A Comparison of Outlier Detection Algorithms for Machine Learning

H. Jair Escalante
hugojair@ccc.inaoep.mx

Computer Science Department
National Institute of Astrophysics, Optics and Electronics
Luis Enrique Erro # 1, Santa María Tonantzintla, Puebla, 72840, México

Abstract. In this paper a comparison of outlier detection algorithms is presented, we present an overview on outlier detection methods and experimental results of six implemented methods. We applied these methods for the prediction of stellar populations parameters as well as on machine learning benchmark data, inserting artificial noise and outliers. We used kernel principal component analysis in order to reduce the dimensionality of the spectral data. Experiments on noisy and noiseless data were performed.

Keywords: Outlier Detection, Machine Learning, Stellar Populations.

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 interpretations, data processing, classifiers and models generated from data as well as decisions made based on data. The corruption can be due to several factors, including: ignorance and human errors, the inherent variability of the domain, rounding errors, transcription error, instrument malfunction and biases. Unusual data also can be due to rare, but correct, behavior, which often result in interesting findings, motivating further investigation. For these reasons it is necessary to develop techniques that allow us to identify unusual data.

Outlier detection algorithms are useful in areas such as: machine learning, data mining, pattern recognition, data cleansing, data warehousing and applications as: credit fraud detection, security systems, medical diagnostic, network intrusion detection and information retrieval. In this work we compared the performance of outlier detection methods for machine learning problems, specifically for the estimation of stellar populations parameters a challenging astronomical domain. The paper is organized as follows. The next section presents a brief related work survey on outlier detection, followed in Section 3 by the description of the astronomical domain used. In Section 4 the compared methods as well as the kernel principal component analysis for dimensionality reduction are briefly described. In Section 5 experimental results are presented and finally in Section 6 we summarize our findings and discuss future directions of this work.

2 Outlier Detection

According to statistical literature[2] "...An outlier is an observation which appears to be inconsistent with the remainder of a set of data...". As the authors state the term *"appears to be inconsistent"* is the main problem when dealing with outliers and this is the problem that outlier detection methods try to solve. The problem has been approached using statistical and probabilistic knowledge, distance and similarity-dissimilarity functions, metrics and kernels, accuracy when dealing with labeled data, association rules, properties of patterns and other specific domain features. *How many outliers are present in a dataset?* this number depends on the domain and conditions in which data were recorded, but in order to give a bound Hampel comments[9, 7] *"altogether 5 to 10% wrong values in a dataset seem to be the rule rather than the exception"* which agrees with [14]. Therefore, every time that we are using a dataset we are facing unusual data.

There is a lot of work on outlier detection including statistical works[2], which use properties of data distribution, as well as probabilistic[19] and bayesian[31] techniques attempting to find the model of the anomalies, however, these approaches are oriented to univariate data or multivariate samples with only a few dimensions, processing time is also a problem when a probabilistic method is used.

Distance based techniques [1, 17, 25] have been proposed, these kind of methods use distance and dissimilarity functions between attributes[17], instances[25] or series of objects[1] in order to detect deviation in data sets, these methods perform well but often they require parameters difficult to set. Recent approaches use kernel properties to compute similarity between objects[32].

Variants and modifications to the support vector machine algorithm have been proposed trying to isolate the outliers class: an interactive method for data cleaning using the optimal margin classifier is presented in [11]; in [28] an algorithm to find the support of a dataset, which can be used to find outliers, is presented; in [34] the sphere with minimal radius enclosing most of the data is found and in [27] the correct class is separated from the origin and from the outlier class for a data set.

Prototype[33] and instance selection[5] implicitly can eliminate instances degrading the performance of instance-based learning algorithms. Some 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[12] and the saturation filtering algorithm[8]. Ensembles of classifiers had been successfully used to identify mislabeled instances in classification problems [6, 36, 7]. Also clustering algorithms implicitly can deal with outliers[23] or de-noise a data set [15].

Principal component analysis have been used to improve the detection [2, 13], by using a combination of the first PC's and the last few PC's. Techniques based on residuals for regression estimation have been proposed [2], but accuracy often is not the best measure to detect outliers.

As we can see there is a wide variety of methods for outlier detection, of course there are many more than the presented here, but we only considered some of the representative techniques. For experiments in this work six methods were implemented, these methods were compared attempting to determine the best technique that allows us to detect abnormalities in a dataset, such methods are: distance based, distance K-based,

a simple statistical approach, the ν-support vector machine, a kernel-based novelty detection method and the one-class algorithm.

