Detecting outliers using transduction and statistical significance testing*

Daniel Barbará
George Mason University
ISE Department, MSN 4A4
Fairfax, VA 22030
dbarbara@gmu.edu

Carlotta Domeniconi
George Mason University
ISE Department, MSN 4A4
Fairfax, VA 22030
carlotta@ise.gmu.edu

James P. Rogers
U.S. Army Engineer Research and Development Center
TEC
7701 Telegraph Road
Alexandria, VA 22315
James.P.Rogers.II@erdc.usace.army.mil

ABSTRACT

Finding points that are outliers with respect to a set of other points is an important task in data mining. Outlier detection can uncover important anomalies in fields like intrusion detection and fraud analysis. In data streaming, the presence of a large number of outliers indicates that the underlying process that is generating the data is undergoing significant changes and the models that attempt to characterize it need to be updated. Although there has been a significant amount of work in outlier detection, most of the algorithms in the literature resort to a particular definition of what an outlier is (e.g., density-based), and use thresholds to detect them. In this paper we present a novel technique to detect outliers that does not impose any particular definition for them. The test we propose aims to diagnose whether a given point is an outlier with respect to an existing clustering model (i.e., a set of points partitioned in groups). However, the test can also be successfully utilize to recognize outliers when the clustering information is not available. This test is based on Transductive Confidence Machines, which have been previously proposed as a mechanism to provide individual confidence measures on classification decisions. The test uses hypothesis testing to prove or disprove whether a point is fit to be in each of the clusters of the model. We demonstrate, experimentally, that the test is highly robust, and produces very few misdiagnosed points, even when no clustering information is available. We also show that the test can be succesfully applied to identify outliers present inside a data set for which no other information is available, thereby provinding the user with a clean data set to identify future outliers. Our experiments also show that even if the data set used to identify further outliers is contaminated with some outliers, the test can perform succesfully.

*This work has been sponsored by NSF grant IIS-0208519

1. INTRODUCTION

Outlier detection is an important data mining task that deals with the discovery of points that are exceptional when compared with a set of observations that are considered "normal." Applications of outlier detection abound in fields such as credit fraud, criminal investigation, spatio-temporal analysis, and computer intrusions. Outlier detection can reveal points that behave "anomalously" with respect to other observations. Examining such points can reveal clues to solve the problem at hand. In other cases, the sudden appearance of a large number of outliers can point to a change in the underlying process that is generating the data. Hawkins [13] defines an outlier as "an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism." This suggests the possibility of detecting outliers by knowing the "mechanism" by which the normal observations were generated and testing points for "membership" to this mechanism. Indeed, that is the path that early work in outlier detection followed (in the statistical community; see [17, 13] for a comprehensive review): postulate a model for the probability distribution of normal points (e.g., a Gaussian model), and compute the likelihood of a point being generated by the postulated model. Unfortunately, coming up with the right model is, at best, as difficult as the original problem of finding outliers, so this approach does not always work well in practice. This approach can be seen as *inducing* a model over the normal data and using it to test points.

Recently, the field of statistical learning theory [27] has developed alternatives to *induction*: instead of using all the available points to induce a model, one can use the data (usually a small subset of it) to estimate unknown properties of points that one wants to test (e.g., membership to a class). This powerful idea leads to elegant algorithms that use standard statistical tests to compute the confidence on the estimation. Using transduction, researchers have built Transductive Confidence Machines (see [8]) which are able to estimate the unknown class of a point and attach confidence to the estimate. The transductive reliability estimation process has its theoretical foundations in the algorithmic theory of randomness developed by Kolmogorov [18]. Unlike traditional methods in machine learning, transduction can offer measures of reliability to individual examples, and uses very broad assumptions (it only assumes that the data points are

independent and generated by the same stochastic mecha-
nism). Thh).rt -1.17.4().)--3-8e1(T)03cra- ffi t -1.17m).a- 0.0025.4().c)31(ha-)243.7(ec)24.9(6448.1(a)1.34-)]TJ 0 - ast17.17toast12eto(s)6.

Let there be m training examples
For $i = 1$ to m do
    Calculate D_i^y and D_i^{-y}
    Calculate α value for each example
Let there be c classes and u be the new example
(to be classified)
For $j = 1$ to c do
    For every training example t, classified as j do
        If $D_{tk}^j > dist(t, u)$
        (if the largest distance to one of the k-neighbors
        in class j for t is bigger than
        the distance of t to the new example u)
            Recalculate the α value for t (with u in j)
    For every training example t, classified as non-j do
        If $D_{tk}^{-j} > dist(t, u)$
        (if the largest distance to one of the k-neighbors
            outside class j for t is bigger than
        the distance of t to the new example u)
            Recalculate the α value for t (with u in j)
    Calculate α value for u
    Compute the p-value of the point for class j
Predict the class with the largest p-value for u
Output as confidence one minus the 2nd p-value

Figure 1: The TCM-kNN algorithm

or more extreme (with respect to the direction indicated by alternative hypothesis) than the observed outcome. So the smaller the p-value is, the smaller is the chance that the test statistic could have assumed a value as incompatible with the null hypothesis if the null hypothesis is true. In our case, the null hypothesis is "class y is a good fit for point i." With these two definitions, the algorithm for TCM-kNN can be described as shown in Figure 1.

In short, the algorithm TCM-kNN (Figure 1) acts as follows. It attempts to place a new point in each class of the problem. While doing that, it may force the updating of some of the α values for the training examples (concretely, this happens whenever the distance between the training example and the new point is less than the largest of the k distances that are used to compute the α). It then computes one p-value for each of the attempts (i.e., for each class placement). It then predicts that the point belongs to the class with the largest p-value, with a confidence equal to the complement of the second p-value.

3. OUR METHOD

Now we adapt the ideas of TCM-KNN for our purposes of determining if a point is an outlier with respect to a clustering model. We can use the ideas of TCM, by computing the strangeness of any point with respect to a cluster y - instead of a class y- by simply considering points in y equivalently to the training examples of a class y. Equally, Definition 2 will allow us to compute the p-value of the decision of placing the new point z_n in cluster y.

