*Artificial Intelligence Review* **22:** 177–210, 2004.



# Class Noise vs. Attribute Noise: A Quantitative Study of Their Impacts

XINGQUAN ZHU* & XINDONG WU
*Department of Computer Science, University of Vermont, Burlington, VT 05405, USA*
*(*author for correspondence, E-mail: xqzhu@cs.uvm. edu)*

**Abstract.** Real-world data is never perfect and can often suffer from corruptions (noise) that may impact interpretations of the data, models created from the data and decisions made based on the data. Noise can reduce system performance in terms of classification accuracy, time in building a classifier and the size of the classifier. Accordingly, most existing learning algorithms have integrated various approaches to enhance their learning abilities from noisy environments, but the existence of noise can still introduce serious negative impacts. A more reasonable solution might be to employ some preprocessing mechanisms to handle noisy instances before a learner is formed. Unfortunately, rare research has been conducted to systematically explore the impact of noise, especially from the noise handling point of view. This has made various noise processing techniques less significant, specifically when dealing with noise that is introduced in attributes. In this paper, we present a systematic evaluation on the effect of noise in machine learning. Instead of taking any unified theory of noise to evaluate the noise impacts, we differentiate noise into two categories: class noise and attribute noise, and analyze their impacts on the system performance separately. Because class noise has been widely addressed in existing research efforts, we concentrate on attribute noise. We investigate the relationship between attribute noise and classification accuracy, the impact of noise at different attributes, and possible solutions in handling attribute noise. Our conclusions can be used to guide interested readers to enhance data quality by designing various noise handling mechanisms.

**Keywords:** attribute noise, class noise, machine learning, noise impacts

## 1. Introduction

The goal of inductive learning algorithms is to form generalizations from a set of training instances such that the classification accuracy on previously unobserved instances is maximized. This maximum accuracy is usually determined by two most important factors: (1) the quality of the training data; and (2) the inductive bias of the learning algorithm. Given a specific learning algorithm, it's obvious that its classification accuracy depends vitally on the quality of the training data. Basically,

the quality of a large real-world dataset depends on a number of issues (Wang et al. 1995, 1996), but the source of the data is the crucial factor. Data entry and acquisition is inherently prone to errors. Many efforts can be put on this front-end process, with respect to reduction in entry errors. However, errors in a large dataset are common and severe, and unless an organization takes extreme measures in an effort to avoid data errors, the field error rates are typically around 5% or more (Wu 1995; Orr 1998; Maletic and Marcus 2000).

The problem of learning in noisy environments has been the focus of much attention in machine learning and most inductive learning algorithms have a mechanism for handling noise. For example, pruning in decision trees is designed to reduce the chance that the trees are overfitting to noise in the training data (Quinlan 1983, 1986a, b). Schaffer (1992, 1993) has made significant efforts to address the impacts of sparse data and class noise for overfitting avoidance in decision tree induction. However, since the classifiers learned from noisy data have less accuracy, the pruning may have very limited effect in enhancing the system performance, especially in the situation that the noise level is relatively high. As suggested by Gamberger et al. (2000), handling noise from the data before hypothesis formation has the advantage that noisy examples do not influence hypothesis construction. Accordingly, for existing datasets, a logical solution to enhance their quality is to attempt to cleanse the data in some way. That is, explore the dataset for possible problems and endeavor to correct the errors. For a real world dataset, doing this task 'by hand' is completely out of the question given the amount of person hours involved. Some organizations spend millions of dollars per year to detect data errors (Redman 1996). A manual process of data cleansing is also laborious, time consuming, and prone to errors. Useful and powerful tools that automate or greatly assist in the data cleansing process are necessary and may be the only practical and cost effective way to achieve a reasonable quality level in an existing dataset.

There have been many approaches for data preprocessing (Wang et al. 1995, 1996; Redman 1996, 1998; Maletic 2000) and noise handling (Little and Rubin 1987; John 1995; Zhao 1995; Brodley and Friedl 1999; Gamberger et al. 1999, 2000; Teng 1999; Allison 2002; Batista and Monard 2003; Kubica and Moore 2003; Zhu et al. 2003a, 2004) to enhance the data quality. Among them, the enhancement could be achieved by adopting some data cleansing procedures, such as eliminating noisy instances, predicting unknown (or missing) attribute values, or correcting noisy values. These methods are efficient in their own scenarios, but some important issues are still open, especially when we

try to view noise in a systematic way and attempt to design generic noise handling approaches. Actually, existing mechanisms seem to be developed without a thorough understanding of noise. To design a good data quality enhancement tool, we believe the following questions should be answered in advance to avoid developing a 'blind' approach, which cannot guarantee its performance all the time.

1. What's noise in machine learning? What's the inherent relationship between noise and data quality?
2. What are the features of noise, and what's their impact with the system performance?
3. What's a general solution in handling noise (especially attribute noise)? Why does it work?

In this paper, we provide a systematic evaluation of the impacts of noise. The rest of the paper is organized as follows. In the next section, we will explain what's noise in machine learning and analyze the relationship between data quality and noise. The design of our experiments and benchmark datasets are introduced in Section 3. We analyze the impacts of class noise and various class noise handling techniques in Section 4. In Section 5, the effects of attribute noise are evaluated and reported, followed by a systematic analysis in handling attribute noise. Conclusion and remarks are given in Section 7.

## 2. Data Quality and Noise

The quality of a dataset can usually be characterized by two information sources: (1) attributes, and (2) class labels. The quality of the attributes indicates how well the attributes characterize instances for classification purpose; and the quality of the class labels represents whether the class of each instances is correctly assigned. When performing classification, we usually select a set of attributes to characterize the target concept (class labels) with the following two assumptions:

(1) Correlations between attributes and the class. The attributes are assumed to be (somewhat) correlated to the class. But being correlated does not necessarily mean that they have the same correlation levels. It is obvious that some attributes have stronger correlations with the class than others, and in such scenarios, those attributes act more importantly in classification.

(2) Weak interactions among attributes. The attributes are assumed to have weak interactions (Freitas 2001) with each other, so the learning algorithms likely ignore these interactions and consider each attribute independently to induce the classifier. This assumption becomes an extreme for Naïve Bayes (*NB*) classifier (Langley et al. 1992) where all attributes are assumed to be independent or conditionally independent (i.e., no interaction at all). For many other greedy induction algorithms, e.g., ID3 (Quinlan 1986a) and CN2 (Clark and Niblett 1989), weak interactions among attributes are actually implicitly adopted, because they usually evaluate one attribute at each time in constructing the classifier and tend to ignore the attribute interactions. Many research efforts have indicated that even though the interactions among attributes extensively exist, the results from these classifiers are surprisingly good, e.g., NB (Domingos and Pazzani 1996) and C4.5 (Quinlan 1993) likely have good performance with normal datasets. However, the existence of attribute interactions actually brings trouble for many classifiers, as shown in Table 1, where a pedagogical example of a logic XOR (eXclusive OR) function is used to demonstrate the impacts of the attribute interactions. It's obvious that many greedy algorithms (e.g., ID3) are likely to be fooled by the interaction between attributes A and B, if they consider only one attribute once a time.

Unfortunately, real-world data does not always comply with the above two assumptions. Given a dataset, it either contains some attributes that have very little correlation with the class, or there may exist strong interactions among attributes. In either case, greedy algorithms' performance decreases. In the worst case, neither of the above assumptions holds.

Accordingly, the quality of a dataset is determined by two, external and internal, factors: the internal factor indicates whether attributes and

*Table 1.* Attribute interaction in a logic XOR function

| Attribute A | Attribute B | Class |
|---|---|---|
| True | True | 0 |
| True | False | 1 |
| False | True | 1 |
| False | False | 0 |

the class are well selected and defined to characterize the underlying theory, and the external factor indicates errors introduced into attributes and the class labels (systematically or artificially). In Hickey (1996), both internal and external factors are used to characterize noisy instances, where noise is anything that obscures the relationship between the attributes and class. Under this scenario, three types of major physical sources of noise are defined: (1) insufficiency of the description for attributes or the class (or both); (2) corruption of attribute values in the training examples; and (3) erroneous classification of training examples. However, for real-world datasets, it is difficult to quantitatively characterize the sufficiency of the description for attributes and the class, therefore, our definition with noise considers only the last two physical sources. More specifically, when an instance becomes problematic in terms of a benchmark theory, due to the incorrectness of attributes or the class, we indicate that the instance contains noise. A similar definition has been used in Quinlan (1986) where non-systematic errors in either attribute values or class information are referred to as noise.

Based on the above observations, the physical sources of noise in machine learning and data mining can be distinguished into two categories (Wu 1995): (a) attribute noise; and (b) class noise. The former is represented by errors that are introduced to attribute values. Examples of those external errors include (1) erroneous attribute values, (2) missing or don't know attribute values, (3) incomplete attributes or don't care values. There are two possible sources for class noise:

(1) Contradictory examples. The same examples appear more than once and are labeled with different classifications.
(2) Misclassifications. Instances are labeled with wrong classes. This type of errors is common in situations that different classes have similar symptoms.

Many research efforts have been made to deal with class noise (John 1995; Zhao 1995; Brodley and Friedl 1999; Gamberger et al. 1999; Gamberger et al. 2000; Zhu et al. 2003a), and have suggested that in many situations, eliminating instances that contain class noise will improve the classification accuracy. However, handling attribute noise is more difficult (Teng 1999; Zhu et al. 2004). Quinlan (1986a) concluded that, 'For higher noise levels, the performance of the correct decision tree on corrupted data was found to be inferior to that of an imperfect decision tree formed from data corrupted to a similar level! The moral seems to be that it is counter-productive to eliminate noise from the attribute information in the training set if these same attributes will be

subject to high noise levels when the induced decision tree is put to use'. From this conclusion, eliminating instances which contain attribute noise is not a good idea, because many other attributes of the instance may still contain valuable information. Accordingly, research on handling attribute noise has not made much progress, except some efforts on handling missing (or unknown) attribute values (Little and Rubin 1987; Allison 2002; Batista and Monard 2003), which were popularized by Cohen and Cohen (1983). Some extensive comparative studies related to missing attribute-value processing can be found in Quinlan (1989), Bruha and Franek (1996), Bruha (2002) and Batista and Monard (2003).

