Outlier Detection with Kernel Density Functions

Longin Jan Latecki[1], Aleksandar Lazarevic[2], and Dragoljub Pokrajac[3]

[1] CIS Dept. Temple University Philadelphia, PA 19122 , USA,
latecki@temple.edu
[2] United Technology Research Center 411 Silver Lane, MS 129-15 East Hartford, CT 06108, USA,
aleks@cs.umn.edu
[3] Dragoljub Pokrajac CIS Dept. CREOSA and AMRC, Delaware State University Dover DE 19901, USA,
dpokrajac@desu.edu

Abstract. Outlier detection has recently become an important problem in many industrial and financial applications. In this paper, a novel unsupervised algorithm for outlier detection with a solid statistical foundation is proposed. First we modify a nonparametric density estimate with a variable kernel to yield a robust local density estimation. Outliers are then detected by comparing the local density of each point to the local density of its neighbors. Our experiments performed on several simulated data sets have demonstrated that the proposed approach can outperform two widely used outlier detection algorithms (LOF and LOCI).

1 Introduction

Advances in data collection are producing data sets of massive size in commerce and a variety of scientific disciplines, thus creating extraordinary opportunities for monitoring, analyzing and predicting global economical, demographic, medical, political and other processes in the World. However, despite the enormous amount of data available, particular events of interests are still quite rare. These rare events, very often called outliers or anomalies, are defined as events that occur very infrequently (their frequency ranges from 5% to less than 0.01% depending on the application). Detection of outliers (rare events) has recently gained a lot of attention in many domains, ranging from video surveillance and intrusion detection to fraudulent transactions and direct marketing. For example, in video surveillance applications, video trajectories that represent suspicious and/or unlawful activities (e.g. identification of traffic violators on the road, detection of suspicious activities in the vicinity of objects) represent only a small portion of all video trajectories. Similarly, in the network intrusion detection domain, the number of cyber attacks on the network is typically a very small fraction of the total network traffic. Although outliers (rare events) are by definition infrequent, in each of these examples, their importance is quite high compared to other events, making their detection extremely important.

Data mining techniques that have been developed for this problem are based on both supervised and unsupervised learning. Supervised learning methods typically build a prediction model for rare events based on labeled data (the training set), and use it to classify each event [1, 2]. The major drawbacks of supervised data mining techniques include: (1) necessity to have labeled data, which can be extremely time consuming for real life applications, and (2) inability to detect new types of rare events. On the other hand, unsupervised learning methods typically do not require labeled data and detect outliers (rare events) as data points that are very different from the normal (majority) data based on some pre-specified measure [3]. These methods are typically called outlier/anomaly detection techniques, and their success depends on the choice of similarity measures, feature selection and weighting, etc. Outlier/anomaly detection algorithms have the advantage that they can detect new types of rare events as deviations from normal behavior, but on the other hand suffer from a possible high rate of false positives, primarily because previously unseen (yet normal) data are also recognized as outliers/anomalies, and hence flagged as interesting.

Outlier detection techniques can be categorized into four groups: (1) statistical approaches; (2) distance based approaches; (3) profiling methods; and (4) model-based approaches. In statistical techniques [3, 6, 7], the data points are typically modeled using a stochastic distribution, and points are labeled as outliers depending on their relationship with the distributional model.

Distance based approaches [8–10] detect outliers by computing distances among points. Several recently proposed distance based outlier detection algorithms are founded on (1) computing the full dimensional distances among points using all the available features [10] or only feature projections [8]; and (2) on computing the densities of local neighborhoods [9, 35]. Recently, LOF (Local Outlier Factor) [9] and LOCI (Local Correlation Integral) [35] algorithms have been successfully applied in many domains for outlier detection in a batch mode [4, 5, 35]. In addition, clustering-based techniques have also been used to detect outliers either as side products of the clustering algorithms (as points that do not belong to clusters) [11] or as clusters that are significantly smaller than others [12].

In profiling methods, profiles of normal behavior are built using different data mining techniques or heuristic-based approaches, and deviations from them are considered as outliers (e.g., network intrusions). Finally, model-based approaches usually first characterize the normal behavior using some predictive models (e.g. replicator neural networks [13] or unsupervised support vector machines [4, 12]), and then detect outliers as deviations from the learned model.

In this paper, we propose an outlier detection approach that can be classified both into statistical and density based approaches, since it is based on local density estimation using kernel functions. Our experiments performed on several simulated data sets have demonstrated that the proposed approach outperforms two very popular density-based outlier detection algorithms, LOF [9] and LOCI [35].

2 Local Density Estimate

We define outlier as an observation that deviates so much from other observations to arouse suspicion that it was generated by a different mechanism [13]. Given a data set $D = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, where n is the total number of data samples in Euclidean space of dimensionality dim, we propose the algorithm that can identify all outliers in the data set D. Our first step is to perform density estimate. Since we do not make any assumption about the type of the density, we use a nonparametric kernel estimate [39] to estimate the density of *majority* data points $q(\mathbf{x})$, also referred to as a ground truth density. Consequently, all data samples that appear not to be generated by the ground truth density $q(\mathbf{x})$ may be considered as potential outliers.

However, it is impossible to directly use density estimate to identify outliers if the estimated distribution is multimodal, which mostly is the case. Data points belonging to different model components may have different density without being outliers. Consequently, normal points in some model components may have lower density than outliers around points from different model components.

In order to detect outliers, we compare the estimated density at a given data points to the average density of its neighbors. This comparison forms the basis of most unsupervised outlier detection methods, in particular of LOF [9]. The key difference is that we compare densities, which have solid statistical foundation, while the other methods compare some local properties that are theoretically not well understood.

