A Comparison of Noise Handling Techniques

Choh Man Teng
cmteng@ai.uwf.edu

Institute for Human and Machine Cognition
University of West Florida
40 South Alcaniz Street, Pensacola FL 32501 USA

Abstract

Imperfections in data can arise from many sources. The quality of the data is of prime concern to any task that involves data analysis. It is crucial that we have a good understanding of data imperfections and the effects of various noise handling techniques. We study here a number of noise handling approaches, namely, robust algorithms that are tolerant of some amount of noise in the data, filtering that eliminates the noisy instances from the input, and polishing which corrects the noisy instances rather than removing them. We evaluated the performance of these approaches experimentally. The results indicated that in addition to the traditional approach of avoiding overfitting, both filtering and polishing can be viable mechanisms for reducing the negative effects of noise. Polishing in particular showed significant improvement over the other two approaches in many cases, suggesting that even though noise correction adds considerable complexity to the task, it also recovers information not available with the other two approaches.

Introduction

Imperfections in data can arise from many sources, for instance, faulty measuring devices, transcription errors, and transmission irregularities. Except in the most structured and synthetic environment, it is almost inevitable that there is some noise in any data we have collected. Data quality is crucial to any task that involves data analysis, and in particular in the domains of machine learning and knowledge discovery, where we have to deal with copious amounts of data. It is thus essential that we have a good understanding of data imperfections and the effects of various noise handling techniques.

We have identified three main approaches to coping with noise, namely, robust algorithms, filtering, and correction. In this paper we study these three approaches experimentally, using a representative method from each approach for evaluation. The effectiveness of the three methods are compared in the setting of classification.

Below we will first discuss the three general approaches to noise handling, and their respective advantages and disadvantages. We will describe one of the more novel approaches, *data polishing*, in more detail. Then we will outline the setup for experimentation, and report the results on predictive accuracy and size of the classifiers built by the three methods in our study. Additional observations are given in the last section.

Approaches to Noise Handling

Noise in a data set can be dealt with in three broad ways. We may leave the noise in, filter it out, or correct it. In the first approach, the data set is taken as is, with the noisy instances left in place. Algorithms that make use of the data are designed to be *robust*; that is, they can tolerate a certain amount of noise. This is typically accomplished by avoiding overfitting, so that the resulting classifier is not overly tuned to account for the noise. This approach is taken by, for example, c4.5 (Quinlan 1987) and CN2 (Clark & Niblett 1989).

In the second approach, the data is *filtered* before being used. Instances that are suspected of being noisy according to certain evaluation criteria are discarded (John 1995; Brodley & Friedl 1996; Gamberger, Lavrač, & Džeroski 1996). A classifier is then built using only the retained instances in the smaller but cleaner data set. Similar ideas can be found in robust regression and outlier detection techniques in statistics (Rousseeuw & Leroy 1987).

In the third approach, the noisy instances are identified, but instead of tossing them out, they are repaired by replacing the corrupted values with more appropriate ones. The corrected instances are then reintroduced into the data set. One such method, called *polishing*, has been investigated in (Teng 1999; 2000).

There are pros and cons to adopting any one of these approaches. Robust algorithms do not require preprocessing of the data, but a classifier built from a noisy data set may be less predictive and its representation may be less compact than it could have been if the data were not noisy. By filtering out the noisy instances from the data, there is a tradeoff between the amount of information available for building the classifier and the amount of noise retained in the data set. Data polishing, *when carried out correctly*, would preserve the maximal information available in the data, approximating the noise-free ideal situation. The benefits are great, but so are the associated risks, as we may inadvertently introduce undesirable features into the data when we attempt to correct it.

We will first outline the basic methodology of polishing in the next section, and then describe the experimental setup for comparing these three approaches to noise handling.

Polishing

Traditionally machine learning methods such as the naive Bayes classifier typically assume that different components of a data set are (conditionally) independent. It has often been pointed out that this assumption is a gross oversimplification; hence the word "naive" (Mitchell 1997, for example). In many cases there is a definite relationship within the data; otherwise any effort to mine knowledge or patterns from the data would be ill-advised.

