Cleaning Training-Datasets with Noise-Aware Algorithms

H. Jair Escalante
Instituto Nacional de Astrofísica Óptica y Electrónica,
Computer Science Department
Tonantzintla, Puebla, 72840, México
hugojair@ccc.inaoep.mx

Abstract

We introduce a novel learning algorithm for noise elimination. Our algorithm is based on the re-measurement idea for the correction of erroneous observations and is able to discriminate between noisy and noiseless observations by using kernel methods. We apply our noise-aware algorithms to several domains including: astronomy, face recognition and ten machine learning benchmark datasets. Experimental results adding noise and useful anomalies to the data show that our algorithm improves data quality, without having to eliminate any observation from the original dataset.

1 Introduction

Real world data are never as good as we would like them to be and often can suffer from corruption that may affect data interpretation, data processing, classifiers and models generated from data as well as decisions based on them. On the other hand, data can also contain useful anomalies, which often result in interesting findings, motivating further investigation. Thus, unusual data can be due to several factors including: ignorance and human mistakes, the inherent variability of the domain, rounding and transcription errors, instrument malfunction, biases and, most important, rare but correct and useful behavior. For these reasons it is necessary to develop techniques that allow us to deal with unusual data.

Data cleaning is a well studied task in many areas dealing with databases, nevertheless, this task requires a large time investment. Indeed, between 30% to 80% of the data analysis task is spent on cleaning and understanding the data [9]. An expert can clean the data, but this requires a large time investment, growing with the number of observations in the data set, which results in expensive costs. From here arises the need to automate this task. However, this is not easy, since useful anomalies and noise may look quite similar to an algorithm. For this reason we need to endow to such algorithm with more human-like reasoning. In this work the re-measurement idea is proposed; this approach consist of detecting suspect data and, by analyzing new observations of these objects, substitute errors while retaining anomalies and correct data for a posterior analysis. This idea is based on the natural way in which a human clarifies his/her doubts when he/she is not sure about the correctness of a datum. When a person is doubtful about an object's observation, a new observation or many more can be obtained to confirm or discard the observer's hypothesis.

The proposed methods could be useful in areas such as machine learning, data mining, pattern recognition, data cleansing, data warehousing and information retrieval. Although in this work we oriented our efforts to improve data quality for machine learning training datasets, we tested our developed method on several domains including: astronomy, face recognition and 10 benchmark datasets. This paper is an extension of previously published work [12, 10], applying the re-measurement idea to the face recognition task, in a realistic scenario.

The paper is organized as follows: in the next section we present a brief survey of related works and in Section 3 the kernel methods that we used are described. Next, we introduce the proposed algorithms in Section 4. The data sets used in this work are presented in Section 5. In Section 6 experimental results evaluating the performance of our algorithms are presented. Finally, we summarize our findings and discuss future directions for this work in Section 7.

2 Related Work

Recent approaches for data cleansing do not distinguish between useful anomalies and noise, they just eliminate the detected suspect data [6, 21, 15, 31, 30, 25, 18]. However, we should not eliminate a datum unless we can determine that it is an invalid one. This often is not possible for several reasons, including: human-hour cost, time investment, ignorance about the domain we are dealing with and even

Seventh Mexican International Conference on Computer Science (ENC'06)
0-7695-2666-7/06 $20.00 © 2006 IEEE

inherent uncertainty. Nevertheless, if we could guarantee that an algorithm will successfully distinguish errors from correct observations, the difficult problem would be solved. As a human does, an algorithm can confirm or discard a hypothesis by analyzing several measurements of the same object.

The idea of requesting new observations as a strategy for data cleansing has been little explored. Here we present some related works that deal with anomaly detection and data cleaning.

In [17] an interactive method for data cleaning that uses the optimal margin classifier (OMC) is presented. The OMC is used to identify suspect data, suspect observations are shown to an expert in the domain, who then decides their validity.

Prototype [29] and instance selection [5] implicitly can eliminate instances degrading the performance of instance-based learning algorithms. Other algorithms saturate a dataset with the risk of eliminating all objects that could define a concept or class, these methods include the use of instance pruning trees [18] and the saturation filtering algorithm [15]. Ensembles of classifiers had been successfully used to identify mislabeled instances in classification problems [7, 31, 8], however, once again the identified instances are deleted from the data set.