One of our main contributions is to provide proper evaluation function that makes outlier detection based on density estimate possible.

There is a large body of published literature on non-parametric density estimation [39]. One of the best-working non-parametric density estimation methods is the variable width kernel density estimator [39]. In this method, given n data samples of dimensionality dim, the distribution density can be estimated as:

$$\tilde{q}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h(\mathbf{x}_i)^{dim}} K(\frac{\mathbf{x}-\mathbf{x}_i}{h(\mathbf{x}_i)}), \tag{1}$$

where K is a kernel function (satisfying non-negativity and normalization conditions) and $h(\mathbf{x}_i)$ are the bandwidths implemented at data points $\mathbf{x}_i$. One of the main advantages of this sample smoothing estimator is that $\tilde{q}(\mathbf{x})$ is automatically a probability density function [39] if K is a probability density function. In our case, K is a multivariate Gaussian function of dimensionality dim with zero mean and unit standard deviation:

$$K(\mathbf{x}) = \frac{1}{(2\pi)^{dim}} \exp(-\frac{||\mathbf{x}||^2}{2}), \tag{2}$$

where $||\mathbf{x}||$ denotes the norm of the vector. The simplest version of the bandwidth function $h(\mathbf{x}_i)$ is a constant function $h(\mathbf{x}_i) = h$, where h is a fixed bandwidth. However, for real data sets, local sample density may vary. Therefore, it is necessary to have a method that is adaptive to the local sample density. This may

be achieved for $h(\mathbf{x}_i) = hd_k(\mathbf{x}_i)$, where $d_k(\cdot)$ denotes the distance to the kth nearest neighbor of point $\mathbf{x}_i$. The usage of the kth nearest neighbor in kernel density estimation was first proposed in [38] (see also [37]).

Since we are interested in detecting outlier data samples based on comparing them to their local neighborhood, the sum in Eq. 1 needs only to be taken over a sufficiently large neighborhood of a point x. Let $mNN(\mathbf{x})$ denotes the m nearest neighbors of a sample $\mathbf{x}$. Thus, from Eq. 1 and 2 we obtain the following formula for distribution density at data sample $\mathbf{x}_j$:

$$\tilde{q}(\mathbf{x}_j) \propto \frac{1}{m} \sum_{\mathbf{x}_i \in mNN(\mathbf{x}_j)} \frac{1}{h(\mathbf{x}_i)^{dim}} K(\frac{\mathbf{x}_j - \mathbf{x}_i}{h(\mathbf{x}_i)})$$

$$= \frac{1}{m} \sum_{\mathbf{x}_i \in mNN(\mathbf{x}_j)} \frac{1}{(2\pi)^{\frac{dim}{2}} h(\mathbf{x}_i)^{dim}} \exp(-\frac{d(\mathbf{x}_j, \mathbf{x}_i)^2}{2h(\mathbf{x}_i)^2}). \tag{3}$$

Here,

$$d(\mathbf{x}_j, \mathbf{x}_i) = ||\mathbf{x}_j - \mathbf{x}_i||^2 \tag{4}$$

is the squared Euclidean distance between samples $\mathbf{x}_i$ and $\mathbf{x}_j$. Restricting the sum in Eq. 1 to a local neighborhood as in Eq. 3 has a computational advantage. While the computation of $\tilde{q}$ for all data points has $O(n^2)$ complexity, the average computation in Eq. 3 can be accomplished in $O(mn \log n)$ time, where n is the number of data samples in a data set D and $O(m \log n)$ refers to the cost of search for m nearest neighbors of a data sample if a hierarchical indexing structure like R-tree is used [46].

Observe that Euclidean distance from Eq. 4 may be very small if there is a neighbor $\mathbf{x}_i$ very close to sample $\mathbf{x}_j$. In such a case, it is possible to misleadingly obtain a large density estimate $\tilde{q}(\mathbf{x}_j)$. To prevent such issues and increase the robustness of the density estimation, following the LOF approach [9], we compute reachability distance for each sample $\mathbf{y}$ with respect to data point $\mathbf{x}$ as follows:

$$\mathbf{rd}_k(\mathbf{y}, \mathbf{x}) = \max(d(\mathbf{y}, \mathbf{x}), d_k(\mathbf{x})), \tag{5}$$

where $d_k(x)$ is the distance to kth nearest neighbor of point x. Eq. 5 prevents the distance from $\mathbf{y}$ to $\mathbf{x}$ to become too small with respect to the neighborhood of point $\mathbf{x}$.

We obtain our local density estimate (LDE) by replacing the Euclidean distance in Eq. 3 with the reachability distance:

$$LDE(\mathbf{x}_j) \propto \frac{1}{m} \sum_{\mathbf{x}_i \in mNN(\mathbf{x}_j)} \frac{1}{(2\pi)^{\frac{dim}{2}} h(\mathbf{x}_i)^{dim}} \exp(-\frac{\mathbf{rd}_k(\mathbf{x}_j, \mathbf{x}_i)^2}{2h(\mathbf{x}_i)^2})$$

$$= \frac{1}{m} \sum_{\mathbf{x}_i \in mNN(\mathbf{x}_j)} \frac{1}{(2\pi)^{\frac{dim}{2}} (h \cdot d_k(\mathbf{x}_i))^{dim}} \exp(-\frac{\mathbf{rd}_k(\mathbf{x}_j, \mathbf{x}_i)^2}{2(h \cdot d_k(\mathbf{x}_i))^2}). \tag{6}$$

The name of local density estimate (LDE) is justified by the fact that we sum over a local neighborhood mNN compared to the sum over the whole data set

commonly used to compute the kernel density estimate (KDE), as shown in Eq. 1.

