Kernel Methods for Anomaly Detection and Noise Elimination

H. Jair Escalante

Instituto Nacional de Astrofísica Óptica y Electrónica
Luis Enrique Erro 1 Puebla, 72840, México
hugojair@ccc.inaoep.mx

Abstract. A kernel-based algorithm for *useful*-anomaly detection and noise elimination is introduced. The algorithm's objective is to improve data quality by correcting wrong observations while leaving intact the correct ones. The proposed algorithm is based on a process that we called "*Re-Measurement*" and it is oriented to datasets that might contain both kinds of rare objects: noise and useful-anomalies. Two versions of the algorithm are presented $R-V1$ and $R-V2$. Both algorithms generate new observations of a suspect object in order to discriminate between erroneous and correct observations. Noise is corrected while outliers are retained. Suspect data is detected by a kernel-based novelty detection algorithm. We presented experimental results of our algorithm, combined with KPCA, in the prediction of stellar population parameters a challenging astronomical domain, as well as in benchmark data.

1 Introduction

Real world data are never as perfect as we would like them to be and often can suffer from corruption that may affect data interpretations, data processing, classifiers and models generated from data as well as decisions made based on data. Affected data can be due to several factors including: ignorance and human errors, the inherent variability of the domain, rounding errors, transcription error, 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 affected data. As we can see corrupted data may be: noise (erroneous data) or anomalies (rare but correct data) and it would be very useful to differentiate between them from the rest of data. An expert can perform this process but it requires a lot of time investment which yields in expensive human-hour costs, from here arises the necessity of automate this task. However this is not an easy task since outliers and noise may look quite similar for an algorithm, then we need to add to such algorithm a more human-like reasoning. In this work the re-measurement idea is proposed, this approach consist of detecting "*suspect*" data and by generating new observations of these objects we can correct errors, while retaining anomalies for posterior analysis. This algorithm could be useful in several research areas, including: machine learning, data mining, pattern recognition, data cleansing, data warehousing, information retrieval and applications such as: security systems and medical diagnostic. In this work we oriented our efforts to improve data quality and prediction accuracy for

© *A. Gelbukh, S. Torres, I. López (Eds.)*
Advances in Computer Science and Engineering
Research in Computing Science 19, 2006, pp. 69-80

machine learning problems, specifically for the estimation of stellar population parameters a domain in which the re-measurement algorithm is suitable to test.

Elimination of suspect objects have been widely used for most of anomaly detection methods [1–6], the popularity of this approach comes from the fact that they can alter calculated statistics, increase prediction error, turn more complex a model based on these data or possibly they introduce a bias in the process to which they are dedicated. However we should not eliminate an observation unless, like an expert, we can determine the incorrectness of the datum. This often is not possible for several reasons: human-hour cost, time investment, ignorance about the domain we are dealing and even uncertainty. Nevertheless if we could guarantee that an algorithm successfully will distinguish errors from rare objects with high confidence level the difficult task would be solved. Like an human does, an algorithm can confirm or discard a hypothesis by analyzing several samples of the same object.

Re-measurement is safer than elimination by several reasons: we can conserve rare objects and decide what to do about them, we can ensure that an anomaly is correct, we can eliminate the wrong objects from our dataset, we can be sure that a common instance will never be affected, all of these reasons make suitable the use of re-measurement instead of elimination in certain domains.

2 Estimation of Stellar Populations Parameters

In most of the scientific disciplines we are facing a massive data overload, astronomy is not the exception. With the development of new automated telescopes for sky surveys, terabytes of information are being generated. Recently machine learning researchers and astronomers have been collaborating towards the goal of automatizing astronomical analysis tasks. Almost all information about a star can be obtained from its spectrum, which is a plot of flux against wavelength. An analysis on galactic spectrum can reveal valuable information about star formation, as well as other physical parameters such as metal content, mass and shape.

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, each with different metallicity and 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 is then $d(\lambda) = f(\lambda)(1 - e^{-r\lambda})$, 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.

Therefore, a simulated galactic spectrum can be built given c_1, c_2, c_3, with $\sum_{i=1}^{3} c_i = 1, c_i > 0$ 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, $a_3 \in \{10^9, 10^{10.2}\}$ 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.