In the outlier/anomaly detection area there are many published works, however, these approaches are intended only for the detection of rare data. The anomaly detection problem has been approached using statistical [3] and probabilistic knowledge [20], distance and similarity-dissimilarity functions [1, 19, 22], metrics and kernels [28], accuracy when dealing with labeled data, association rules, properties of patterns and other specific domain features.

Variants and modifications to the support vector machine algorithm have been proposed, trying to isolate the outlier class: in [27] an algorithm to find the support of a dataset, which can be used to find outliers, is presented; in [30] the sphere with minimal radius enclosing most of the data is found and in [25] the correct class is separated from the origin and from the outlier class for a given data set.

There are many more methods for anomaly detection than the presented here, however, we have only presented some of the representative ones. What is important to notice is that at the moment there are automated approaches for data cleaning that are concerned with the elimination of useful data.

3 Kernel Methods

Kernel methods have been shown to be useful tools for pattern recognition, dimensionality reduction, denoising, and image processing. In this work we use kernel methods for dimensionality reduction, novelty detection and anomaly-noise differentiation.

3.1 Kernel PCA

Stellar populations data are formed with instances with dimensionality $d = 12134$, therefore, in order to perform experiments in feasible time we need a method for dimensionality reduction. Kernel principal component analysis (KPCA) [26] is a relative recent technique, which takes the classical PCA technique to the feature space, taking advantage of "kernel functions". This feature space is obtained by a mapping from the linear input space to a commonly nonlinear feature space F by $\Phi : \mathbf{R}^N \rightarrow F, x \mapsto X$.

In order to perform PCA in F, we assume that we are dealing with centered data, using the covariance matrix in F, $\overline{C} = \frac{1}{l}\sum_{j=1}^{l}\Phi(\mathbf{x}_j)\Phi(\mathbf{x}_j)^T$, we need to find $\lambda \geq 0$ and $\mathbf{v} \in F\backslash\{0\}$ satisfying $\lambda\mathbf{V} = \overline{C}\mathbf{V}$. After some mathematical manipulation and defining a $M \times M$ matrix K by

$$K_{i,j} := (\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)) \tag{1}$$

the problem reduces to $\lambda\alpha = K\alpha$, knowing that there exist coefficients $\alpha_i (i = 1, \ldots, l)$ such that $\lambda\mathbf{V} = \sum_{i=1}^{l}\lambda_i\Phi(\mathbf{x}_i)$.

Depending on the dimensionality of the dataset, matrix K in (1) could be very expensive to compute, however, a much more efficient way to compute dot products of the form $(\Phi(\mathbf{x}), \Phi(\mathbf{y}))$ is by using kernel representations $k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}))$, which allow us to compute the value of the dot product in F without having to carry out the expensive mapping Φ.

Not all dot product functions can be used, only those that satisfy Mercer's theorem [16]. In this work we used a polynomial kernel (Eq. 2).

$$k(\mathbf{x}, \mathbf{y}) = ((\mathbf{x} \cdot \mathbf{y}) + 1)^d \tag{2}$$

3.2 Kernel based novelty detection

In order to develop an accurate noise-aware algorithm we need first a precise method for novelty detection. We decided to use a novelty detection algorithm that has outperformed others in an experimental comparison [11]. This algorithm presented in [28] computes the center of mass for a dataset in feature space by using a kernel matrix K, then a threshold t is fixed by considering an estimation error (Eq. 3) of the empirical center of mass, as well as distances between objects and such center of mass in a dataset.

$$t = \sqrt{\frac{2 * \phi}{n}} * \left(\sqrt{2} + \sqrt{\ln\frac{1}{\delta}}\right) \tag{3}$$

where $\phi = \max(diag(K))$, and K is the kernel matrix of the dataset with size $n \times n$; δ is a confidence parameter

for the detection process. Therefore, the observations with distances to the center of mass exceeding t, are considered outliers.

This is an efficient and very precise method; for this work we used a polynomial kernel function (Eq. 2) of degree 1.

4 Noise-Aware Algorithms

There are many domains and practical applications in which a re-measurement algorithm is suitable to use, however, it is necessary to emphasize that such algorithms are suitable for certain type of domains and applications. Also may exist some domains in which the re-measuring process will result in unfavorable consequences, therefore, we should be careful about the problems to which we might apply this algorithm. In general, a re-measurement algorithm can be applied to any domain in that the re-measuring process is affordable and feasible, domains that require of highly reliable information, domains in which the novelty is more useful than the rest of the objects and domains in which decisions made on data are crucial. Suitable domains include, but are not limited to: medical diagnosis, security systems, scientific and commercial applications as well as information retrieval.