LDE is not only computationally more efficient than the density estimate in Eq. 1 but yields more robust density estimates. LDE is based on the ratio of two kinds of distances: the distance from a point $\mathbf{x}_j$ to its neighbors $\mathbf{x}_i$ and distances of the neighboring points $\mathbf{x}_i$ to their k-th neighbors. Namely, the exponent term in Eq. 6 is a function of the ratio $\frac{\mathbf{rd}_k(\mathbf{x}_j,\mathbf{x}_i)}{d_k(\mathbf{x}_i)}$, which specifies how is the reachability distance from $\mathbf{x}_j$ to $\mathbf{x}_i$ related to the distance to the k-th nearest neighbor of $\mathbf{x}_i$. In fact, we use $d_k(\mathbf{x}_i)$ as a "measuring unit" to measure the Euclidean distance $d(\mathbf{x}_j, \mathbf{x}_i)$. If $d(\mathbf{x}_j, \mathbf{x}_i) \leq d_k(\mathbf{x}_i)$, then the ratio $\frac{\mathbf{rd}_k(\mathbf{x}_j,\mathbf{x}_i)}{d_k(\mathbf{x}_i)}$ is equal to one (since $\mathbf{rd}_k(\mathbf{x}_j, \mathbf{x}_i) = d_k(\mathbf{x}_i)$), which yields the maximal value of the exponential function ($\exp(-\frac{1}{2h^2})$). Conversely, if $d(\mathbf{x}_j, \mathbf{x}_i) > d_k(\mathbf{x}_i)$, then the ratio is larger than one, which results in smaller values of the exponent part.

The bandwidth h specifies how much weight is given to $d_k(\mathbf{x}_i)$. The larger h, the more influential are the k nearest neighbors that are further away. The smaller h, the more we focus on k nearest neighbors.

Observe that we compare a given point $\mathbf{x}_j$ to its neighbors in $mNN(\mathbf{x}_j)$. It is important that the neighborhood $mNN(\mathbf{x}_j)$ is not too small (otherwise, the density estimation would not be correct). Overly large m does not influence the quality of the results, but it influences the computing time (to retrieve m nearest neighbors).

Having presented an improved local version of a nonparametric density estimate, we are ready to introduce our method to detect outliers based on this estimate. In order to be able to use LDE to detect outliers, the local density values $LDE(\mathbf{x}_j)$ need to be related to the LDE values of neighboring points. We define **Local Density Factor (LDF)** at a data point as the ratio of average LDE of its m nearest neighbors to the LDE at the point:

$$LDF(\mathbf{x}_j) \propto \frac{\sum_{\mathbf{x}_i \in mNN(\mathbf{x}_j)} \frac{LDE(\mathbf{x}_i)}{m}}{LDE(\mathbf{x}_j) + c \cdot \sum_{\mathbf{x}_i \in mNN(\mathbf{x}_j)} \frac{LDE(\mathbf{x}_i)}{m}}. \tag{7}$$

Here, c is a scaling constant (in all our experiments we used $c = 0.1$). The scaling of LDE values by c is needed, since $LDE(\mathbf{x}_j)$ may be very small or even equal to zero (for numerical reasons), which would result in very large or even infinity values of LDF if scaling is not performed, i.e., if $c = 0$ in Eq. 7. Observe that the LDF values are normalized on the scale from zero to $1/c$. Value zero means that $LDE(\mathbf{x}_j) \gg \sum_{\mathbf{x}_i \in mNN(\mathbf{x}_j)} \frac{LDE(\mathbf{x}_i)}{m}$ while value $1/c$ means that $LDE(\mathbf{x}_j) = 0$. The higher the LDF value at a given point (closer to $1/c$) the more likely the point is an outlier.

The normalization of LDE values makes possible to identify outliers with a threshold $LDF(\mathbf{x}_j) > T$ chosen independently for a particular data set.

Observe that it is possible to use the Eq. 6 with covariance matrix of the Gaussian that automatically adjusts to the shape of the whole neighborhood mNN. Let $\mathbf{\Sigma}_i$ be the covariance matrix estimated on the m data points in $mNN(\mathbf{x}_i)$. If we use a general Gaussian kernel with covariance matrices $\mathbf{\Sigma}_i$,

then Eq. 3 becomes:

$$\tilde{q}(\mathbf{x}_j) \propto \sum_{i=1}^{n} \frac{1}{h^{dim}|\mathbf{\Sigma}_i|^{\frac{1}{2}}} \exp(-\frac{d_{\Sigma}(\mathbf{x}_j, \mathbf{x}_i)^2}{2h^2}), \tag{8}$$

where $d_{\Sigma}(\mathbf{x}, \mathbf{y})^2 = (\mathbf{x} - \mathbf{y})^T \mathbf{\Sigma}_i^{-1} (\mathbf{x} - \mathbf{y})$ is the Mahalanobis distance of vectors $\mathbf{x}$ and $\mathbf{y}$. It can be shown that

$$d_{\Sigma}(\mathbf{x}, \mathbf{y})^2 = (\mathbf{x}^* - \mathbf{y}^*)^T \cdot (\mathbf{x}^* - \mathbf{y}^*). \tag{9}$$