Polishing takes advantage of this interdependency between the components of a data set to identify the noisy elements and suggest appropriate replacements. Rather than utilizing the features only to predict the target concept, we can just as well turn the process around and utilize the target together with selected features to predict the value of another feature. This provides a means for identifying noisy elements together with their correct values. Note that except for totally irrelevant elements, each feature would be at least related to some extent to the target concept, even if not to any other features.

The basic algorithm of polishing consists of two phases: prediction and adjustment. In the *prediction* phase, elements in the data that are suspected of being noisy are identified together with a nominated replacement value. In the adjustment phase, we selectively incorporate the nominated changes into the data set. In the first phase, the predictions are carried out by systematically swapping the target and particular features of the data set, and performing a ten-fold classification using a chosen classification algorithm for the prediction of the feature values. If the predicted value of a feature in an instance is different from the stated value in the data set, the location of the discrepancy is flagged and recorded together with the predicted value. This information is passed on to the next phase, where we institute the actual adjustments.

Since the polishing process itself is based on imperfect data, the predictions obtained in the first phase can contain errors as well. We should not indiscriminately incorporate all the nominated changes. Rather, in the second phase, the *adjustment* phase, we selectively adopt appropriate changes from those predicted in the first phase, using a number of strategies to identify the best combination of changes that would improve the fitness of a datum. We perform a ten-fold classification on the data, and the instances that are classified incorrectly are selected for adjustment. A set of changes to a datum is acceptable if it leads to a correct prediction of the target concept by all ten classifiers obtained from the ten-fold process.

Further details of polishing can be found in (Teng 1999; 2000).

Experimental Setup

Below we report on an experimental study of three representative mechanisms of the the noise handling approaches we have discussed, and compare their performance on a number of test data sets.

The basic learning algorithm we used is c4.5 (Quinlan 1993) the decision tree builder. Three noise handling mechanisms were evaluated in this study.

Robust : c4.5, with its built in mechanisms for avoiding overfitting. These include, for instance, post-pruning, and stop conditions that prevent further splitting of a leaf node.

Filtering : Instances that have been misclassified by the decision tree built by c4.5 are discarded, and a new tree is built using the remaining data. This is similar to the approach taken in (John 1995).

Polishing : Instances that have been misclassified by the decision tree built by c4.5 are polished, and a new tree is built using the polished data, according to the mechanism described in the previous section.

Twelve data sets from the UCI Repository of machine learning databases (Murphy & Aha 1998) were used. These are shown in Table 1. The training data was artificially corrupted by introducing random noise into both the attributes and the class. A noise level of $x\%$ means that the value of each attribute and the target class is assigned a random value $x\%$ of the time, with each alternative value being equally likely to be selected.

The actual percentages of noise in the data sets are given in the columns under "Actual Noise" in Table 1. These values are never higher, and in almost all cases lower, than the advertised $x\%$, since the original noise-free value could be selected as the random replacement as well. Also shown in Table 1 are the percentages of instances with at least one corrupted value. Note that even at fairly low noise levels, the majority of instances contain some amount of noise.

Results

We performed a ten-fold cross validation on each data set, using the above three methods (robust, filtering, and polishing) in turn to obtain the classifiers. In each trial, nine parts of the data were used for training, and the remaining one part was held for testing. We compared the classification accuracy and size of the decision trees built. The results are summarized in Tables 2 and 3.

Table 2 shows the classification accuracy and standard deviation of trees obtained using the three methods. We compared the methods in pairs (robust vs. filtering; robust vs. polishing; filtering vs. polishing), and differences that are significant at the 0.05 level using a one-tailed paired t-test are marked with an $*$. (An "$*$" indicates the latter method performed better than the former in the pair being compared; a "$-*$" denotes that the difference is "reverse": the former method performed significantly better than the latter.)