An interesting fact from real-world data is that the class information is usually much cleaner than what we thought; and it is the attributes that usually need to be cleaned. Take a medical dataset as an example. The doctors would likely put more attention and more care on the class label for the following reasons:

(1) In comparison with the unique class label, a dataset usually has more attributes, some of which can be of little use.
(2) For some attributes, their values are simply not available in many situations. For example, when we identify genes with similar cellular functions, it's usual that in a single experiment only a small portion of proteins have reactions. For proteins having no reaction, their attribute values become unavailable.

The above analysis likely indicates something embarrassing: we paid much attention on class noise that has already been emphasized; on the other hand, we generously ignored attribute noise brought by original carelessness. Are attributes less important than class labels, so we can ignore noise introduced to them? This paper will view attribute noise from different perspectives. We will demonstrate that in terms of data quality and classification accuracy, both attributes and class are important. By an extensive evaluation of their impacts, we can have a clear guidance in designing more efficient noise-handling mechanisms, especially for attribute noise that is introduced by erroneous attribute values. Instead of taking any unified theory of noise to evaluate the noise impacts, like Hickey (1996) did, we differentiate noise into two categories: class noise and attribute noise (based on the physical sources of noise), and analyze their impacts on the system performance separately, because for real-world datasets it is actually difficult (if not impossible) to work out a unified theory of noise (which combines errors in attributes and the class). In the following sections, we will systematically analyze the effects of noise handling for efficient learning.

We focus on attributes noise, because little research has been conducted in this regard.

## 3. Experiment Settings and Benchmark Datasets

The results presented in this paper are based on 17 datasets of which 16 were collected from the UCI repository (Blake and Merz 1998) and 1 from the IBM synthetic data generator (IBM Synthetic Data), as shown in Table 2. Numerous experiments were run on these datasets to assess the impact of the existence of noise on learning, especially on classification accuracy. The majority of experiments use C4.5, a program for inducing decision trees (Quinlan 1993).

For most of the datasets we used, they don't actually contain noise, so we use manual mechanisms to add both class noise and attribute noise. For class noise, we adopt a pairwise scheme (Zhu et al. 2003a): given a pair of classes $(X, Y)$ and a noise level $x$, an instance with its label $X$ has an $x \cdot 100\%$ chance to be corrupted and mislabeled as $Y$, so

*Table 2.* Benchmark datasets for our experiments

| Dataset | Instances | Number of nominal attributes | Number of numerical attributes | Attribute number | Class number |
|---|---|---|---|---|---|
| Adult | 48842 | 8 | 6 | 14 | 2 |
| Car | 1728 | 6 | 0 | 6 | 4 |
| CMC | 1473 | 7 | 2 | 9 | 3 |
| Connect-4 | 67557 | 42 | 0 | 42 | 3 |
| Credit-app | 690 | 9 | 6 | 15 | 2 |
| IBM | 9000 | 3 | 6 | 9 | 4 |
| Krvskp | 3196 | 36 | 0 | 36 | 2 |
| LED24 | 1000 | 24 | 0 | 24 | 10 |
| Monk3 | 432 | 6 | 0 | 6 | 2 |
| Mushroom | 8124 | 22 | 0 | 22 | 2 |
| Nursery | 12960 | 8 | 0 | 8 | 5 |
| Sick | 3772 | 22 | 7 | 29 | 2 |
| Splice | 3190 | 60 | 0 | 60 | 3 |
| Tictactoe | 958 | 9 | 0 | 9 | 2 |
| Vote | 435 | 16 | 0 | 16 | 2 |
| WDBC | 569 | 0 | 30 | 30 | 2 |
| Wine | 178 | 13 | 0 | 13 | 3 |

does an instance of class $Y$. We use this method because in realistic situations, only certain types of classes are likely to be mislabeled. Meanwhile, with this scheme, the percentage of the entire training set that is corrupted will be less than $x \cdot 100\%$, because only some pairs of classes are considered problematic. In the sections below, we corrupt only one pair of classes (usually the pair of classes with the highest proportions of instances). Meanwhile, we only report the value $x$ of class noise (which is not the actual class noise level in the dataset) in all tables and figures below.

For attribute noise, the error values are introduced into each attribute with a level $x \cdot 100\%$ (Zhu et al. 2004). This is consistent with the assumptions in Section 2, where the interactions among attributes are assumed to be weak. Consequently, the noise introduced into one attribute usually has not much correlation with noise from other attributes. To corrupt each attribute (e.g., $A_i$) with a noise level $x \cdot 100\%$, the value of $A_i$ is assigned a random value approximately $x \cdot 100\%$ of the time, with each possible value being approximately equally likely to be selected. For a numerical attribute, we select a random value that is between the maximal and the minimal. With this scheme, the actual percentage of noise is always lower than the theoretical noise level, as sometimes the random assignment would pick the original value (especially for nominal attributes). Note that, however, even if we exclude the original value from the random assignment, the extent of the effect of noise is still not uniform across all components. Rather, it is dependent on the number of possible values in the attribute or class. As the noise is evenly distributed among all values, this would have a smaller effect on attributes with a larger number of possible values than those attributes that have only two possible values (Teng 1999).

The above mechanism implies that we only deal with completely random attribute noise (Howell 2002), which means the probability that an attribute ($A_i$) has noise is unrelated to any other attribute. For example, if Whites were more likely to omit reporting income than African Americans, we would not have attribute noise that were completely random because noise with 'income' would be correlated with 'ethnicity'. If noise among attributes is introduced with correlations, the situation becomes more complicated, and this is beyond the coverage of this manuscript.

## 4. Impact of Class Noise

To evaluate the impact of class noise, we have executed our experiments on the above benchmark datasets, where various levels of class noise

(and no attribute noise) are added. We then adopt various learning algorithms to learn from these noisy datasets and evaluate the impact of class noise on them. We demonstrate one set of representative results in Figure 1 (from the car dataset), where the $x$-axis indicates the class noise level, and the $y$-axis represents the classification accuracy from different types of classifiers trained from the noise corrupted and manually cleaned training set respectively (evaluated with the same test set). As we can see from Figure 1, when the noise level increases, all classifiers trained from the noise corrupted training set suffer from decreasing the classification accuracy dramatically, where the classification accuracies decline almost linearly with the increase of the noise level. We have used five classification algorithms, C4.5 (Quinlan 1993), C4.5 rules (Quinlan 1993), HCV (Wu 1995), 1R (Holte 1993) and Prism (Cendrowska 1987) in our experiments. On the other hand, the classifiers from the manually cleaned training set (in which instances containing class noise are removed) will have their classification accuracies improved comprehensively. We have executed the same experiments on all other datasets and found that the above conclusion holds for almost all datasets – the existence of class noise will decrease classification accuracy, and removing those noisy instances will improve the classification accuracy. In other words, cleaning the training data will result in a higher predictive accuracy with learned classifiers. Even though the use of pruning and learning ensembles of many existing learning algorithms partially addresses the impact of class noise, class noise can still drastically affect

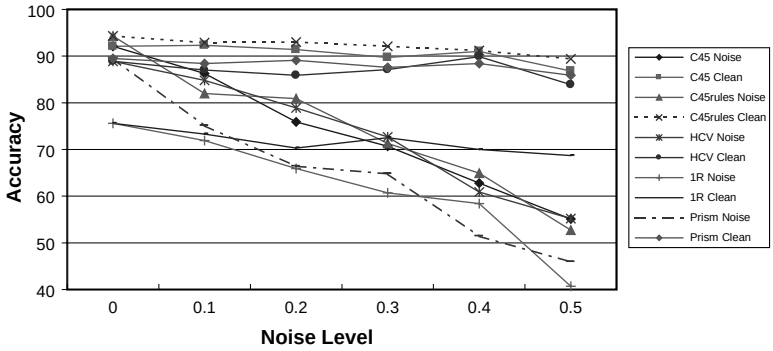


*Figure 1.* Classification accuracy of various classifiers trained from noise corrupted and manually cleaned training sets, where 'Λ Noise' indicates the classifier 'Λ' is trained from a noise corrupted training set and 'Λ Clean' represents the classifier 'Λ' trained from a cleaned training set. All results are evaluated from the test dataset (Car dataset from the UCI data repository).

the classification accuracy, as long as the noise exists in the training set. In addition to the classification accuracy, the research from Brodley and Friedl (1999) and Zhu et al. (2003a) suggested that class noise handling could shrink the size of the decision tree and save the time in training a classifier comprehensively. Therefore, many research efforts have been conducted in handling class noise for effective learning, where one of the most important questions is how to figure out the noisy instances.

To distinguish 'noisy' instances from normal cases, various strategies have been designed. Among them, the most general techniques are motivated by the intention of removing outliers in regression analysis (Weisberg, 1980). An outlier is a case that does not follow the same model as the rest of the data and appears as though it comes from a different probability distribution. As such, an outlier does not only include erroneous data but also surprisingly correct data. In John (1995), a robust decision tree was presented, and it took the idea of pruning one step further: training examples that are misclassified by the pruned tree, are also globally uninformative. Therefore, after pruning a decision tree, the misclassified training examples should be removed from the training set and the tree needs to be rebuilt using this reduced set. This process is repeated until no more training examples are removed. With this method, the exceptions to the general rules are likely to be removed without any hesitation; hence, this scheme runs a high risk of removing both exceptions and noise.

Instead of employing outlier filtering schemes, some researchers believe that noise can be characterized by various measures. Guyon et al. (1996) provided an approach that uses an information criterion to measure an instance's typicality; and atypical instances are then presented to a human expert to determine whether they are mislabeled errors or exceptions. However, they noted that because their method is an on-line method it suffers from ordering effects. Oka and Yoshida (1993, 1996) designed a method that learns generalizations and exceptions separately by maintaining a record of the correctly and incorrectly classified inputs in the influence region of each stored example. The mechanism for distinguishing noise from exceptions is based on a user-specified parameter, which is used to ensure that each stored example's classification rate is sufficiently high. Unfortunately, as concluded in Brodley and Friedl (1999), this approach has only been tested on artificial datasets. The method in Srinivasan et al. (1992) uses an information theoretic approach to detect exceptions from noise during the construction of a logical theory. Their motivation is that there is no

mechanism by which a non-monotonic learning strategy can reliably distinguish true exceptions from noise. The noise detection algorithm of Gamberger et al. (2000) is based on the observation that the elimination of noisy examples, in contrast to the elimination of examples for which the target theory is correct, reduces the CLCH value of the training set (CLCH stands for the Complexity of the Least Complex correct Hypothesis). They call their noise detection algorithm a Saturation Filter since it employs the CLCH measure to test whether the training set is saturated, i.e., whether, given a selected hypothesis language, the dataset contains a sufficient number of examples to induce a stable and reliable target theory.