Here,

$$\mathbf{x}^* \equiv (\mathbf{\Lambda}^T)^{-\frac{1}{2}} \mathbf{V}^T (\mathbf{x}^* - \mu), \tag{10}$$

where $\Lambda = diag(\lambda_1, \ldots, \lambda_k)$ is the diagonal matrix of eigenvalues and $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$ is the matrix od corresponding eigenvectors of $\mathbf{\Sigma}_i$ and μ is the mean of the vectors in the mNN neighborhood. Therefore, Eq. 8 can be, using Eq. 9 and Eq. 4 represented in the form:

$$\tilde{q}(\mathbf{x}_j) \propto \sum_{i=1}^{n} \frac{1}{h^{dim}|\mathbf{\Sigma}_i|^{\frac{1}{2}}} \exp(-\frac{d(\mathbf{x^*}_j, \mathbf{x^*}_i)^2}{2h^2}). \tag{11}$$

Now, analogous to Eq.6, we may generalize the LDE measure to:

$$LDE(\mathbf{x}_j) \propto \frac{1}{m} \sum_{\mathbf{x}_i \in NN(\mathbf{x}_j)} \frac{1}{(2\pi)^{\frac{dim}{2}} h^{dim}|\mathbf{\Sigma}_i|^{\frac{1}{2}}} \exp(-\frac{\mathbf{rd}_k(\mathbf{x}_j^*, \mathbf{x}_i^*)^2}{2h^2}) \tag{12}$$

Equation 12 can be replaced within Eq. 7 to obtain generalized measure of the local density factor.

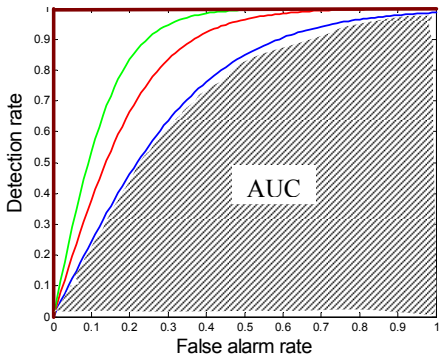


Fig. 1. The ROC curves for different detection algorithms

3 Performance Evaluation

Outlier detection algorithms are typically evaluated using the detection rate, the false alarm rate, and the ROC curves [44]. In order to define these metrics, let us look at a confusion matrix, shown in Table 1. In the outlier detection problem, assuming class "C" as the outlier or the rare class of the interest, and "NC" as a normal (majority) class, there are four possible outcomes when detecting outliers (class "C")-namely true positives (TP), false negatives (FN), false positives (FP) and true negatives (TN). From Table 1, detection rate and false alarm rate may be defined as follows:

$$DetectionRate = \frac{TP}{TP + FN}$$
$$FalseAlarmRate = \frac{FP}{FP + TN}.$$

Table 1. Confusion matrix defines four possible scenarios when classifying class "C"

	Predicted Outliers –Class C	**Predicted Normal Class–NC**
Actual Outliers –Class C	True Positives (TP)	False Negatives (FN)
Actual Normal –Class NC	False Positives (FP)	True Negatives (TN)

Detection rate gives information about the relative number of correctly identified outliers, while the false alarm rate reports the number of outliers misclassified as normal data records (class NC). The ROC curve represents the trade-off between the detection rate and the false alarm rate and is typically shown on a $2 - D$ graph (Fig. 1), where false alarm rate is plotted on x-axis, and detection rate is plotted on y-axis. The ideal ROC curve has 0% false alarm rate, while having 100% detection rate (Fig. 1). However, the ideal ROC curve is hardly achieved in practice. The ROC curve can be plotted by estimating detection rate for different false alarm rates (Fig. 1). The quality of a specific outlier detection algorithm can be measured by computing the area under the curve (AUC) defined as the surface area under its ROC curve. The AUC for the ideal ROC curve is 1, while AUCs of "less than perfect" outlier detection algorithms are less than 1. In Figure 1, the shaded area corresponds to the AUC for the lowest ROC curve.

4 Experiments

In this section, we compare the performance of the proposed LDF outlier detection measures (Eq. 7) to two state of the art outlier detection algorithms LOF [9]

and LOCI [35] on several synthetic data sets. In all of our experiments, we have assumed that we have information about the normal behavior (normal class) and rare events (outliers) in the data set. However, we did not use this information in detecting outliers, i.e. we have used completely unsupervised approach.

Recall that LOF algorithm [9] has been designed to properly identify outliers as data samples with small local distribution density, situated in vicinity of dense clusters. To compare LOF to our proposed LDF algorithm, we created two data sets $Dataset1$ and $Dataset2$. $Dataset1$ shown in Fig. 2(a) has two clusters of non-uniform density and sizes (with 61 and 27 data samples) and two clear outliers A and B (marked with stars in Fig. 2(a)). Data sample C does not belong to the second cluster, but as argued in [9] is should not be regarded as an outlier, since its local density is similar to its neighbors' local densities. Although points A and C have equal distances to their closest clusters ($cluster1$ and $cluster2$ correspondingly), the difference in clusters density suggests that A is an outlier while C is a normal data point. Recall that one of the main motivations for LOF in [9] is based on a data set of this kind.

As shown in Fig. 3, both methods LOF and the proposed LDF correctly identify the outliers A and B in $Dataset1$ without classifying C as an outlier (as in all figures presented here, the larger the circle and the darker its color, the higher the outlier factor value). However, observe that LOF assigns a significantly smaller LOF value to point B than A. This is counter intuitive, since point B is definitely the strongest outlier, and may lead to incorrect outlier detection results.

We illustrate the main problem of LOF on the second data set with two clusters of different densities shown in Fig. 2(b). The data set contains 41 points in sparse $cluster1$, 104 points in the dense $cluster2$, and four outstanding outliers A, B, C and$D$ (marked with stars). While samples C and D are clearly outliers, we regard samples A and B as outliers in analogy to sample A from $Dataset1$ (see Fig. 2(a)). Like sample A in $Dataset1$, their local density is lower then the local density of their neighbors from $cluster1$. In other words, samples A and B are too far from the closet cluster to be regarded as normal data points. However, the outlier values for points C and D should be significantly larger than for points A and B.