Of the three methods studied, we can establish a general ordering of the resulting predictive accuracy. Except for the nursery data set at the 0% noise level, and the zoo data set at the 10% noise level, in all other cases, where there was a significance difference, polishing gave rise to a higher classification accuracy than filtering, and filtering gave rise to a

Data Set	Noise Level	Actual Noise	Instances with Noise
audiology	0%	0.0%	0.0%
	10%	5.2%	96.0%
	20%	10.6%	100.0%
	30%	15.5%	100.0%
	40%	21.4%	100.0%
car	0%	0.0%	0.0%
	10%	6.8%	39.4%
	20%	14.6%	66.8%
	30%	21.4%	82.9%
	40%	28.4%	90.5%
LED-24	0%	0.0%	0.0%
	10%	5.3%	74.8%
	20%	10.3%	94.2%
	30%	15.4%	98.3%
	40%	20.9%	99.9%
lenses	0%	0.0%	0.0%
	10%	5.8%	29.2%
	20%	13.3%	50.0%
	30%	15.6%	45.8%
	40%	23.3%	75.0%
lung cancer	0%	0.0%	0.0%
	10%	7.3%	100.0%
	20%	14.8%	100.0%
	30%	21.9%	100.0%
	40%	27.5%	100.0%
mushroom	0%	0.0%	0.0%
	10%	7.3%	82.5%
	20%	14.8%	97.4%
	30%	22.1%	99.8%
	40%	29.5%	100.0%
nursery	0%	0.0%	0.0%
	10%	6.9%	47.4%
	20%	13.9%	74.2%
	30%	20.8%	87.7%
	40%	27.9%	94.7%
promoters	0%	0.0%	0.0%
	10%	7.6%	97.2%
	20%	14.3%	100.0%
	30%	22.0%	100.0%
	40%	29.9%	100.0%
soybean	0%	0.0%	0.0%
	10%	5.6%	85.7%
	20%	11.5%	96.2%
	30%	17.0%	99.6%
	40%	22.9%	100.0%
splice	0%	0.0%	0.0%
	10%	7.4%	98.9%
	20%	15.0%	100.0%
	30%	22.5%	100.0%
	40%	30.1%	100.0%
vote	0%	0.0%	0.0%
	10%	6.4%	66.2%
	20%	12.9%	89.2%
	30%	19.8%	97.7%
	40%	26.0%	99.8%
zoo	0%	0.0%	0.0%
	10%	5.4%	63.4%
	20%	11.8%	87.1%
	30%	15.6%	98.0%
	40%	20.0%	100.0%

Table 1: Noise characteristics of data sets at various noise levels.

higher classification accuracy than c4.5 alone. The results suggested that both filtering and polishing can be effective methods for dealing with imperfections in the data. In addition, we also observed that polishing outperformed filtering in quite a number of cases in our experiments, suggesting that *correcting* the noisy instances can be of a higher utility than simply identifying and tossing these instances out.

Now let us look at the average size of the decision trees built from data processed by the three methods. The results are shown in Table 3. There is no clear trend as to which method performed the best, but in almost all cases, the smallest trees were given by either filtering or polishing. Which of the two methods worked better in terms of reducing the tree size seemed to be data-dependent. In about half of the data sets, polishing gave the smallest trees at all noise levels, while the results were more mixed for the other half of the data sets.

To some extent it is expected that both filtering and polishing would give rise to trees of a smaller size than plain c4.5. By eliminating or correcting the noisy instances, both of these methods strive to make the data more uniform. Thus, fewer nodes are needed to represent the cleaner data. In addition, the data sets obtained from filtering are smaller in size than both the corresponding original and polished data sets, as some of the instances have been eliminated in the filtering process. However, judging from the experimental results, this did not seem to pose a significant advantage for filtering.

Remarks

We studied experimentally the behaviors of three methods of coping with noise in the data. Our evaluation suggested that in addition to the traditional approach of avoiding overfitting, both filtering and polishing can be viable mechanisms for reducing the negative effects of noise. Polishing in particular showed significant improvement over the other two approaches in many cases. Thus, it appears that even though noise correction adds considerable complexity to the task, it also recovers information not available with the other two approaches.