In Brodley and Friedl (1996, 1999), general noise elimination approaches are simplified as a filtering model, where noise classifiers learned from corrupted datasets are used to filter and clean noisy instances, and the classifiers learned from cleaned datasets are used for data classification. Based on this filtering model, they proposed a noise identification approach where noise is characterized as the instances that are incorrectly classified by a set of trained classifiers. A combination of the saturation filter (Gamberger et al. 2000) and the filtering operation (Brodley and Friedl 1996) was reported in Gamberger et al. (1999), and a *Classification Filter* (CF) scheme was suggested for noise identification.

To handle class noise from large, distributed datasets, a *Partitioning Filter* (PF) was reported in Zhu et al. (2003a), where noise classifiers learned from small subsets are integrated together to identify noisy instances. As concluded from the comparative studies (Zhu et al. 2003b) and demonstrated in Tables 3–5, where OG indicates the classification accuracy of the classifier learned from the original noisy training set (without any noise elimination), CF represents the accuracy from the Classification Filter, and PF denotes the results from the Partitioning Filter, PF exhibits a better performance than CF in higher noise-level environments. In addition to the classification accuracy, PF also achieves comprehensive time efficiency in comparison with CF, as shown in Table 6.

## 5. Impact of Attribute Noise

For attribute noise, the situations are much more complicated than class noise. In Quinlan (1983, 1986a, b), extensive experiments were executed to evaluate the problem of learning from noisy environments. It was

*Table 3*. Experimental comparison between Classification Filter and Partitioning Filter on classification accuracy (Krvskp, Car, Nursery and WDBC)

| Noise (%) | Krvskp (%) | | | Car (%) | | | Nursery (%) | | | WDBC (%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OG | CF | PF | OG | CF | PF | OG | CF | PF | OG | CF | PF |
| 5 | 96.6 | 98.5 | 97.9 | 91.5 | 91.8 | 91.3 | 95.8 | 96.9 | 96.2 | 92.6 | 92.2 | 93.9 |
| 15 | 88.1 | 97.5 | 96.3 | 82.7 | 88.7 | 88.6 | 90.4 | 96.5 | 94.3 | 90.6 | 91.5 | 92.4 |
| 25 | 76.7 | 96.4 | 95.2 | 76.8 | 83.8 | 86.4 | 83.5 | 94.9 | 93.3 | 88.3 | 90.1 | 91.1 |
| 35 | 68.3 | 93.1 | 93.6 | 67.5 | 78.1 | 82.7 | 77.5 | 90.4 | 92.7 | 82.7 | 84.7 | 84.9 |
| 40 | 60.7 | 83.1 | 84.8 | 61.8 | 69.7 | 81.8 | 72.7 | 83.1 | 92.3 | 78.6 | 79.2 | 79.7 |

*Table 4*. Experimental comparison between Classification Filter and Partitioning Filter on classification accuracy (Splice, Credit-app, Connect-4 and Tic-tac-toe)

| Noise (%) | Splice (%) | | | Credit-app (%) | | | Connect-4 (%) | | | Tic-tac-toe (%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OG | CF | PF | OG | CF | PF | OG | CF | PF | OG | CF | PF |
| 5 | 89.1 | 92.6 | 91.8 | 81.9 | 85.3 | 85.6 | 73.2 | 75.8 | 75.7 | 83.5 | 83.9 | 83.8 |
| 15 | 85.6 | 92.1 | 91.4 | 73.7 | 84.6 | 86.7 | 68.2 | 74.7 | 75.1 | 76.3 | 79.2 | 78.8 |
| 25 | 82.1 | 91.2 | 89.7 | 66.7 | 83.4 | 85.2 | 61.6 | 71.8 | 72.5 | 69.1 | 72.5 | 73.4 |
| 35 | 77.6 | 89.1 | 86.4 | 61.5 | 80.5 | 83.9 | 55.8 | 68.8 | 69.7 | 61.8 | 62.6 | 64.7 |
| 40 | 75.5 | 87.4 | 80.9 | 58.2 | 79.1 | 81.4 | 51.6 | 66.5 | 67.9 | 57.8 | 61.1 | 62.7 |

*Table 5*. Experimental comparison between Classification Filter and Partitioning Filter on classification accuracy (Monks-3, IBM-Synthetic, Sick and CMC)

| Noise (%) | Monks-3 (%) | | | IBM-Synthetic (%) | | | Sick (%) | | | CMC (%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OG | CF | PF | OG | CF | PF | OG | CF | PF | OG | CF | PF |
| 5 | 96.8 | 99.2 | 97.3 | 88.5 | 92.7 | 91.8 | 97.0 | 98.1 | 98.1 | 49.2 | 52.2 | 53.5 |
| 15 | 89.2 | 98.0 | 96.9 | 83.6 | 91.4 | 90.9 | 93.2 | 97.6 | 97.9 | 48.8 | 52.5 | 52.8 |
| 25 | 82.7 | 91.9 | 90.8 | 76.4 | 89.2 | 90.3 | 91.4 | 96.3 | 95.8 | 44.9 | 49.3 | 49.7 |
| 35 | 67.3 | 79.2 | 80.1 | 63.7 | 83.6 | 80.2 | 83.7 | 95.5 | 94.7 | 42.8 | 47.1 | 47.8 |
| 40 | 63.1 | 71.4 | 67.5 | 53.1 | 63.7 | 66.3 | 77.5 | 88.6 | 86.9 | 43.3 | 46.0 | 47.6 |

suggested that 'for higher noise levels, the performance of a correct decision tree on corrupted test data was found to be inferior to that of an imperfect decision tree formed from data corrupted to a simi-

*Table 6*. Execution time comparison between Classification Filter and Partitioning Filter (Mushroom dataset)

| Methods | Execution time at different noise levels (seconds) | | | | |
|---|---|---|---|---|---|
| | 0% | 10% | 20% | 30% | 40% |
| CF | 18.2 | 159.3 | 468.6 | 868.4 | 1171.2 |
| PF | 5.3 | 12.8 | 19.8 | 22.8 | 29.6 |

lar level! The moral seems to be that it is counter-productive to eliminate noise from the attribute information in the training set if these same attributes will be subject to high noise levels when the induced decision tree is put to use'. Intuitively, it seems that this concludes that instead of bringing more benefits, more troubles would be introduced if we attempt to handle attribute noise. Nevertheless, these evaluations focused more on learning with the existence of noise, rather than from the noise handling point of view, meanwhile many issues about attribute noise remain unclear, and deserve a comprehensive evaluation.

### 5.1. *Effects of attribute noise with classification accuracy*

Our first set of experiments is executed by using a set of cross-evaluations, as shown in Figure 2. Given a dataset $D$, we first split it into a training set $X$, and a test set $Y$ (using a cross-validation mechanism). We train a classifier $C$ from $X$, use $C$ to classify instances in $Y$, and denote the classification accuracy by *CvsC* (i.e., Clean training set vs. Clean test set). We then manually corrupt each attribute with a noise $x \cdot 100\%$ and construct a noisy training set $X'$ (from $X$). We learn classifier $C'$ from $X'$, use $C'$ to classify instances in $Y$ and denote the classification accuracy by *DvsC* (i.e., Dirty training set vs. Clean test set). In addition, we also add the corresponding levels ($x \cdot 100\%$) of attribute noise into test set $Y$ to produce a dirty test set $Y'$, and use classifiers $C$ and $C'$ to classify instances in $Y'$. We denote the classification accuracies by *CvsD* and *DvsD* respectively (i.e., Clean training set vs. Dirty test set, Dirty training set vs. Dirty test set). For each dataset, we execute 10-fold cross validation 10 times, and use the average accuracy as the final result, as demonstrated in Figure 3, on 16 datasets.

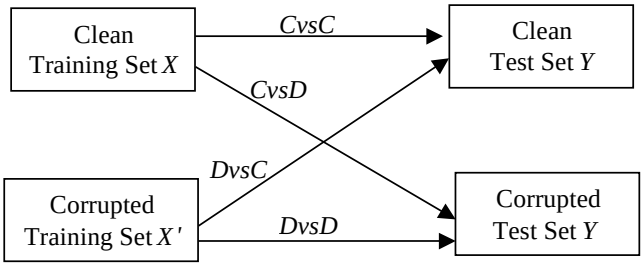


*Figure 2.* Cross-evaluations in exploring the effects of attribute noise with classification accuracy.

From the experimental results in Figure 3, we can draw several conclusions as follows:

1. The highest classification accuracy (when evaluating at different noise levels) is always from the classifier trained from the clean training set in classifying a clean test set, i.e., *CvsC*, which implies that the existence of attribute noise does bring some troubles in term of classification accuracy, even though we still do not know how attribute noise behaves with different learning algorithms and datasets. As we can see from Figure 3, when the noise level goes higher, the decreasing of classification accuracy (*CvsD*, *DvsC* or *DvsD*) can be observed from all 16 benchmark datasets, no matter whether attribute noise is introduced to the training set or test set, or both.
2. The lowest classification accuracy (when evaluating at different noise levels) usually comes from the classifier trained from the corrupted training set in classifying a corrupted test set (*DvsD*). This implies that in a noisy environment, adopting some attribute noise handling mechanisms will likely enhance the classification accuracy, in comparison with unprocessed noisy datasets.
3. If the test set does not contain any attribute noise, adopting cleaning attribute noise on the training set can always improve the classification accuracy remarkably. Comparing curves *CvsC* and *DvsC* in Figure 3, we can find that at all noise levels, the value of *CvsC* is always higher (or much higher) than the corresponding value of *DvsC*. Actually, this assumption has been implicitly taken by Teng (1999) in her noise polishing approach. However, for real-word datasets, this assumption can be too strong, and the fact is we never know whether a test set is clean or not. Therefore, a more realistic assumption is that attribute noise may exist in the test set too.
4. In the case that attribute noise exists in the test set, if we can handle (correct) attribute noise in the test set, the classification accuracy can

also be improved comprehensively, even if the classifier is trained from a noise corrupted training set. Comparing curves *DvsC* and *DvsD* in Figure 3, one can find that even though the training set remains unchanged, cleaning attribute noise from the test set can always improve the classification accuracy. The reason is that although the training set is corrupted, we can still learn a partially correct theory. When applying this theory on corrected test instances, we can still get good results, in comparison with applying this theory on corrupted test instances. However, handling noise in test instances seems odd and does not make much sense in many situations, because learning algorithm cannot simply modify the user's input to fit