Before introducing the noise-aware algorithms, the *re-measuring* process must be clarified. Given a set of instances: $X = \{x_1, x_2, \ldots, x_n\}$, with $x_i \in \mathbf{R}^n$ (generated from a known and controlled process by means of measurement instruments or human recording), we have a subset $S \subset X$ of instances x_i^s with $S = \{x_1^s, x_2^s, \ldots, x_m^s\}$ and $m << n$ that, according to a method for anomaly detection are suspect to be incorrect observations. Therefore, the re-measuring process consists of generating another observation $x_i^{s'}$ for each of the m objects, in the same conditions and using the same configuration that when the original observations were made.

For example, in case our observations were pictures for face recognition, the re-measuring process would consists of taking another picture for every suspect person-object in our data set. The new pictures should be taken to the original objects (persons) in a place with similar illumination, in the same pose and using the same zoom and camera resolution.

In Figure 1 the base noise-aware algorithm is shown. The data preprocessing module includes dimensionality reduction, scaling data, feature selection or similar necessary processes. In this work we applied KPCA for dimensionality reduction of the astronomical domain; reducing the dimensionality from 12134 features to 100 principal components. For the faces data we reduced the data from vectors of size 10304 to small vectors of size 50.

In the next step, suspect data are identified by using the

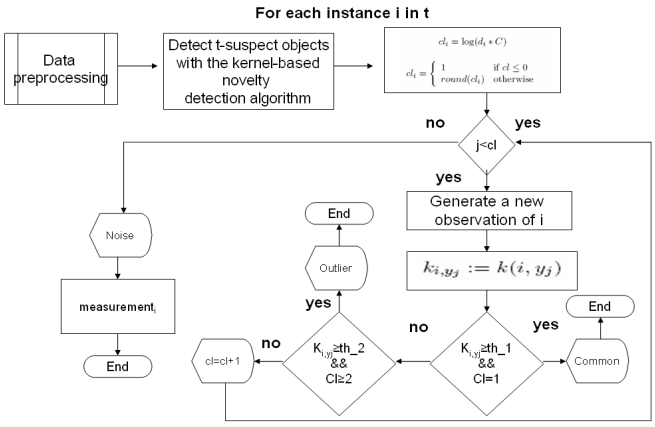

Figure 1. Block diagram of the base noise-aware algorithm.

anomaly detection method described in Section 4.2. Then, a confidence level cl is calculated; this cl indicates how rare an object is, and it can be used to determine the number of new measurements to obtain for each of the suspicious instances; the value of cl is 1 for common instances and around 2 to 5 for noise and outliers. cl is obtained from the distance of the suspect instances to the center of mass (in the feature space) of the full data set, and it is defined in (4).

$$cl_i = \begin{cases} 1 & \text{if } \log(d_i * C) \leq 0 \\ \text{round}(\log(d_i * C)) & \text{otherwise} \end{cases} \quad (4)$$

Where d_i is the distance in feature space of the suspect instance x_1^s to the center of mass of the full data set, and C is a scaling constant. d_i Is obtained from the kernel-based novelty detection algorithm.

Once we know how rare an object is, we do request new measurements for each suspect object using the cl value. The more an object is the higher its cl value is. Then, we proceed to the identification of noisy observations by comparing the original observation and its cl re-measurements.

For the anomaly-noise discrimination we decided to use a kernel, since kernels can be used to calculate similarity between objects [16]. Several kernels were tested, but the kernel that best distinguished among dissimilar instances was the extended radial basis function (Eq. 5) with $\sigma = 0.25$, this value was chosen experimentally.

$$k(x, y) = \exp\left(\frac{-\sqrt{\|x - y\|^2}}{2\sigma^2}\right) \quad (5)$$

Based on the result of applying the above kernel to the original observation and its new measurements we generated

0-7695-2666-7/06 $20.00

simple rules to discriminate among noise, outliers and common instances. Our approach is based on the idea that having several observations of the same object can be very helpful in deciding if an observation is correct or it can be noisy. That is, for correct observations the original measurement and its re-measurements must be very similar (if not identical), while for noisy observations the original measurement and its new ones should differ one of another. Assuming that any (or some) of the measurements, for that object, is correct.

