X = B \qquad (5)$$

where

$$A = \begin{bmatrix} 1 - p_i + \frac{p_i}{M_i} & \cdots & \frac{p_i}{M_i} & \cdots & \frac{p_i}{M_i} \\ \cdots & & & & \\ \frac{p_i}{M_i} & \cdots & 1 - p_i + \frac{p_i}{M_i} & \cdots & \frac{p_i}{M_i} \\ \cdots & & & & \\ \frac{p_i}{M_i} & \cdots & \frac{p_i}{M_i} & \cdots & 1 - p_i + \frac{p_i}{M_i} \end{bmatrix} \quad B = \begin{bmatrix} |D_{i1}| \\ \cdots \\ |D_{ij}| \\ \cdots \\ |D_{i_{M_i}}| \end{bmatrix}.$$

As we hold the corrupted dataset D, we can easily calculate the value of $|D_{ij}|$, $j = 1, \ldots, M_i$ (the number of instances in D which contain the attribute value a_{ij}). Because noise level p_i is known as well, (5) is just a set of linear functions consisting of M_i variables $|E_{ij}|$, $j = 1, \ldots, M_i$, and M_i functions, which can be easily solved to estimate $|E_{ij}|$, i.e., the number of instances in E which contain the attribute value a_{ij}.

The results from (5) can only estimate the number of instances with respect to (w.r.t.) each attribute value, regardless of the class label. This information is not sufficient to solve our problem, as NB needs to estimate the conditional probability given a particular class $Y = c_l$, $P(X|Y = c_l)$. For this purpose, we transform (5) by pushing constraints onto the class labels.

By assuming that the number of instances in E, which contain the attribute value a_{ij} and the class label c_l, is denoted by $|E_{ij}^{c_l}|$, and the same type of instances in the corrupted dataset D is denoted by $|D_{ij}^{c_l}|$, (5) can be rewritten as follows:

$$\begin{aligned}
&A \cdot (X_1 + X_2 + \cdots + X_l + \cdots + X_L) \\
&\qquad = B_1 + B_2 + \cdots + B_l + \cdots + B_L \qquad (6)
\end{aligned}$$

where

$$\begin{aligned}
X_l &= [\,|E_{i1}^{c_l}| \quad |E_{i2}^{c_l}| \quad \ldots \quad |E_{iM_i}^{c_l}|\,] \\
B_l &= [\,|D_{i1}^{c_l}| \quad |D_{i2}^{c_l}| \quad \ldots \quad |D_{iM_i}^{c_l}|\,]^{\mathrm{T}} \\
B &= B_1 + B_2 + \cdots + B_L.
\end{aligned}$$

Equation (6), however, is unsolvable, as there are $L \cdot M_i$ variables $(|E_{ij}^{c_l}|, l = 1, \ldots, L, j = 1, \ldots, M_i, i \in [1, M])$ but M_i functions only, although we know exactly the values of $B_1, \ldots, B_L$. An alternative is to decompose (6) into a series of linear functions associated to each single class, as denoted by

$$\begin{cases} A \cdot X_1 = B_1 \\ A \cdot X_2 = B_2 \\ \cdots \\ A \cdot X_L = B_L \\ X_1 + X_2 + \cdots + X_L = X. \end{cases} \qquad (7)$$

The rationale of (7) lies in the assumption that errors are randomly and independently distributed across all attributes; thus, instances in each class suffer from almost the same level of errors. Once the number of instances is large enough, estimating the attribute value distribution from an instance subset or from the whole dataset does not bring much difference. In the case that errors exist in the class label as well, the decomposed equations in (7) may still hold, as long as noise is randomly distributed across all classes.

Since (7) estimates the attribute value distributions w.r.t. each class, it may possibly result in higher estimation errors for classes with a very limited number of examples. Considering a binary-class dataset where the class containing the least number of instances is defined by the minority class and the other class is defined by the majority class, because the minority class has a very limited number of instances, it is hard to assess whether noise in this small number of instances indeed complies with the transformation model in Fig. 1. As a result, for the minority class, the estimated values $(|E_{ij}^{c_l}|, l = 1, \ldots, L, j = 1, \ldots, M_i)$ can be seriously biased. Although we cannot do much to improve this shortfall, NB has inherently accommodated this issue. In (1), the *priori* probability $P(c_l)$ also takes part in the final decision, and the final decision error is the product between

Procedure: ErrorAwareNaiveBayesClassification()

Input: (1) D (a noisy dataset); (2) p_i, i=1,.., M (noise level for each attribute)

Output: Polished Naive Bayes model.

(1). **For** each class c_l, l=1, …, L

(2). Calculate class priori probability $P(c_l)$

(3). **For** attribute a_i, i=1, …, M

(4). Calculate attribute distribution values, $|D_{ij}^{c_l}|$, j=1, .., M_i from D

(5). Solve Eq. (7) and acquire estimated $|E_{ij}^{c_l}|$, j=1,.., M_i.

(6). **End For**

(7). **End For**

(8). Take estimated $|E_{ij}^{c_l}|, j$=1, .., M_i; i=1, .., M; l=1, …, L, as conditional probabilities, and combine with priori probability $P(c_l)$ to form an error aware Naive Bayes classifier.

Fig. 2. EA-NB classification model.

the bias of the conditional probability $\text{Bias}(P(X|Y=c_l))$ and the *priori* probability $P(c_l)$. Although classes with a small number of training examples may have a larger $\text{Bias}(P(X|Y=c_l))$, they actually have less $P(c_l)$. Consequently, the bias from the minority classes can be controlled and should not bring a large impact to the final results.

C. EA-NB Classification

With the aforementioned analysis, we can estimate the value of the original attribute distribution $(|E_{ij}^{c_l}|, l=1,\ldots,L, j=1,\ldots,M_i, i=1,\ldots,M)$, w.r.t. the constraint of each class c_l, $l=1,\ldots,L$. This value can be directly used as the conditional probability $P(X|Y=c_l)$. As NB assumes that all attributes are conditionally independent given the class label, we can repeat the same process for each attribute and use the estimated conditional probabilities for final classification. The pseudocode of the whole algorithm is shown in Fig. 2.

Because the EA-NB model in Fig. 2 processes each attribute independently, errors inside each attributes are also considered independently. In other words, even if different attributes suffer from different levels of errors, EA-NB would still be able to restore data distributions for each of them.

D. Data Distribution Restoration From a General Transformation Model

The transformation model in Fig. 1 assumes that errors are uniform across all attribute values, and data distributions are consequently restored from a similar model to solve our problem. In reality, errors can be biased toward some specific attribute values, i.e., some attribute values tend to be more error prone than others. We illustrate, in this section, that as long as the statistical error information across different attribute values is known in advance, EA-NB can still be applied to solve the problem. More specifically, given an attribute a_i with M_i values $a_{i1}, a_{i2}, \ldots, a_{iM_i}$, assuming that errors across the M_i attribute values are nonuniform and that an attribute value a_{ij} has q_{jk}^i probability of being corrupted as another value a_{ik}, as shown in Fig. 3, then for any particular attribute value a_{ij}, its probability of remaining the same value unchanged is $q_{jj}^i = 1-(q_{j1}^i+\cdots+q_{jk}^i+\cdots+q_{jM_i}^i), k\neq j$. A general transmission matrix q^i for attribute a_i can then be denoted by (8), where the diagonal elements denote the probabilities that an attribute value remains the same value unchanged.