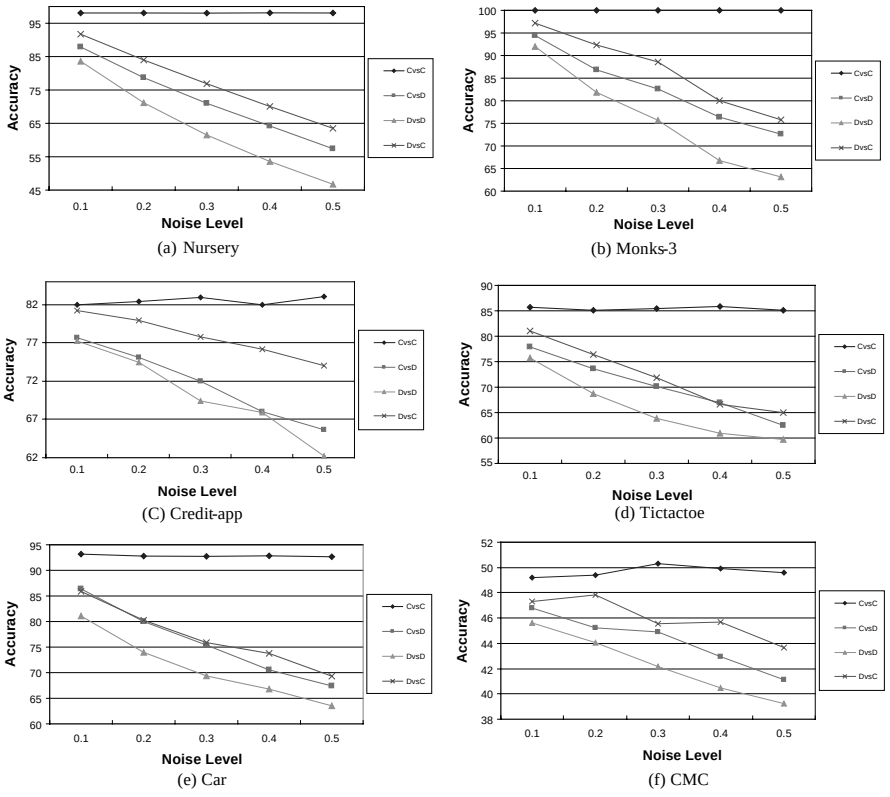


*Figure 3*. Experimental results of cross-evaluations in exploring the effects of attribute noise with classification accuracy: *x*-axis denotes the attribute noise level and *y*-axis represents the classification accuracy, each curve means the result from one methodology (as introduced in Figure 2).

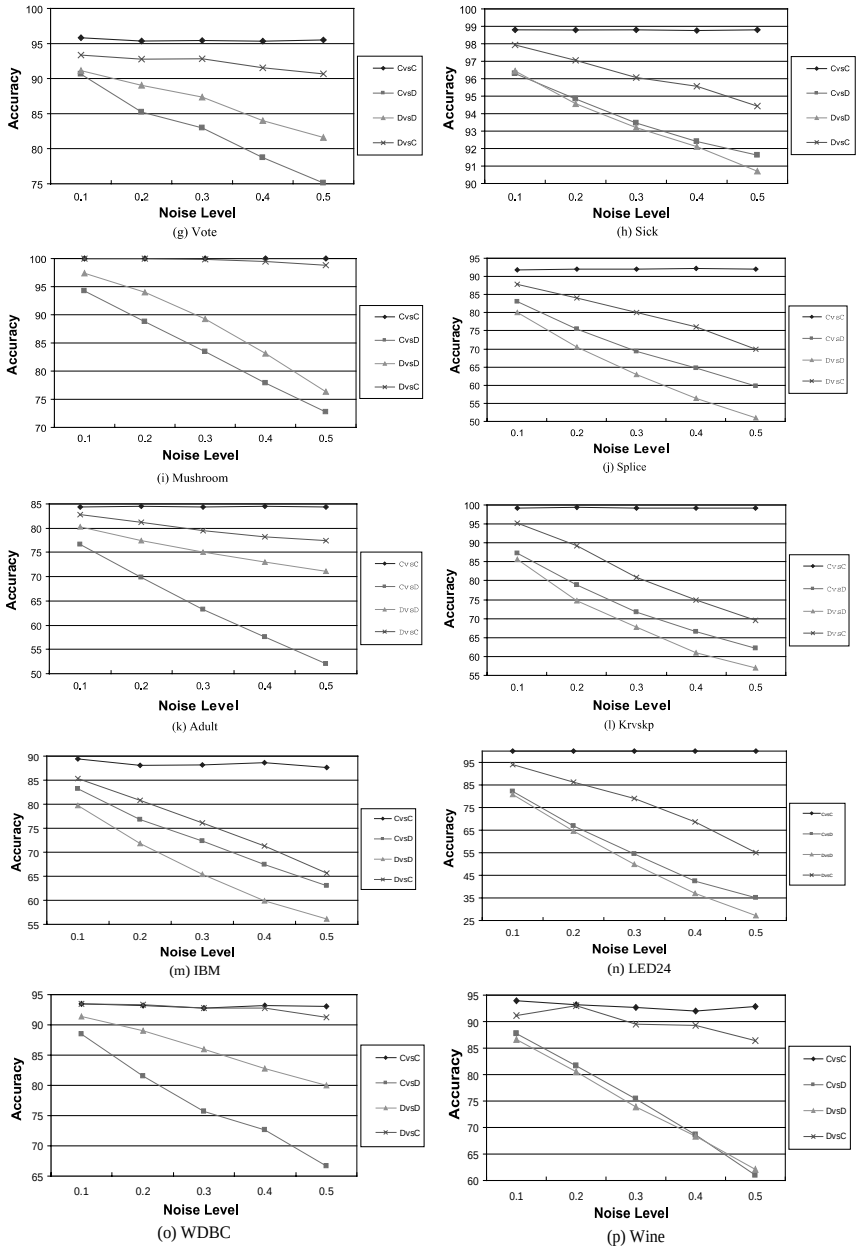


*Figure 3*. (Continued)

it with its own model, even if this model has a 100% accuracy. In the next subsection, we will discuss that noise handling in a test set can act as a data recommendation tool to enhance the data quality.

5. If we accept the restriction that the system can do nothing with the noise in the test set, cleaning noise from the training set will still have a reasonable chance to enhance the classification accuracy. Comparing curves *CvsD* and *DvsD* in Figure 3, with all 16 benchmark datasets, cleaning attribute noise from the training set has increased the classification accuracy for 12 datasets. For the other four datasets (Adult, WDBC, Mushroom, and Vote), adopting data cleaning on the training set will cause more troubles.

The above conclusions suggest that noise handling from the training set may provide a good solution in enhancing the classification accuracy. Instead of eliminating instances that contain attribute noise, correcting attribute noise seems more promising.

### 5.2. *Experimental evaluations from partially cleaned noisy datasets*

Experiments in Section 5.1 assume that we can identify and correct attribute noise from the training (or test) sets with 100% accuracy. Even though the results suggest that noise correction could benefit classification accuracy remarkably, the above assumption is simply too strong, because in many situations, we obviously cannot identify and correct all noisy instances. Accordingly, we execute another set of experiments, where we add the same level ($x \cdot 100\%$) of noise into both training and test sets, but we assume that we can only identify and clean a certain portion ($\beta \cdot 100\%$), $\beta = [0.2, 0.8]$, of attribute noise. As shown in Figure 4, the corresponding classification accuracies are denoted by *PvsP*, *PvsD*, *DvsP*, and *DvsD* respectively. The experimental results are reported in Figures 5–9, which are evaluated from 5 representative datasets.

In Figures 5–9, we set attribute noise ($x \cdot 100\%$) in original datasets (training and test sets) to two levels: $x = 0.25$ and $x = 0.4$, and randomly correct $\beta \cdot 100\%$ of the attribute noise, $\beta = [0.2, 0.8]$. We then evaluate the relationship between noise cleaning and the classification accuracy. In all figures from 5 to 9, (a) and (b) represent the results from the datasets corrupted with 25 and 40% attribute noise respectively. (We have performed experiments with other noise levels, and they basically support all conclusions below). From the results in Figures 5–9, an obvious conclusion is that *even partially correcting attribute noise can benefit the classification accuracy*.

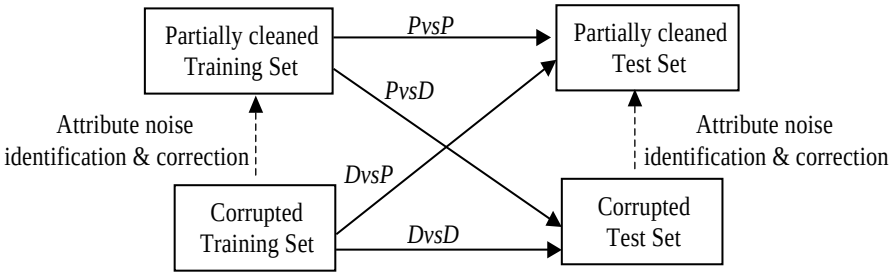


*Figure 4*. Cross evaluation in exploring the impact of attribute noise from partial cleaned dataset.

As shown in Figure 5(a) (Monks-3 dataset), when 25% attribute noise is added to both training and test sets, the classification accuracy from *DvsD* (datasets without any noise handling mechanism) is 79.34%. If we can clean 20% of the attribute noise from the training set (keeping the test set as it was), the classification accuracy (*PvsD*) increases to 81.27%. Moreover, in addition to cleaning from the training set, if we can clean 20% of attribute noise from the test set, the accuracy (*PvsP*) increases to 83.39%. When the percentage of cleaned noise goes higher and higher, more and more improvement could be achieved.

