Noise-Aware Algorithms for Analysis of Galactic Spectra

H. Jair Escalante[1] and Olac Fuentes[2]

[1] Instituto Nacional de Astrofísica Óptica y Electrónica
Luis Enrique Erro 1 Puebla, 72840, México
hugojair@ccc.inaoep.mx
[2] University of Texas at El Paso
500 West University Avenue, El Paso TX, USA
ofuentes@utep.edu

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 the prediction of stellar population parameters, a challenging astronomical problem. Experimental results adding noise and useful anomalies to the data show that our algorithm provides a significant reduction in error, 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 [1]. 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 suspects of

© A. Gelbukh, S. Torres, I. López (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 19, 2006, pp. 81-92

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. In this work we oriented our efforts to improve data quality and prediction accuracy for machine learning problems, specifically, for the estimation of stellar population parameters, an interesting domain in which an algorithm based on re-measuring is suitable to test.

The paper is organized as follows: in the next section we present a brief survey of related works. In Section 3 we introduce the astronomical domain used in this work; in Section 4 the kernel methods that we used are described. In Section 5 the proposed algorithms are introduced. 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 [2–8]. However, we should not eliminate a datum unless we can determine that it is invalid. This often is not possible for several reasons, including: human-hour cost, time investment, ignorance about the domain we are dealing with and even 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 [9] 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 [10] and instance selection [11] 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 [8] and the saturation filtering algorithm [4]. Ensembles of classifiers had been successfully used to identify mislabeled instances in classification problems [12, 5, 13], 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 [14] and probabilistic knowledge [15], distance and similarity-dissimilarity functions [16–18], metrics and kernels [19], 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 [20] an algorithm to find the support of a dataset, which can be used to find outliers, is presented; in [6] the sphere with minimal radius enclosing most of the data is found and in [7] 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 Estimation of Stellar Populations Parameters

In most of the scientific disciplines we are facing a massive data overload, and astronomy is not the exception. With the development of new automated telescopes for sky surveys, terabytes of information are being generated. Such amounts of information need to be analyzed in order to provide knowledge and insight that can improve our understanding about the evolution of the universe. Such analysis becomes impossible using traditional techniques, thus automated tools should be developed. Recently, machine learning researchers and astronomers have been collaborating towards the goal of automating astronomical data analysis tasks. Such collaborations have resulted in the automation of several astronomical tasks. These works include galaxy classification [21], prediction of stellar atmospheric parameters [22] and estimation of stellar population parameters [23].

In this work we applied our algorithms for the prediction of stellar population parameters: ages, relative contribution, metal content, reddening and redshift. In the remaining of this section the data used are briefly described.

3.1 Analysis of Galactic Spectra

Almost all the relevant information about a star can be obtained from its spectrum, which is a plot of flux against wavelength. An analysis of a galactic spectrum can reveal valuable information about stellar formation, as well as other physical parameters such as metal content, mass and shape. The accurate knowledge of these parameters is very important for cosmological studies and for the understanding of galaxy formation and evolution. Template fitting has been used to carry out estimates of the distribution of age and metallicity from spectral data. Although this technique achieves good results, it is very expensive in terms of computing time and therefore can be applied only to small samples.

Modeling Galactic Spectra Theoretical studies have shown that 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 1, with their respective metallicity, together with a model of the effects of interstellar dust in these individual spectra. Interstellar dust absorbs energy preferentially at short wavelengths, near the blue end of the visible spectrum, while its effects on longer wavelengths, near the red end of the spectrum, are small. 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. This is taken as evidence that the universe is expanding and that it started in a Big Bang. More distant objects generally exhibit larger redshifts; these more distant objects are also seen as they were further back in time, because the light has taken longer to reach us.