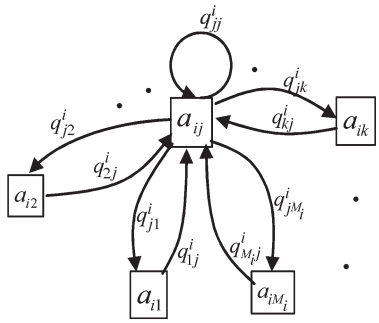

Fig. 3. General transformation model for the attribute value a_{ij}.

$$q^i = \begin{bmatrix} q_{11}^i,\ldots,q_{1k}^i,\ldots,q_{1M_i}^i \\ \ldots \\ q_{k1}^i,\ldots,q_{kk}^i,\ldots,q_{kM_i}^i \\ \ldots \\ q_{M_i1}^i,\ldots,q_{M_ik}^i,\ldots,q_{M_iM_i}^i \end{bmatrix} \tag{8}$$

where

$$q_{jj}^i = 1-\left(q_{j1}^i+\cdots+q_{jk}^i+\cdots+q_{jM_i}^i\right), \quad j=1,\ldots,M_i,\ k\neq j. \tag{9}$$

Considering (8) and the data restoration model in (4), we can easily derive a data restoration model, as denoted by

$$\begin{aligned}
&|D_{i1}| = |E_{i1}| \cdot q^i_{11} + \cdots + |E_{ij}| \cdot q^i_{1j} + \cdots + |E_{i_{M_i}}| \cdot q^i_{1M_i} \\
&\cdots \\
&|D_{ij}| = |E_{i1}| \cdot q^i_{j1} + \cdots + |E_{ij}| \cdot q^i_{jj} \cdots + |E_{i_{M_i}}| \cdot q^i_{jM_i} \\
&\cdots \\
&|D_{i_{M_i}}| = |E_{i1}| \cdot q^i_{M_i1} + \cdots + |E_{ij}| \\
&\qquad \cdot q^i_{M_{ij}} + \cdots + |E_{i_{M_i}}| \cdot q^i_{M_iM_i}. \qquad (10)
\end{aligned}$$

Because the statistical error information is assumed to be available in advance, (10) can be transformed into a set of linear functions as the ones shown in (5) [the difference is that matrix A is now given by (8)]. Thus, the approach that we discussed in Section III-B can still be used to solve the problem. However, different from the error model in Fig. 1, the model in Fig. 3 indicates that the overall error rate for attribute a_i is determined by the error rate w.r.t. each attribute value a_{ij}, along with the percentage of instances having different attribute values. From (10), we can directly estimate the number of instances w.r.t. each attribute value a_{ij}, $|E_{ij}|$. Based on all estimated values, we can denote the overall error rate of attribute a_i, p_i, by (11), where N is the total number of instances in the training set, and $q^i_{jj} = 1 - (q^i_{j1} + \cdots + q^i_{jk} + \cdots + q^i_{jM_i})$, $j = 1, \ldots, M_i, k \neq j$. The expression p_i is given in (11), shown at the bottom of the page.

The model in (8) can be reasonably relaxed by assuming that transformation probability values between two attribute values a_{ij} and a_{ik} are symmetric, i.e., $q^i_{jk} = q^i_{kj}$, which leads to a symmetric q^i matrix. In Section IV-D, we will report experimental results from the general transformation model and assess the algorithm performance under circumstances such as a data transformation model is slightly and severely inaccurate.

IV. Experimental Evaluations

To evaluate the performance of the proposed EA-NB classification design, we implemented both NB and EA-NB. In our implementation, most NB classifications are based on the discriminant function in (2), and in the case that class distributions of the dataset become undistinguishable (e.g., when datasets have many attributes), we use (3) instead. We evaluate our approach on ten benchmark datasets from the UCI database repository [20], where each numerical attribute is discretized with equal-width discretization approach. Although other complex discretization methods [46], such as equal frequency or supervised discretization algorithms [47], are reported to have a better performance than equal-width discretization, since we are interested in the relative improvement of EA-NB in comparison with NB, we believe that the impact of the inferior discretization model can be ignored as long as we are using the same discretization method for both EA-NB and NB. For this reason, we chose a simple equal-width discretization method instead where each attribute was discretized into a fixed number of bins (ten bins in our experiments).

The main characteristics of our benchmark datasets and their tasks are described in Table II. The number of instances in these datasets varies from about 100 (zoo) to about 50 000 (adult), and the number of attribute values varies from 6 (car) to 60 (splice), which gives us a chance to observe EA-NB's performance from different perspectives, e.g., the performance on very sparse to relatively dense datasets.

The datasets in the UCI database repository have been carefully examined by domain experts; thus, they do not contain much noise (at least we do not know which instances and which attribute values are erroneous). For comparative studies, we adopt both the random corruption model in Fig. 1 and the general transformation model in Fig. 3 to manually inject errors into the attributes, and then, we observe the performance of different methods on corrupted datasets. For simplicity, majority of the experimental results are based on the random corruption model in Fig. 1 (thus, we do not need to take care of the transformation probabilities across different attribute values but to simply specify a noise level value for each attribute). For a more extensive comparison, in Section IV-D, we will also report the general transformation model-based (Fig. 3) results from two datasets.

With random corruption model in Fig. 1, given a noise level p_i, an attribute value a_{ij} has a chance of p_i to be changed to any other random value (including itself). Thus, the actual noise level in a_i is $p_i - p_i/M_i$, which is always lower than the designed value. With the same noise level p_i, the more the number of attribute values, the higher the overall noise level in the attribute. As we assume that noise is uniformly distributed among all attribute values, it would bring a much smaller impact on attributes with a large number of values than those that have, for example, only two attribute values. On the other hand, to estimate the original attribute distribution from a corrupted dataset, a higher estimation accuracy is expected from the attribute with a smaller number of attribute values compared with other attributes in the same dataset. The reason is that when the number of attribute values increases, it will be more and more difficult to generate a truly random distribution across all values, given a limited number of training examples. Thus, the noisy dataset might be biased and does not comply with the intended error distributions. We will discuss this part of the results in Section IV-A.

With the general transformation model in Fig. 3, the actual noise level in an attribute a_i is determined by (11). In our experimental results in Section IV-D, we follow (8) and control

$$p_i = \frac{|E_{i1}| \cdot \left(1 - q^i_{11}\right) + \cdots + |E_{ij}| \cdot \left(1 - q^i_{jj}\right) + \cdots + |E_{iM_i}| \cdot \left(1 - q^i_{M_iM_i}\right)}{N} \qquad (11)$$

TABLE II
BENCHMARK DATASETS USED FOR EVALUATION

Dataset	# of Classes	# of Attributes		# of Instances	Simple description
		Nominal	Continuous		
Adult	2	8	6	48,842	Predicting whether a person's income exceeds $50K/yr based on census data
Car	4	6	0	1,728	Predicting a car into one of the four categories: unacceptable, acceptable, good, v-good, by using six input attributes: buying, maint, doors, persons, lug_boot, safety.
Glass	7	0	9	214	The classification of 6 types of glasses defined in terms of their oxide content.
Krvskp	2	36	0	3,196	Predicting "white-can-win" or "white-cannot-win" by using board-descriptions for each chess endgame.
Led24	10	24	0	1,000	Predicting 10 digital numbers (0-9) by using the value of 7 LED light-emitting diodes plus 17 random attributes.
Mushroom	2	22	0	8,124	Classifying whether a mushroom is poisonous or edible by using its physical characteristics.
Splice	3	60	0	3,190	Given a sequence of DNA, predict the boundaries between exons (the parts of the DNA sequence retained after splicing) and introns (the parts of the DNA sequence that are spliced out)
Nursery	5	8	0	12,960	Ranking applications for nursery schools (no-recommend, recommend, …, special-priority) based on an applicant's background information
Wine	3	0	13	178	A chemical analysis of wines grown in the same region in Italy but derived from three different cultivars (based on the quantities of 13 constituents found in each of the three types of wines)
Zoo	7	15	2	101	Classifying animals into 1 of the 5 categories

TABLE III
CLASSIFICATION ACCURACY COMPARISON

Dataset	p_i (%)	*Original*		*Corrupted*		*EA_NB*	*p-value (EA_NB vs Corrupted NB)*	*Cleansing NB*
		NB	*C4.5*	*NB*	*C4.5*			
Adult	10	81.67	84.78	80.79	83.94	80.20	0.4E-00	77.94
	30			79.41	80.25	80.29	**4.6E-02**	77.92
	50			78.92	74.33	80.08	**3.4E-03**	78.66
Krvskp	10	87.93	99.64	86.17	96.53	87.61	**1.3E-02**	79.40
	30			82.31	82.62	86.47	**5.8E-03**	79.38
	50			79.67	71.07	85.09	**6.5E-04**	78.74
LED24	10	100.0	100.0	100.0	94.41	100.0	0.5E-00	100.0
	30			98.97	78.62	99.96	**2.3E-02**	99.98
	50			93.62	55.59	98.68	**3.7E-03**	96.00
Nursery	10	92.07	98.93	91.03	91.97	90.93	0.5E-00	87.05
	30			90.61	77.18	90.38	0.3E-00	87.28
	50			89.44	63.71	90.52	0.2E-00	87.23
Wine	10	94.96	93.58	95.02	92.61	95.01	1.2E-01	94.06
	30			92.63	90.32	94.09	**1.9E-02**	89.22
	50			88.71	85.14	92.23	**2.3E-03**	72.03
Zoo	10	96.32	92.64	90.21	92.01	94.37	**4.4E-02**	89.87
	30			89.13	87.07	91.54	**1.4E-02**	88.54
	50			84.02	76.63	86.07	**2.7E-02**	82.97

the transformation matrix and then use EA-NB to restore the original data distributions and build an NB classifier.