One might wonder why we did not use as a baseline for evaluation the "perfectly filtered" data sets, namely, those data sets with all known noisy instances removed. (It is possible in this setting, since we added the noise into the data sets ourselves.) While such a data set would be perfectly clean, it would also be very small. Table 1 shows the percentages of instances with at least one noisy element. Even at the 10% noise level, in the majority of data sets more than 50% of the instances are noisy. This percentage grows very quickly to almost 100% as the noise level increases. Thus, we would have had only very little data to work with if we

Data Set	Noise Level	Classification Accuracy ± Standard Deviation			Significant Difference		
		Robust	Filtering	Polishing	Robust/ Filtering	Robust/ Polishing	Filtering/ Polishing
audiology	0%	78.0 ± 7.7%	77.5 ± 7.8%	80.2 ± 7.0%			
	10%	73.0 ± 7.9%	73.0 ± 7.8%	73.0 ± 5.8%			
	20%	67.8 ± 8.0%	67.7 ± 5.9%	70.9 ± 5.2%		*	
	30%	54.8 ± 11.5%	54.5 ± 11.5%	61.6 ± 7.8%		*	*
	40%	33.6 ± 11.7%	42.5 ± 8.0%	48.2 ± 5.7%	*	*	
car	0%	93.2 ± 1.7%	92.8 ± 1.6%	92.9 ± 1.6%			
	10%	86.3 ± 2.6%	86.3 ± 2.5%	86.7 ± 2.9%			
	20%	83.6 ± 3.3%	83.6 ± 3.2%	84.3 ± 2.7%		*	*
	30%	76.5 ± 2.3%	76.5 ± 2.0%	80.6 ± 2.5%		*	*
	40%	74.5 ± 2.1%	74.4 ± 2.2%	77.1 ± 2.0%		*	*
LED-24	0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%			
	10%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%			
	20%	92.3 ± 4.3%	95.5 ± 3.8%	97.2 ± 2.2%	*	*	
	30%	76.2 ± 4.7%	83.2 ± 3.9%	90.0 ± 3.3%	*	*	*
	40%	49.4 ± 5.9%	53.8 ± 6.1%	68.1 ± 5.3%	*	*	*
lenses	0%	83.3 ± 30.7%	83.3 ± 30.7%	86.7 ± 30.5%			
	10%	50.0 ± 24.7%	50.0 ± 24.7%	78.3 ± 31.7%		*	*
	20%	55.0 ± 28.9%	55.0 ± 28.9%	73.3 ± 32.7%		*	*
	30%	48.3 ± 32.9%	48.3 ± 32.9%	56.7 ± 41.6%			
	40%	58.3 ± 38.2%	58.3 ± 38.2%	60.0 ± 30.9%			
lung cancer	0%	50.0 ± 23.9%	47.5 ± 22.4%	54.2 ± 31.0%			