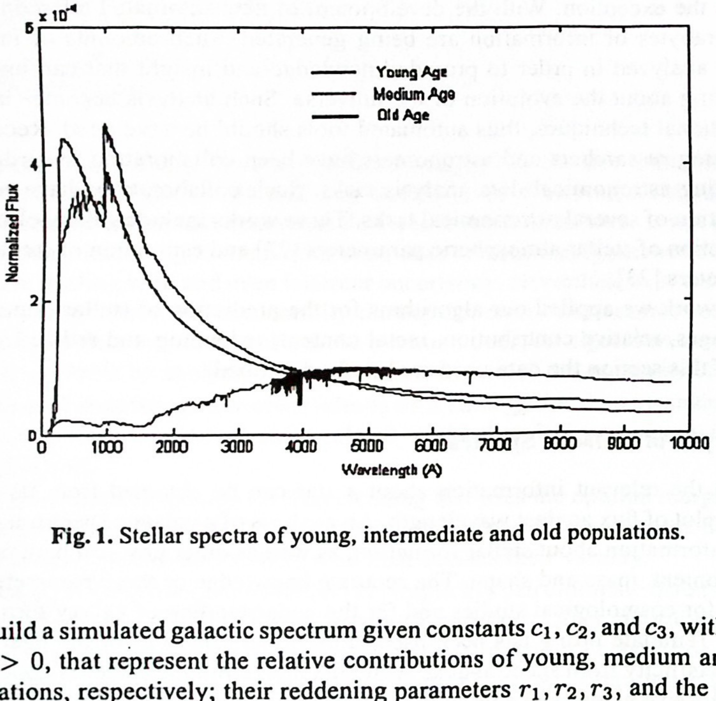

Fig. 1. Stellar spectra of young, intermediate and old populations.

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,

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

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.

4 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.

4.1 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) [24] 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 $\mathbf{v} \in F \setminus \{0\}$ 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 [25]. In this work we used a polynomial kernel (Eq. 2).

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

4.2 Kernel based novelty detection

In order to develop an accurate nose-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 [26]. This algorithm presented in [19] 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. This is an efficient and very precise method; for this work we used a polynomial kernel function (Eq. 2) of degree 1.

5 Noise-Aware Algorithms

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. Then, 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.

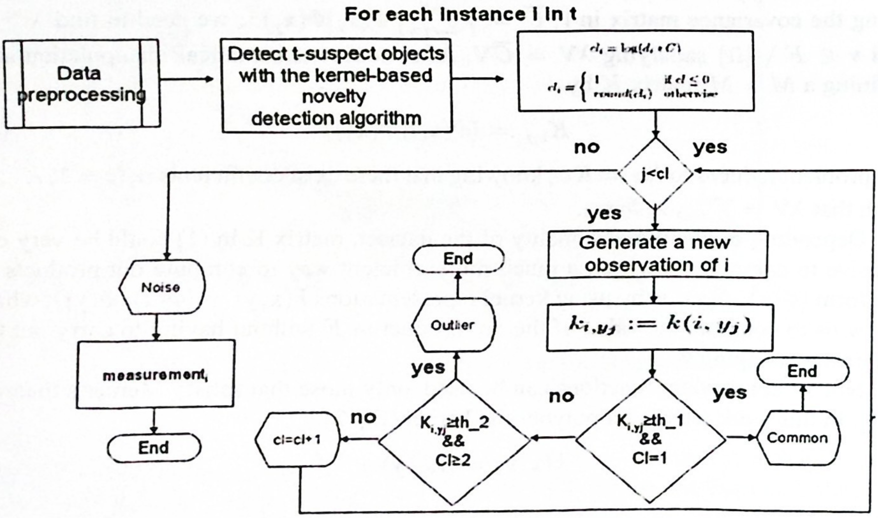
For each Instance I In t
Data preprocessing
Detect t-suspect objects with the kernel-based novelty detection algorithm
$cl_i = \log(d_i \cdot C)$
$cl_i = \begin{cases} 1 & \text{if } cl \le 0 \\ round(cl_i) & \text{otherwise} \end{cases}$
no
yes
j<cl
yes
Generate a new observation of i
End
Noise
Outlier
$k_{i,y_j} := k(i, y_j)$
measurement$_i$
yes
no
yes
End
no
End
cl=cl+1
$K_{i,y_j} \ge$ th_2 && Cl≥2
$K_{i,y_j} \ge$ th_1 && Cl=1
Common

Fig. 2. Block diagram of the base noise-aware algorithm.

In Figure 2 the base noise-aware algorithm is shown. The data preprocessing module includes dimensionality reduction, scaling data, feature selection or similar necessary processes. In the next step, suspect data are identified by using an anomaly detection method. 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. *cl* is obtained from the distance of the suspect instances to the center of mass of the full data set, and it is defined in Eq. (4).