3 Estimation of Stellar Population 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. 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.

Almost all information relevant 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 star 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.

3.1 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 3.1, 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 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. In a more realistic scenario we consider the redshift, that tell 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. Redshift can be due to several factors including movement of the source, expansion of space, gravitational effects. In this work we considered a non-relativistic formula to simulate redshift in spectra.

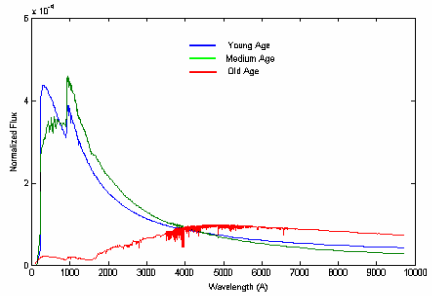

Fig. 1. Stellar spectra with ages: young, intermediate and old.

We build a simulated galactic spectrum 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,

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

with $m = [0.0004, 0.004, 0.008, 0.02, 0.05]$ in solar units and $m_1 \leq m_2 \leq m_3$, finally we add an artificial redshift Z by:

$$\lambda = \lambda_0(Z + 1), 0 < Z \leq 2$$

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 contribution (c_1, c_2, c_3) and redshift Z, starting from the spectra.

4 Methods

In this section the kernel principal component analysis is introduced. Also the compared methods: distance based, distance K-based, statistical, kernel-based, ν-SVM and one-class SVM are briefly described.

4.1 Kernel PCA

The stellar populations domain is a dataset of dimensionality $d = 12134$, then in order to perform experiments in feasible time we need a method for dimensionality reduction. Kernel principal component analysis (KPCA)[29] is a relative recent technique that takes the classical PCA technique to the feature space, taking advantage of "kernel functions". KPCA works in feature space F rather than in input space $\mathbf{R}^N$, this feature space F is obtained by a mapping from input linear space to a commonly nonlinear feature space F by $\Phi : \mathbf{R}^N \rightarrow F, x \mapsto X$. Then, in order to perform PCA on 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[1] and defining a MXM matrix K by

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

the problem reduces to:

$$l\lambda\alpha = K\alpha$$

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

$$k(x, y) = (\Phi(\mathbf{x}), \Phi(\mathbf{y})) \tag{2}$$

which allow to compute the value of the dot product $(\Phi(\mathbf{x}), \Phi(\mathbf{y}))$ in F without having to carry out the expensive mapping Φ. Not all dot product functions can be used, but just a set of them: the dot product functions which satisfy the Mercer's theorem [10]. This theorem implies that if k is a continuous kernel of a positive integral operator, there exist a mapping into a space where k acts as a dot product[29]. Commonly used kernels are polynomial (3) and gaussian (4), recent kernel functions include: sigmoid, spline, wavelet, laplace, fourier and anova functions among others.

$$k(x, y) = ((\mathbf{x}.\mathbf{y}) + 1)^d = (\Phi(\mathbf{x}), \Phi(\mathbf{y})) \tag{3}$$

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) = (\Phi(\mathbf{x}), \Phi(\mathbf{y})) \tag{4}$$

KPCA has the same goals that PCA and can be applied to the same problems with a comparable, and often superior, performance [26]. Furthermore, recently KPCA had been used in denoising applications [21, 26] with good results.

4.2 Distance Based Method

Knorr et al [16] tried to unify the notion of outliers by introducing the notion of distance based outliers defined as follows: *"An object O in a dataset T is a $DB(p, D)-$outlier if at least fraction p of the objects in T lies greater than distance D from O"*. The authors show [16–18] that this notion unifies the notions of outliers in data with different distributions (normal, poisson and t-student) and for regression models as well as for residual approaches. The method consists of calculating the distances of each object with the rest and comparing the distances with parameter D. The outliers will be the set of objects which distance with fraction p of data exceeds D. Parameters p and D need to be specified by the user; the value of p is commonly near to 1, it specifies the percentage

[1] For a detailed explanation of the method we refer the reader to[29]

of data that most be at distance D or higher; the value of D depends of the specific problem; often it is difficult to specify the value of D; commonly, the Euclidean distance is used as distance function. We implemented the nested loop algorithm presented in [17], which is the appropriate for data sets with dimensions higher than 5 [18]. For experiments we used values for p of 0.95 and 0.98, and the value of D was specified by trial and test experimentation for each dataset.