However, we need to realize a fundamental difference between our problem and that solved by TCM. In TCM we are always sure that the point we are examining *belongs* to one of the classes. In our problem, we are trying to determine if the point in question is an outlier, and hence does not belong to any of the clusters that we have. This has consequences in the outcome of the calculation of α, if we follow the Equation we presented in Section 2. The α computed for an outlier (a point that does not belong to any of the clusters) will be the ratio between two large numbers (the distances from the point in question to those in any of the clusters are large). In some cases, this ratio will be small enough to be comparable to the α values for points already in the cluster, leading to false negatives. (This is indeed the case in practice, as our experience has shown.) Instead, we propose to use a modified definition of α, as follows:

Definition 3. Strangeness with respect to a cluster: The strangeness α_i of a point i with respect to a cluster y is defined as:

$$\alpha_{iy} = \sum_{j=1}^{K} D_{ij}^y$$

This new definition of strangeness will make the strangeness value of a point far away from the cluster considerably larger than the one for points already inside the cluster.

Using the α values, we can compute a series of p-values for the new point, one for each cluster $y = 1, \cdots, c$ (where c

Given a point u under consideration:
    1. Compute the p-value of i with respect
    to clusters $1, \cdots, c$.
    2. Sort the p-value list in descending order.
    3. Call p_{max} the highest p-value, and
    p_{next} the next in the list.
    4. If $p_{max} \leq \tau$, reject all
    the null hypotheses H_0^y, for $y = 1, \cdots, c$,
    and therefore declare i an outlier
    with confidence $1 - \tau$.
    5. else, reject all the alternative hypotheses,
    (the point belongs to a cluster in the model)

Figure 2: Algorithm to compute the fitness of a point i with respect to the existing clusters

their **entire set** of null hypothesis rejected are considered outliers.

The operation of changing the α values for some of the examples in the clusters (whenever the new point is closer to the example than at least one of the example's k nearest neighbors), is potentially a costly one. Instead of just keeping the α values for all the points in the clusters, it requires maintaining the information of the k-nearest neighbors for each point, so when a new point is under consideration, these lists can be examined and the distances to the k-nearest neighbors compared to the distance to the new point. However, since we are only interested in diagnosing whether the new point is an outlier or not, we can do away with all this information and drop that operation all together, working instead with the original α values of the clustered points. In doing so, we only risk working with some value of α that is larger than it really ought to be. To understand this, consider a point u to be tested. If u is close to at least one point i in cluster y, in such a way that there exist a point j in y, among the k-nearest neighbors of i, such that $d(i, j) > d(i, u)$. In that case, u will replace j among the nearest neighbors of i when tested in cluster y, making the α_i smaller than the value previously calculated (without the inclusion of u in the cluster). That would then make i less strange than previously calculated. So, if we do not modify this α value, the algorithm would tend to consider i more strange than it really is, increasing the p-value for u. In other words, we risk lowering the chance of declaring that new point an outlier. However, this only happens if the new point is close enough to examples in the cluster (indeed, closer than some of the neighbors of the example), and by definition, that point should not qualify as an outlier (there are points already in the cluster that are farther away than this one). So, we can sacrifice precision in the calculation of the p-values to gain speed in the diagnose of the outliers. Experimental results have shown that this change has no effect in the capacity of the algorithm of detecting true outliers. It is important to remark that this argument only holds if **every new point is considered in isolation**. If the new point is incorporated in any of the clusters, the α values of the other points already there may need to be recomputed. We are only interested here in processing every point in a new batch **separately**, and decide, individually whether each point is an outlier under the current clustering model or not. Once this decision is made for the whole batch of points, one needs to undergo a process by which the new batch of points is incorporated to the cluster-

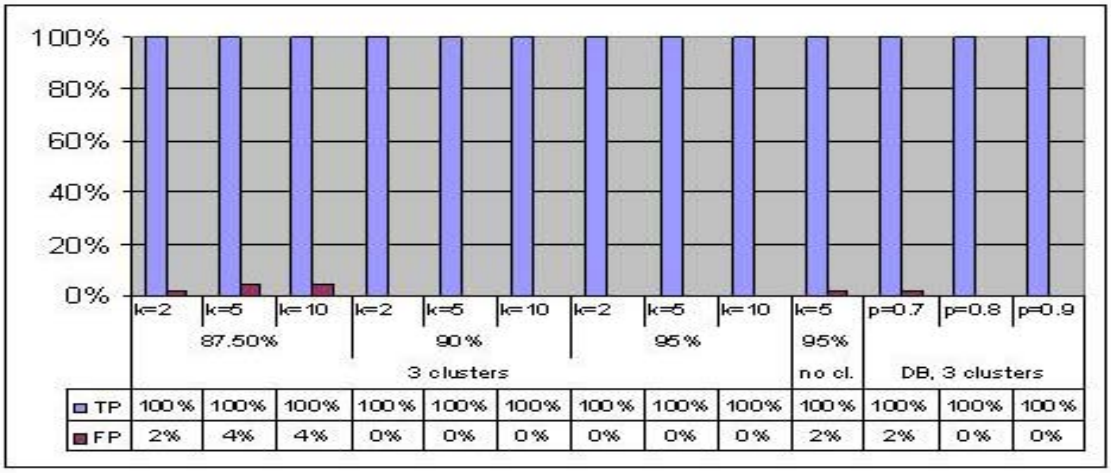


Figure 3: Results of the synthetic data. Each bar indicates the percentage of True Positives or False Positives detected by the corresponding test for the given number of nearest neighbors used (K). The two bars with the label "no cl." indicate the True Positives and False Positives when no clustering information is used ($K = 5$).