The majority of our experiments are designed to assess the performance of the proposed EA-NB in noisy environments in comparison with the original NB classifiers trained from the same dataset. For each experiment, we perform a tenfold cross validation ten times and use the average accuracy as the final result. In each run, the dataset is randomly (with a proportional partitioning scheme) divided into a training set and a test set. The error corruption model was applied to the training set, and this corrupted dataset was used to build the NB and EA-NB classifiers. All the learners are tested on the test set to evaluate their performance. In the following sections, we will mainly analyze the results on several representative datasets. The summarized results are reported in Table III.

A. Attribute Value Distribution Estimation

We first conduct an empirical study to assess the proposed effort in estimating the original attribute value distribution $|E_{ij}^{c_l}|$, as it is the key to ensure EA-NB's success. For this purpose, we adopt the following two approaches to characterize the estimation errors: a global estimation error and an estimation

error w.r.t. the number of attribute values, which are defined in (12) and (13), respectively.

Global estimation error:

$$\text{OrgGlbErr} = \frac{1}{L \cdot M} \sum_{l=1}^{L} \sum_{i=1}^{M} \left(\frac{1}{M_i} \sum_{j=1}^{M_i} \left\| E_{ij}^{c_l} | - | D_j^{c_l} \right\| \right)$$

$$\text{EstGlbErr} = \frac{1}{L \cdot M} \sum_{l=1}^{L} \sum_{i=1}^{M} \left(\frac{1}{M_i} \sum_{j=1}^{M_i} \left\| E_{ij}^{c_l} | - | \tilde{E}_{ij}^{c_l} \right\| \right) \quad (12)$$

Estimation error w.r.t. the number of attribute values:

$$\text{OrgAttErr}(n) = \frac{1}{L \cdot M(n)} \sum_{l=1}^{L} \sum_{i=1}^{M} \left\{ \frac{1}{M_i} \sum_{j=1}^{M_i} \left\| E_{ij}^{c_l} | - | D_{ij}^{c_l} \right\| \right\}_{A_i \in \Omega(n)}$$

$$\text{EstAttErr}(n) = \frac{1}{L \cdot M(n)} \sum_{l=1}^{L} \sum_{i=1}^{M} \left\{ \frac{1}{M_i} \sum_{j=1}^{M_i} \left\| E_{ij}^{c_l} | - | \tilde{E}_{ij}^{c_l} \right\| \right\}_{A_i \in \Omega(n)} \quad (13)$$

where $\Omega(n) = \{A_i | M_i = n, i = 1, \ldots, M\}$, and $M(n)$ is the number of attributes in the subset $\Omega(n)$.

As defined by (11), OrgGlbErr indicates the average differences between the distributions of the clean dataset ($E_{ij}^{c_l}$) and the corrupted dataset ($D_{ij}^{c_l}$) w.r.t. the class c_l. EstGlbErr represents the average differences between the estimated distributions ($\tilde{E}_{ij}^{c_l}$) and the distribution of the clean dataset ($E_{ij}^{c_l}$). This value indicates, on average, how close is the estimated distribution to the true value. Meanwhile, we also adopt an *error w.r.t. the number of attribute values* to characterize the estimation error from the attributes with different numbers of attribute values. Given a value n, we first build a subset $\Omega(n)$, which consists of all attributes with their attribute value number equaling to n, and the number of attributes in $\Omega(n)$ is denoted by $M(n)$. EstAttErr(n) therefore calculates the estimation errors from the attributes with the same number of values. We hope that this measure can help us explore what types of attributes are more difficult for estimation.

We use mushroom dataset as our testbed. The reason is twofold: 1) The instance number in mushroom (more than 8000) is good enough to simulate the random corruption, as a small number of instances often cannot capture randomization effectively, and 2) the number of attribute values in mushroom varies from 2 to 12, which is perfect to assess the estimation error for attributes with different numbers of attribute values.

Fig. 4 shows the global estimation errors from the mushroom dataset at different noise levels p_i. As noise continuously increases, the differences between the original and the corrupted datasets (OrgGlbErr) linearly increase. With the proposed effort, we can certainly reduce the amount of errors, where the improvement could be as significant as ten times better (e.g., when $p_i = 0.2$, OrgGlbErr and EstGlbErr are equal to 0.196 and 0.019, respectively). When noise becomes radical, EstGlbErr value slightly deteriorates. One possible reason is that randomness has dominated the corrupted dataset D, and it

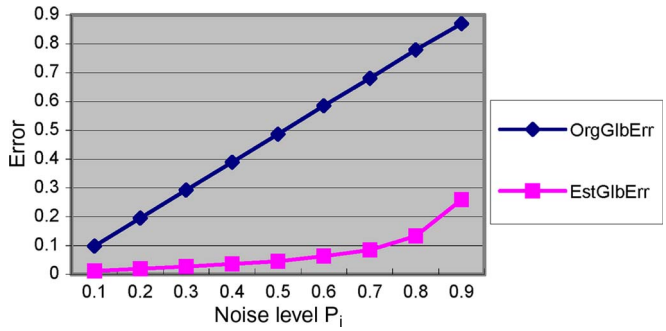

Fig. 4. Global estimation error (mushroom dataset).

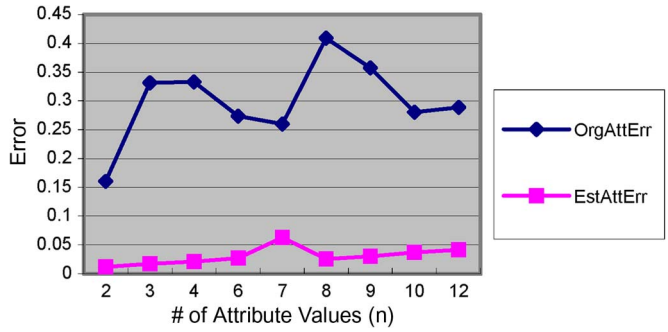

Fig. 5. Estimation error w.r.t. the number of attribute values ($P_i = 0.3$, mushroom dataset).

eventually impacts on the restoring procedures which work on D. However, even if with 90% of attribute noise, the errors of the restored distribution can still be three times less than those of the unprocessed corrupted dataset D.

We present the estimation error w.r.t. the number of attribute values in Fig. 5, where the number of attribute values varies from 2 to 12 (except 5 and 11, as the mushroom dataset does not have attributes with these two numbers of values), and the noise level p_i is set to 0.3. As we can see, the effectiveness of the proposed effort can be observed from all types of attributes, regardless of how many possible values they have. Because actual noise level in the dataset is $p_i - p_i/M_i$, theoretically, the more the number of attribute values, the higher level the noise is actually contained in the dataset, given the same noise level p_i. However, Fig. 5 shows that the impact from this factor does not appear to be significant, as OrgAttErr(n) does not show any trend of increase across all the values of n. This indicates that the variance of noise, which is caused by different numbers of attribute values, can be ignored. On the other hand, EstAttErr(n) in Fig. 5 shows a clear trend of increase as the number n becomes larger. This trend can be further verified in Fig. 6, where we compare the results at three noise levels ($p_i = 0.1$, 0.3, and 0.5). This indicates that restored distributions for attributes with a large number of attribute values can be less accurate in comparison with the attributes with a small number of possible values. As with a large number of attribute values, the corruption model has more choices in flipping the attribute value, which will ultimately bring more difficulty to the restoration process. In Figs. 5 and 6, the abnormal value from $n = 7$ was caused by the only attribute containing missing

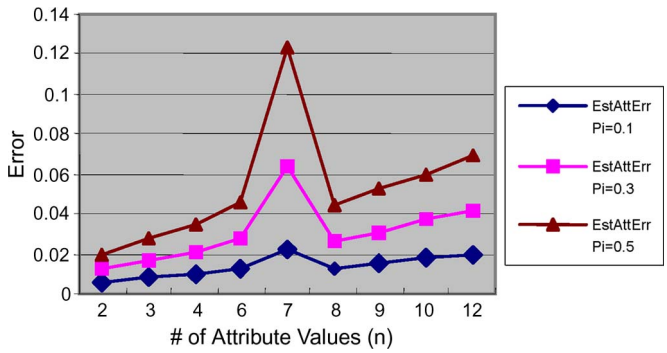

Fig. 6. Estimation error comparison w.r.t. the number of attribute values (mushroom dataset).

values, which is the stalk root, which contains seven possible values (including the missing value).