We also provide the results from an exceptional dataset – Vote, where handling attribute noise from the training set (only) likely decreases the classification accuracy. As shown in Figure 9, one can find that in the same way as we have concluded from the same dataset in Section 5.1, if we correct attribute noise from the training set only, it likely decreases classification performance. However, among all 16 benchmark datasets, only a small portion of them exhibit such an abnormal characteristic, and most support our conclusion that

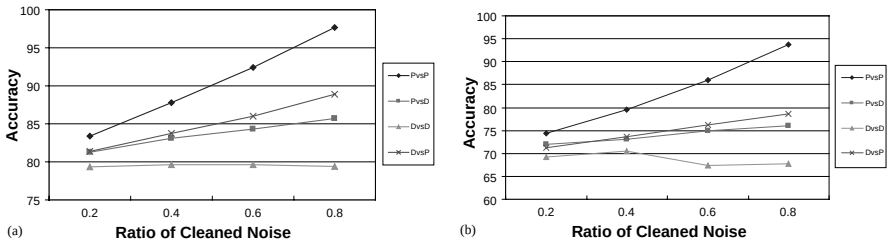


*Figure 5*. Experimental results of partial attribute noise cleaning from Monks-3 dataset: (a) the original datasets are corrupted with 25% attribute noise; (b) the original datasets are corrupted with 40% attribute noise.

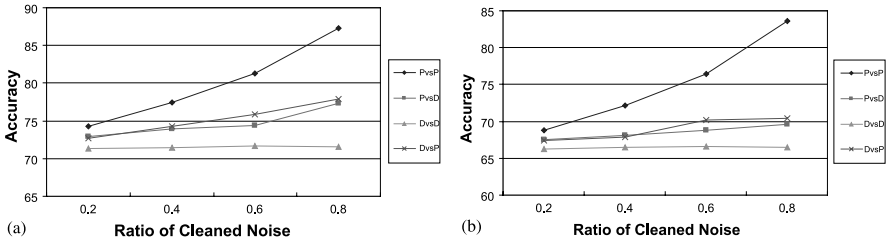


*Figure 6*. Experimental results of partial attribute noise cleaning from Car dataset: (a) the original datasets are corrupted with 25% attribute noise; (b) the original datasets are corrupted with 40% attribute noise.

correcting attribute noise from the training set likely enhances the classification accuracy.

Another interesting observation from Figures 5–9 is that, *in comparison with noise handling from the training set, correcting attribute noise from the test set usually brings more benefits (more accuracy improvement)*. Comparing curves *PvsD* and *DvsP*, on average, a 2–5% more improvement could be found from *DvsP*. It means that more improvement has been achieved through noise correction in the test set, even if the classifier is learned from a corrupted training set (without any noise handling mechanism). However, correcting the test set means that we need to modify instances in the user's hand, which seems dangerous and unreasonable. Because an algorithm can always change the user's instances to fit them with its own model from which the system has a high confidence, this may actually lose valuable information from the user. One can imagine that a classifier can change all outliers into instances that the system can classify well. However, these negative comments do not necessarily mean that we can do nothing in cleaning the test set. Actually, we can take the attribute noise correction

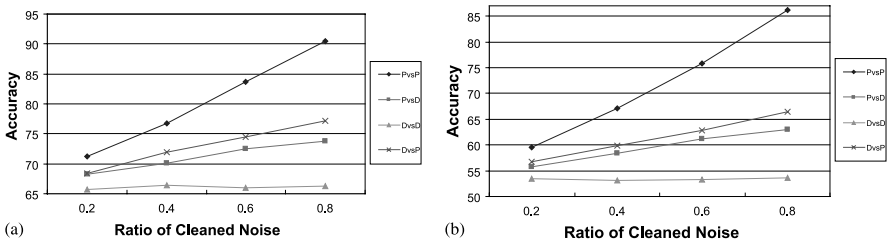


*Figure 7*. Experimental results of partial attribute noise cleaning from Nursery dataset: (a) the original datasets are corrupted with 25% attribute noise; (b) the original datasets are corrupted with 40% attribute noise.

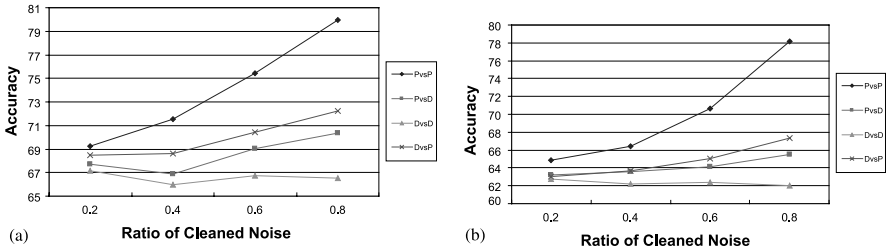


*Figure 8*. Experimental results of partial attribute noise cleaning from Tictactoe dataset: (a) the original datasets are corrupted with 25% attribute noise; (b) the original datasets are corrupted with 40% attribute noise.

mechanism as a recommendation system, provide the users with problematic instances and their attribute values, recommend and suggest the users that a more reasonable value could be assigned for the suspicious attribute, under the context of the instance. By doing this, it is the user who draws the final decision in making any change, and the system just acts as a recommendation tool. Consequently, the user could be involved in an active manner in enhancing the data quality, and obviously it's more efficient than any manual data cleansing scheme.

### 5.3. *Impact of attribute noise from different attributes*

As we have investigated above, the impact of attribute noise could be crucial in the term of the classification accuracy. Other research efforts have also indicated that the existence of attribute noise could result in a larger tree size (Teng 1999). Given all these facts, one intuitive argument might be: *if we introduce noise into attributes, does noise of different*

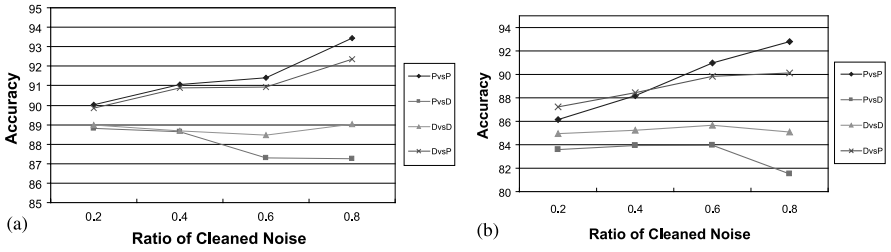


*Figure 9*. Experimental results of partial attribute noise cleaning from Vote dataset: (a) the original datasets are corrupted with 25% attribute noise; (b) the original datasets are corrupted with 40% attribute noise.

*Table 7.* $\chi^2$ values between attributes and class (Monks-3 dataset)

| $\chi^2$ | Attribute 1 | Attribute 2 | Attribute 3 | Attribute 4 | Attribute 5 | Attribute 6 |
|---|---|---|---|---|---|---|
| Class | 0.427 | 136.999 | 0.224 | 2.626 | 133.171 | 0.199 |

*Table 8.* $\chi^2$ values between attributes and class (Car dataset)

| $\chi^2$ | Attribute 1 | Attribute 2 | Attribute 3 | Attribute 4 | Attribute 5 | Attribute 6 |
|---|---|---|---|---|---|---|
| Class | 151.839 | 115.177 | 9.623 | 296.755 | 44.776 | 383.260 |

*attributes behave in the same way? If not, what's the relationship between the noise of each attribute and the system performance* ?

To explore answers for these questions, we execute the following experiments. Given a dataset $D$, we split it into a training set $X$ and a test set $Y$ (using a cross-validation mechanism). We then re-perform the experiments in Section 5.1, with the following changes:

1. When adding attribute noise, instead of introducing noise to all attributes, we corrupt only one attribute at each time, and the remaining attributes are unchanged.
2. Instead of testing all four methodologies (*DvsD*, *DvsC*, *CvsC* and *CvsD*), we only evaluate the results from *DvsD* and *DvsC*.

We have executed our experiments on all 17 benchmark datasets, and provide results from four representative datasets, which are Monks-3, Car, Nursery and Tic-tac-toe. The results are shown in Figures 10–13, where $x$-axis represents the noise levels of the attribute, $y$-axis indicates

*Table 9.* $\chi^2$ values between attributes and class (Nursery dataset)

| $\chi^2$ | Att1 | Att2 | Att3 | Att4 | Att5 | Att6 | Att7 | Att8 |
|---|---|---|---|---|---|---|---|---|
| Class | 954.62 | 2512.76 | 79.31 | 175.46 | 265.48 | 61.11 | 241.51 | 11084.76 |

*Table 10.* $\chi^2$ values between attributes and class (Tictactoe dataset)

| $\chi^2$ | Att1 | Att2 | Att3 | Att4 | Att5 | Att6 | Att7 | Att8 | Att9 |
|---|---|---|---|---|---|---|---|---|---|
| Class | 15.38 | 8.39 | 16.22 | 8.83 | 92.30 | 8.16 | 15.04 | 7.44 | 15.76 |

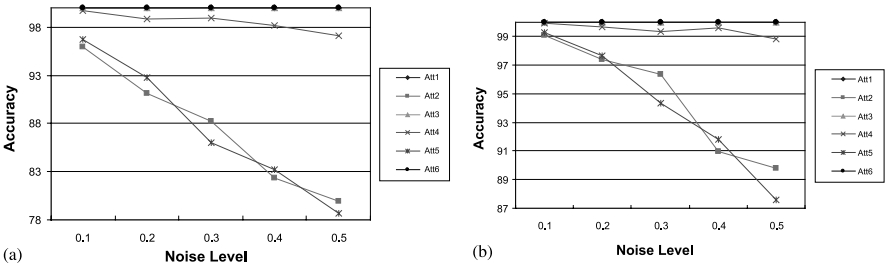


*Figure 10*. The impact of attribute noise at different attributes vs. the system performance from Monks-3 dataset, where *AttX* means (only) the attribute *X* is corrupted: (a) *DvsD* – Noisy training set vs. Noisy test set; (b) *DvsC* Noisy training set vs. Clean test set.

the corresponding classification accuracy, and each curve in the figures represents the results evaluated from one attribute. From results in Figures 10–13, we may find that noise has various impacts with different attributes. Comparing different attributes with the same noise level, it's obvious that some attributes are more sensitive to noise, i.e., introducing a small portion of noise could decrease the classification accuracy significantly, such as attributes 2 and 5 in Figure 10. On the other hand, introducing noise to some attributes does not have much influence with the accuracy (even not at all), such as attributes 1, 3 and 6 in Figure 10.