3 Methods

Kernel methods have demonstrated been useful tools for pattern recognition, dimensionality reduction, denoising, and image processing. In this work we used kernel methods for dimensionality reduction of spectral data. Also we used a kernel-based method for novelty detection. Furthermore the re-measurement algorithm differentiates anomalies from noise by using a kernel. In this section KPCA and the algorithm for anomaly detection used are briefly described.

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) [7] 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 [8]. 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 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 [9]. This algorithm presented in [10] 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.

4 Re-Measurement Algorithm

Before introducing the re-measurement algorithm, the concept of the "re-measurement" process should 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 every $x_i^s, i = \{1, 2, \ldots, m\}$, is suspect to be a incorrect observation. 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.

The idea of re-measurement is based on the natural way in which a human clarifies his doubts; when a person is doubtful about the correctness of a datum he/shee can check the datum's validity by analyzing several observations of the same object. For example, in case our observations were pictures for face recognition, the re-measuring process would consists of taking another picture for every suspect object in our data set.

The re-measurement algorithm uses a confidence level value (*cl*) which tell us how rare a suspect object is. *cl* can indicates the number of re-measurements to perform for

Table 1. The $R - V1$ algorithm

```
Generate a dataset T which columns are attributes and rows instances
not-outlier-th:=0.99; outlier-th:=0.8;

0. Obtain the principal components ($PC_T$) for T
1. Identify t-suspect observations from $PC_T$

2. For each $i \in t$

    - $cl_i := \ln(d_i * C)$
    - $measurements_i := cl_i$-new observations of object $i$
    - $k_{avg} := \frac{1}{cl} \sum_j^{cl} k(i, y_j)$,

    - if ($k_{avg}$ ≥not-outlier-th and $cl = 1$): return(not-outlier)
    - else if($k_{avg}$ ≥outlier-th):return(outlier)
    - else return(noise)

3. For each $i \in t$ labeled as noise : $PC_T i := measurements_{irand}$
```

each suspect instance. cl value is obtained from the distance of the suspect objects to the center of mass in the feature space and it is defined in (4),

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

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 [8]. Several kernels were tested but the kernel that best distinguished between instances was the extended radial basis function (5) with $\sigma = 0.25$. This kernel returns a 1 value if the instances (x, y) are identical or a value between $(0, 1)$, if they are different, that indicates how similar objects (x, y) are, near to 1 indicates more similitude.

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

Using this property of the kernel we generated simple rules to differentiate between noise, anomalies and common instances.

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

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. In Table 1 the re-measurement algorithm applied to the prediction of stellar population parameters is presented, in step 0 we used KPCA as a preprocessing step. Next we applied the KB-novelty detection algorithm and with a little modification we forced the algorithm to return the top t-farther objects with their distance to the center of mass. cl-value is calculated and cl-new observations are generated and stored in $measurements_i$, then we calculate k_{avg}. This k_{avg} is compared with our thresholds and the algorithm decide the type of object, finally the erroneous objects are substituted by a random sample in $measurements_i$.

4.1 Reducing the Number of Re-Measurements

The proposed algorithm performs well but it requires of nearby 5 new samples to identify anomalies and above 2 for noisy objects, in some domains (including astronomy) the generation of a new instances is expensive and obtaining 5 or 4 new measurements is complicated. A little modification in the algorithm will overcome this difficulty, by a slightly change in our rules and by verifying them each time that a new measurement is generated we would need only a new sample to identify common instances and anomalies and at most 2 more to detect noise, we will call this algorithm $R-V2$, the new rules are:

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

In Figure 1 the modification to the algorithm is shown. This time cl is used to complement the basic conditionals. Anomalies and common instances will be detected with only a new sample by using cl, while noise will be re-sampled a few times to discard confusions, finally all noise is substituted by a random sample.

5 Experimental Results

In order to test the performance of the re-measurement algorithms some experiments were performed. The stellar populations domain was used in the following way: in each experiment a dataset of 200 spectra is generated, 5% of this data is affected with additive(normal distributed) extreme noise (2.5% with positive mean and 2.5% with a negative one), another 5% of the data is shifted by a factor ($f \in R : 1 < f < 10$) simulating useful-anomalies.