LOF was not able to detect points A and B as outliers for any value of its parameter k. We illustrate this fact in Fig. 4 for $k = 5$ and 20. Observe also that for larger k values, the LOF value of point C actually decreases. In contrast, as shown in Fig.5(a), LDF is able to clearly identify all four outliers. Fig. 5 also illustrates a multiscale behavior of LOF as a function of the bandwidth parameter h. For small values of h, more weight is given to close neighbors of a sample, while for larger values of h, the more distant neighbors also receive higher weight. In other words, with smaller h values, we have higher sensitivity to local situations, and therefore are able to detect all four outliers in Fig. 5(a) for $h = 0.5$. In contrast, with larger h, we smooth local variations. Consequently, for $h = 5$, LDF detects only two outliers, while for $h = 1$, LDF detects all four outliers, while assigning higher LDF values for the two clear outliers C and D.

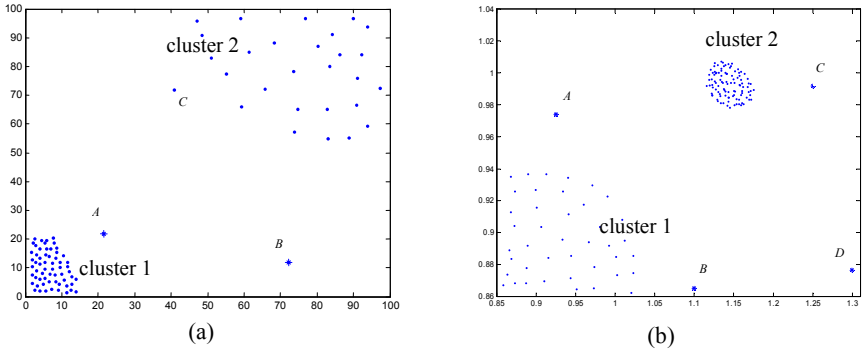


Fig. 2. Two simulated data sets with two clusters of different densities. (a) *Dataset*1: Two outliers A and B are marked with stars. (b) *Dataset*2: Four outliers are marked with A, B, C, and D

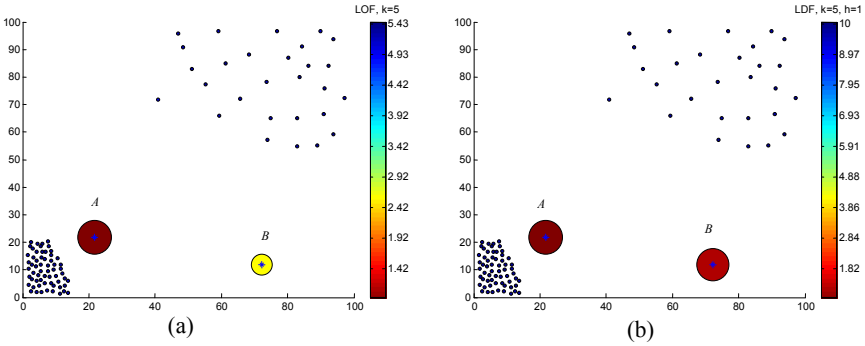


Fig. 3. Results on two cluster data set in Fig. 2(a) for $k = 5$: (a) LOF. (b) LDF

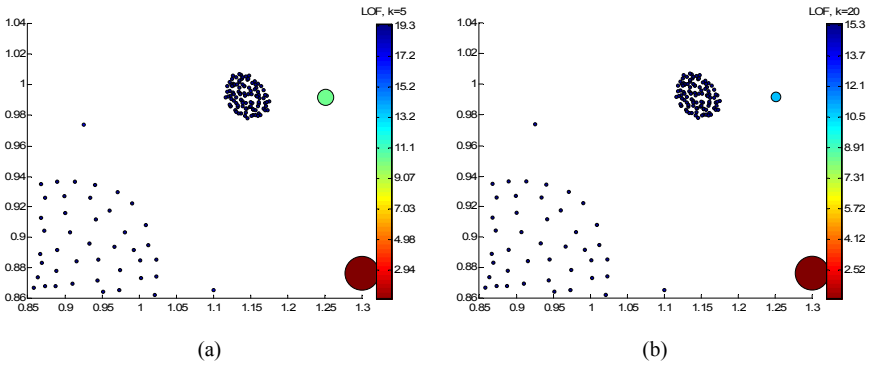
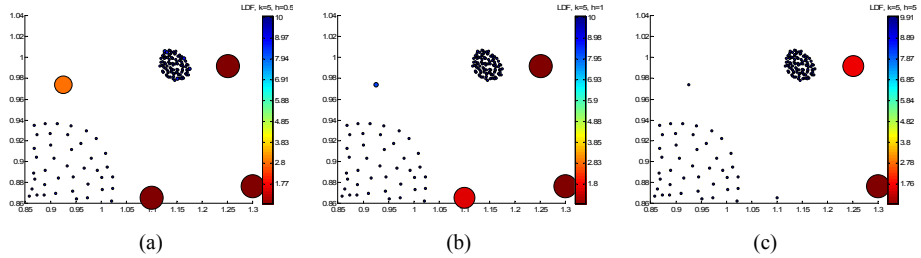


Fig. 4. LOF results on the data set in Fig. 2(b) for $k = 5$ and 20.