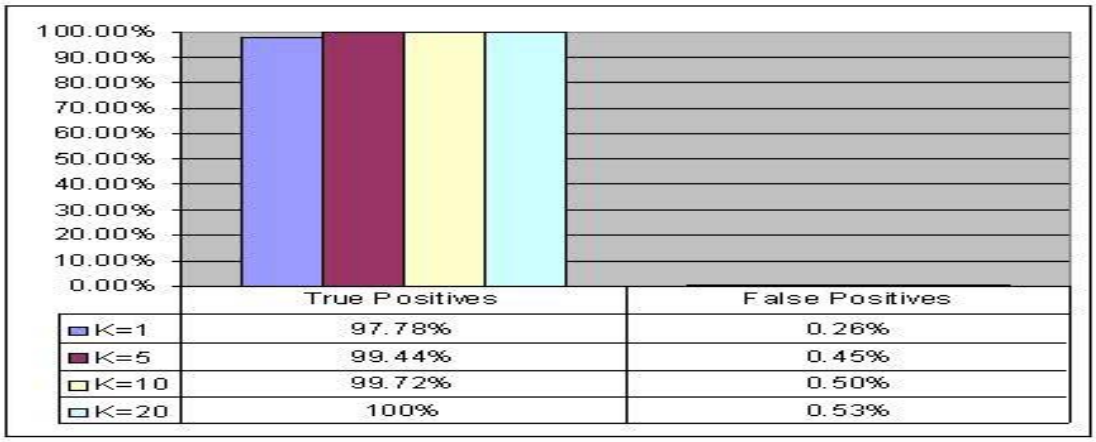


Figure 4: Results of using our technique on the bookstore data. Each bar indicates the percentage of True Positives or False Positives detected by the corresponding test for the given number of nearest neighbors used (k).

type of Iris plant. There are 50 records of each class in the set. We chose this set, as one of the classes (Iris Setosa) is highly separable from the other two. Taking away the class attribute, we form two clusters with 45 records of each the other two classes (Iris Virginica and Iris Versicolor) and use the records of the Setosa class, plus the remaining records of the other two classes (5 each) as tests for our outlier detection algorithm. Since the Setosa records are sufficiently separated from records of the other two classes, a good outlier detection technique should be able to recognize all of them as outliers. Meanwhile, the records of the other two classes should not be flagged as outliers. The Setosa records were tested against those two clusters, using our algorithm, resulting in all (100 %) of the records declared as outliers. The other 10 records were not flagged as outliers. We have computed for each Setosa point the distance to the closest centroid (of the two clusters representing the Versicolor and Virginica classes), and found out that these distances are in a small range: 2.86 to 3.79, with a mean of 3.24, and a small standard deviation: 0.20. The range for the "normal" data (Versicolor and Virginica points) is 0.23 to 1.80, with a mean of 0.73, and a standard deviation of 0.37. To maintain a confidence of 95% we have set the $\tau = 0.025$, since $0.975^2 \approx 0.95$.

These results hold even when we do not use the clustering information (i.e., we put all the 90 records of Virginica and Versicolor in one cluster). Without the clustering information, we used $\tau = 0.05$. Throughout the experiment (for both, clustering and non-clustering cases, we used $k = 5$.

5.3 The on-line bookstore data

This data set was generated from an e-commerce workload and has been used previously in [19]. The logs correspond to a couple of weekdays in which a large number of HTTP requests were processed. Entries corresponding to images, errors, etc were deleted and the URLs of the remaining entries in the log were mapped to one of 12 e-business functions such as "add item to cart," and "pay." A session vector was generated for each session. This vector indicates the number of times that each of the 12 different functions (e.g., Search, Browse, Add) was invoked during the session. So, this data set is 12 dimensional. The "ground truth" in this experiment consists of the domain knowledge that robot sessions are those with a total number of "clicks" bigger than or equal to 50. Those records were separated from the data set, along with some non-outliers used to test our algorithm. The basic clustering model was found using K-means over a set of records that do not contain any robot sessions, using $c = 3$. The members of each cluster differ on the intensity of the sessions: the first cluster contains records with low activity (few clicks), the second cluster, records with moderate activity, and the third one with high activity. The last two clusters contain records of sessions in which the "Add" function was performed considerably more often than in the sessions represented by the first cluster records. (Cluster 3 sessions correspond to "heavy buyers" in the bookstore.)

We employed 101,808 records to form the original cluster. Then we tested the technique with 65,536 records, of which 360 were known to be outliers (robot sessions) and the rest (65,176) non-outliers. Figure 4 shows the results of running our technique on this data set. We used $\tau = 0.017$ to obtain a total confidence level of 0.95 ($0.983^3 \approx 0.95$). A large percentage of the outliers are detected by the test, while the false positive rate is kept low. The results throughout the range of K are stable. Figure 8 shows the histogram of distances to the closest centroid for the non-outliers and outliers in the test data. The distributions are very different, with the bulk of the outliers concentrating at distances bigger than 7, while most of the non-outliers have distances smaller than 7.

We also conducted experiments varying the clustering model: using K-Means and different seeds (we tried 3 different seeds), we were able to obtain different clusterings for the baseline data. We compared the diagnoses of our method for each clustering, and found out that on the average 98% of the test points received the same diagnoses in the presence of two different clusterings of the same baseline data. This proves that the method is very robust with respect to changes in the clustering model.

In order to compare our technique with DB for this data set, we performed Principal Component Analysis and represented the data using the first four principal components (we do this, since the code we have for DB only handles 4 dimensions). Figure 5 shows the results for our technique and DB. The results are comparable to those shown in Figure 4 (the fraction of true positives is less than 1 % smaller than the one found in the case of the full data set; the fraction of false positives is around 1 % larger than the one found in the case of the full data set). For DB, we found that even decreasing p it was only possible to find a fraction of the true outliers (less than 7%). The technique however, does not introduce any false positives for this data set.

Figure 6 shows the results of experiments that do not use the clustering information. In the first one (labelled 'original' in the figure), we conducted the test on the original bookstore data without the cluster information. The True Positive rate is almost as good as the one obtained using the clusters (99.72%), while the False Positive rate is bigger but still very manageable (4.97%). The second test uses the 4-dimensional data obtained by PCA. The results are comparable to the original data (TP=100%, FP=4.90%). The last two tests were conducted by using a sample of the original data (as the normal set). The sample sizes were 1% and 10% respectively. The results show perfect recognition of the outliers in both cases (TP=100%), with a very slight increase of the False Positive rate (5.37% and 5.80%, for the 1% and 10% sample respectively). These experiments demonstrate that when the data set is large, it is possible to use a sample of the normal data to capture outliers without significant loss of accuracy.