4.3 Distance K-Based Method

Ramaswamy et al [25] proposed a more general algorithm trying to avoid parameters difficult to set. The distance k^{th}-outliers notion D_n^k: *"Given a input data set with N points, parameters n and k, a point O is a D_n^k outlier if there are no more than n-1 other points O' such that $D^k(O') > D^k(O)$"*. $D^k(O)$ is the distance of O to their k^{th} nearest neighbor. Objects are sorted by its distance to its k^{th} nearest neighbor. However, it is possible that the k^{th} nearest neighbor of a point P, lies at a large distance from P, but the distances of P to its m-nearest neighbors, with $m < k$, are much more smaller than distance from P to its k^{th} neighbor, then P is not necessarily an outlier. Instead of using the D_n^k method we propose a variation (DK) considering the average distance of a point to its k-nearest neighbors. With this little modification to the original algorithm, outliers are such points farther from their k-nearest neighbors and not only from the k^{th} nearest neighbor. Here the objects with highest distance to their k-nearest neighbors are ranked. The top n points in this ranking are considered to be outliers. Again we used the Euclidean distance between instances. In this approach parameters n and k need to be specified but this is an easy task, since we can specify the probable number of outliers present in a dataset n, and the neighbors to consider k, for all experiments $k = 3$ and $n = 10\%$ of the total objects.

4.4 Simple Statistical Method

Many statistical approaches perform very well if we know the distribution of the data (normal, Poisson, t-student, etc), however, real datasets either do not follow any distribution or it is to difficult to find it. Independently of the distribution of data, statisticians have widely used the mean and standard deviation to identify outliers[2]. Therefore, we introduced the following definition of statistical based outliers, independent of any distribution: *"An object O is a $ST - (k, \rho)$ outlier if at least k attributes of O have higher values than ρ standard deviations from the mean"*. Those objects that have k attribute values outside a determined number of standard deviations ρ from the mean of that attribute are labeled as outliers. Clearly, for normal distributed data the value of $\rho = 3$ and for k a value near to the dimensions of data, however, for other data sets values are difficult to set.

4.5 Kernel-Based Novelty Detection Method

This method presented in [32] is a method that uses kernels, it calculates the center of mass for a dataset in feature space by using a kernel matrix K like in (1). A threshold

is fixed by considering both: an estimation error (Equation 5) of the empirical center of mass and the distances between objects and such center of mass in a dataset.

$$\sqrt{\frac{2 * \phi}{n}} * (\sqrt{2} + \sqrt{\ln \frac{1}{\delta}}) \tag{5}$$

where $\phi = \max(diag(K))$, and K is the kernel matrix of the dataset with size nXn, δ is a confidence parameter for the detection process. This is a kernel-based method of easy implementation, efficient and very precise, for this thesis we used a polynomial kernel, see Equation (3), to perform the mapping to feature space.

4.6 ν and One-class SVM

The support vector machine algorithm (SVM) is a training algorithm that find a separating hyperplane, which maximizes the distance of such hyperplane with respect to a set of examples mapped into a feature space making use of a kernel. Based on the structural risk minimization principle, the SVM algorithm minimizes both: the empirical risk of a dataset and a bound on the VC-dimension that controls the complexity of the learning machine[2], the complexity of the model is controlled by a constant C.

Since the apparition the support vector algorithm[4, 35] a lot of modifications and extensions have been proposed[22], one of such modifications is the ν-SVM algorithm, which was first proposed for regression[24] tasks and then extended for classification problems[28]. This algorithm substitutes the parameter C on the classical SVM algorithm by a ν-parameter ($\nu \in (0, 1]$), which interpretation is as an upper bound in the fraction of outliers and a lower bound in the fraction of support vectors[28]. Advantages of the ν-SVM over the classical SVM for regression are the elimination of parameters C and $\epsilon-$bound that often are difficult to set. In the ν-SVM one can differentiate outliers from support vectors by checking the value of the lagrange multiplier of each example, such examples with lagrangian α_i equal to $\frac{1}{n}$ are outliers while α's different from zero and lower than $\frac{1}{n}$ are support vectors.

Another variant of the SVM that combines approaches presented in [34, 28] is presented in [27], in this work a modification in constrains of the ν-SVM equations leads to maximize the distance of a hyperplane from the origin and then a fraction of data will lie beyond that hyperplane while allowing some outliers (between the origin and the hyperplane).