If an object is correct, based on the below decision rules, the algorithm leaves that object intact. Otherwise, the noisy observation is substituted by one in the new measurements, according to some approach depending on the application and data. The generated decision rules were:

$$O = \begin{cases} not-outlier & \text{if } k_{avg} \geq 0.99 \text{ and } cl = 1 \\ outlier & \text{if } k_{avg} \geq 0.8 \text{ and } cl \geq 2 \\ noise & \text{otherwise} \end{cases}$$

where $k_{avg} = \frac{1}{cl} \sum_{j=1}^{cl} k(x, y_j)$, is the average of the kernel evaluations given a suspect instance x and its cl new measurements $y_1, \ldots, y_{cl}$ as inputs. k_{avg} tells us how similar are the original observation and the corresponding cl new measurements for an object. The values for the thresholds worked well for the astronomical domain we used. A small modification was done for the other domains, although this is not a difficult task.

As we can see, outliers and common instances will be detected with only a new observation, while noise will be remeasured a few times, finally all of the noise is substituted. Depending on the type of data and their application we can substitute noisy observations in different ways. Trying that the substitution method used can retain useful information only, eliminating the noise. When no information is given about the data we can substitute noisy (original) observations by a random measurement. Other substitution methods are for example substituting the original observation by the average of re-measurements or even creating a new observation by combining randomly the attribute's values of the measurements, this and other substitution methods are proposed in [11]. For this work we used the random substitution method for the UCI data and in the Olivetti dataset too. For the astronomical domain we substituted the noisy observations by the average of re-measurements, since we know that the noise in the astronomical data is Gaussian.

In summary, the algorithm first preprocess the data (if necessary), then it detects the suspect data using the kernel-based novelty detection algorithm. Then a cl value is calculated and cl new measurements are requested for each suspect observation. Next, we apply the decision rules to discriminate correct and erroneous observations. Finally, noisy observations are substituted while correct data remain not affected. The algorithm from Figure 1 can be used for cleaning datasets, eliminating all of the noise and retaining correct observations. However the usefulness of this sort of algorithms is not limited to data cleaning. In [12] a modification of this algorithm is used to strengthen a learning algorithm, obtaining promising results.

5 Training-Datasets used

In this paper we applied our noise-aware algorithm to clean some machine learning datasets. Previous work on an astronomical domain as well as on benchmark data has been published already [12, 10, 11]. With this paper we are going one step further, applying the algorithm to a benchmark dataset for face recognition: the Olivetti faces database. Furthermore, we compare the performance of noise-aware algorithms through several datasets for different applications. In the remaining of this section we briefly describe the used datasets.

5.1 Stellar Populations Dataset

We used an astronomical dataset: *the stellar populations dataset*, which has been widely used in machine learning [14, 12, 11], for testing the performance of our algorithm. The estimation of stellar populations parameters can reveal useful information for astronomers, providing knowledge and insight about the evolution of the universe. We perform experiments on this dataset affecting the data in two different ways trying to model the real behavior of this domain. In the following the domain we used is described.

A galactic spectrum can be modeled with good accuracy as a linear combination of three spectra, corresponding to young, medium and old stellar populations, see Figure 2, with their respective metallicity, together with a model of the effects of interstellar dust in these individual spectra. This effect is called reddening in the astronomical literature. Let $f(\lambda)$ be the energy flux emitted by a star or group of stars at wavelength λ. The flux detected by a measuring device can be approximated as $d(\lambda) = f(\lambda)(1 - e^{-r\lambda})$, where r is a constant that defines the amount of reddening in the observed spectrum and depends on the size and density of the dust particles in the interstellar medium.

We also need to consider the redshift, which tells us how the light emitted by distant galaxies is shifted to longer wavelengths, when compared to the spectrum of closer galaxies. We build a simulated galactic spectrum given constants c_1, c_2, and c_3, with $\sum_{i=1}^{3} c_i = 1$ and $c_i > 0$, that represent the relative contributions of young, medium and old stellar populations, respectively; their reddening parameters r_1, r_2, r_3, and the ages of the populations $a_1 \in \{10^6, 10^{6.3}, 10^{6.6}, 10^7, 10^{7.3}\}$ years, $a_2 \in \{10^{7.6}, 10^8, 10^{8.3}10^{8.6}\}$ years, and $a_3 \in \{10^9, 10^{10.2}\}$ years,