5.4 Texture data

This is one of the real data sets of the Elena project, which can be found in [5]. The data set was generated by the Laboratory of Image Processing and Pattern Recognition (INPG-LTIRF) in Grenoble, France, using as the original source the material in [4], and referenced in [9, 10]. The data set contains a large number of classes (11) and a high dimensionality (40). The original aim was to distinguish between 11 different textures (Grass lawn, Pressed calf leather, Handmade paper, Raffia looped to a high pile, Cotton canvas, etc.), each pattern (pixel) being characterized by 40 attributes built by the estimation of fourth order modified moments in four orientations: 0, 45, 90 and 135 degrees. Again, we discard the class attribute and look to detect the points in one class as outliers with respect to the

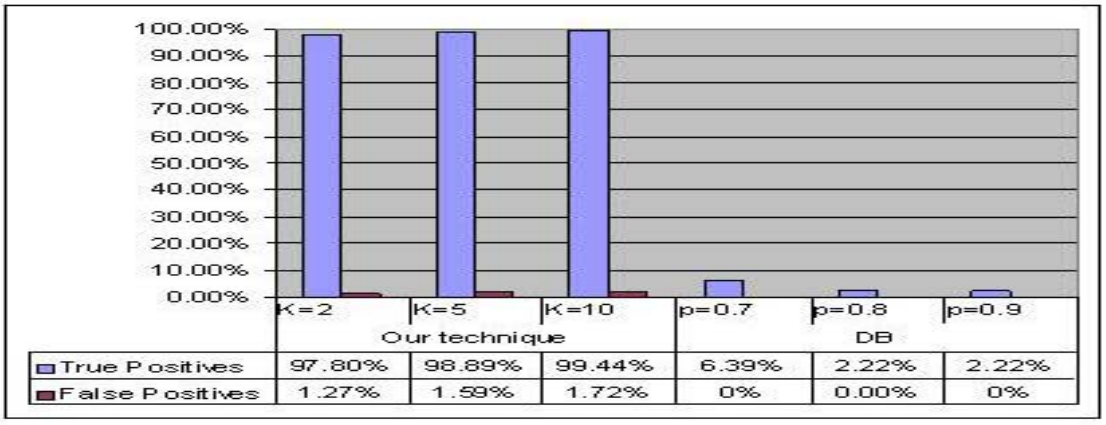


Figure 5: Results of the bookstore data subjected to PCA, where the four principal components are used. Each bar indicates the percentage of True Positives or False Positives detected by the corresponding test for the given number of nearest neighbors used (k).

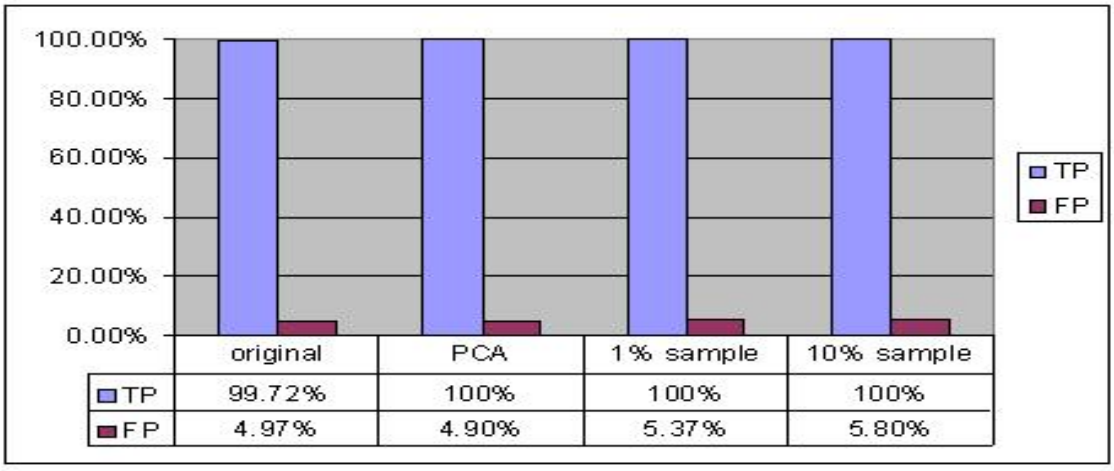


Figure 6: Results of using our technique on the bookstore data without clustering information. The first test is done over the original data; the second over the data subjected to PCA, where the four principal components are used; the last two tests were conducted using samples of the original data (1% and 10% respectively). Each bar indicates the percentage of True Positives or False Positives detected by the corresponding test for the corresponding value of p used.

points in other classes. In concrete, this experiment used all the records of the texture data as the initial points, with the exception of: a) those that belong to one of the classes (500), b) A sample of five percent (223 in total) of records of other classes (which were then included in the test set). Each of the other classes was represented by a cluster, 10 clusters , 4,250 records in total. With $K = 5$ and $\tau = 0.005$ (to reach a final confidence of 95%, as $0.995^{10} \approx 0.95$), we obtain a true positive rate of 98.4%, with no false positives. When the clustering information is ignored, the true positive rate goes to 100% with 4.93% false positives. (See Figure 7.)

Figure 9 shows the histograms of distances to the closest centroid for the points that are declared as outliers and those that are not in the test set. It is easy to see that the two distributions are radically different with little overlap on the distances.

In order to compare our technique with DB for this data set, we performed Principal Component Analysis and represented the data using the first four principal components. Figure 10 shows the results for our technique and for DB. Although the percentage of true positives (outliers) captured is smaller than that obtained when using the entire data set, the technique still captures a large fraction of the outliers, with no false positives to report. For DB, since the 500 points of class 13 are somewhat close to each other, it is necessary to lower the value of p to capture them as outliers. But by doing so, false positives are introduced.

5.5 National Hockey League data

We performed a test using NHL player's statistics (they can be obtained from Web sites such as nhlstatistics.hypermart.net) for the 1994 season. The reason for this test is that we wanted to compare the behavior of our technique with DB and the NHL 1994 data set is one of the case studies they investigated. We use a four-dimensional description for each player with the following attributes: the plus-minus statistic (indicates how many more event-strength goals were scored by the team when the player was in the ice), the number of penalty minutes, the percentage points, and goals scored. After tuning the parameters for the distance-based outliers code, we obtained seven outliers for the distance-based code. We took these seven records and added 27 other players (selected at random) to perform a test with our technique. The rest of the players in the data set (834) were grouped into 3 clusters (using K-means). We used $\tau = 0.025$ for a total confidence of 95 %. Our technique found the seven outliers and added 3 more to the list. (For a total of 10 outliers out of the 34 records tested.) Figure 11 shows the histograms of the distance of the points to the closest. centroid. The figure shows that the outliers found by our technique are indeed radically different than the points declared non-outliers, justifying the finding of the extra 3 outliers.