In summary, our empirical study here indicates that the proposed effort can indeed estimate the original data distribution from noise-corrupted datasets, which essentially assures the performance of EA-NB. In addition, we observed that the noise variance caused by different numbers of attribute values, under the same level of random corruption, does not bring much impact and, therefore, can be ignored. Meanwhile, as the number of attribute values increases, it will continuously bring more difficulty to our algorithm and result in higher estimation errors.

B. Classification Accuracy Comparisons Under a Uniform Corruption Model

To evaluate the performance of EA-NB under a uniform corruption model (Fig. 1), we design the following experiments. Given a dataset E, we first train an NB classifier and denote its classification accuracy by "Original." We then introduce a certain level of noise into E to build a corrupted dataset D and learn another NB classifier from D with its performance denoted by "Corrupted." With D and a noise level p_i, we can build an EA-NB classifier, which is represented as "EA-NB." Since in noisy environments, data cleansing is often adopted to enhance data quality and improve the classification accuracy, we therefore apply a data cleansing method [5] on D to remove all misclassified examples, and we build another NB classifier from the cleansed dataset. The performance of this NB classifier is expressed as "Cleansing."

We compare the performance of the aforementioned four classifiers at different noise levels $p_i \in [0.1, 0.5]$ and report the detailed results from four representative datasets in Fig. 7, where the x-axis represents the noise level p_i, and the y-axis indicates the classification accuracy. The summarized results from six other datasets are reported in Table III. To justify the performance of the NB classifier implemented by ourselves and to show that NB is indeed robust in noisy environments, we also report the results of C4.5 in Table III so that we can comparatively study NB and EA-NB. To ensure that the observations we made are statistically significant, we report the t-test results (p-value) between the accuracies of NB and EA-NB by using accuracies from a ten-time cross validation. A statistically significant difference (less than 5%) is marked in bold text in Table III.

As shown in Fig. 7, errors have a negative impact on the learners built from noisy datasets. This is a common sense as corrupted datasets no longer reveal genuine data distributions and will confuse the NB classifiers from making correct decisions. It is worth noting that different datasets react differently to the same level of noise. A small portion of noise can seriously deteriorate an NB learner (e.g., for the car and splice datasets in Fig. 7), or a significant amount of noise may still do not have much impact at all (e.g., for the adult and nursery datasets in Table III). We believe that this is an intrinsic feature of a dataset, which is determined by factors such as the instance numbers and the complexity of the concepts in the dataset. Meanwhile, as NB is a typical statistical learner, noise normally does less harm to it compared with other nonstatistical learning mechanisms (as shown in Table III, where C4.5 usually deteriorates much faster than NB). Generally, for a dataset with a large number of instances and containing a significant amount of redundancy, the existence of errors does less harm, as the genuine data distributions can be restored from just a small portion of the data. On the other hand, for a dataset with a very limited number of instances and when each instance appears to be necessary for classification, adding a small amount of errors can make considerable changes to the NB classifier because errors in this case can easily modify data distributions and confuse NB learners. Overall, our observation concludes that NB is relatively robust to data errors compared with its other peer C4.5. This conclusion is consistent with the observations from [37].

When noise is introduced to the attributes, data cleansing is not an effective solution to improve data mining performance. For many datasets that we used, the learners trained from the cleansed dataset "Cleansing" are inferior to the ones trained from the original noisy datasets "Corrupted," where the results of "Cleansing" can be as worse as 7% less than the accuracy of "Corrupted" (the absolute accuracy difference). This complies with the previous observations from Quinlan [16]. The negative impact of data cleansing may come from two possible reasons: 1) Removing suspicious instances, which do not comply with the existing model, may inevitably eliminate good examples and incur information loss, and 2) just because some attribute values are erroneous, it does not necessarily mean that the whole instance is useless, and many other attribute values of the noisy instance may still benefit the learning theory; therefore, it cannot be simply removed. The impact of these two factors becomes extremely clear if the accuracy of the underlying learner is low, as shown in Fig. 7(d). If a learner has only 50% accuracy, it means that half of the removed instances are actually good, and this explains the reason why data cleansing may introduce information loss.

When incorporating statistical error information for the EA-NB classification, we can achieve, on average, a good improvement in the classification accuracy (as shown in "EA-NB" and "Corrupted"). Take the car dataset in Fig. 7(a) as an example, where the accuracy of EA-NB is 10% higher than the learner from the corrupted dataset. Similar results can be observed from many benchmark datasets, where EA-NB, on average,

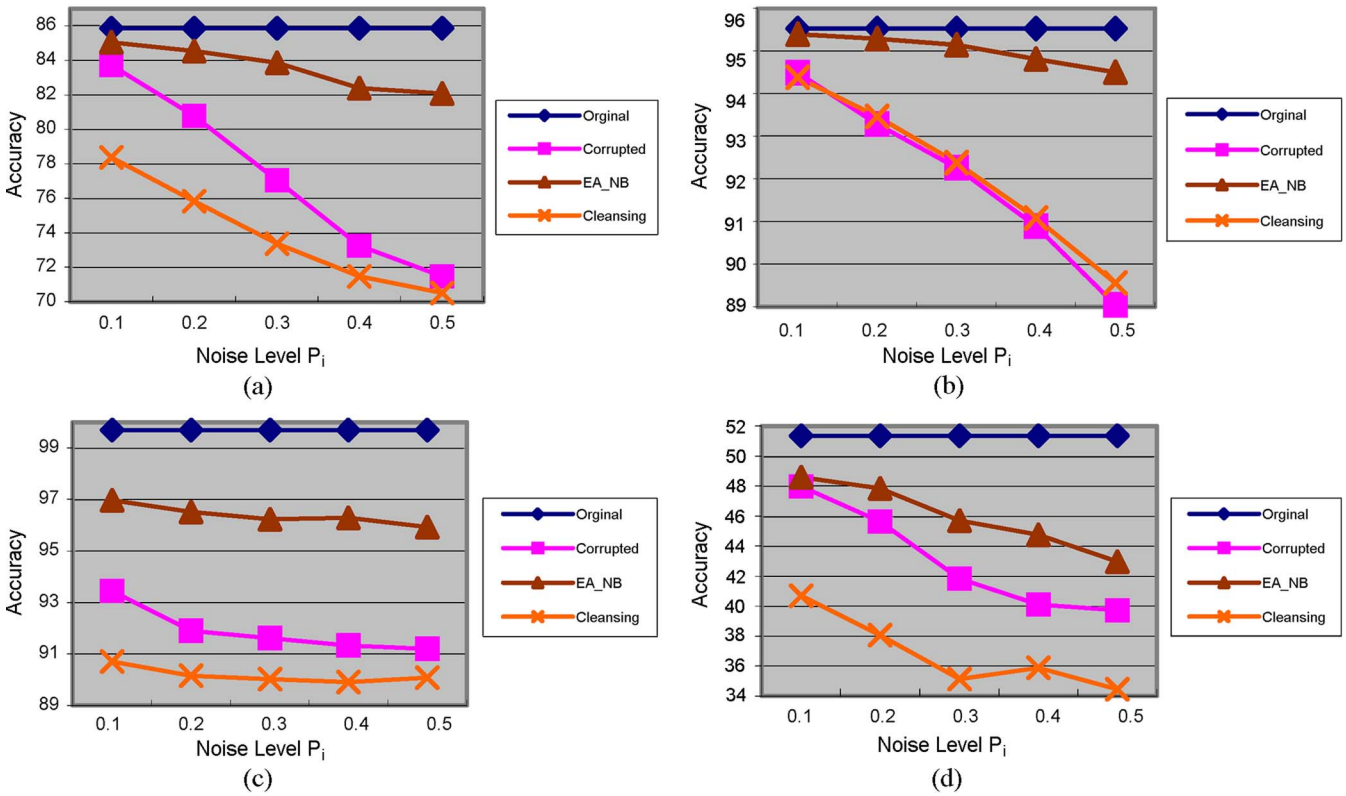

Fig. 7. Classification accuracy comparison. (a) Car dataset. (b) Splice dataset. (c) Mushroom dataset. (d) Glass dataset.

performs better than a learner built from corrupted datasets. From $3 \times 6 = 18$ observations in Table III, we can find that 12 out of the 18 observations are statistically significant. Because EA-NB has higher mean accuracy on these 12 observations, it is safe for us to conclude that EA-NB is significantly better than NB on these datasets. The results from Fig. 7 and Table III suggest that the higher the noise level in the datasets, the more improvement can be observed (when the noise level is less than 50%). This indicates that although data errors continuously bring an impact to the learning theory, having a data mining process aware of the data errors can reduce noise impact and enhance mining results.