We compared accuracy using the mean absolute error (M.A.E.) obtained by a classifier builded with locally weighted linear regression (LWLR)[11]. LWLR belongs to the family of instance-based learning algorithms, these algorithms build query specific local models, which attempt to fit the training examples only in a region around a query point. For this work we considered a neighborhood of 80 points to approximate the target function.

In Table 2 percentage reduction error is presented for algorithms $R-V1$ and $R-V2$, we reported the average of 5 runs using a 10-fold cross validation.

There is an important reduction of error when we used KPCA, the maximum error reduction is attained when all of the suspect objects are eliminated, although we are loosing useful information too. Our algorithms reach accuracy only 2% down the

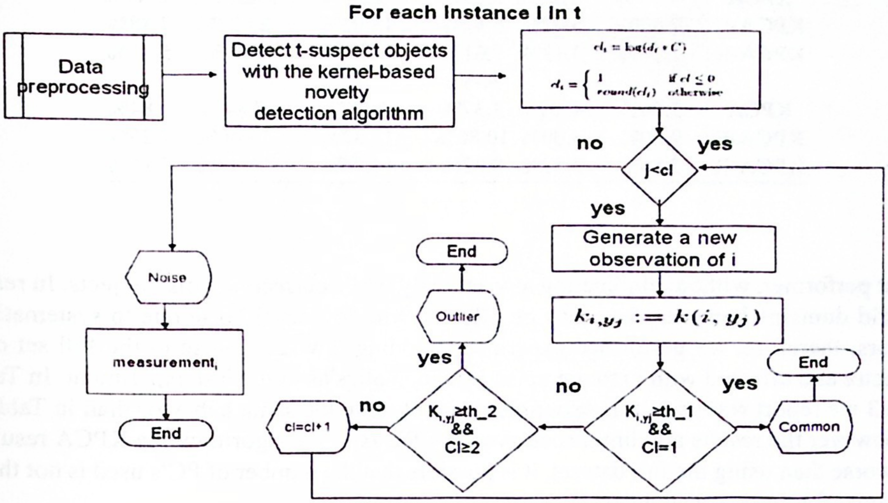
For each Instance I In t
Data preprocessing
Detect t-suspect objects with the kernel-based novelty detection algorithm
$cl_i = \log(d_i * C)$
$cl_i = \begin{cases} 1 & \text{if } cl \le 0 \\ round(cl_i) & \text{otherwise} \end{cases}$
no
yes
j<cl
yes
Generate a new observation of i
$k_{i,y_j} := k(i, y_j)$
End
Noise
Outlier
measurement,
yes
End
no
yes
End
no
$K_{i,y_j} \ge th\_2$ && Cl≥2
$K_{i,y_j} \ge th\_1$ && Cl=1
cl=cl+1
Common

Fig. 1. Block diagram of the $R - V2$ algorithm

Table 2. Reduction percentage of M.A.E. for the prediction of stellar populations parameters regarding as baseline the M.A.E. obtained when the full-affected dataset was used, compared with using 10-KPCA, 10-KPCA when all suspect data is eliminated (KPCA-E) and using 10-KPCA with the re-measurement algorithm (KPCA-R), clean data was used

Method	Reddening	Metal	Ages	Contributions	Redshift	Average
			R-V1			
KPCA	17.29%	10.17%	13.58%	20.21%	-2.39%	11.29%
KPCA-E	20.88%	16.40%	19.19%	33.23%	29.90%	19.76%
KPCA-R	19.82%	14.30%	16.68%	27.14%	13.68%	16.13%
			R-V2			
KPCA	13.97%	1.57%	14.00%	20.60%	-4.30%	7.56%
KPCA-E	20.85%	7.29%	17.69%	31.85%	29.44%	14.96%
KPCA-R	16.98%	6.69%	15.53%	25.18%	16.24%	12.34%

Table 3. Reduction percentage of M.A.E. for the prediction of stellar population parameters, noisy data was used