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

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.

For the anomaly-noise discrimination we decided to use a kernel, since kernels can be used to calculate similarity between objects [25]. Several kernels were tested, but the kernel that best distinguished among dissimilar instances was the extended radial basis function (Eq. 5) with $\sigma = 0.25$.

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

We generated simple rules to discriminate among noise, outliers and common instances. If an object is correct, the algorithm leaves that object intact, otherwise, the noisy observation is substituted by one in the new measurements. 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. As we can see, outliers and common instances will be detected with only a new observation, while noise will be re-measured a few times, finally all of the noise is substituted by a correct measurement or in other approach by the average of the re-measurements.

The algorithm from Figure 2 can be used for cleaning datasets, eliminating all of the noise and retaining correct observations. Now we have to describe how to take advantage of it to improve the results of a machine learning task.

In Figure 3, the base noise-aware algorithm is adapted to predict the stellar population parameters in the astronomical data, using locally-weighted linear regression LWLR [27], a well known machine learning algorithm.

We have divided the data cleaning process into two phases: training and testing. Data cleaning in training is just what we have descibed before in the base algorithm. Data cleaning for testing data is somewhat different, in this setting we have a new data set of p (unseen) new observations. Then, the algorithm uses the distance of each test observation to the center of mass of the improved training set to determine the set of suspicious test data. Suspect observations are re-measured. Then, the erroneous observations are differentiated from correct observations and wrong data are substituted by the average of their measurements, while for correct rare observations the original measurement was used. In the case of correct observations we could also use the average of the measurements, which, as we will see, results in better accuracy in experiments with noisy data.

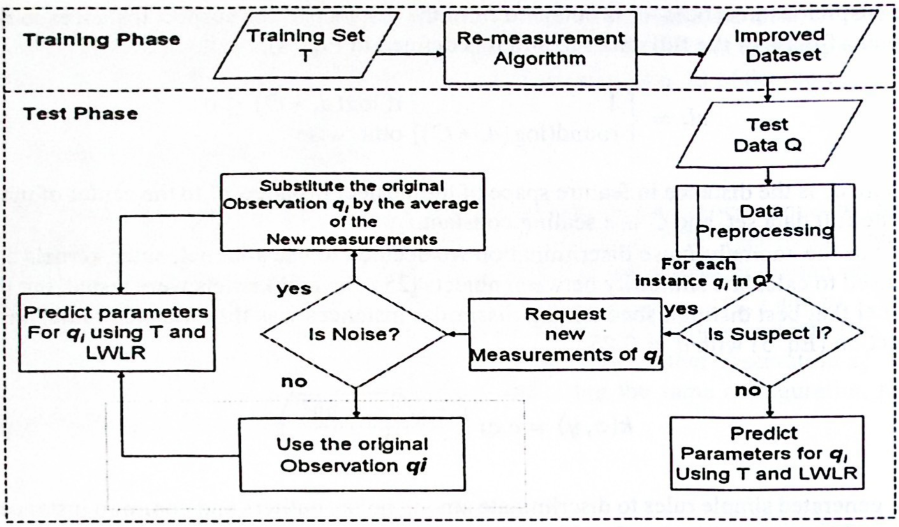
Training Phase
Training Set T
Re-measurement Algorithm
Improved Dataset
Test Phase
Test Data Q
Data Preprocessing
For each instance q_i in Q
Is Suspect i?
yes
no
Request new Measurements of q_i
Is Noise?
yes
no
Substitute the original Observation q_i by the average of the New measurements
Predict parameters For q_i using T and LWLR
Use the original Observation *qi*
Predict Parameters for q_i Using T and LWLR

Fig. 3. Block diagram of a noise-aware machine learning algorithm.

6 Experimental Results

We performed several experiments in order to test the performance of our algorithms. In each experiment we generated a dataset of 200 observations for training and 3 datasets of 100 observations for testing. We used LWLR as learning algorithm considering a neighborhood of 80 objects. All results presented here are averages over the three test datasets.

In the first experiment we tested the base noise-aware algorithm inserting noise and outliers to the datasets. For this experiment all of the data were affected with low-level noise; 5% of the data were affected with extreme gaussian noise with zero mean, and varying the value of σ^2, as shown in Figure 4. Furthermore, 5% of the data were affected by inserting useful anomalies.