Because EA-NB relies on the statistical error information to ensure the success of the algorithm, a limited number of training instances may be insufficient to restore the original data distribution. For this purpose, we intentionally selected three datasets with a small number of instances (glass, wine, and zoo). Our results from glass (214 examples and 7 classes), wine (178 examples and 3 classes), and zoo (101 instances and 7 classes) indicate that even with a very limited number of instances, the proposed effort can still function well and achieve impressive results. The performance of EA-NB is not just determined by the total number of training examples in the dataset but also relies on the number of attribute values for each attribute. Take the zoo dataset as an example. If we evenly divide 101 instances by seven classes, each class would only have about 15 instances on average. An attribute with five attribute values would have three instances for each attribute, respectively. This would lead to a severe bias for EA-NB to estimate the original data distribution even if we know the parameters of the data transformation model perfectly. In reality, the 15 nominal attributes of zoo are all Boolean, and the first two classes contain about 60.4% of instances in the dataset, which makes an accurate data restoration for EA-NB possible. In short, our results from the three spare datasets indicate that it is possible for the EA data mining to receive a good performance on small datasets, although the nature of EA data mining suggests that it prefers datasets with a relatively large number of training examples.

C. Classification Performance Under Inexact Noise Levels

EA-NB considers the noise level p_i as *priori* knowledge given by users. In reality, a user-specified p_i value may be different from the actual noise level in the dataset. If a tiny difference between the user-specified value and the actual noise level in the database would bring a considerable impact to the system performance, we should then find solutions to enhance the robustness of our algorithm. For this purpose, we adopt the following approach to perturb the user-specified noise level p_i.

Given a noise level p_i, we first use this value to construct a noisy dataset D. When learning an EA-NB classifier from D, we intentionally change the noise level p_i as $p_i + \vartheta \cdot d \cdot p_i$, where ϑ is a random variable with a 50% chance of equaling to $+1$ or -1, respectively, and d is another random variable which controls the difference (the absolute value) between the user-provided value and the genuine noise level in attribute a_i. This approach simulates situations where users can only roughly guess the noise level in each

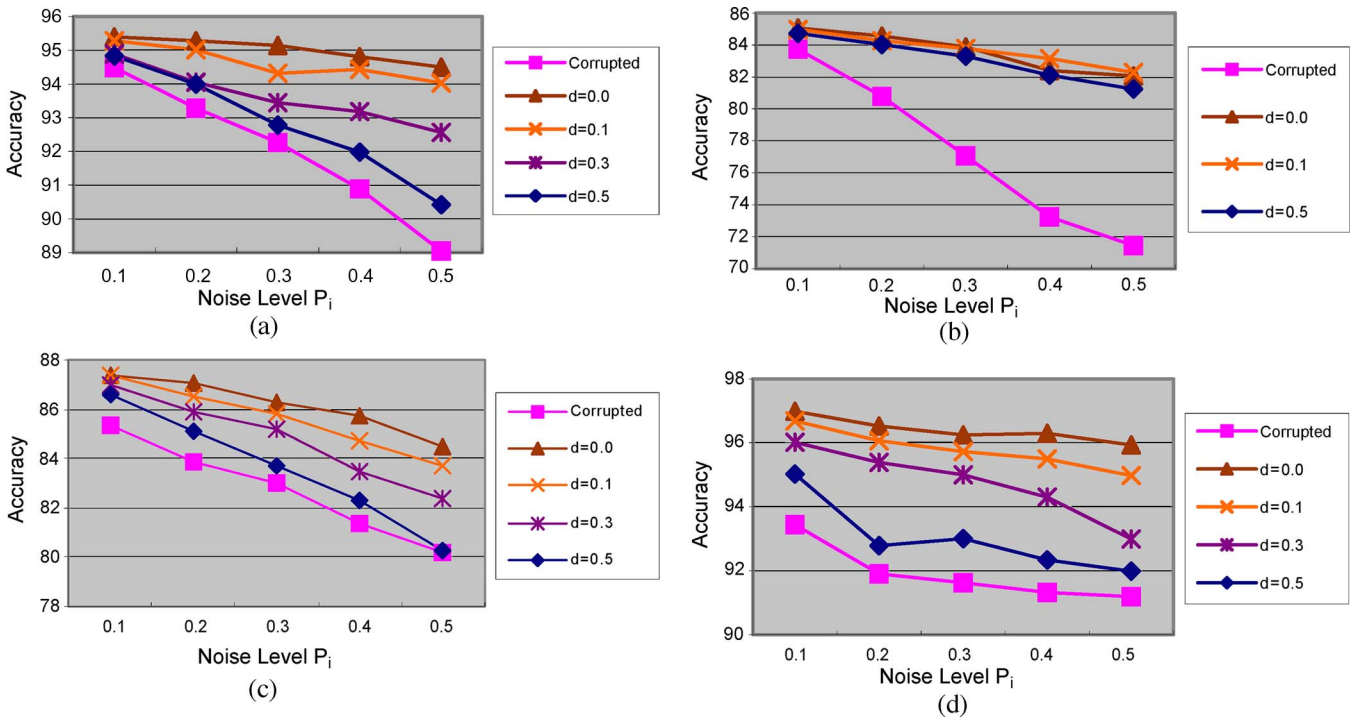

Fig. 8. Classification performance comparisons under inexact noise levels (uniform transformation model). (a) Splice dataset. (b) Car dataset. (c) Krvskp dataset. (d) Mushroom dataset.

attribute. Note that with the aforementioned perturbation, the noise uncertainty range $(\vartheta \cdot d \cdot p_i)$ is proportional to the actual noise level of attribute a_i, such that the relative values of the inexact noise levels for different noise levels (p_i) are consistent.

In Fig. 8, we show the results from four representative datasets, namely, Krvskp, car, splice, and mushroom, with different values of d. We set the value of d from 0 to 0.5, which means that the perturbation amplitude varies from 0% to 50% of the original noise level, and we provide the results at four values, namely, 0, 0.1, 0.3, and 0.5 (as different d values do not result in significant changes in Fig. 8(b), we ignore the result from $d = 0.3$).

As shown in Fig. 8, when the user-specified noise level is different from the actual noise level in the database, EA-NB deteriorates for sure, as inaccurate noise knowledge misleads EA-NB to build a biased model. For a slightly different noise value, e.g., $d = 0.1$, the impact is not significant. However, the higher the amplitude of the perturbation, the more severely the system will deteriorate. Depending on the intrinsic characteristics of each dataset, the deterioration of the system performance varies significantly. Take datasets in Fig. 8 as examples. When the maximum perturbation amplitude is 50%, i.e., $d = 0.5$, the system from the car dataset deteriorates only 0.8% in its performance in comparison with the results without any perturbation. On the other hand, with the same level of perturbation, the results from Krvskp are almost totally ruined and become approximately the same as the results from the corrupted dataset. Determining what types of datasets are more sensitive to such a perturbation is a nontrivial task and requires intensive studies on the complexity of the underlying concepts, the data redundancy, and the interactions among attributes, which are beyond the coverage of this paper. However, our observations from four representative datasets indicate that, as long as user-specified values are close to the actual noise level in the dataset (e.g., no more than 30%), the proposed effort can still achieve impressive results and outperform a learner trained from noise-corrupted datasets. This shows that EA-NB is pretty robust in reality and can accommodate deviations in the users' input for effective mining.