5.6 Cleaning data

In this section we show the results of experiments aimed to clean a data set containing outliers, without relying on the presence of an outlier-free sample. To do so, we used the texture data, and "contaminated" the normal data with a number of outliers (from the class chosen to act as outliers). Then, use this data both as the "normal" and test sets. The only difference is that, in order to be transductive, whenever we test a data point we compute the α values of the training set **without including that point**. The results , shown in Figure 12, indicate that we can effectively determine almost all the outliers present in the data set, especially when we use a high value of K. As expected intuitively, the more outliers we have in the set, the larger K needs to be. Of course, we do not know how many outliers we have in the data set in advance, so a "rule of thumb" has to be used to select K. (In nearest-neighbor classification, it is customary to set K as high as 10% of the number of examples in the data set.) The experiments show that a relatively low value of K (100, or around 2.3%) is enough to catch a high volume of the outliers (more than 95%) while maintaining a low false positive rate (less than 4%).

5.7 Using a contaminated data set

In this section we show what happens if the "normal" data is contaminated with outliers and yet it is used to further detect other outliers. We conducted these experiments using the texture data and contaminating the "normal" data with outliers from the chosen class. We then used this data set as the basis for diagnosing the test set, without the outliers that were already used to contaminate the "normal" data. That is the test set contains the rest of the examples from the chosen class (not in the "normal" set) plus 223 examples of non-outliers.

Figure 13 shows the results of these experiments. The number of outliers used to contaminate the normal set is shown below the row with the values of K. Again, we diagnose a high percentage of the outliers in the test data (more than 95%) with a low false positive rate (less than 4%) when a value of K, commensurable with the number of contaminants is used. As we do not know this value, a "rule of thumb" value for K must be used, but the experiments show that a relative low value of K (100, or 2.3% of the normal data set examples) can effectively be used to diagnose further outliers.

6. RELATED WORK

Early work in outlier detection was done in the field of statistics (see [17, 13]). However, these methods largely work with univariate data, and all of them assume knowledge of the underlying distribution of the data, which in practice is very restrictive. More recently a technique for spatial outlier detection was proposed in [24]. The method uses the difference between the attribute value at location x and the average attribute value of x's neighbors to determine if x is an outlier. It works only for univariate data (one attribute besides the spatial coordinates) , and assumes a normal distribution for the non-spatial attribute value.

An exception to the univariate restriction in the statistics techniques is the work of Rousseeuw et al (whose most recent survey can be found in [22]). Their approach is to find a robust fit model, i.e., one that is similar to the fit that would have been found without the outliers, and use it to diagnose which points do not fit the model well. Their choice is to compute the *Minimum Covariance Determinant* estimator or **MCD**, which can be briefly described as finding a subset of the examples in the data whose covariance matrix has the lowest possible determinant. Their algorithm, *FAST-MCD* uses randomization to speed up the high-complexity calculation. Outliers are detected by computing the Mahalanobis distance (using the covariance matrix computed by MCD) and finding points that are "too far away" from the robust centroid. One can argue that estimating a mean

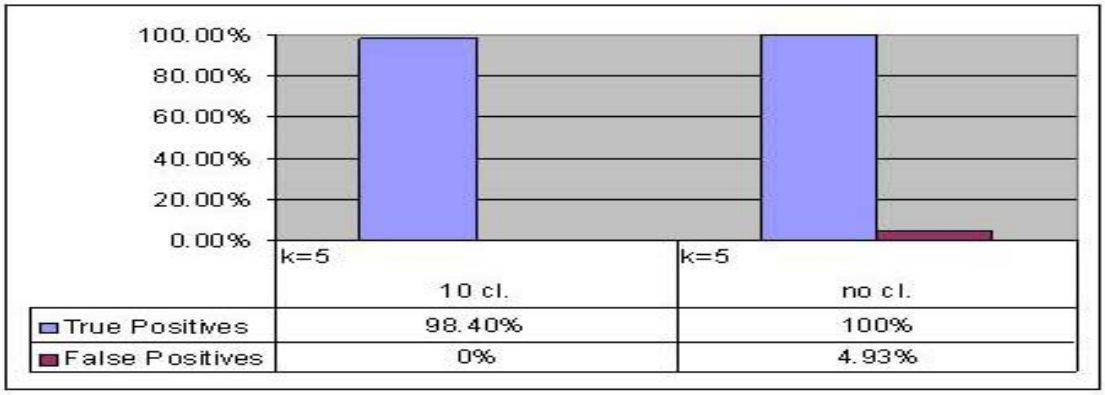


Figure 7: Results of our technique on the texture data with and without clusters.

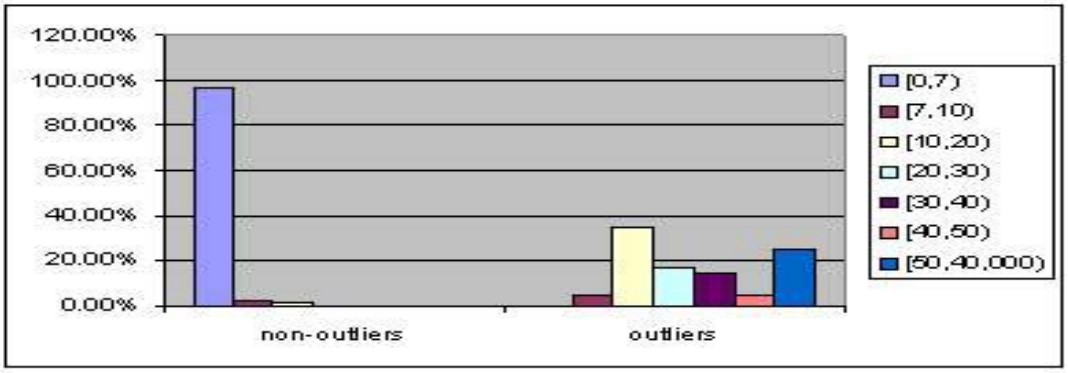


Figure 8: Distance histogram for the bookstore data. Each bar shows the percentage of points within each group (non-outliers, outliers) with distances to the closest centroid in the range indicated.