	10%	30.8 ± 24.2%	43.3 ± 31.1%	46.7 ± 23.6%		*	
	20%	45.8 ± 37.5%	39.2 ± 32.9%	54.2 ± 37.1%			
	30%	57.5 ± 26.0%	55.0 ± 27.7%	55.0 ± 27.7%			
	40%	50.8 ± 16.9%	56.7 ± 20.0%	63.3 ± 24.8%		*	
mushroom	0%	100.0 ± 0.0%	100.0 ± 0.0%	100.0 ± 0.0%			
	10%	99.9 ± 0.1%	99.9 ± 0.1%	100.0 ± 0.1%			
	20%	99.7 ± 0.4%	99.7 ± 0.4%	100.0 ± 0.1%		*	*
	30%	98.9 ± 0.6%	98.9 ± 0.6%	99.3 ± 0.6%		*	*
	40%	98.6 ± 0.5%	98.6 ± 0.5%	98.8 ± 0.5%			
nursery	0%	97.0 ± 0.4%	96.8 ± 0.4%	96.8 ± 0.3%	−*	−*	
	10%	94.4 ± 0.4%	94.5 ± 0.5%	94.5 ± 0.4%			
	20%	90.9 ± 0.6%	91.0 ± 0.6%	91.2 ± 0.8%	*	*	*
	30%	90.0 ± 0.7%	90.0 ± 0.7%	90.3 ± 1.0%	*		
	40%	87.4 ± 1.1%	87.4 ± 1.1%	88.1 ± 0.7%		*	*
promoters	0%	75.6 ± 13.5%	75.6 ± 13.5%	77.5 ± 16.7%			
	10%	73.0 ± 12.5%	73.0 ± 12.5%	80.4 ± 10.4%		*	*
	20%	65.9 ± 9.0%	66.8 ± 8.2%	78.5 ± 12.7%		*	*
	30%	55.6 ± 10.3%	59.3 ± 14.1%	67.9 ± 15.0%		*	*
	40%	55.6 ± 14.7%	57.4 ± 14.7%	58.5 ± 14.8%		*	
soybean	0%	92.1 ± 2.0%	91.8 ± 2.2%	92.1 ± 1.8%			
	10%	86.2 ± 4.9%	85.7 ± 4.7%	88.7 ± 2.3%			
	20%	83.0 ± 3.3%	82.5 ± 4.6%	85.8 ± 3.6%		*	*
	30%	72.2 ± 6.1%	76.7 ± 3.7%	76.7 ± 4.4%	*	*	
	40%	50.7 ± 8.4%	51.2 ± 5.5%	55.4 ± 3.7%		*	*
splice	0%	94.0 ± 1.3%	94.1 ± 1.5%	94.2 ± 1.4%			
	10%	89.3 ± 1.8%	89.6 ± 1.6%	91.8 ± 1.5%		*	*
	20%	83.1 ± 2.1%	83.5 ± 1.7%	88.3 ± 1.5%		*	*
	30%	73.1 ± 4.0%	73.1 ± 3.2%	83.8 ± 2.6%		*	*
	40%	61.6 ± 2.9%	63.9 ± 2.2%	72.3 ± 3.0%	*	*	*
vote	0%	94.7 ± 2.0%	94.7 ± 2.0%	94.7 ± 2.5%			
	10%	94.7 ± 2.5%	94.7 ± 2.5%	95.2 ± 3.0%			
	20%	94.0 ± 2.1%	93.6 ± 1.7%	95.4 ± 2.3%		*	*
	30%	92.9 ± 3.2%	92.7 ± 2.9%	90.6 ± 5.5%			
	40%	92.4 ± 3.1%	92.4 ± 3.1%	92.8 ± 3.9%			
zoo	0%	92.2 ± 7.2%	92.2 ± 7.2%	93.1 ± 6.4%			
	10%	91.2 ± 8.1%	88.3 ± 9.3%	95.2 ± 7.7%	−*		*
	20%	83.2 ± 7.8%	84.2 ± 6.5%	85.2 ± 9.1%			
	30%	77.4 ± 11.4%	79.3 ± 9.2%	86.2 ± 4.7%		*	*
	40%	78.3 ± 11.5%	80.3 ± 9.8%	87.1 ± 7.8%		*	*