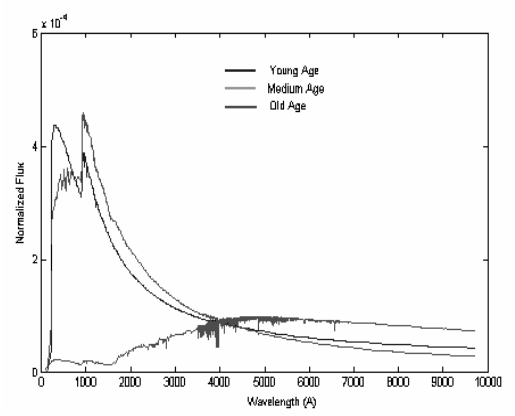

Figure 2. Stellar spectra of young, intermediate and old populations.

$$g(\lambda) = \sum_{i,m=1}^{3} c_i s(m_i, a_i, \lambda)(1 - e^{r_i \lambda})$$

with $m \in \{0.0004, 0.004, 0.008, 0.02, 0.05\}$ in solar units and $m_1 \geq m_2 \geq m_3$, finally we add an artificial redshift Z by:

$$\lambda = \lambda_0(Z + 1), 0 < Z \leq 1$$

Therefore, the learning task is to estimate the parameters: reddening (r_1, r_2, r_3), metallicities (m_1, m_2, m_3), ages (a_1, a_2, a_3), relative contributions (c_1, c_2, c_3), and redshift Z, from the spectra. For each experiment we considered a dataset of 200 different spectra.

The stellar populations dataset was affected in two ways. On the one hand, the noise was simulated by inserting additive extreme noise with extreme negative and positive means. Outliers, were simulated by scaling the observations by a factor $(1 < f \leq 9)$. We will call this dataset spp_1, see Figure 3.

On the other hand, we also affected the stellar populations dataset in a very realistic way. Noise were simulated by adding gaussian noise (with zero mean), varying the standard deviation, see Figure 4. Useful anomalies were simulated in a realistic way. Commonly, redshift values lie in the range $(0 \leq Z \leq 0.4)$; redshifts higher than 1 are useful anomalies for astronomers. In astronomy, locating galaxies with redshifts over 2 is very useful for galaxy evolution research. We simulated in 5% of the data redshifts between 2 and 4 $(2 \leq Z \leq 4)$. We call this dataset spp_2.

5.2 Affecting the UCI datasets

In order to test the generality of our approach we decide to use benchmark data. To this end we decided to use data from a widely used machine learning repository: UCI

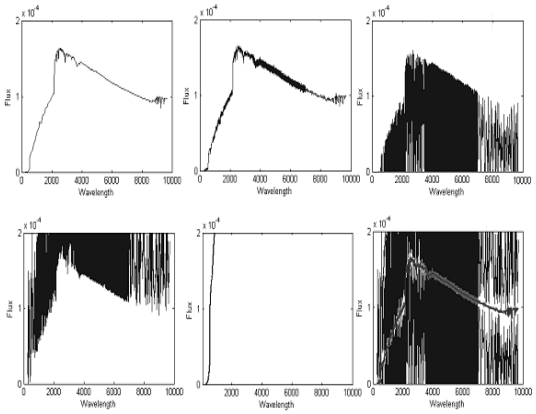

Figure 3. Sample spectra with the different levels of noise added. In all of the figures, the noise is Gaussian with zero mean and varying the standard deviation in each case.

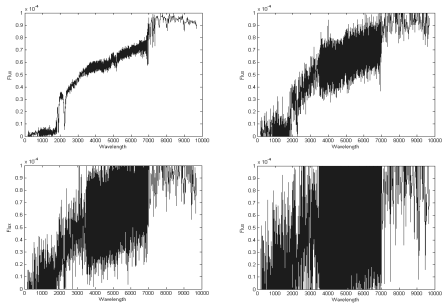

Figure 4. Sample spectra with the different levels of noise added. In all of the figures, the noise is Gaussian with zero mean and varying the standard deviation in each case.