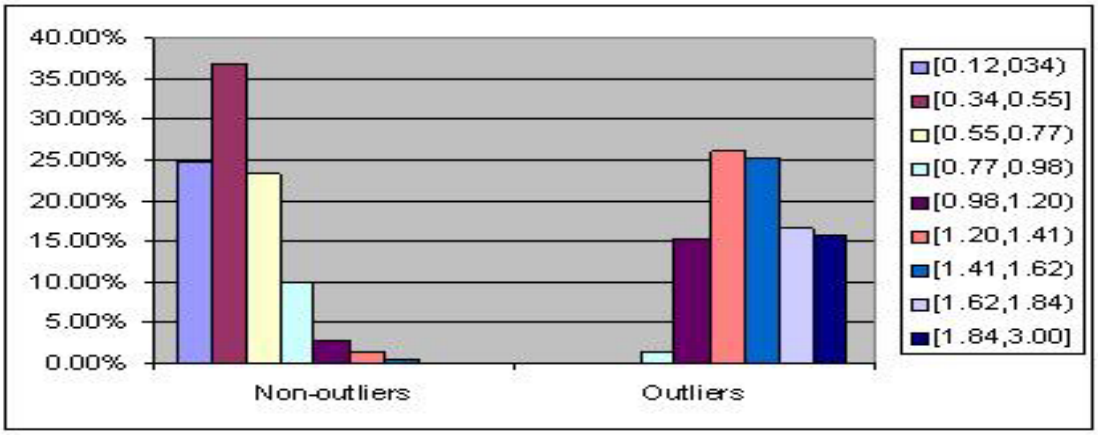


Figure 9: Histogram of distances to the closest centroid for the texture data. Each bar shows the percentage of points in the group that have distances to the closest centroid in the range indicated.

and covariance matrix from the data (or rather in MCD, a sub-sample of the data) corresponds to assuming a Gaussian distribution of the data. Also, looking for points with large Mahalanobis distance that is "too large" calls for a threshold, i.e., it is equivalent to consider as outliers those points that are 3 or more standard deviations away from the centroid, which is the common practice of univariate statistical techniques. Moreover, in [12], which uses a technique based on MCD to find multiple clusters and diagnose outliers, the authors argue that not every data set will give rise to an obvious separation of outliers and non-outliers by using robust estimators.

A large body of work has been published in the area of discovering outliers with respect to clustering models (see [6, 20, 23, 25, 11]). However, most of these algorithms do not aim at the discovery of outliers, but rather offer ways to deal with them. And, in all the cases, the discovery of outliers is done by careful setting of the algorithm parameters. Some outlier detection schemes that do not assume a clustering model or a known distribution have been proposed. They fall under two categories. The first is *distance-based* techniques (see [1, 16]. The second is *density-based* techniques (see [3, 15]). Again, in all these algorithms one must threshold parameters to obtain the set of outliers.

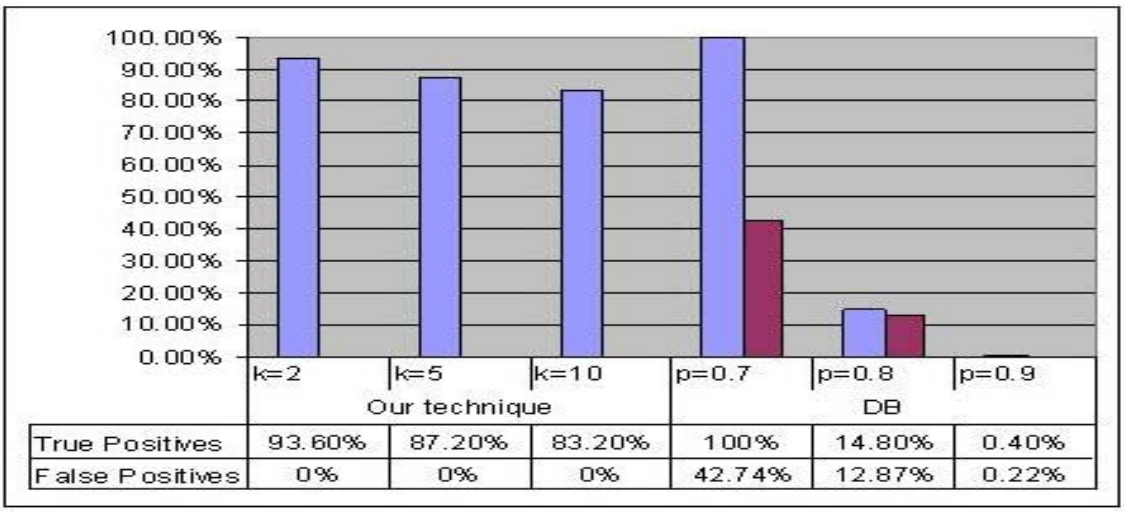
The method of [16], which we use here to provide a comparison to ours, uses distance and density calculations to discover the outliers in a data set. Succinctly, the method aims to answer a nearest neighbor query with radius D for each point in the data set and decide if there are enough neighbors to the point in this D-neighborhood. This decision is controlled by a parameter p, which thresholds the minimum fraction of records that must be found outside of the D-neighborhood for the point to be an outlier. These leaves two parameters to be adjusted (D and p). However, the code of DB overwrites a badly chosen D by computing a reasonable one using sampling. So we concentrated in varying the choice of p in our experiments.

7. CONCLUSIONS

We have presented here a novel technique to detect outliers based on statistical testing and the application of transduction. The technique does not make assumptions about the data distribution and only requires that the number of neighbors (K) utilized in the distance calculation and the confidence level be provided. We have shown using extensive experimentation that the technique is robust with respect to the choice of K and that it obtains extremely good results for the standard choice of 95 % confidence level. We observe that the definition of strangeness does depend in turn on the distance metric used. In this work we have fixed the metric to be the Euclidian distance to allow a fair comparison with the DB technique. Additional distance measures will be investigated in our future work.