Summarizing, the main difference between the ν-SVM and the one-class SVM is the modification in the constrains for the optimization process. Furthermore, the one-class algorithm does not needs the labels of the training examples opposite to the ν-SVM, which is applicable only in supervised learning.

5 Experimental Results

In this section experimental results comparing the performance of the above described methods are presented. We used benchmark data sets from the UCI repository[3] and

[2] For a detailed description of the SVM algorithm and the variants presented here we refer the reader to [30, 22]

artificially we added outliers and noise randomly in order to determine which of the methods performs better. Also we used the stellar populations data set, in Table 5 the used data sets are described.

Dataset	I	D	O
Triazines	1116	60	56
Pyrimidines	220	28	15
WaveformII	1000	21	50
Musk	1000	166	50
Ionosphere	351	34	18
Stellar Population Data	200	10	10

Table 1. Description of datasets for the outlier detection experiments. I=number of cases, D=dimensionality of data and O=Number of outliers present (inserted) in the dataset.

The UCI datasets were normalized to the range [0,1] and we affected the data with 2 types of errors. In the first one we added "gaussian" noise with $\mu = 0, \sigma = 1$ multiplied by an equal constant for all datasets. The second form is inserting simulated outliers to the data, we multiplied the data by a factor in order to simulate rare objects. Accuracy seems to be not appropriate for evaluating methods for anomaly detection, therefore, we used a measure based on recall $\mathbf{R} = \frac{TP}{(TP+FN)}$ and precision $\mathbf{P} = \frac{TP}{(TP+FP)}$ called F-measure [20]

$$F = \frac{2 * R * P}{(R + P)} \tag{6}$$

where TP is for true positives, FP is for false positives and FN is for false negatives. The $F-$measure express with a real number in [0,1] the performance of an outlier detection method based on the detection rate and the precision obtained by such method, a 1 $F-$measure value indicates perfect performance while a 0 value means that the method did not detected any of the outliers present in data. In table 5 the $F-$measure value is presented for the UCI data, also the average for each method is reported. For each dataset and method we report results with additive noise (A) and simulated outliers (O).

The best performer is the kernel-based novelty detection approach which shows perfect performance four times, the method shows poor performance on the ionosphere data, although this method detected 100% of the outliers present in the dataset, the problem was that also detected several false positives. The statistical method and the DK method also show regular performance on all datasets. The worse performer was ν-SVM which only detected a few of true outliers, the one-class method as well as the DB approach detected almost 100% of the outliers in data but the false positives rate was so high.

In table 5 the performance of the methods for the stellar populations dataset is presented. We generated a dataset of 200 spectra for each experiment; we used KPCA with a polynomial kernel of degree 1; for dimensionality reduction, 10 PCs were used. Additionally and considering the ideas in [2, 13] for outlier detection we used 10 first

Dataset	T	DB	DK	ST	ν-SVM	KB	One-Class
Triazines	A	0.633	0.587	0.402	0	0.883	0.647
	O	0.640	0.778	0.779	0.103	1	0.034
Pyridines	A	0.4	0.444	0.222	0.129	0.629	0.490
	O	0.4	0.667	0.8	0.278	0.966	0.078
Waveform	A	0.175	0.667	0.863	0.014	0.990	0.662
	O	0.16	0.667	1	0.039	1	0.039
Ionosphere	A	0.275	0.075	1	0.2	0.226	0.182
	O	0.205	0.566	0.941	0	0.516	0.067
Musk	A	0.543	0.667	0.980	0	1	0.649
	O	0.617	0.667	1	0	1	0.078
Average	-	0.405	0.578	0.799	0.127	0.821	0.293

Table 2. F-measure value obtained by the tested methods on the UCI data, for additive noise(A) and simulated outliers(O)

components and the last 5 components in order to detect outliers difficult to identify. Experiments with high and low additive(A) noise as well as artificial outliers(O) were conducted. Average of each method is also presented. Experimental results show that the use of the 5 last components returned by KPCA does not improves accuracy detection. Also we can see that performance of all methods degrade on low-level additive noise, which could be due to the fact that the noise level is so low that there is not much difference between affected data and clean objects.