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 $(2 \leq Z \leq 4)$.

The experiment consists of applying the algorithm from Figure 3, to the prediction of the stellar population parameters, using a training dataset previously improved with the algorithm from Figure 2. Results of these experiments are shown in Table 1; the mean absolute error (M.A.E.) reduction is presented. We report results using different configurations for training and testing. We can see that the best results are those obtained when the training set has been improved with our algorithm. The best result was obtained when the original (affected) test data were used, however, there is not a sig-

Table 1. Percentage of M.A.E. reduction for the different configurations on the training and test sets. Noisy is the original (affected) data set, and noise-aware is the data that have been previously improved with our algorithm. The first column indicates the training data used, while the first row indicates the test data used.

Training/Test	*Noisy*	*Noise-Aware*
Noisy	0	0.01
Noise-Aware	4.19	3.46

nificant difference. What is important to notice is that an improvement in the training set results in an improvement of the prediction accuracy in the test sets. Something remarkable, that is not shown in the tables, is that the noise-aware algorithm detected 14 of the 15 total artificially-added anomalies on the test datasets. Furthermore, 100% of the noisy observations were corrected, which would result in data quality improvement without a loss of useful information. In order to determine how much the heuristics implemented in the noise-aware algorithms help to improve the accuracy, we performed another experiment. In the following experiments we compared the performance of our algorithm with one that re-measures randomly, without repetition; again, we divided the data into training and test sets. For these experiments, all of the data sets were affected with 4 different noise levels (gaussian, with mean zero and varying the standard deviations), see Figure 4. The experiment consists of comparing the noise-aware algorithm form Figure 3 with one that randomly chooses instances to re-measure. In this scenario, we have the capability of performing R new measurements. Therefore, the random method performs a new measurement of R objects chosen randomly, without repetition. On the other hand, the noise-aware algorithm (Figure 3) iterates on the data

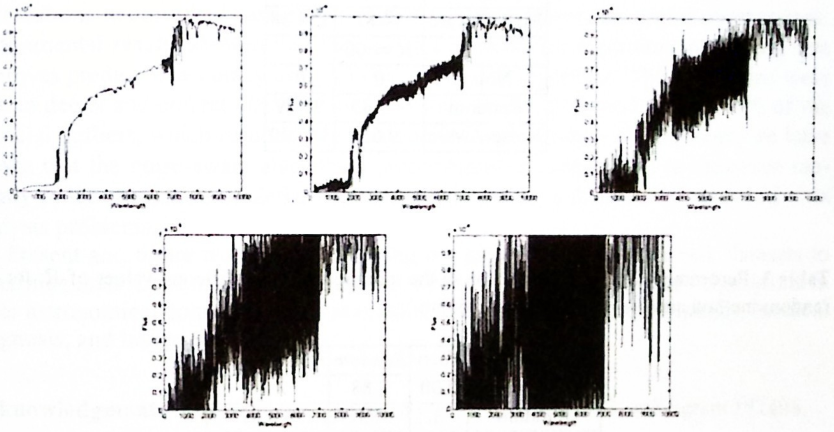

Fig. 4. Sample spectra with the different levels of noise added. In all of the figures, the noise is Gaussian with zero mean and varying the standard deviation in each case.

set, until R re-measurements are made. That is, in each iteration the algorithm identifies, re-measures and corrects erroneous observations. We substituted the noisy observations by the average of the new measurements, due to the nature of the noise added. The results for the training phase, with $R = 200, 100, 66$, are presented in Table 3. We can

Table 2. Percentage of M.A.E. reduction, Noisy is the original (affected) dataset; noise-aware is the dataset that has been improved with our algorithm; random is the dataset improved with the method that re-measures randomly.

	$R = 200$	
	%	Time
Noisy	0	0
Random	−6.35	273.86
Noise-Aware	15.54	298.56
	$R = 100$	
Noisy	0	0
Random	−7.11	138.9
Noise-Aware	14.82	154.38
	$R = 66$	
Noisy	0	0
Random	−1.39	90.79
Noise-Aware	9.65	147.40

Table 3. Percentage of M.A.E. reduction in the training phase for different values of R, for the random method and the noise-aware algorithm.