We have shown that discarding the clustering information has little effects on the results (generally a small increase on false positives), so the method can be used also in cases when the cluster information is not available. (And therefore we can claim that the method is robust even if the clusters are not.) We have compared the technique with the distance-based algorithm (DB) presented in [16], and the results show that our technique sometimes outperforms DB and is never outperformed by it.

We have also shown that it is possible to use our method to effectively clean a data set from outliers, thereby providing a sample of data that is, at least in a large percentage, free



	k=2	k=5	k=10	p=0.7	p=0.8	p=0.9
True Positives	93.60%	87.20%	83.20%	100%	14.80%	0.40%
False Positives	0%	0%	0%	42.74%	12.87%	0.22%

Figure 10: Results for the texture data subjected to PCA, where the four principal components are used. Each bar indicates the percentage of True Positives or False Positives detected by the corresponding test for the given number of nearest neighbors used (k).

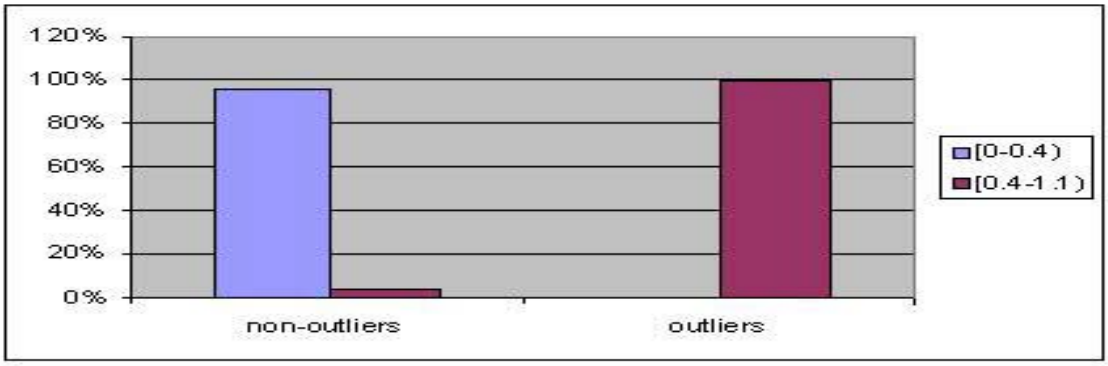


Figure 11: Histogram of distances to the closest centroid for the NHL data. Each bar shows the percentage of points in the group that have distances to the closest centroid in the range indicated.

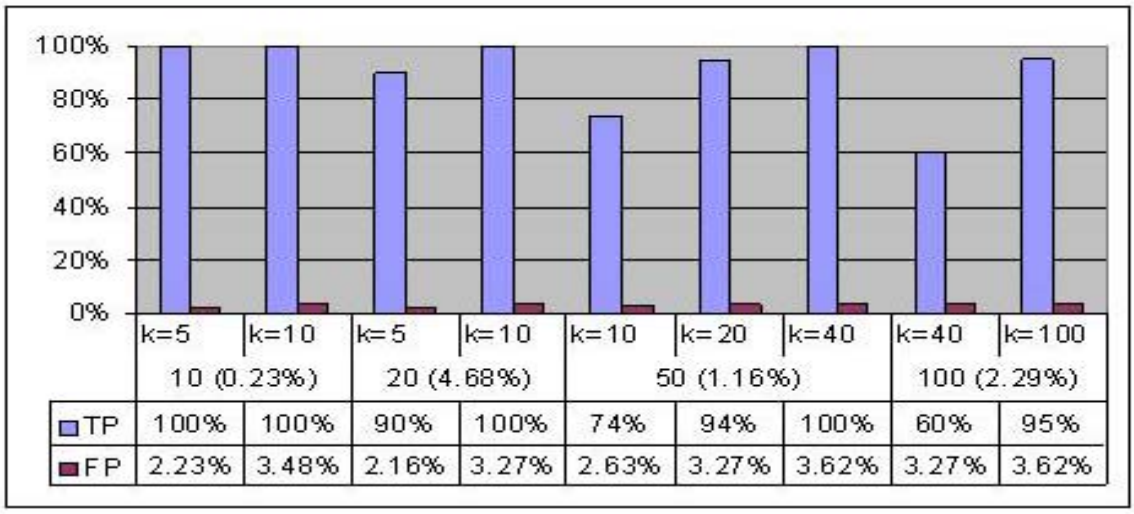


Figure 12: Results of cleaning the texture data contaminated with a number of outliers of the chosen class. The number of outliers in the data set, and its corresponding percentage of the entire set are shown below the row that indicates the value of K used. Each bar indicates the percentage of True Positives or False Positives detected by the corresponding test for the given number of nearest neighbors used (k).

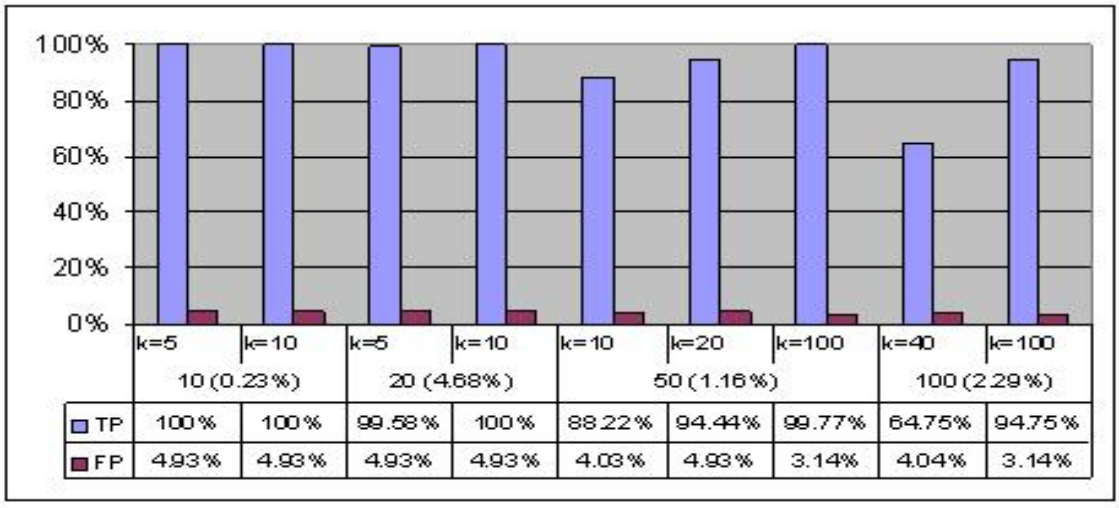


Figure 13: Results of diagnosing the texture data by using a contaminated "normal" set. The number of outliers in the "normal"data set, and its corresponding percentage of the entire set are shown below the row that indicates the value of K used. Each bar indicates the percentage of True Positives or False Positives detected by the corresponding test for the given number of nearest neighbors used (k).

of outliers. Moreover, we have demonstrated that even if the "normal" data set is contaminated by outliers, it can be used to diagnose further outliers.