T	DB	DK	ST	KB	OC	Av.
A-15-High	0	0.9	0	1	0.4615	0.4523
A-10-High	0.7407	1	1	1	0.4737	0.8429
A-15-Low	0	0.2	0	0.05	0.0976	0.0695
A-10-Low	0	0	0	0.0588	0.0526	0.0223
O-15	0	0.8	0	1	0.4390	0.4478
O-10	0	0.9	0	1	0.2286	0.4257
Average	0.123	0.633	0.167	0.668	0.292	0.4257

Table 3. Outlier detection for the stellar populations data, F-measure value is reported, first column specifies the type of noise (A or O), the components used (10 for first 10-PCs and 15 for the first 10 plus the last 5 components) and the level of additive noise (high or low).

It would be very useful if we could differentiate between rare correct objects and highly noise observations by developing an algorithm able to do such task. However, before developing such algorithm we need to select the best method that allows us to detect both: rare objects (outliers), and highly noisy observations in a more real environment. In order to determine which method will performs better in such scenario the next experiment was performed. A dataset of 200 spectra was generated, then we add a distribution of noise and outliers to all of the data in the following way: 90% of data were affected with gaussian noise with $\mu = 0$, $\sigma = 1$ multiplied by a small factor, simulating

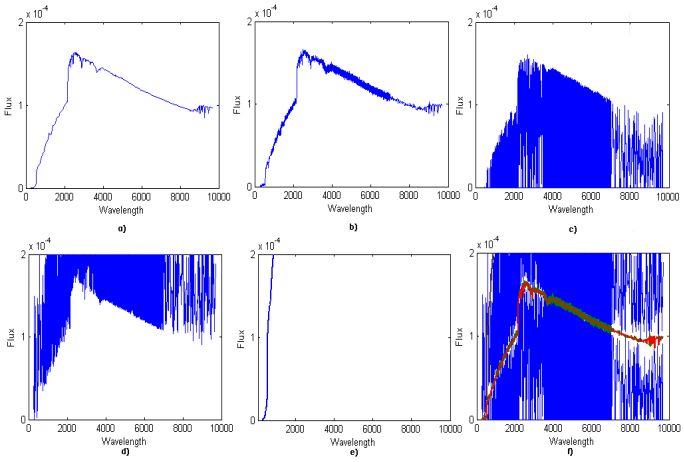

Fig. 2. A randomly selected spectrum a)original, b)affected with low-level noise, c)affected with negative extreme noise, d)affected with positive extreme noise, e) shifted spectrum, simulating an outlier, and f) all in one plot.

systematic errors. Also we affected the data with two distributions of extreme noise with positive and negative means, inserted with probability $p = 0.05$, shifted data simulating outliers were introduced by shifting a spectrum by a factor $(f \in R : 1 < f < 10)$, inserted with probability $p = 0.05$, see figure 5 to observe a sample spectra.

We compared accuracy using M.A.E. obtained by a classifier builded with locally weighted linear regression (LWLR). Outliers detected by each method are eliminated and a classifier is built on the rest of data, M.A.E. obtained is compared with the performance of KPCA using the 10 principal components and all data.

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. Learning with these algorithms consists of simply storing some or all of the training examples and postpone any generalization until a new instance must be classified. For this work we considered a neighborhood of 80 points to approximate the target function.

Results of this experiment are presented in table 5. From this table the best performer using the $F-$measure is the ST method, the kernel-based novelty detection algorithm reach a similar performance. Indeed the kernel-based method detected 1 outlier more than the ST method. Comparing accuracy we can see that the best performer is the kernel-based method, moreover, this method is the more efficient. Comparing TP, FP, $F-$measure, accuracy and processing time the best performer is again the kernel-based algorithm.

Results on data from the UCI repository and on the astronomical domain suggest that the most suitable technique to detect abnormality is the kernel-based algorithm for novelty detection. This algorithm identifies both kinds of observations: rare and highly-

noisy even in the presence of low-level noise, by investing the minor processing time and improving estimation accuracy of a classifier.

P	DB	DK	ST	ND	OC
TP	20	15	18	19	12
FP	23	15	4	2	14
(F_m)	0.6349	0.6	0.9473	0.89	0.5333
Red %	7.5%	3%	-3.2%	8.3%	0.9%
Time	1.793	0.421	0.561	0.38	0.601

Table 4. Performance of the outlier detection methods when the data is noise affected, true positives, false positives, $F-measure\ (F_m)$, average error reduction percentage (Red) and processing time are showed.