Training/Test	*Noisy*	*Random*	*Noise-Aware*
Noisy	0.00	2.88	2.12
Random	−3.4	−5.86	−2.07
Noise-Aware	6.15	7.01	6.61

see from Table 3 that there is a clear improvement by using our algorithm instead of the one that re-measures randomly. Indeed, when the random method is used there is a slight decrease in accuracy. The improvement is large when we iterate our algorithm until 200 new measurements are made. Moreover, the difference in processing time is small. The performance of the algorithms in the test sets can be seen in Table 2. Again, we presented different configurations for training and testing. From Table 2, we can observe that the best result was obtained when we used the improved training data. For testing, the best result was obtained when the random algorithm was used. However, the difference in accuracy is small. We performed the same experiment but instead of using the original measurement for low and medium noise affected observations, we used the average of the new-measurements. Results of this experiment are shown in Table 4. We can see that there is a clear improvement in our algorithm when all of the suspect data

Table 4. Percentage of M.A.E. reduction for the different configurations of training and test sets. In this experiment all of the suspect observations were substituted by the average of the new measurements in the noise-aware algorithm.

Training/Test	*Noisy*	*Random*	*Noise-Aware*
Noisy	0.00	0.21	2.81
Random	−2.46	−2.7	−1.18
Noise-Aware	5.69	6.74	10.88

were substituted. With this modification, the best result is obtained when both training and testing data were improved with our algorithm. The improvement is around 11% in accuracy. The behavior of the random method was similar to that in Table 2.

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. Experimental results showed that the use of a noise-aware algorithm in training sets improves prediction accuracy using LWLR as learning algorithm. The algorithms were able to detect and correct 100% of the erroneous observations and around 90% of the artificial outliers, which resulted in a data quality improvement. Furthermore, we have shown that the noise-aware algorithms outperformed a method that re-measures randomly in the prediction of stellar population parameters, a difficult astronomical data analysis problems.

Present and future work includes testing our algorithms on 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.

References

1. Tamraparni Dasu and Theodore Johnson. *Exploratory Data Mining and Data Cleaning*. Probability and Statistics. Wiley, 2003.
2. Carla Brodley. Identifying mislabeled training data. In *Journal of Artificial Intelligence Research*, volume 11, pages 131–167, 1999.
3. 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.
4. 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.
5. 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.

6. 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. ISBN 2-600049-9-X.
7. 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.
8. George H. John. Robust decision trees: Removing outliers from databases. In *Proceedings of the 1st. Int. Conf. on KDDM*, pages 174–179, 1995.
9. 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.
10. David B. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *ICML*, pages 293–301, 1994.
11. 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.
12. Carla E. Brodley and Mark A. Friedl. Identifying and eliminating mislabeled training instances. In *AAAI/IAAI, Vol. 1*, pages 799–805, 1996.
13. David Clark. Using consensus ensembles to identify suspect data. In *KES*, pages 483–490, 2004.
14. Vic Barnett and Toby Lewis. *Outliers in Statistical Data*. John Wiley and Sons, 1978. ISBN 0-471-99599-1.
15. J. Kubica and A. Moore. Probabilistic noise identification and data cleaning. In *Technical Report CMU-RI-TR-02-26, CMU*, 2002.
16. 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.
17. 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.
18. 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. ISBN 1-58113-218-2.
19. John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
20. B. Schölkopft, A. Smola, R. Williamson, and P. Bartlett. New support vector algorithms. In *Neural Computation*, volume 12, pages 1083 – 1121, 2000.
21. Jorge de la Calleja and Olac Fuentes. Machine learning and image analysis for morphological galaxy classification. *Monthly Notices of the Royal Astronomical Society*, 349:87–93, 2004.
22. Olac Fuentes and Ravi K. Gulati. Prediction of stellar atmospheric parameters using neural networks and instance-based learning. In *Experimental Astronomy 12:1*, pages 21–31, 2001.
23. 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.
24. 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.
25. Ralf Herbrich. *Learning Kernel Classifiers*. MIT press, first edition, 2002. ISBN 0-262-08306-X.
26. H. Jair Escalante. Noise-aware machine learning algorithms. Master's thesis, Instituto Nacional de Astrofisica Óptica y Electrónica, January 2006.
27. Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1-5):11–73, 1997.