(a) (b) (c)

Fig. 5. LDF results on two cluster data set in 2(b) for $k = 5$ and bandwidth (a) $h = 0.5$, (b) $h = 1$, (c) $h = 5$

To further compare the results of the proposed algorithm with existing algorithms [9, 35], we generated synthetic data sets similar to those used in [35] (original data from this reference were not available to us). The data set *Dens* contains two uniformly distributed rectangular clusters (coordinates $(12, 22; 15, 25)$ and $(80, 120; 30, 70)$ respectively) with 200 samples in each and one outlier at coordinates $(32, 20)$. The second data set $Multimix$ contains a Gaussian cluster, two uniform clusters, 3 outstanding outliers (with coordinates $(80, 110)$, $(85, 110)$ and $(20, 50)$ and three points linearly positioned on top of the uniform cluster. The Gaussian cluster has 250 samples with mean at (20,110) and diagonal covariance matrix with both variances equal to 5. The first uniform cluster has 400 samples uniformly distributed in the rectangle $(130, 150; 95, 105)$. The second uniform cluster had 200 points uniformly distributed in the circle with center at $(80, 50)$ and radius 20.

In Fig. 6(a,b), we demonstrate the performance of the LDF algorithm with parameters $h = 1, k = 10, m = 30$ on these data sets. We compare results of the proposed algorithm with LOF [9]. Fig. 6(c,d) contains results of executing LOF algorithm for the same value of $k = 10$.

As we can see, the proposed LDF and the LOF algorithm performed similarly. LDF values for samples on the boundaries of the Gaussian cluster of $Multimix$ tend to be higher, but the computed high rank correlation [48] between LDF and LOF values (0.85) indicates similar order performance (since the outlier detection is performed by thresholding). We also compare the performance of the proposed algorithm with exact LOCI algorithm with parameters suggested in [35]. LOCI results are shown in Fig. 6(e,f) for $n_{min} = 20, \alpha = 2, k_\sigma = 2$. The visualization in Fig. 6(e,f) is different from (a-d), since LOCI outputs only a binary classification for each data point (outlier or not an outlier). As can be clearly seen, LOCI has trouble with data points on cluster boundaries. It tends to identify samples on boundaries of clusters as outliers.

,

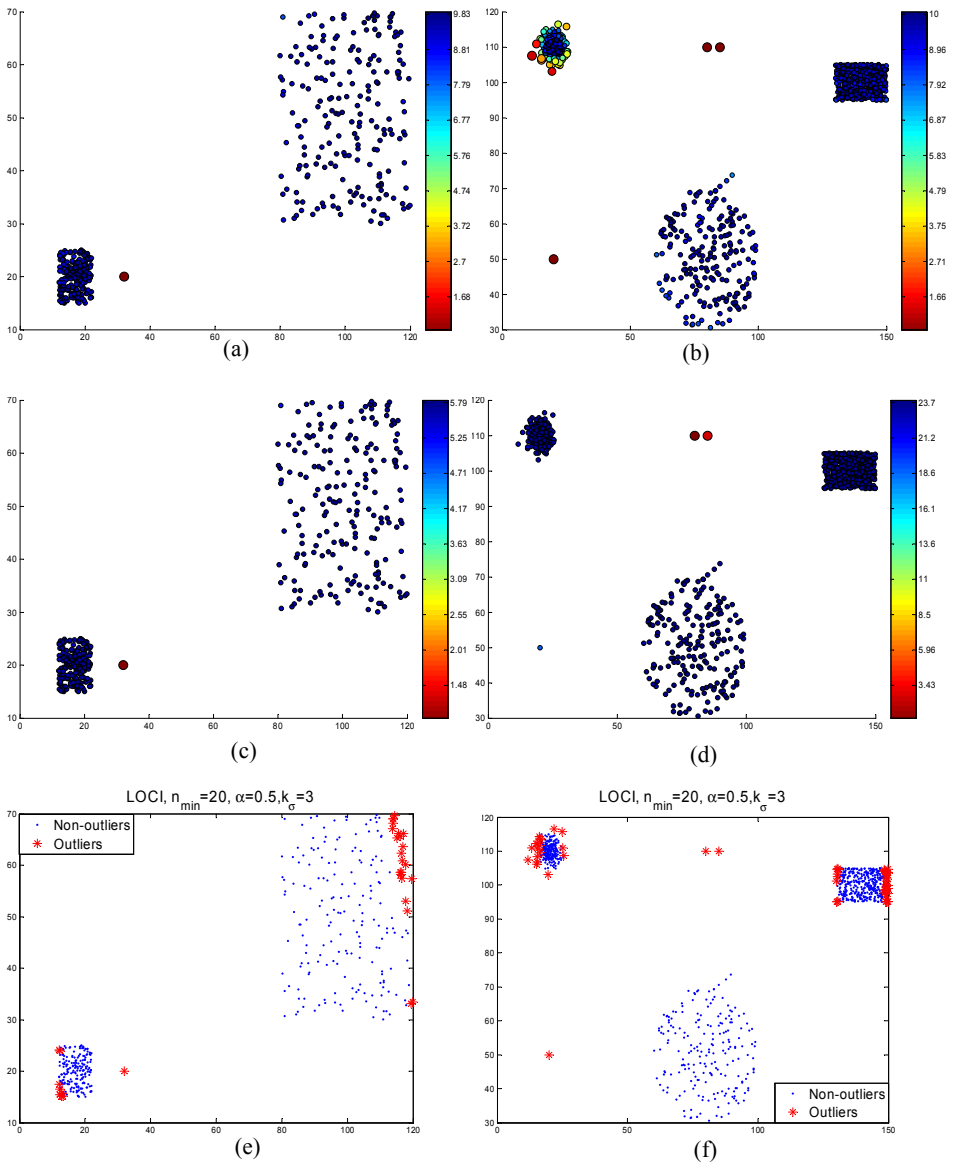


Fig. 6. LDF results on synthetic data sets (a) *Dens* (b) *Multimix*. Corresponding results for LOF are in (c) and (d), and for LOCI in (e) and (f)