Table 1. UCI Datasets description

ID	Name	$\#_{Cases}$	Output	Affected
W	Wine	$178 - 13$	3-Discrete	18
G	Glass	$214 - 9$	Real	21
H	Boston Housing	$506 - 13$	Real	51
A	Auto	$32 - 7$	Real	3
I	Iris	$150 - 4$	3-Discrete	15
M	Machine CPU	$209 - 6$	Real	21
L	Lymphography	$148 - 18$	4-Discrete	15
C	Breast Cancer	$683 - 9$	2-Discrete	68
B	Bio Med	$194 - 5$	2-Discrete	19
Ab	Abalone	$1000 - 8$	Real	100

0-7695-2666-7/06 $20.00

Figure 5. Sample image for the experiment 1, left original image, center a simulated useful-anomaly and right a noisy image

Figure 6. Sample image with the following effects (from left to right)negative, oil paint, cutting regions, emboss, edge detection, decreasing color depth, twirl and pixelizing

[4]. We selected 5 regression datasets and 5 classification datasets, which are described in Table 1.

These UCI datasets were affected in the simplest way: noise were simulated by inserting 30% of additive noise, while for outliers we scaled observations by a factor of 3.

5.3 Olivetti faces database

We decided to use another benchmark dataset: the Olivetti Research Ltd (ORL) database of faces. The dataset consist of 400 images, 10 each of 40 different subjects. The subjects are in frontal position, see Figure 5 for a sample image. There are 10 images of 40 people for a total of 400 pictures with a resolution of $r = 112_X 92$, 8-bit grey levels, we refer the reader to [24] for a more complete description of the dataset.

For the faces dataset we used two approaches for affecting the data. In the first one, the noise was gaussian with zero mean and standard deviation of 0.5 added to the selected images. For the outliers we affected some images simulating people with sunglasses, just like in [23], see Figure 5 for a sample of this approach, which we named $Faces_1$.

The second approach aimed to deal with more general types of artificial noise. Outliers were simulated as in Figure 5, however, the noise were simulated by affecting the images with the following effects: negative, oil paint, cutting regions, emboss, edge detection, decreasing color depth, twirl and pixelizing. see Figure 6. We named this dataset $Faces_2$

6 Experimental Results

We performed several experiments in order to test the performance of the method described in the Section 4. We used the machine learning training-datasets outlined in the last Section. For each dataset we performed 3 experiments and the average is presented in the tables. For every experiment each dataset was affected in a similar way: 5% of the data were affected with noise and another 5% of the data were affected by inserting outliers (*useful-anomalies*). The difference lies in the way that the noise and outliers were generated for each dataset, see Section 5 for a description of this process.

In this work we used locally weighted linear regression (LWLR) as our learning algorithm. LWLR is part of the group of instance based learning algorithms. This kind of algorithm stores some or all of the training examples and then postpone any generalization effort until a new instances class needs to be predicted. At this moment the algorithm attempt to fit the training examples only in a region around the query point[1]. We choose LWLR as our machine learning algorithm because its training time is equal to zero. We perform many experiments, so with LWLR we avoid the training process time. Furthermore results from [13] show that LWLR outperforms artificial neural networks and self organizing maps in the prediction of stellar atmospheric parameters.

Each experiment consists of applying the algorithm from Figure 1 to each of the datasets. We compare some parameters such as: percentage of outliers (O.D.) and noise detected (N.D.), confusions between noise and outliers, number of new measurements needed for noisy observations (CLN), percentage of true positives (TP) and false positives (FP), as well as recall, precision and F measure value. Furthermore, we reported the percentage of accuracy improvement using the LWLR learning algorithm after applying our noise-aware algorithm; taking as baseline the accuracy of the learning algorithm in the affected dataset without any processing.

In Table 2 the results of the above described parameters for all the datasets are presented.

As we can see, most of outliers and noise were detected. For the case of outliers the method failed in the spps datasets, however, the percentages are still very high. For the case of noise, the method performed perfect in all but one dataset: the $Faces_2$. This is due to the heterogeneity with which the noise were simulated, see Figure 6. However, the result is promising since in a real application we will not have all this sort of noise in the same dataset (with high probability).