However, until now, the intrinsic relationship between noise of each attribute and the classification accuracy is unclear, and we still have no idea about what types of attributes are sensitive to noise and why they are more sensitive than others. Therefore, we adopt the $\chi^2$ test ($\chi^2$) from statistics (Everitt 1977) to analyze the correlations between each attribute and the class label. Essentially, the test is a widely used method for

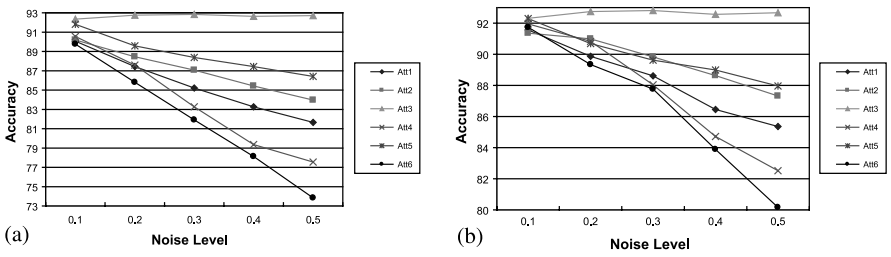


*Figure 11*. The impact of attribute noise at different attributes vs. the system performance from Car dataset, where *AttX* means (only) the attribute *X* is corrupted: (a) *DvsD* – Noisy training set vs. Noisy test set; (b) *DvsC* Noisy training set vs. Clean test set.

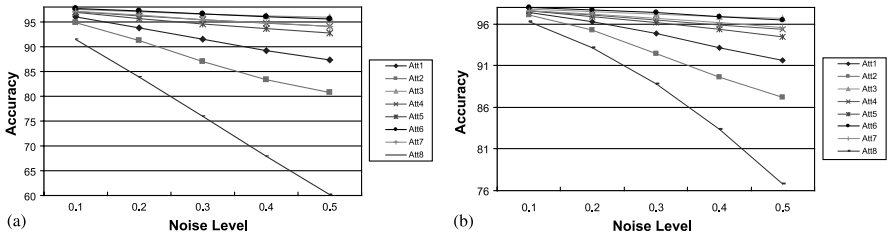


*Figure 12.* The impact of attribute noise at different attributes vs. the system performance from Nursery dataset, where *AttX* means (only) the attribute *X* is corrupted: (a) *DvsD* – Noisy training set vs. Noisy test set; (b) *DvsC* Noisy training set vs. Clean test set.

testing independence and/or correlation between two vectors. It is based on the comparison of observed frequencies with the corresponding expected frequencies. The closer the observed frequencies are to the expected frequencies, the greater is the weight of evidence in favor of independence. As shown in Equation (1), let $f_0$ be an observed frequency, and $f$ be an expected frequency. The $\chi^2$ value is defined by Equation (1):

$$\chi^2 = \sum \frac{(f_0 - f)^2}{f}. \tag{1}$$

A $\chi^2$ value of 0 implies the corresponding two vectors are statistically independent with each other. If it is higher than a certain threshold value (e.g., 3.84 at the 95% significance level (Everitt 1977)), we usually reject the independence assumption between two vectors. In other words, the higher the $\chi^2$ value, the higher the correlation between the corresponding vectors.

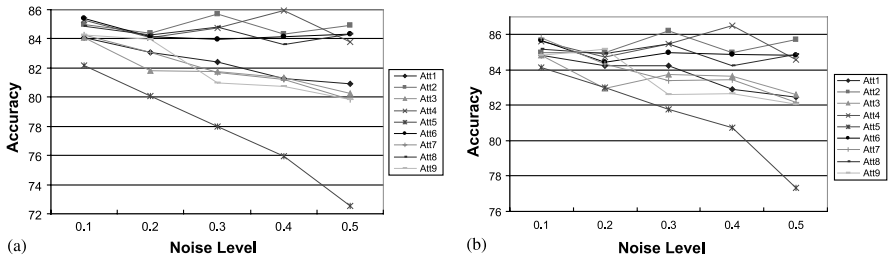


*Figure 13.* The impact of attribute noise at different attributes vs. the system performance from Tictactoe dataset, where *AttX* means (only) the attribute *X* is corrupted: (a) *DvsD* – Noisy training set vs. Noisy test set; (b) *DvsC* Noisy training set vs. Clean test set.

To execute the $\chi^2$ test between an attribute ($A_i$) and the class label ($C$), we take each of them as a vector, and calculate how many instances contain the corresponding values. For any dataset, we execute the $\chi^2$ test between each attribute and class, and provide the results in Tables 7–10. After we compare the results from Figures 10–13 and the corresponding $\chi^2$ values in Tables 7–10, some interesting conclusions can be drawn as follows:

1. The noise of different attributes has different impact with the system performance. The impact of the attribute noise critically depends on the dependence between the attribute and class.
2. Given an attribute $A_i$ and class $C$, the higher the correlation between $A_i$ and $C$, the more impact could be found from this attribute ($A_i$), if we introduce noise into $A_i$. As demonstrated in the Car dataset (Figure 11), where attribute 6 has the highest $\chi^2$ value with $C$, adding noise into attribute 6 has the largest impact (in the term of the accuracy decrease) in comparison with all other attributes (when the same noise level is added to each attribute). The same conclusion could be drawn from all other datasets.
3. If attribute $A_i$ has very small correlation with class (or not at all), introducing noise into $A_i$ usually has not much impact with the system performance. As demonstrated in the Monks-3 dataset (Figure 10), where attributes 1, 3 and 6 have very small $\chi^2$ values with the class (according to the assumption of Everitt (1977), all these three attributes are independent with the class C). Adding noise into these three attributes have no impact with the system performances, i.e., no matter how much noise has been introduced into these attributes, it would not affect the classification accuracy. Also, the same conclusion could be drawn from all other datasets.

The above conclusions indicate that the impact of noise from different attributes varies significantly with the classification accuracy, determined by the correlation between the corresponding attribute and class. This implies that when handling attribute noise, it's not necessary to deal with all attributes, and we may focus on some noise sensitive attributes only.

### 5.4. *Attribute noise vs. class noise: which is more harmful?*

As we have indicated in the above sections, both attribute noise and class noise could bring negative impacts with the classification accuracy. We have also concluded that noise from different attributes varies

significantly with the system performance. Then, one intuitive question might be: in comparison with attribute noise and class noise, which is more harmful? To resolve this question, we execute the following experiments.

Given dataset $D$, we split it into a training set $X$ and a test set $Y$ (using a cross-validation mechanism), and then adopt the following five mechanisms:

1. We corrupt the class in $X$ only (using the mechanism introduced in Section 3) to construct a noisy dataset $X'$, and use the classifier learned from $X'$ to classify instances in $Y$. We denote the accuracy from this approach by the 'Class' curve in Figure 14.
2. We corrupt the most noise-sensitive attribute in $X$ only. The most noise-sensitive attribute means the attribute that has the highest Chi-square value with the class (as analyzed in Section 5.3). We use the acquired classifier to classify instances in $Y$, and denote the accuracy by the '*Att_S*' curve in Figure 14.
3. We corrupt the most noise-sensitive attribute in $X$, and in addition, the same level of attribute noise is introduced into the same attribute of $Y$. We denote the accuracy from this approach by the '*Att_S_Test*' curve in Figure 14.
4. We corrupt all attributes in $X$, and denote the accuracy from this approach by '*Att_A*' in Figure 14.
5. We corrupt all attributes in $X$, and in addition, the same noise level is introduced to all attributes of $Y$. We denote the accuracy from this approach by '*Att_A_Test*' in Figure 14.

In Figure 14, if we assume the test set $Y$ does not contain any noise, then the curves 'Class', 'Att_S' and 'Att_A' will indicate such scenarios. Comparing all these tree curves, we can find that the values of 'Class' are almost always the lowest at any noise level. This indicates that when the test set does not contain noise, class noise is more harmful than attribute noise, no matter whether the attribute noise is introduced to a single attribute or all attributes. When the test set $Y$ does contain certain levels of noise, we may find that the influence of attribute noise could be more severe than class noise. As shown in Figure 14(b), when the noise level is less than 30%, introducing noise to all attributes in the training and test sets (*Att_A_Test*) shows more negative impacts than introducing the same level of class noise. However, when the noise level goes higher, the impact of class noise becomes worse. In Figure 14(c), we find that the class noise is only worse than '*Att_S*', which is actually misled by our class noise corruption mechanism. When

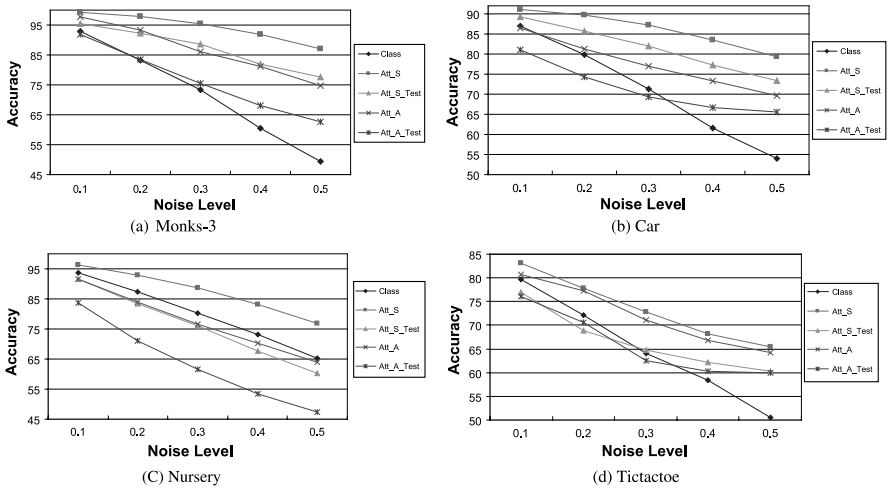


*Figure 14*. Experimental comparisons of impacts of attribute noise and class noise with classification accuracy, where (a), (b), (c) and (d) represent the results evaluated from four different datasets.