Method	Reddening	Metal	Ages	Contributions	Redshift	Average
R-V1						
KPCA	-1.71%	-1.25%	3.09%	-1.24%	-12.01%	-0.47%
KPCA-E	7.00%	5.41%	6.42%	13.02%	23.23%	7.88%
KPCA-R	2.24%	3.13%	7.61%	7.76%	5.29%	5.41%
R-V2						
KPCA	2.66%	-0.62%	3.37%	3.07%	7.87%	1.84%
KPCA-E	9.80%	4.00%	10.80%	18.32%	25.40%	9.29%
KPCA-R	7.88%	3.57%	8.82%	11.46%	16.48%	7.16%

best performer, without eliminating any anomaly while correcting noisy objects. In real world domains however, data may be affected with low-level noise due to systematic errors, therefore, we performed experiments adding low-level noise to the full set of spectra and affected with extreme noise and anomalies as in the last experiment. In Table 3 we report results of this experiment, we observe the same behavior than in Table 2 however the results are diminished even, for the $R - V1$ algorithm, the KPCA result is worse than using the full dataset, it is possible that the number of PC's used is not the optimal for these affected data.

Accuracy improvement is significant when we used the re-measurement algorithm, however if we want to analyze data quality, accuracy may not be the best measure to compare. In Table 4 performance of the re-measurement algorithms $R - V1$ and $R - V2$ is shown.

As we see both algorithms detected and corrected 100% of the noise and none instance was confused. The anomaly detection rate was high although no perfect. There are not neither false anomalies nor false noisy objects detected. CLC is the *cl* value for common instances detected as suspicious and its value is obviously 1. CLO is the *cl* value for anomalies and it is of almost 5 for the $R - V1$ algorithm and of 1 for $R - V2$, this means that only a new sample was needed for identify anomalies and common instances while for the case of noisy objects *cl* value (CLN) is of 1.5. This results on the *cl* values confirm that the selection of *cl* (4) is adequate. Processing time decreases about 25% for the $R - V2$ algorithm in this artificial dataset which yields in saving some seconds, although for real data the decrement could be of hours.

Last three rows on Table 4 show the performance of the KB-algorithm for novelty detection. We present the F-measure value obtained by such algorithm. This measure is based on recall $\mathbf{R} = \frac{TP}{(TP+FN)}$ and precision $\mathbf{P} = \frac{TP}{(TP+FP)}$ and it is defined as $F = \frac{2*R*P}{(R+P)}$, where TP is for true positives, TN is for true negatives, FP is for false positives and FN is for false negatives. $F-$measure express with a real number in [0,1] the performance of an outlier detection method. We forced the novelty detection algorithm to return the top 30 points farther the center of mass and this is a reason of because $F-$measure value is not perfect, however a look on the TP and FP rates is more useful.

Besides the good performance of the re-measurement algorithms in the astronomical domain, we had doubts about the performance of our algorithms in other domains.

Table 4. Performance of the re-measurement algorithms: $R - V1$ and $R - V2$

Algorithm		R-V1		R-V2
Parameter / Data	Clean	Noisy	Clean	Noisy
Re-Measurement				
Anomalies Detected	90%	100%	80%	86.6%
Noise Detected	100%	100%	100%	100%
Confused	0	0	0	0
False Anomalies	0.33	1.33	0	0
False Noise	0	0	0	0
CLC-value	1	1	1	1
CLO-value	4.86	4.26	1	1
CLN-value	1.5	1.37	1.6	1.5
Time(s)	77.57	87.64	55.34	57.32
Novelty Detection				
TP	19.33	20	19	20
FP	0.67	0	1	0
F-measure	0.77	0.8	0.76	0.8

For this reason we performed experiments on ten data sets from the UCI repository[12], the datasets used are briefly described in Table 5.

In this experiments we used only the $R - V2$ algorithm since it is the best performer on the above experimentation, moreover we performed experiments with noise only, since it allow us to simulate the re-measurement process. Each data set was normalized to the range $[0, 1]$ and it was affected as with the astronomical domain. Results on accuracy for these datasets are show in Table 8, while the performance results are presented in tables 6 and 7.