Table 2: Classification accuracy with standard deviation. An "*" indicates a significant improvement of the latter method over the former at the 0.05 level. A "−*" indicates a "reverse" significant difference: the former method performed better than the latter.

Data Set	Noise Level	Robust	Filtering	Polishing
audiology	0%	50.5	45.4	47.0
	10%	66.6	44.3	56.2
	20%	90.3	58.8	90.9
	30%	125.0	76.8	121.6
	40%	143.3	89.3	130.1
car	0%	173.4	169.1	169.1
	10%	108.5	105.1	102.1
	20%	103.7	104.0	102.7
	30%	88.8	81.7	104.0
	40%	41.6	41.3	61.2
LED-24	0%	19.0	19.0	19.0
	10%	78.2	69.4	34.8
	20%	193.4	134.6	85.6
	30%	335.8	231.0	162.4
	40%	490.8	347.2	317.8
lenses	0%	6.4	6.4	5.0
	10%	4.2	4.0	6.7
	20%	3.5	3.5	3.4
	30%	3.8	3.8	7.3
	40%	3.9	3.9	4.5
lung cancer	0%	19.0	15.0	14.2
	10%	20.6	19.0	23.0
	20%	15.8	15.4	15.4
	30%	12.2	11.0	11.8
	40%	16.6	15.0	16.2
mushroom	0%	30.6	30.6	30.6
	10%	214.3	207.2	109.9
	20%	268.3	244.8	124.6
	30%	345.0	325.5	129.3
	40%	535.2	519.2	286.4
nursery	0%	508.4	473.3	464.1
	10%	315.3	304.5	286.4
	20%	177.0	167.6	149.5
	30%	174.8	164.9	133.6
	40%	258.0	235.3	212.6
promoters	0%	21.4	21.4	12.2
	10%	21.8	21.8	12.2
	20%	28.2	28.2	15.0
	30%	32.2	29.0	17.4
	40%	34.6	32.6	34.2
soybean	0%	94.1	88.7	91.2
	10%	157.9	131.6	162.7
	20%	202.5	146.7	189.2
	30%	278.5	184.0	218.7
	40%	328.7	209.2	289.4
splice	0%	171.8	168.2	156.2
	10%	318.2	294.6	120.6
	20%	537.4	512.2	233.4
	30%	836.2	793.4	335.4
	40%	1143.0	1036.6	680.2
vote	0%	14.5	14.5	5.8
	10%	17.8	17.2	13.6
	20%	20.8	19.0	11.8
	30%	43.3	38.5	24.7
	40%	27.1	27.1	19.6
zoo	0%	17.8	17.6	16.2
	10%	24.2	21.8	21.2
	20%	20.6	19.8	22.2
	30%	30.2	21.2	30.0
	40%	35.4	25.8	30.0

Table 3: Average size of the pruned decision trees

had opted for a perfectly filtered data set. Note that, however, ironically we may end up in this situation if our filtering technique becomes too effective.

While we have evaluated the performance of the noise handling methods against each other, there is no reason why we cannot combine these methods in practice. The different methods address different aspects of data imperfections, and the combined mechanism may be able to tackle noise more effectively than any of the individual component mechanisms alone. However, first we need to achieve a better understanding of the behaviors of these mechanisms before we can utilize them properly.

References

Brodley, C. E., and Friedl, M. A. 1996. Identifying and eliminating mislabeled training instances. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*.

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

Gamberger, D.; Lavrač, N.; and Džeroski, S. 1996. Noise elimination in inductive concept learning: A case study in medical diagnosis. In *Proceedings of the Seventh International Workshop on Algorithmic Learning Theory*, 199–212.

John, G. H. 1995. Robust decision trees: Removing outliers from databases. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, 174–179.

Mitchell, T. M. 1997. *Machine Learning*. McGraw-Hill.

Murphy, P. M., and Aha, D. W. 1998. UCI repository of machine learning databases. University of California, Irvine, Department of Information and Computer Science. `www.ics.uci.edu/~mlearn/MLRepository.html`.

Quinlan, J. R. 1987. Simplifying decision trees. *International Journal of Man-Machine Studies* 27(3):221–234.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Rousseeuw, P. J., and Leroy, A. M. 1987. *Robust Regression and Outlier Detection*. John Wiley & Sons.

Teng, C. M. 1999. Correcting noisy data. In *Proceedings of the Sixteenth International Conference on Machine Learning*, 239–248.

Teng, C. M. 2000. Evaluation noise correction. In *Lecture Notes in Artificial Intelligence: Proceedings of the Sixth Pacific Rim International Conference on Artificial Intelligence*. Springer-Verlag.