6 Conclusions

In this paper six methods for outlier detection were compared. Using benchmark datasets and a synthetic astronomical domain, several experiments were performed. The best performer using the $F-$measure in most of experiments was the kernel-based novelty detection method. In this method the center of mass for a kernel matrix is computed and a threshold for normal behavior is fixed, the method is effective and very simple. Results using the synthetic spectra show that the kernel-based method shows perfect performance on high-level noise and when data contains simulated outliers. DK and ST methods showed regular performance too. The use of the last few components returned by KPCA does not improves accuracy on detection for the tested methods. Future directions of this work include the use of the kernel-based algorithm with an outlier accommodation algorithm, and also propose the algorithm as a preprocessing step for machine learning problems.

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

References

1. A. Arning, R. Agrawal, and P. Raghavan. A linear method for deviation detection in large databases. In *KDDM*, pages 164–169, 1996.
2. V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley and Sons, 1978.
3. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
4. Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *COLT*, pages 144–152, 1992.
5. H. Brighton and C. Mellish. Advances in instace selection for instance-based learning algorithms. In *Proc. of the 6th ICKDDM*.
6. Carla E. Brodley and Mark A. Friedl. Identifying and eliminating mislabeled training instances. In *AAAI/IAAI, Vol. 1*, pages 799–805, 1996.
7. D. Clark. Using consensus ensembles to identify suspect data. In *KES*, pages 483–490, 2004.

8. 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.
9. F. R. Hampel. Robust estimation: A condensed partial survey.
10. R. Herbrich. *Learning Kernel Classifiers*. MIT press, first edition, 2002.
11. N. Matic I. Guyon and V. Vapnik. Discovering informative patterns ans data cleaning. In *Advances in Knowledge Discovery and Data Mining*, page 181.
12. George H. John. Robust decision trees: Removing outliers from databases. In *Proc. of the 1st ICKDDM*, pages 174–179, 1995.
13. I. T. Jolliffe. *Principal Component Analysis*. Springer Verlag-New York, 2nd edition, 2002.
14. Andrian Marcus Jonathan I. Maletic. Data cleansing: Beyond integrity analysis. In *Proc. of The Conf. on Information Quality (IQ2000)*, pages 200–209, 2000.
15. Frank Klawonn. Noise clustering with a fixed fraction of noise. In *Advances in Soft Computing: Applications and Science in Soft Computing*, pages 133–138. Springer, 2003.
16. Edwin M. Knorr and Raymond T. Ng. A unified notion of outliers: propieties and computation, 1997.
17. Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proc. 24th VLDB*, pages 392–403, 24–27 1998.
18. Edwin M. Knorr, Raymond T. Ng, and Vladimir Tucakov. Distance-based outliers: Algorithms and applications. *VLDB Journal: Very Large Data Bases*, 8(3–4):237–253, 2000.
19. J. Kubica and A. Moore. Probabilistic noise identification and data cleaning, 2002.
20. Aleksandar Lazarevie, Jaidep Srivastava, and Vipin Kumar. Data mining for analysis of rare events. Slides of PAKDD tutorial, 2004.
21. S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de–noising in feature spaces. In *NIPS 11*, 1999.
22. K.-R. Müller, S. Mika, G. Rätsch, , and K. Tsuda. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.
23. R. T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In *20th ICVLDB*, pages 144–155, 1994.
24. S. olkopf, B. Smola, and R. Williamson. Shrinking the tube: A new support vector regression algorithm, 1999.
25. Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. pages 427–438, 2000.
26. B. Schölkopf, S. Mika, A. Smola, G. Rätsch, and K.-R. Müller. Kernel PCA pattern reconstruction *via* approximate pre-images. In *Proc. of the 8th ICAN*, pages 147–152, 1998.
27. B. Schölkopf, o Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution, 1999.
28. B. Schölkopf, o Smola, R. Williamson, and P. Bartlett. New support vector algorithms, 2000.
29. B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. In *Neural Computation 10*, pages 1299–1319, 1998.
30. Bernhard Schölkopf and Alex Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
31. S. Schwarm and S. Wolfman. Cleaning data with bayesian methods, 2000.
32. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
33. David B. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *ICML*, pages 293–301, 1994.
34. 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.
35. Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.
36. Sofie Verbaeten and Anneleen Van Assche. Ensemble methods for noise elimination in classification problems. In *Multiple Classifier Systems*. Springer, 2003.