5 Conclusions

A novel outlier detection framework is presented that is closely related to statistical nonparametric density estimation methods. Experimental results on several synthetic data sets indicate that the proposed outlier detection method can result in better detection performance than two state-of-the-art outlier detection algorithms (LOF and LOCI). Data sets used in our experiments contained different percentage of outliers, different sizes and different number of features, thus providing a diverse test bed and illustrating wide capabilities of the proposed framework. Although performed experiments have provided evidence that the proposed method can be very successful for the outlier detection task, future work is needed to fully characterize the method in real life data, especially in very large and high dimensional databases, where new methods for estimating data densities are worth considering. It would also be interesting to examine the influence of irrelevant features to detection performance of LDF method as well as to investigate possible algorithms for selecting relevant features for outlier detection task.

6 Acknowledgments

D. Pokrajac has been partially supported by NIH (grant $\sharp 2P20RR016472-04$), DoD/DoA (award 45395-MA-ISP) and NSF (awards $\sharp$ 0320991, $\sharp$0630388, $\sharp$ HRD-0310163). L. J. Latecki was supported in part by the NSF Grant IIS-0534929 and by the DOE Grant $DE-FG52-06NA27508$.

We would like to thank Guy Shechter for providing his KDTREE software in Matlab and for extending it by a KDNN function.

7 Reproducible Results Statement

All data sets used in this work are available by emailing the authors.

References

1. M. Joshi, R. Agarwal, V. Kumar, P. Nrule, Mining Needles in a Haystack: Classifying Rare Classes via Two-Phase Rule Induction, In Proceedings of the ACM SIGMOD Conference on Management of Data, Santa Barbara, CA, May 2001.
2. N. Chawla, A. Lazarevic, L. Hall, K. Bowyer, SMOTEBoost: Improving the Prediction of Minority Class in Boosting, In Proceedings of the Principles of Knowledge Discovery in Databases, PKDD-2003, Cavtat, Croatia, September 2003.
3. V. Barnett and T. Lewis, Outliers in Statistical Data. New York, NY, John Wiley and Sons, 1994.
4. A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava, V. Kumar, A comparative study of anomaly detection schemes in network intrusion detection, In Proceedings of the Third SIAM Int. Conf. on Data Mining, San Francisco, CA, May 2003.

5. A. Lazarevic, V. Kumar, Feature Bagging for Outlier Detection, In Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Chicago, IL, August 2005.
6. N. Billor, A. Hadi and P. Velleman BACON: Blocked Adaptive Computationally-Efficient Outlier Nominators, Computational Statistics and Data Analysis, **34**, pp. 279-298, 2000.
7. E. Eskin, Anomaly Detection over Noisy Data using Learned Probability Distributions, In Proceedings of the Int. Conf. on Machine Learning, Stanford University, CA, June 2000.
8. Aggarwal, C. C., Yu, P. Outlier detection for high dimensional data. In Proceedings of the ACM SIGMOD International Conference on Management of Data, 2001.
9. M. M. Breunig, H.P. Kriegel, R.T. Ng and J. Sander, LOF: Identifying Density Based Local Outliers, In Proceedings of the ACM SIGMOD Conference, Dallas, TX, May 2000.
10. E. Knorr and R. Ng, Algorithms for Mining Distance based Outliers in Large Data Sets, In Proceedings of the Very Large Databases (VLDB) Conference, New York City, NY, August 1998.
11. D. Yu, G. Sheikholeslami and A. Zhang, FindOut: Finding Outliers in Very Large Datasets, The Knowledge and Information Systems (KAIS), **4**, 4, October 2002.
12. E. Eskin, A. Arnold, M. Prerau, L. Portnoy and S. Stolfo. A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data, in Applications of Data Mining in Computer Security, Advances In Information Security, S. Jajodia D. Barbara, Ed. Boston: Kluwer Academic Publishers, 2002.
13. S. Hawkins, H. He, G. Williams and R. Baxter, Outlier Detection Using Replicator Neural Networks, In Proc. of the 4th Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK02), Aix-en-Provence, France, 170-180, September 2002.
14. G. Medioni, I. Cohen, S. Hongeng, F. Bremond and R. Nevatia. Event Detection and Analysis from Video Streams, IEEE Trans. on Pattern Analysis and Machine Intelligence, **8**(23), 873-889, 2001.
15. S.-C. Chen, M.-L. Shyu, C. Zhang, J. Strickrott: Multimedia Data Mining for Traffic Video Sequences. MDM/KDD 2001: 78-86.
16. Shu-Ching Chen, Mei-Ling Shyu, Chengcui Zhang, Rangasami L. Kashyap: Video Scene Change Detection Method Using Unsupervised Segmentation And Object Tracking. ICME 2001
17. Y. Tao, D. Papadias, X. Lian, Reverse kNN search in arbitrary dimensionality, In Proceedings of the 30th Int. Conf. on Very Large Data Bases, Toronto, Canada, September 2004.
18. Amit Singh, Hakan Ferhatosmanoglu, Ali Tosun, High Dimensional Reverse Nearest Neighbor Queries, In Proceedings of the ACM Int. Conf. on Information and Knowledge Management (CIKM'03), New Orleans, LA, November 2003.
19. I. Stanoi, D. Agrawal, A. E. Abbadi, Reverse Nearest Neighbor Queries for Dynamic Databases, ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, Dalas, TX, May 2000.
20. J. Anderson, Brian Tjaden, The inverse nearest neighbor problem with astrophysical applications. In Proceedings of the 12th Symposium of Discrete Algorithms (SODA), Washington, DC, January 2001
21. D. Pokrajac, L. J. Latecki, A, Lazarevic et al. Computational geometry issues of reverse-k nearest neighbors queries, Technical Report TR-CIS5001, Delaware State University 2006.