There was not any confusion between noise and outliers, which is very important if we want to retain the cor-

[1] We refer the reader to [2] for a detailed explanation of the method

P D	spp_1	spp_2	UCI	$Faces_1$	$Faces_2$
O.D. (%)	86.6%	73.33%	99.61%	100%	100%
N.D. (%)	100%	100%	99.3%	100%	65%
Confusions	0	0	0	0	0
CLN	1.5	5	3.75	3.2	3.15
TP	100%	90%	99.19%	100%	82.5%
FP	0	10%	0.81%	0	17.5%
R	1	0.9	0.9	1	0.825
P	0.66	0.47	0.66	0.66	0.55
F	0.8	0.61	0.79	0.8	0.66
Red (%)	7.16%	-5.88%	27.17%	-1%	-0.75%

Table 2. Performance of the noise-aware algorithms for data cleaning in the different datasets

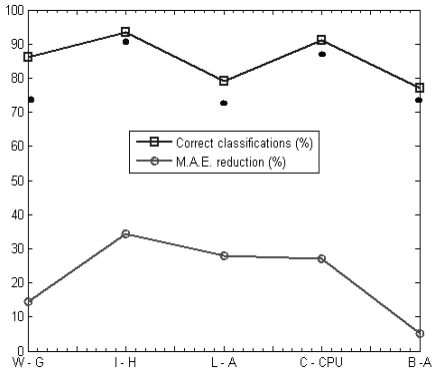

Figure 7. Prediction accuracy improvement in term of percentage for the UCI datasets. Line with circles, represents the percentage of error reduction for regression datasets. Line with squares, represents the percentage of correct classifications for classification datasets, the dots in black are the baseline percentages of correct classifications.

rect data. The number of new measurements required by the method ranges from 1.5 to 5, though in average 3.2 re-measurements were needed, which makes feasible the application of this algorithm in some domains.

The following rows in Table 2 show us the performance of the kernel-based novelty detection algorithm from Section 3.2. The recall ($\mathbf{R} = \frac{TP}{(TP+FN)}$) is always close to 1, which means that almost all of the affected data (noise and outliers) were detected. However, the precision $\mathbf{P} = \frac{TP}{(TP+FP)}$ is low, which means that several false data were labeled as suspect. These results on recall (R) and precision (P), are combined in a well known metric from information retrieval called $F - measure$, $F = \frac{2*R*P}{(R+P)}$, which in Table 2 tells us that the method had a regular performance. For the purposes of noise-aware algorithms the novelty detection algorithm is very appropriate, since we need a method able to detect all (or most) of the suspect data, while the effects of a low precision rate do not affect the performance of noise-aware algorithms.

Finally, in the last row of Table 2 we can judge the improvement in terms of accuracy. The use of our method, improved the prediction accuracy in 2 of 5 datasets. The best result was attained in the UCI datasets; an analysis of these results in more detail can be seen in Figure 7. From this Figure we can see that in all of the datasets there is a clear error reduction when we use our method, although this happen only in the UCI datasets.

The results on the ssp_2, $Faces_1$ and $Faces_2$ datasets are disconcerting. Why accuracy is decreased if we have improved data quality?, this can be due to several factors: the bias of the learning algorithm, over-fitting, the noise could made easy the selection neighbors for the LWLR algorithm or the use of KPCA is not optimal. We do not know exactly what is the cause of decrement in accuracy and this can motivate future research.

Besides our method does not improve the prediction accuracy clearly, our method improves data quality by correcting erroneous observations while retaining all of the correct instances. Therefore, we can argue that our method improves data quality. Furthermore, it as been proved that a noise-aware algorithm can improve prediction accuracy if it is used to such end [12].

7 Conclusions and Future Work

We have presented the re-measuring idea as a method for the correction of erroneous observations in corrupted datasets without eliminating potentially useful observations. The algorithm was able to detect and correct 100% of the erroneous observations and around 90% of the artificial outliers for most of the tested datasets, which resulted in a data quality improvement. We performed experiments in several domains including face recognition, an astronomical domain and ten benchmark datasets from the UCI machine learning repository showing the generalization ability of noise-aware algorithms. The experiments we performed simulated real behavior of the domains used and they can be used in other several domains.

Present and future work includes testing our algorithms on other benchmark datasets to determine their scope of applicability. Also, we plan to apply noise-aware algorithms in other astronomical domains as well as in other areas, including bioinformatics, medical diagnosis, and image analysis.

Acknowledgments

This work was partially supported by CONACYT under grant

181498. Special thanks to Dr. Olac Fuentes by its support and kindness, as well as to Veronica Rodriguez for the patronage.

References

[1] Andreas Arning, Rakesh Agrawal, and Prabhakar Raghavan. A linear method for deviation detection in large databases. In *Knowledge Discovery and Data Mining*, pages 164–169, 1996.