We have shown that, for large data sets, is possible to drastically improve the performance of the algorithm by using sampling, while maintaining good results. We want to experiment in future work with the idea of representing the clusters of normal data (or the whole data) by means of carefully selected *representatives* (instead of uniformly sampling the data). We expect that this technique will result in a decrease in false positives with respect to the rates obtained by sampling.

8. REFERENCES

[1] Bay, S.D., and Schwabacher, M. ((2003) Mining distance-based outliers in near linear time with randomization and a simple pruning rule. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining,* 29-38.

[2] Baseball Reference. http://www.baseball-reference.com/

[3] Breunig, M., Kriegel, H., Ng, R., Sander, J. (2000) LOF: Identifying Density-Based Local Outliers. *Proc. of the ACM SIGMOD Conference on Management of Data,* 427-438.

[4] Brodatz, P. (1966) Textures: A Photographic Album for Artists and Designers, Dover Publications,Inc.,New York.

[5] Elena project data. ftp://ftp.dice.ucl.ac.be/pub/neural-nets/ELENA/databases/

[6] Ester, M., Kriegel, H., Sander, J., and Xu, X. (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. *Proc. of the 2nd Intl. Conference on Knowledge Discovery and Data Mining.* 226-231.

[7] Edgington, E. (1980) Randomization Tests. New York: Marcel Dekker.

[8] Gammerman, A., and Vovk, V. (2002) Prediction algorithms and confidence measures based on algorithmic randomness theory. Theoretical Computer Science. **287**, 209-217.

[9] Guerin-Dugue, A., and Aviles-Cruz, C. (1993) High Order Statistics from Natural Textured Images, *ATHOS workshop on System Identification and High Order Statistics*. Sophia-Antipolis, France, September 1993.

[10] Guerin-Dugue, A. et al., (1995) Deliverable R3-B4-P - Task B4: Benchmarks, Technical report, Elena-NervesII "Enhanced Learning for Evolutive Neural Architecture", ESPRIT-Basic Research Project Number 6891, June 1995

[11] Guha, S., Rastogi, R., and Shim, K. (1998) CURE: an efficient clustering algorithm for large databases. *Proc. of ACM SIGMOD Conf. on Management of Data,* 73-84.

[12] Hardin, J., and Rocke, D.M. (2004) Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. Computational statistics and data analysis, Vol 44, pp. 625-638, 2004.

[13] Hawkins, D. (1980) Identification of Outliers. Chapman and Hall, London.

[14] Neville, J., Jensen, D., Friedland, L., and Hay, M. (2003) Learning Relational Probability Trees, *Proc. of the 9th ACM-SIGKDD Intl. Conference on Knowledge Discovery and Data Mining,* Washington, DC, 625 - 630.

[15] Jin, W., Tung, A., and Han, J. Mining (2001) Top-n local outliers in large databases. *Proc. of the Intl. Conference on Knowledge Discovery and Data Mining,* 293-298.

[16] Knorr, E., Ng, R. (1998) Algorithms for Mining Distance-Based Outliers in Large Datasets. *Proceedings of the 24th Intl. Conference on Very Large Databases,* 392-403.

[17] Lewis, B.V. (1994) Outliers in Statistical Data. John Wiley.

[18] Li, M., and Vitanyi, P. (1997) Introduction to Kolmogorov Complexity and its Applications. 2nd Edition, Springer Verlag.

[19] Menasce, D., Abraho, B., Barbará, D., Almeida, V., Ribeiro, F. (2002) Fractal Characterization of Web Workloads. *Proceedings of the "Web Engineering" Track of WWW2002, Honolulu, Hawaii, USA*, 7-11

[20] Ng, R., and Han, J. (1994) Efficient and effective clustering methods for spatial data mining. *Proceedings of the 20th Intl. Conference on Very Large Data Bases*, 144-155.

[21] Proedru, K., Nouretdinov, I., Vovk, V., Gammerman, A. (2002) Transductive confidence machine for pattern recognition. *Proc. 13th European conference on Machine Learning.* **2430**, 381-390.

[22] Hubert, M., Rousseeuw, P.J, and Van Aelst, S. (2005) Multivariate Outlier Detection and Robustness. In *Handbook of Statistics*, Vol. 24, C.R. Rao, E. Wegman, J. Solka, editors. Elsevier, 2005.

[23] Sheikholesami, G., Chatterjee, S., and Zhang, A. (1998) WaveCluster: a multi-resolution clustering approach for very large spatial databases. *Proc. of the 24th Intl. Conference on Very Large Databases.* 428-439.

[24] Shekhar, S., Lu, C.T., and Zhang, P. (2003) A Unified Approach to Spatial Outliers Detection, *Geoinformatica*,**7(2)**, June, 2003.

[25] Tang, J., Chen, Z., Fu, A., and Cheung, D. (2002) Enhancing Effectiveness of Outlier Detection for Low Density Patterns. *Proc. of PAKDD'02*, 535-548.

[26] UCI Machine Learning Repository. http://www.ics.uci.edu/ mlearn/MLRepository.html

[27] Vapnik. V. (1998) Statistical Learning Theory, New York: Wiley.

[28] Vovk, V., Gammerman, A., and Saunders, C. (1999) Machine learning applications of algorithmic randomness. *Proceedings of the 16th Intl. Conference on Machine Learning.* 444-453.

[29] Ho, S.S., and Wechsler, H. (2003) Transductive Confidence Machine for Active Learning, *Int. Joint Conf. on Neural Networks*, Portland, OR.