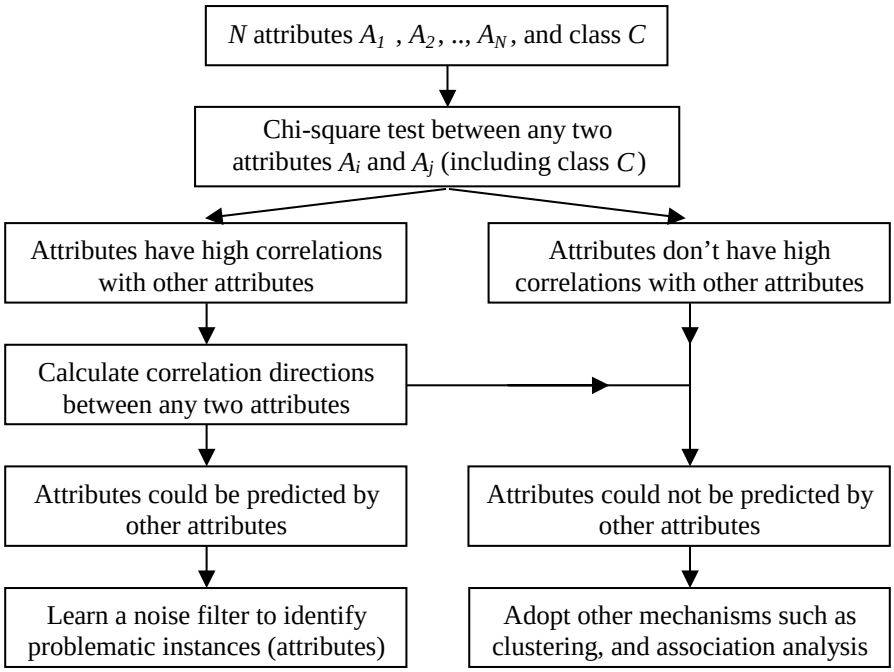


*Figure 15*. A systematic analysis in handling attribute noise.

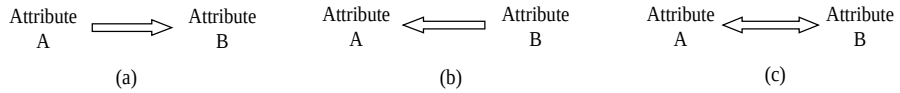


*Figure 16*. Correlation directions between two attributes: (a) one-way correlation from *A* to *B*; (b) one-way correlation from *B* to *A*; (c) dual correlation between *A* and *B*.

introducing class noise, we adopt a pair-wise mechanism, which corrupts only one pair of classes (usually the pair of classes having the highest proportion of instances). With the 'Nursery' dataset, it has five classes, and the selected pair of classes contains less than 50% of instances, so even we introduce 50% of class noise, the actually noise level is just about half of the intended noise level. However, even in this case, we can still find that class noise is worse than '*Att_S*', and almost equal to '*Att_A*'.

The results from Figure 14 suggest that in many situations, class noise is likely more severe than attribute noise, especially when the test set does not contain noise or we have a very limited number of class labels. The answer for why attribute noise looks less harmful than class noise may come from the following two reasons:

1. In comparison with the unique class label, we usually have multiple attributes, and the number of attributes could vary from several to more than a hundred. Consequently, the noise introduced from one or a limited number of attributes may have a very limited impact with the system performance.
2. When adopting various learning algorithms, the attributes vary differently in terms of importance in constructing the classifier. There is no doubt that the class label is the most important one, and consequently, the noise introduced to class attribute will have the largest impact; on the other hand, some attributes may have a very limit contribution in constructing the classifier (or even not at all), and the noise introduced into these attributes will have a very limited impact on the performance of the classifier. Therefore, on average, attribute noise is less harmful than class noise.

### 5.5. *Discussion on attribute noise handling*

After the above systematic and extensive evaluations, we need to answer the following question: Given instances in a dataset, how can we possibly identify problematic attributes and handle the noise to enhance the

data quality?' The question is more oriented from a noise-cleansing point of view: Given a dataset $D$, how can we possibly adopt various techniques to identify noisy instances? Meanwhile, instead of eliminating them, we will try to predict and assign a 'correct' (or less harmful) value for those problematic attributes. In this paper, we will not provide any solid solution in handling attribute noise, instead, we provide a systematic analysis on design different approaches.

In Section 3, we have mentioned that researches on identifying instances with class noise has been widely conducted, where the concept of 'noise filter' (Brodley and Friedl 1996) has been introduced. Given a dataset $D$, we can split it into $N$ subsets, a classifier ($C_k$) trained from the aggregation of any $N-1$ subsets is used to identify instances in the excluded subset. A noisy instance is an example with its class label different from the classification of the classifier $C_k$. Based on existing research efforts, one intuitive solution in identifying and correcting attribute noise may be taking each attribute as the class label and applying a 'noise filter' on each of them.

Given a dataset $D$, with an instance $I_k$ in $D$ denoted by $N$ attribute $A_1, A_2, \ldots, A_N$ and one class label $C$, we first split $D$ into a training set $X$ and a test set $Y$ (using cross-validation). To handle attribute noise contained in attribute $A_1$, we switch class $C$ with attribute $A_1$, and use attributes $A_2, A_3, \ldots, A_N$ plus $C$ as the attributes (so we still have $N$ attributes). We train a classifier $T_1'$ by using these $N$ attributes (and all instances in $X$), and then apply this classifier on $Y$ to predict the value of attribute $A_1$ of all instances in $Y$ (at the current stage, we consider nominal attribute values only). By applying the same mechanism on all attributes $A_2, A_3, \ldots, A_N$, we can acquire the corresponding classifiers $T_2', T_3', \ldots, T_N'$. Given an instance $I_k$ in the test set, if its attribute value on $A_i$ is different from the classification result of the classifier, e.g $T_i'$, it will imply that the attribute value of $A_i$ of $I_k$ is likely problematic. And accordingly, we may use the classification result of $T_i'$ as the 'corrected' value for $A_i$ of $I_k$. A similar mechanism has been conducted by Teng (1999) in her data polishing scheme.

Data polishing (Teng 1999) is attractive and efficient in handling many datasets, but its performance from many other datasets could be problematic, especially when we take a closer look with the mechanism itself. To identify and correct noise from an attribute, e.g., $A_i$, the mechanism switches $A_i$ with the class, and uses all other attributes plus the class to train a classier which is used to predict the 'correct' value of $A_i$. Unfortunately, this procedure may seriously break the assumptions of classification which were stated in Section 2:

(a) The attributes should somewhat correlate to the class.
(b) The attributes are conditionally independent with each other.

By switching the class with $A_i$, we may find that neither of the assumptions above could be guaranteed from the newly constructed dataset. If that's the case, how can we possibly trust that classifier $T_i'$ can make right predictions? When we construct a dataset for classification purpose, the above two assumptions play an important role with the classification accuracy. Accordingly, violating these assumptions by switching the class with $A_i$, we may find that there may not many attributes that correlate with the newly constructed class ($A_i$), if not at all. In such a scenario, it's actually not a classification problem, then how can we possibly adopt classification mechanisms to handle it?

Accordingly, we need a systematic analysis in guiding attribute noise handling. As shown in Figure 14, given a dataset $D$ which has $N$ attributes, our objective is to separate all attributes into two categories: Attributes which could be predicted by other attributes, and attributes which could not be predicted by other attributes. For attributes which belong to the first category, we may adopt a noise filtering mechanism to figure out problematic instances (or attributes), and for attributes which could not be predicted by other attributes, the classification mechanism may not work (or cannot work out good results), and we may need to consider other possible solutions such as clustering (Davé 1991), $k$ nearest neighborhood (Huang and Lee 2001), or association analysis (Ragel and Cremilleus 1999), which have been suggested for missing values prediction and outliers detection. Interested readers may refer to a noise handling biography (Höppner 2003) for more noise handling references in this regard.

To facilitate our objective in Figure 15, we first execute a $\chi^2$ test between any two attributes (including the class label). Those attributes, which have low correlations with all other attributes, are first selected, because having low correlations with others implies that they cannot be predicted by using any learning theory with a high accuracy. For those remaining attributes, just because they have high correlations with others, does not necessarily mean that they could be predicted, because correlations could happen in just one way, i.e., given $A$ we can predict $B$ does not imply that given $B$ we can predict $A$ too. So we need to figure out the correlation directions between attributes. Given two attributes $A$ and $B$, we can possibly classify their correlation directions into three categories: One-way correlation from $A$ to $B$, one-way correlation from $B$ to $A$, and dual correlations between $A$ and $B$, as shown in Figure 16.

To classify the correlation direction between $A$ and $B$, we adopt the Information Gain (Hunt et al. 1966) measure, also known as mutual information between $A$ and $B$, given by Equation (2).

$$\text{Gain}_B(A) = I(A;B) = \sum_a \sum_b P(a,b) \log \frac{P(a,b)}{P(a)P(b)}, \qquad (2)$$

where $a$ and $b$ indicate the attribute values of attributes $A$ and $B$ respectively. The information gain can be regarded as a measure of the strength of a 2-way interaction between $A$ and $B$. The larger the value $\text{Gain}_B(A)$, the more confident the one-way correlation from $A$ to $B$ will hold. The same conclusion could be drawn from $\text{Gain}_A(B)$ (it is likely that $\text{Gain}_B(A) \neq \text{Gain}_A(B)$ in most cases). Consequently, a threshold mechanism will indicate the correlation direction between any two attributes. After we determine the correction directions between any two attributes, an attribute that has no correction direction towards it will be removed and taken as one of the attributes which could not be predicted by classification mechanisms. All other remaining attributes would be taken as the predictable attributes (worse or better) by employing noise filtering approaches.

To figure out whether an attribute could be predicted by other attributes or not, the data polishing mechanism (Teng 1999) adopted an accuracy-oriented mechanism. It first switches attribute $A_i$ and class attribute $C$, and constructs a classifier $T_i$. If the accuracy of $T_i$ is relatively high, it will conclude that $A_i$ is predictable by other attributes. Basically, this mechanism works in may situations, because if $A_i$ has higher correlations with other attributes, $T_i$ likely has a higher accuracy. However, all these evaluations are based on the performance of the adopted learning algorithm, where various features, such as high noise levels and different learning theories, could impact the accuracy of $T_i$. With the analysis above, we have adopted a statistical tool to indicate whether an attribute is predictable or not in advance. Consequently, it provides some benchmarks in guiding attribute noise handling, and it is more reliable than polishing like mechanisms.