[2] Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1-5):11–73, 1997.

[3] Vic Barnett and Toby Lewis. *Outliers in Statistical Data*. John Wiley and Sons, 1978.

[4] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[5] Henry Brighton and Chris Mellish. Advances in instance selection for instance-based learning algorithms. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining*, pages 153–172, 2003.

[6] Carla Brodley. Identifying mislabeled training data. In *Journal of Artificial Intelligence Research*, volume 11, pages 131–167, 1999.

[7] Carla E. Brodley and Mark A. Friedl. Identifying and eliminating mislabeled training instances. In *AAAI/IAAI, Vol. 1*, pages 799–805, 1996.

[8] David Clark. Using consensus ensembles to identify suspect data. In *KES*, pages 483–490, 2004.

[9] Tamraparni Dasu and Theodore Johnson. *Exploratory Data Mining and Data Cleaning*. Probability and Statistics. Wiley, 2003.

[10] H. Jair Escalante. Kernel methods for anomaly detection and noise elimination. In *Computing and Systems, CORE 2006*. IPN, 2006.

[11] H. Jair Escalante. Noise-aware machine learning algorithms. Master's thesis, Instituto Nacional de Astrofísica Óptica y Electrónica, January 2006.

[12] H. Jair Escalante and Olac Fuentes. Analysis of galactic spectra using noise-aware learning algorithms. In *19th International FLAIRS Conference*. AAAI, 2006.

[13] Olac Fuentes. Neural networks and instance-based learning for the prediction of stellar atmospheric parameters. In *Proceedings of the IASTED International Conference on Artificial Intelligence and Soft Computing ASC2001*, volume 10, 2001.

[14] Olac Fuentes, Thamar Solorio, Roberto Terlevich, and Elena Terlevich. Analysis of galactic spectra using active instance-based learning and domain knowledge. In *Proceedings of IX IBERAMIA, Puebla, Mexico*. Lecture Notes in Artificial Intelligence 3315, 2004.

[15] Dragan Gamberger, Nada Lavrač, and Ciril Grošelj. Experiments with noise filtering in a medical domain. In *Proceedings of the 16th International Conference on Machine Learning*, pages 143–151. Morgan Kaufmann, San Francisco, CA, 1999.

[16] Ralf Herbrich. *Learning Kernel Classifiers*. MIT press, first edition, 2002.

[17] N. Matic I. Guyon and V. Vapnik. Discovering informative patterns ans data cleaning. In *Advances in Knowledge Discovery and Data Mining*, pages 181–203, 1996.

[18] George H. John. Robust decision trees: Removing outliers from databases. In *Proceedings of the 1st. Int. Conf. on KDDM*, pages 174–179, 1995.

[19] Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24th International Conference in Very Large Data Bases, VLDB*, pages 392–403, 1998.

[20] J. Kubica and A. Moore. Probabilistic noise identification and data cleaning. In *Technical Report CMU-RI-TR-02-26, CMU*, 2002.

[21] Raymond T. Ng and Jiawei Han. Efficient and effective clustering methods for spatial data mining. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 144–155. Morgan Kaufmann Publishers, 1994.

[22] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 427–438, Dallas, Texas, USA, 2000. ACM.

[23] Volker Roth. Outlier detection with one-class kernel fisher discriminants. In *Advances in Neural Information Processing Systems*, volume 17, pages 1169–1176. MIT Press, 2005.

[24] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *nd IEEE Workshop on Applications of Computer Vision*, 1994.

[25] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution. In *Technical Report 99-87, Microsoft Research*, 1999.

[26] B. Schölkopf, A. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. In *Neural Computation*, volume 10, pages 1299–1319, 1998.

[27] B. Schölkopft, A. Smola, R. Williamson, and P. Bartlett. New support vector algorithms. In *Neural Computation*, volume 12, pages 1083 – 1121, 2000.

[28] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[29] David B. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *ICML*, pages 293–301, 1994.

[30] D. Tax and R. Duin. Data domain description using support vectors. In *Proceedings of the European Symposium on Artificial Neural Networks*, pages 251–256, 1999.

[31] Sofie Verbaeten and Anneleen Van Assche. Ensemble methods for noise elimination in classification problems. In *Multiple Classifier Systems*, volume 2709 of *Lecture Notes in Computer Science*, pages 317–325. Springer, 2003.