As we can see the $R - V2$ algorithm performance on UCI data is similar to the observed in the astronomical data. There is an accuracy improvement in all of the datasets

Table 5. UCI Datasets description

ID	Name	$\#_{Cases} - \#_{Features}$	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

Table 6. Performance of the $R-V2$ algorithm for the UCI datasets. We present: CLC, CLO and CLN as before, outliers detected (O.D.), noise detected (N.D.), confusions(Conf.), false outliers (F.O.) and false noise (F.N.)

Dataset	W	G	H	A	I	M	L	C	B	Ab
CLN	3.33	3.66	3.58	3.67	4.29	4.15	2.92	3.82	4.21	3.88
O.D.(%)	100	100	96.15	100	100	100	100	100	100	100
N.D.(%)	100	100	98.67	100	100	100	100	100	96.67	98
F.O.	0	0	0	0	0	3	0	0	2	11

Table 7. Performance of the kernel-based novelty detection algorithm used. We present the number of suspect observations detected, true positives and false negatives, recall, precision and $F-$measure

Dataset	W	G	H	A	I	M	L	C	B	Ab
Suspect	27	32	76	5	23	31	22	102	29	150
TP	18	21	49.66	3	15	21	15	68	18.66	99
FN	0	0	1.33	0	0	0	0	0	0.33	1
Rec	1	1	0.97	1	1	1	1	1	0.98	0.99
Prec	0.76	0.66	0.65	0.6	0.65	0.68	0.68	0.67	0.64	0.66
F	0.8	0.79	0.78	0.75	0.79	0.81	0.81	0.8	0.78	0.79

Table 8. Reduction percentage of M.A.E. for each dataset, when suspect data is eliminated(E) and when we used the $R-V2$ algorithm, compared with the prediction of LWLR using the full data. In last 3 rows, novelty detection algorithm performance is presented

Dataset	W	G	H	A	I	M	L	C	B	Ab
					Red %					
E	0.54	15.76	15.77	24.61	5.23	13.89	6.29	1.83	0.85	5.86
$R-V2$	10.86	14.74	34.54	28.37	4	28.62	5.63	1.75	2.24	5.08

when we used $R - V2$ even in some results our algorithm improved the elimination of suspect data. The algorithm needed only a new sample to identify outliers and common instances and nearby 4 to detect noise. There are not confusions and the false outliers rate was low, although we had false outliers only in two datasets. Performance of the kernel-based algorithm for novelty detection again is almost perfect.

6 Conclusions

We have introduced the re-measurement process as an option for useful-anomaly and noise differentiation. Two kernel based algorithms were presented, $R - V2$ needs only a new sample to identify anomalies and at most two for noisy objects. Anomalies remain unaffected while noise is substituted in an almost automated process (an user may be needed to generate the new measurements). Our algorithms are model data independent and can be generalized for non real valued domains, since they are based on kernels.

Experimental results on an astrophysics domain as well as on benchmark data are presented, our algorithm combined with KPCA improves prediction accuracy and data quality for the astronomical domain, while for the UCI data the same pattern is observed showing the generalization ability of our algorithm. This algorithm could be useful in domains requiring of highly-reliable data or in those in which the novelty is more interesting than the rest of the objects.

Acknowledgments This work was partially supported by CONACYT under grant 181498.

References

1. Carla Brodley. Identifying mislabeled training data. *JAIR*, 11:131–167, 1999.
2. R. T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In *20th ICVLDB*, pages 144–155, 1994.
3. D. Gamberger, N. Lavrač, and C. Grošelj. Experiments with noise filtering in a medical domain. In *Proc. 16th ICML*, pages 143–151. Morgan Kaufmann, San Francisco, CA, 1999.
4. David Tax and Robert 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.
5. B. Schölkopf, o Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution, 1999.
6. George H. John. Robust decision trees: Removing outliers from databases. In *Proc. of the 1st ICKDDM*, pages 174–179, 1995.
7. B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. In *Neural Computation 10*, pages 1299–1319, 1998.
8. R. Herbrich. *Learning Kernel Classifiers*. MIT press, first edition, 2002. ISBN 0-262-08306-X.
9. H. Jair Escalante. Resampling algorithms for machine learning. Master's thesis, Instituto Nacional de Astrofísica Óptica y Electrónica, to appear, 2005.
10. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
11. Christopher Atkeson, Andrew Moore, and Stefan Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1-5):11–73, 1997.
12. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.