## 6. Conclusions

The focus of this paper was to evaluate the impact of noise on learning, by measuring two types of noise, class noise and attribute noise, from 17 datasets. We demonstrated that the impacts of noise are severe in

many circumstances. In addition to investigating the role of noise in learning, we have put much emphasis on how to handle different types of noise. Meanwhile, the paper has paid more attention to attribute noise than class noise, because the later has been extensively addressed in the literature. The conclusions from our experiments can be summarized as follows:

1. Eliminating instances containing class noise will likely enhance the classification accuracy.
2. In comparison with class noise, the attribute noise is usually less harmful, but could still bring severe problems to learning algorithms.
3. When handling attribute noise, noise correction will likely enhance the accuracy of learned classifiers.
4. In comparison with noise handling from the training set, cleaning noise from the test set usually brings more benefits (in terms of classification accuracy), even if the classifier is learned from a noise corrupted training set (without any noise handling mechanisms).
5. In the case that noise handling from a test set is forbidden, cleaning attribute noise from a training set will still likely enhance the classification accuracy comprehensively, no matter whether the test set contains noise or not.
6. In most situations, the noise from different attributes behaves differently with the system performance. The higher the correlation between an attribute and the class, the more negative impact the attribute noise may bring. Accordingly, it is not necessary for a noise handling mechanism to take care of every attribute, and handing noise on noise-sensitive attributes would be more important.
7. To identify and correct attribute noise, we can adopt some learning algorithms to learn a noise filter. However, analyzing correlations among attributes in advance is necessary in this case, and it could tell whether a specific attribute is predictable by using other attributes and the class, because an attribute with low correlations with others simply cannot be predicted by any learning theory.
8. More experiments should be conducted on identifying and correcting those attributes that have low correlations with others.

With these conclusions, instead of adopting some 'blind' noise handling mechanisms, interested readers can design their own noise handling approaches to enhance data quality from their own perspectives.

## Acknowledgements

This research is supported by the U.S. Army Research Laboratory and the U.S. Army Research Office under grant number DAAD19-02-1-0178.

## References

Allison, P. D. (2002). *Missing Data*. Thousand Oaks, CA: Sage.

Bansal, N., Chawla, S. & Gupta, A. (2000). Error Correction in Noisy Datasets Using Graph Mincuts. *Project Report, Carnegie Mellon University*, http://www.cs.cmu.edu/~15781/web/proj/chawla.ps.

Batista, G. & Monard, M. C. (2003). An Analysis of Four Missing Data Treatment Methods for Supervised Learning. *Applied Artificial Intelligence* **17**: 519–533.

Blake, C. L. & Merz, C. J. (1998). UCI Repository of Machine Learning Databases, http://www.ics.uci.edu/~mlearn/MLRepository.html

Brodley, C. E. & Friedl, M. A. (1996). Identifying and Eliminating Mislabeled Training Instances. *Proc. of 13th National Conf. on Artificial Intelligence*, 799–805.

Brodley, C. E. & Friedl, M. A. (1999). Identifying Mislabeled Training Data. *Journal of Artificial Intelligence Research* **11**: 131–167.

Bruha, I. & Franek, F. (1996). Comparison of Various Routines for Unknown Attribute Value Processing the Covering Paradigm. *International Journal of Pattern Recognition and Artificial Intelligence* **10**(8): 939–955.

Bruha, I. (2002). Unknown Attributes Values Processing by Meta-learner. *Foundations of Intelligent Systems*, *13th International Symposium*, 451–461.

Cendrowska, J. (1987). Prism: An Algorithm for Inducing Modular Rules. *International Journal of Man-Machines Studies* **27**: 349–370.

Clark, P. & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning* **3**(4): 261–283.

Cohen, J. & Cohen, P. (1983). *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences* (2nd ed.), Hillsdale, NJ: Erlbaum.

Davé, R. (1991). Characterization and Detection of Noise in Clustering. *Pattern Recognition Letter* **12**: 657–664.

Domingos, P. & Pazzani, M. (1996). Beyond Independence: Conditions for the Optimality of Simple Bayesian Classifier. *In Proceedings of the 13th International Conference on Machine Learning*, pp. 105–112.

Everitt, B. S. (1977). *The Analysis of Contingency Tables*. Chapman and Hall.

Freitas, A. (2001). Understanding the Crucial Role of Attribute Interactions in Data Mining. *Artificial Intelligence Review* **16**(3): 177–199.

Gamberger, D., Lavrac, N. & Groselj, C. (1999). Experiments with Noise Filtering in a Medical Domain. *Proc. of 16th ICML Conference*, *San Francisco*, *CA*, 143–151.

Gamberger, D., Lavrac, N. & Dzeroski, S. (2000). Noise Detection and Elimination in Data Preprocessing: experiments in medical domains. *Applied Artificial Intelligence* **14**: 205–223.

Guyon, I., Matic, N. & Vapnik, V. (1996). Discovering Information Patterns and Data Cleaning. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, pp. 181–203.

Hickey, R. (1996). Noise Modeling and Evaluating Learning from Examples. *Artificial Intelligence* **82**(1–2): 157–179.

Holte, R. C. (1993). Very Simple Classification Rules Perform well on Most Commonly Used Datasets. *Machine Learning* **11**: 1993.

Höppner, F. (2003). A Biography Index of References Related to Noise Handling, http://public.rz.fhwolfenbuettel. de/~hoeppnef/bib/keyword/NOISE-HANDLING.html

Howell, D. C. (2002). Treatment of Missing Data, *Technical Report, University of Vermont,* http://www.uvm. edu/~dhowell/StatPages/More_Stuff/Missing_Data/Missing.html

Huang, C. & Lee, H. (2001). A grey-based Nearest Neighbor Approach for Predicting Missing Attribute Values. *Proc. of 2001 National Computer Symposium*, Taiwan, NSC-90-2213-E-011-052.

Hunt, E. B., Martin, J. & Stone, P. (1966). *Experiments in Induction*. New York: Academic Press.

IBM Synthetic Data. IBM Almaden Research, Synthetic classification data generator, http://www.almaden.ibm.com/software/quest/Resources/datasets/syndata.html#classSynData

John, G. H. (1995). Robust Decision Trees: Removing Outliers from Databases. *Proc. of the First International Conference on Knowledge Discovery and Data Mining*. AAAI Press, pp. 174–179.

Kubica, J. & Moore, A. (2003). Probabilistic Noise Identification and Data Cleaning. *Proceedings of Third IEEE International Conference on Data Mining, Florida.*

Langley, P., Iba, W. & Thompson, K. (1992). An Analysis of Bayesian Classifiers. *Proceedings of AAAI-92*, 223–228.

Little, R. J. A. & Rubin, D. B. (1987). *Statistical Analysis with Missing Data*. Wiley: New York.

Maletic, J. & Marcus, A. (2000). Data Cleansing: Beyond Integrity Analysis. *Proceedings of the Conference on Information Quality* (IQ2000).

Oak, N & Yoshida, K. (1993). Learning regular and irregular examples separately. *Proc. of IEEE International Joint Conference on Neural Networks*, 171–174.

Oak, N. & Yoshida, K. (1996). A noise-tolerant hybrid model of a global and a local learning model. *Proc. of AAAI-96 Workshop: Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithm*, 95–100.

Orr, K. (1998). Data Quality and Systems Theory. *CACM* **41**(2): 66–71.

Quinlan, J. R. (1983). Learning from Noisy Data. *Proceedings of the Second International Machine Learning Workshop*, University of Illinois at Urbana-Champaign.

Quinlan, J. R. (1986a). Induction of Decision Trees. *Machine Learning* **1**(1): 81–106.

Quinlan, J. R. (1986b). The Effect of Noise on Concept Learning. In Michalski, R. S., Carboneel, J. G. & Mitchell, T. M. (eds.), *Machine Learning*, Morgan Kaufmann.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Quinlan, J. R. (1989). Unknown Attribute Values in Induction. *Proceedings of 6th International Workshop on Machine Learning*, 164–168.

Ragel, A. & Cremilleus, B. (1999). MVC – a preprocessing method to Deal with Missing Values. *Knowledge-Based Systems*, 285–291.

Redman, T. (1998). The Impact of Poor Data Quality on the Typical Enterprise. *CACM* **41**(2): 79–82.

Redman, T. (1996). *Data Quality for the Information Age*. Artech House.

Schaffer, C. (1992). Sparse Data and the Effect of Overfitting Avoidance in Decision Tree Induction. *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI), San Jose, CA*. pp. 147–152.

Schaffer, C. (1993). Overfitting Avoidance as Bias. *Machine Learning* **10**: 153–178.

Srinivasan, A., Muggleton, S. & Bain, M. (1992). Distinguishing Exception from Noise in Non-monotonic Learning. *Proc. of 2nd Inductive Logic Programming Workshop*, pp. 97–107.

Teng, M. (1999). Correcting Noisy Data. *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 239–248.

Wang, R., Storey, V. & Firth, C. (1995). A Framework for Analysis of Data Quality Research. *IEEE Transactions on Knowledge and Data Engineering* **7**(4): 623–639.

Wang, R., Strong, D. & Guarascio, L. (1996). Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems* **12**(4): 5–34.

Weisberg, S. (1980). *Applied Linear Regression*. John Wiley and Sons, Inc.

Wu, X. (1995). *Knowledge Acquisition from Databases*. Ablex Pulishing Corp.

Zhao, Q. & Nishida, T. (1995). Using Qualitative Hypotheses to Identify Inaccurate Data. *Journal of Artificial Intelligence Research* **3**, pp.119–145.

Zhu, X., Wu, X. & Chen, S. (2003a). Eliminating class noise in large datasets. *Proceedings of the 20th ICML International Conference on Machine Learning, Washington D.C*. pp. 920–927.

Zhu, X., Wu, X. & Chen, Q. (2003b). Identifying Class Noise in Large, Distributed Datasets. *Technical Report, University of Vermont*, http://www.cs.uvm.edu/tr/CS-03-12.shtml.

Zhu, X., Wu, X. & Yang, Y. (2004). Error Detection and Impact-sensitive Instance Ranking in Noisy Datasets. In *Proceedings of 19th National conference on Artificial Intelligence (AAAI-2004), San Jose, CA*.