D. Classification Accuracy Comparison Under a General Transformation Model

To assess EA-NB's performance under a general transformation model, we design the following experiments. For any attribute value a_{ij} of a_i in the dataset, we first assume that its error rate is $\alpha \cdot 100\%$, but the errors are not uniformly distributed across all other attribute values. For this purpose, we randomly choose another attribute value of a_i (excluding a_{ij}), for example, a_{ik}, and we assign a transformation probability $\alpha/2$ to q^i_{jk}, i.e., $q^i_{jk} = \alpha/2$. This means that an instance containing value a_{ij} has $q^i_{jk} = \alpha/2$ probability of being corrupted as value a_{ik}. For any remaining attribute of a_i (excluding a_{ij} and a_{ik}), for example, a_{il}, we assign a uniform transformation probability to it, i.e., $q^i_{jl} = \alpha/2(M_i - 2)$, which means that a_{ij} has a uniform probability of $q^i_{jl} = \alpha/2(M_i - 2)$ to be transformed to any other attribute $a_{il}, l = 1, 2, \ldots, M_i, l \neq j, l \neq k$. If an attribute has two attribute values only, we will assign $q^i_{jl} = \alpha$ to ensure that the sum of the transformation probabilities is equal to one. Take an attribute a_i with four attribute values $\{1, 2, 3, 4\}$ as an example, where an attribute value has about $\alpha \cdot 100\%$ errors on average. For an attribute value "one," by assuming that we

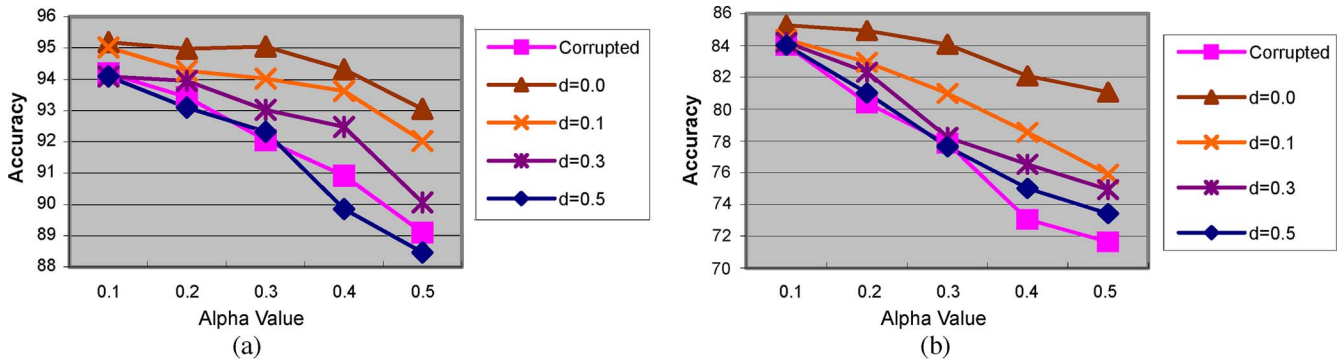

Fig. 9. Classification performance comparison under inexact noise levels (general transformation model). (a) Splice dataset. (b) Car dataset.

randomly select another attribute value "three" and assign $\alpha/2$ to it, then the transmission matrix for a_{11} is denoted by

$$q_1^i = [1-\alpha, \alpha/4, \alpha/2, \alpha/4]. \tag{14}$$

We continue the same process for all other attribute values, and eventually, the transmission matrix for attribute a_i is denoted by

$$q^i = [\, q_1^i, \quad q_2^i, \quad q_3^i, \quad q_4^i \,]^{\mathrm{T}}. \tag{15}$$

We generate this q^i matrix in each run of the cross validation and use it to corrupt the learning set. Because of the random process in choosing the attribute value a_{ik}, the q^i matrix in each run of the cross validation is not the same. When restoring data distribution, we also assume the existence of some uncertainties in the user-specified q^i matrix, which is given by

$$\tilde{q}^i = [\, q_1^i, \quad q_2^i, \quad q_3^i, \quad q_4^i \,]^{\mathrm{T}} + [\, \beta_1^i, \quad \beta_2^i, \quad \beta_3^i, \quad \beta_4^i \,]^{\mathrm{T}} \tag{16}$$

where

$$\beta_1^i = [0, \vartheta \cdot d \cdot \alpha/4, \vartheta \cdot d \cdot \alpha/2, \vartheta \cdot d \cdot \alpha/4] \tag{17}$$

$$\tilde{q}_{jj}^i = 1 - \left(\tilde{q}_{j1}^i + \cdots + \tilde{q}_{jk}^i + \cdots + \tilde{q}_{jM_i}^i\right), \quad j = 1, \ldots, M_i, \quad k \neq j. \tag{18}$$

In (17), ϑ is a random variable with a 50% chance of equaling to $+1$ or -1, respectively, and d is a random variable specified by the user to control the accuracy of the user-specified $\tilde{q}^i$ matrix (compared with the genuine transformation matrix q^i). If $d = 0$, the user-specified matrix $\tilde{q}^i$ is identical to the underlying data transformation model q^i. Because of the random process in (17), the sum of the pertubated probabilities in each row of $\tilde{q}^i$ may not be equal to one. Thus, we use (18) to ensure that the sum of each attribute's transformation probabilities is equal to one.

In Fig. 9, we show the results from two datasets with different parameter settings, where the x-axis (alpha value) denotes the value of α. Comparing the curves of $d = 0$ in Fig. 9 with the curves of $d = 0$ in Fig. 8(a) and (b), we can find that the results from the general transmission matrix are slightly worse than those from a uniform error corruption model. We believe that this is mainly because the complexity of the underlying data correction model (compared with the previous simple uniform model) raises a new challenge for EA-NB to exactly restore the original data distribution. In addition, when errors are no longer uniformly distributed across all attributes, it might lead to high estimation errors on some particular attribute values, which might be more informative for classification (compared with other attribute values). As a result, the overall classification accuracy will drop accordingly. However, the results in Fig. 9 indicate that EA-NB is still capable of restoring the original data distributions most of the time, unless the user-specified model is severely inaccurate, such as $d = 0.5$, where the accuracy deterioration can be extremely severe for some datasets, e.g., for the splice dataset which has 60 nominal attributes where each has eight attribute values. This indicates that for a general transmission model, the EA-NB has a high requirement in terms of the accuracy of the model specified by the users, and a carelessly specified model, which is significantly different from the underlying data characteristics, may lead to inferior decision models (compared with the ones built from the original datasets).

V. Discussion

A. Extension to Other Learning Algorithms

To demonstrate the idea of EA data mining, we have materialized an NB-based algorithm. We believe that the same idea can be extended to many other learning algorithms, as long as the learning algorithm is based on statistical data analysis to induce decision theories, such as the most popular decision-tree algorithms ID3/C4.5 [15], [16] and CART [21]. In this section, we discuss the feasibility of extending the proposed design to other algorithms.

For decision-tree construction, ID3/C4.5 and CART adopt information gain, gain ratio, or the gini index, respectively, to evaluate each attribute, and they select the most informative attribute once a time to split data into smaller subsets. This procedure is repeated until all instances in each subset belong to one class or some stopping criteria are met. To ensure a good performance, an accurate calculation of information gain and gini index values is crucial, as incorrectly calculated values lead to poor splitting and decrease the system performance. With the statistical error information, we can restore the original information gain or gini index values, which is similar to what we have done with NB.

Take the gini index in CART as an example. For a dataset S with N instances and L classes, the gini index $\text{Gini}(S)$ is defined as $\text{Gini}(S) = 1 - \sum_{l=1}^{L} f_l^2$, where f_l is the relative frequency of class l in S. With certain splitting criteria T, if we split S into two subsets S_1 and S_2 with sizes N_1 and N_2, respectively, the gini index $\text{Gini}(S, T)$ is defined as

$$\text{Gini}_{\text{Split}}(S, T) = \frac{N_1}{N}\text{Gini}(S_1) + \frac{N_2}{N}\text{Gini}(S_2). \tag{19}$$

To find the "best" splitting attribute, we have to enumerate all possible splitting points (determined by the attribute values) for each attribute, produce a pair of subsets S_1 and S_2, and choose the one with the smallest gini index for splitting.