22. J. Conway, N. H. Sloane, Sphere Packings, Lattices and Groups, Springer, 1998.
23. F. P. Preparata, M. I. Shamos, "Computational Geometry: an Introduction", 2nd Printing, Springer-Verlag 1988
24. N. Roussopoulos, S. Kelley and F. Vincent, Nearest neighbor queries, 71-79, Proceedings of the ACM SIGMOD Conference, San Jose, CA, 1995
25. N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: an efficient and robust access method for points and rectangles. SIGMOD Rec., **19**(2):322-331, 1990.
26. S. Berchtold, D. A. Keim, and H.-P. Kriegel. The X-tree: An index structure for highdimensional data. In T. M. Vijayaraman, A. P. Buchmann, C. Mohan, and N. L. Sarda, editors, Proceedings of the 22nd International Conference on Very Large Databases, pages 28-39, San Francisco, U.S.A., 1996. Morgan Kaufmann Publishers.
27. R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In VLDB '98: Proceedings of the 24rd International Conference on Very Large Data Bases, pages 194-205, San Francisco, CA, USA, 1998. Morgan Kaufmann.
28. D. DeMenthon, L. J. Latecki, A. Rosenfeld, and M. Vuilleumier Stckelberg: Relevance Ranking of Video Data using Hidden Markov Model Distances and Polygon Simplification. Proc. of the Int. Conf. on Visual Information Systems, Lyon, France, Springer-Verlag, pp. 49-61, 2000.
29. L. J. Latecki, R. Miezianko, V. Megalooikonomou, D. Pokrajac, "Using Spatiotemporal Blocks to Reduce the Uncertainty in Detecting and Tracking Moving Objects in Video," Int. Journal of Intelligent Systems Technologies and Applications. **1**(3/4), pp. 376-392, 2006.
30. I. T. Jolliffe. Principal Component Analysis, 2nd edition. Springer Verlag, 2002.
31. R. P. Lippmann, D. J. Fried, I. Graf, J. et al, Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation, In Proc. DARPA Information Survivability Conf. and Exposition (DISCEX) 2000, **2**, pp. 12-26, IEEE Computer Society Press, 2000.
32. Tcptrace software tool, www.tcptrace.org.
33. UCI KDD Archive, KDD Cup 1999 Data Set, www.ics.uci.edu/ kdd/databases/kddcup99/kddcup99.html
34. J. Tang, Z. Chen, A. Fu, D. Cheung, Enhancing Effectiveness of Outlier Detections for Low Density Patterns, In Proceedings of the Sixth Pacific-Asia Conference on Knowledge Discovery and Data Mining, (PAKDD), Taipei, May, 2002.
35. S. Papadimitriou, H. Kitagawa, P. B. Gibbons, C. Faloutsos: LOCI: Fast Outlier Detection Using the Local Correlation Integral. Proc. of the 19th Int. Conf. on Data Engineering (ICDE'03), Bangalore, India, March 2003.
36. S. D. Bay, M. Schwabacher, Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In Proceedings of the Ninth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, New York, NY, 2003.
37. L. Breiman, W. Meisel, and E. Purcell. Variable kernel estimates of multivariate densities. Technometrics, **19**(2):135-144, 1977.
38. D. O. Loftsgaarden and C. P. Quesenberry. A nonparametric estimate of a multivariate density function. Ann. Math. Statist., **36**:1049-1051, 1965.
39. G. R. Terrell and D. W. Scott. Variable kernel density estimation. The Annals of Statistics, **20**(3):1236-1265, 1992.

40. M. Maloof, P. Langley, T. Binford, R. Nevatia and S. Sage, Improved Rooftop Detection in Aerial Images with Machine Learning, Machine Learning, **53**, 1-2, pp. 157 - 191, October-November 2003.
41. R. Michalski, I. Mozetic, J. Hong and N. Lavrac, The Multi-Purpose Incremental Learning System AQ15 and its Testing Applications to Three Medical Domains, In Proceedings of the Fifth National Conference on Artificial Intelligence, Philadelphia, PA, 1041-1045, 1986.
42. P. van der Putten and M. van Someren, CoIL Challenge 2000: The Insurance Company Case, Sentient Machine Research, Amsterdam and Leiden Institute of Advanced Computer Science, Leiden LIACS Technical Report 2000-09, June, 2000.
43. L. Ertoz, Similarity Measures, PhD dissertation, University of Minnesota, 2005.
44. F. Provost and T. Fawcett, Robust Classification for Imprecise Environments, Machine Learning, **42**, 3, pp. 203-231, 2001.
45. C. Blake and C. Merz, UCI Repository of machine learning databases, http://www.ics.uci.edu/ mlearn/MLRepository.html, 1998.
46. N. Roussopoulos, S. Kelly and F. Vincent, "Nearest Neighbor Queries," Proc. ACM SIGMOD, pp. 71-79, 1995.
47. J. Devore, "Probability and Statistics for Engineering and the Sciences," 6th edn., 2003.
48. W. J. Conover, "Practical Nonparametric Statistics," 3rd edn., 1999.