In noisy environments, erroneous attribute values produce incorrect class frequencies in the splitted subsets S_1 and S_2 and, therefore, damage the true gini index values. With error information p_i, we can estimate the original gini index values through the following three steps.

1) Given a dataset S, for each class c_l, $l = 1, \ldots, L$, we adopt (7) to estimate the attribute value's original distribution $E_{ij}^{c_l}$, $j = 1, \ldots, M_i$; $i = 1, \ldots, M$.
2) Use (4) or (10) to estimate the original distribution of attribute A_i, $|E_{ij}|$, $j = 1, \ldots, M_i$.
3) The modified gini index of S, w.r.t. each possible split of attribute A_i, is denoted by

$$\text{Gini}_{\text{Split}}(S, a_{ij}) = \frac{|E_{ij}|}{N}\left(1 - \sum_{l=1}^{L}\frac{|E_{ij}^{c_l}|}{|E_{ij}|}\right) + \frac{N - |E_{ij}|}{N} \times \left(1 - \sum_{l=1}^{L}\frac{\sum_{k=1}^{M_i}|E_{ik}^{c_l}|,\ k \neq j}{N - |E_{ij}|}\right). \tag{20}$$

4) Enumerate all possible splits for all attributes, and choose the one with the smallest value for splitting.

We believe that the same approach is valid for information-gain (or information-gain-ratio)-based algorithms as well. However, because the statistical error information is not directly beneficial for an accurate estimation of the genuine attribute values of each particular instance, for algorithms like instance-based learning [18], e.g., K-nearest neighbor classification, where the decision is derived directly from each single instance (instead of statistical information of the instances), the concept of EA data mining might not be easily materialized.

B. Limitations of the Proposed Approach

The limitations of the proposed EA data mining design are mainly derived from its dependence on the presumed error model. This can be elaborated by the following two aspects.

First, both our uniform corruption model in Fig. 1 and general transmission model in Fig. 3 are based on an assumption that the statistical error information is given in advance, which may not be always the case in reality, although we have argued in Section I that such information is indeed available in many applications. When such information is unavailable, the applicability of the proposed design is invalid. Although we have demonstrated that roughly guessed error information may still lead to improved data mining results for EA-NB, our experimental results in Fig. 9 indicated that EA-NB could deteriorate the system performance if the user-provided model is severely biased or inaccurate. However, just like a cost-sensitive learning algorithm [44] needs user-specified cost values, such as misclassification cost or test cost, to fulfill its goal, the EA data mining relies on the assumption that users are capable of providing the statistical error models beforehand.

Second, the data distribution restoration models that we proposed in Section III can only handle nominal attribute values, and a data discretization process is thus needed for numerical attributes. This leads to additional computations and also suffers from potential information loss. This limitation, we believe, can be overcome by using simple statistical analysis, which is similar to the privacy-preserving data-mining-based approaches [3]. For example, if we know in advance that errors in each attribute comply with a normal distribution with a mean μ and a standard deviation δ, such knowledge can be directly used to restore the original distribution of the data, from which accurate decision trees or NB models can be constructed (although it is not feasible for us to accurately estimate the values of each instance). The effectiveness of this type of solutions has been evaluated by existing privacy-preserving data mining algorithms [3].

There would be very little we can do if we know nothing about the underlying data, and because of this, the proposed EA data mining framework takes the assumption that the statistical error information is known beforehand. We argued in Section I that in many applications, such information is available or can be easily acquired with trivial endeavors. Our results in Section IV supported our solutions when the data errors indeed comply with our assumptions. In addition, our results also demonstrated that a slight violation of our assumption does not lead to fatal results but may still receive a good performance. Note that the traditional data mining framework (without error awareness) separates the data preprocessing (data enhancement) module from the succeeding mining process, and this paper on EA data mining provides a new way to seamlessly unify them into one framework. Future research may focus on extending and materializing the idea of EA data mining to other algorithms, such as support vector machines (in addition to the ones we discussed in Section V-A), and on considering error models which are closer to real-world applications.

VI. Conclusion

In this paper, we have proposed an EA data mining framework which seamlessly unifies statistical error information and a data mining algorithm for effective learning. The proposed effort makes use of noise knowledge to modify the model built from noise-corrupted data, and it has resulted in a substantial improvement in comparison with the models built from the original noisy data and the noise-cleansed data. The novel features that distinguish the proposed effort from existing endeavors are twofold: 1) we unify noise knowledge and a general data mining algorithm into a unique structure, whereas existing data mining activities often have no awareness of the underlying

data errors, and 2) instead of polishing noisy data, like many cleansing-based approaches do, we take advantage of the noise knowledge to polish the model trained from noisy data sources, and therefore, the original data are well maintained.

While the solutions presented in this paper are based on the NB classification only, incorporating noise knowledge into the mining process for EA data mining is an essential idea that we try to deliver here. Our experimental results indicated that when data contain a certain level of erroneous attribute values, data cleansing may not be a good solution to improve the data mining performance due to reasons such as information loss incurred by data cleansing. If statistical errors were known in advance, the proposed EA data mining framework, which utilizes noise knowledge to supervise the model construction, demonstrated to be a promising solution to solve the problem, as it can bridge the gap between the data imperfections and the mining process to enhance the system performance and it avoids possible information loss incurred by data cleansing.

Data mining from noisy information sources involves three essential tasks [49]: noise identification, noise profiling, and noise-tolerant mining. Data cleansing deals with noise identification. The EA data mining framework designed in this paper makes use of the statistical noise knowledge for noise-tolerant mining. Between noise identification and noise-tolerant mining, how to profile the noise identified from the noisy data in a given domain for the purpose of effective noise-tolerant mining is another challenging problem. In [50], a specific type of noise knowledge, which is called associative corruption rules, is modeled and studied with experimental results. Like the statistical error information used in this paper, the associative corruption rules are also used to model nonrandom structured noise. How to deal with different types of noise, namely, random or structured, for noise-tolerant mining is still an open research issue.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers whose constructive comments and advice helped improve this paper.

REFERENCES

[1] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study of their impacts," *Artif. Intell. Rev.*, vol. 22, no. 3/4, pp. 177–210, Nov. 2004.

[2] D. Luebbers, U. Grimmer, and M. Jarke, "Systematic development of data mining-based data quality tools," in *Proc. 29th VLDB*, Berlin, Germany, 2003.

[3] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proc. ACM SIGMOD*, 2000, pp. 439–450.

[4] W. Du and Z. Zhan, "Using randomized response techniques for privacy-preserving data mining," in *Proc. 9th ACM SIGKDD*, 2003, pp. 505–510.

[5] X. Zhu, X. Wu, and Q. Chen, "Eliminating class noise in large datasets," in *Proc. ICML*, 2003, pp. 920–927.

[6] C. Brodley and M. Friedl, "Identifying mislabeled training data," *J. Artif. Intell. Res.*, vol. 11, pp. 131–167, 1999.

[7] M. Teng, "Correcting noisy data," in *Proc. ICML*, 1999, pp. 239–248.

[8] I. Fellegi and D. Holt, "A systematic approach to automatic edit and imputation," *J. Amer. Stat. Assoc.*, vol. 71, no. 353, pp. 17–35, Mar. 1976.

[9] D. Rubin, *Multiple Imputation for Nonresponse in Surveys*. New York: Wiley, Jun. 2004.

[10] *DNA Microarray Data Analysis*, CSC—Scientific Computing Ltd., Espoo, Finland, 2005. [Online]. Available: http://www.csc.fi/oppaat/siru/

[11] R. Wang, V. Storey, and C. Firth, "A framework for analysis of data quality research," *IEEE Trans. Knowl. Data Eng.*, vol. 7, no. 4, pp. 623–639, Aug. 1995.

[12] X. Zhu, X. Wu, and Y. Yang, "Error detection and impact-sensitive instance ranking in noisy datasets," in *Proc. AAAI*, 2004, pp. 378–384.

[13] X. Zhu and X. Wu, "Cost-constrained data acquisition for intelligent data preparation," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1542–1556, Nov. 2005.

[14] F. Ferri, J. Albert, and E. Vidal, "Considerations about sample-size sensitivity of a family of edited nearest-neighbor rules," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 5, pp. 667–672, Oct. 1999.

[15] J. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.

[16] J. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[17] J. Fuernkranz, "Integrative windowing," *J. Artif. Intell. Res.*, vol. 8, pp. 129–164, 1998.

[18] D. Aha, D. Kibler, and M. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, Jan. 1991.

[19] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Mach. Learn.*, vol. 29, no. 2/3, pp. 103–130, Nov./Dec. 1997.

[20] C. Blake and C. Merz, *UCI Repository of Machine Learning Databases*, 1998.

[21] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Pacific Grove, CA: Wadsworth & Brooks, 1984.

[22] M. Berry and G. Linoff, *Mastering Data Mining*. New York: Wiley, 1999.

[23] J. R. Quinlan, "The effect of noise on concept learning," in *Machine Learning*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds. San Mateo, CA: Morgan Kaufmann, 1986.

[24] J. R. Quinlan, "Learning from noisy data," in *Proc. 2nd Int. Mach. Learn. Workshop*, Urbana-Champaign, IL, 1983.

[25] D. Gamberger, N. Lavrac, and C. Groselj, "Experiments with noise filtering in a medical domain," in *Proc. 16th ICML Conf.*, San Francisco, CA, 1999, pp. 143–151.

[26] G. H. John, "Robust decision trees: Removing outliers from databases," in *Proc. 1st Int. Conf. Knowl. Discovery Data Mining*, 1995, pp. 174–179.

[27] J. Kubica and A. Moore, "Probabilistic noise identification and data cleaning," in *Proc. ICDM Conf.*, Melbourne, FL, 2003.

[28] I. Guyon, N. Matic, and V. Vapnik, "Discovering informative patterns and data cleaning," in *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Menlo Park, CA: AAAI/MIT Press, 1996, pp. 181–203.

[29] Y. Yang, X. Wu, and X. Zhu, "Dealing with predictive-but-unpredictable attributes in noisy data sources," in *Proc. 8th Eur. Conf. PKDD*, Pisa, Italy, 2004.

[30] U. Fayyad, "Data mining and knowledge discovery: Making sense out of data," *IEEE Intell. Syst.*, vol. 11, no. 5, pp. 20–25, Oct. 1996.

[31] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, *CRISP-DM 1.0: Step-by-Step Data Mining Guide*. Chicago, IL: CRISP-DM Consortium/SPSS Inc., 2000. [Online]. Available: http://www.crisp-dm.org/CRISPWP-0800.pdf

[32] W. Sarle, "Prediction with missing inputs," in *Proc. 4th Joint Conf. Inf. Sci.*, 1998, pp. 399–402.

[33] J. Beaumont, "On regression imputation in the presence of nonignorable nonresponse," in *Proc. Survey Res. Methods Section*, 2000, pp. 580–585.

[34] L. Coppola, M. Zio, O. Luzi, A. Ponti, and M. Scanu, "Bayesian networks for imputation in official statistics: A case study," in *Proc. DataClean Conf.*, 2000, pp. 30–31.

[35] M. Hu, S. Salvucci, and M. Cohen, "Evaluation of some popular imputation algorithms," in *Proc. Survey Res. Methods Section*, 1998, pp. 308–313.

[36] K. Lakshminarayan, S. Harp, and T. Samad, "Imputation of missing data in industrial databases," *Appl. Intell.*, vol. 11, no. 3, pp. 259–275, Nov. 1999.

[37] P. Langley, W. Iba, and K. Thompson, "An analysis of Bayesian classifiers," in *Proc. 10th Nat. Conf. Artif. Intell. (AAAI)*, San Jose, CA, Jul. 1992.

[38] L. Ferrari and S. Aitken, "Mining housekeeping genes with a naive Bayes classifier," *BMC Genomics*, vol. 7, no. 277, Oct. 2006. DOI: 10.1186/1471-2164-7-277.

[39] A. D. Chapman, "Principles and methods of data cleaning—Primary species and species-occurrence data," in "Report for the Global Biodiversity Information Facility," GBIF, Copenhagen, Denmark, 2005. Version 1.0.

[40] M. Hernandez and S. Stolfo, "The merge/purge problem for large databases," *ACM SIGMOD Rec.*, vol. 24, no. 2, pp. 127–138, May 1995.
[41] J. Garcke and M. Griebel, "Data mining with sparse grids using simplicial basis functions," in *Proc. 7th ACM SIGKDD Conf.*, 2001, pp. 87–96.
[42] Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in *Proc. ACM SIGMOD Conf.*, 2005, pp. 37–48.
[43] A. McCallum and K. Nigam, "A comparison of event models for naive Bayes text classification," in *Proc. AAAI Workshop Learn. Text Categorization*, 1998, pp. 41–48.
[44] P. Domingos, "MetaCost: A general method for making classifiers cost sensitive," in *Proc. 5th Int. Conf. Knowl. Discovery Data Mining*, 1999, pp. 155–164.
[45] C. Shearer, "The CRISP-DM model: The new blueprint for data mining," *J. Data Warehous.*, vol. 5, no. 4, pp. 13–22, 2000.
[46] U. Fayyad and K. Irani, "On the handling of continuous-valued attributes in decision tree generation," *Mach. Learn.*, vol. 8, no. 1, pp. 87–102, Jan. 1992.
[47] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proc. 12th Int. Conf. Mach. Learn.*, 1995, pp. 194–202.
[48] X. Zhu and X. Wu, "Error awareness data mining," in *Proc. IEEE Int. Conf. GRC*, Atlanta, GA, May 10–12, 2006.
[49] X. Wu, "Class noise vs attribute noise: Their impacts, detection and cleansing," in *Proc. 11th PAKDD*, Nanjing, China, May 22–25, 2007, pp. 7–8.
[50] Y. Zhang and X. Wu, "Noise modelling with associative corruption rules," in *Proc. 7th IEEE ICDM*, Omaha, NE, Oct. 28–31, 2007.

Xindong Wu received the Ph.D. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K.

He is a Professor and the Chair of the Department of Computer Science, University of Vermont, Burlington. He is also currently with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China. His research interests include data mining, knowledge-based systems, and Web information exploration. He has published extensively in these areas in various journals and conferences, including the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING; IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE; *ACM Transactions on Information Systems*; *Data Mining and Knowledge Discovery*; *Knowledge and Information Systems*; International Joint Conference on Artificial Intelligence; Annual Conference of the Association for Advancement of Artificial Intelligence; International Conference on Machine Learning; ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD); IEEE International Conference on Data Mining (ICDM); and International World Wide Web Conference, as well as 14 books and conference proceedings.

Dr. Wu is the Editor-in-Chief of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (by the IEEE Computer Society), the Founder and Current Steering Committee Chair of the IEEE ICDM, an Honorary Editor-in-Chief of *Knowledge and Information Systems* (by Springer), and a Series Editor of the *Springer Book Series on Advanced Information and Knowledge Processing* (AI&KP). He was the Program Committee Chair for ICDM'03 and is the Program Committee Co-Chair for KDD-07 (the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining). He has been an invited/keynote speaker at numerous international conferences, including PAKDD-07, IEEE EDOC'06, IEEE ICTAI'04, IEEE/WIC/ACM WI'04/IAT'04, SEKE 2002, and PADD-97. He was the recipient of the 2004 ACM SIGKDD Service Award, the 2006 IEEE ICDM Outstanding Service Award, and the 2005 Chaired Professor in the Cheung Kong (or Yangtze River) Scholars Program at the Hefei University of Technology sponsored by the Ministry of Education of China and the Li Ka Shing Foundation.

Xingquan Zhu received the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2001.

From February 2001 to October 2002, he was a Postdoctoral Associate with the Department of Computer Science, Purdue University, West Lafayette, IN. From October 2002 to July 2006, he was a Research Assistant Professor with the Department of Computer Science, University of Vermont, Burlington. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton. Since 2000, he has been publishing extensively, including over 60 refereed papers in various journals and conference proceedings. His research interests include data mining, machine learning, bioinformatics, multimedia systems, and information retrieval.