Outlier Detection with Globally Optimal Exemplar-Based GMM

Conference Paper · April 2009

DOI: 10.1137/1.9781611972795.13 · Source: DBLP

3 authors, including:

Longin Jan Latecki
Temple University
349 PUBLICATIONS **11,645** CITATIONS

SEE PROFILE

David Pokrajac
Delaware State University
140 PUBLICATIONS **1,783** CITATIONS

SEE PROFILE

SDM 2009

Outlier Detection with Globally Optimal Exemplar-Based GMM

Xingwei Yang Longin Jan Latecki* Dragoljub Pokrajac †

Abstract

Outlier detection has recently become an important problem in many data mining applications. In this paper, a novel unsupervised algorithm for outlier detection is proposed. First we apply a provably globally optimal Expectation Maximization (EM) algorithm to fit a Gaussian Mixture Model (GMM) to a given data set. In our approach, a Gaussian is centered at each data point, and hence, the estimated mixture proportions can be interpreted as probabilities of being a cluster center for all data points. The outlier factor at each data point is then defined as a weighted sum of the mixture proportions with weights representing the similarities to other data points. The proposed outlier factor is thus based on global properties of the data set. This is in contrast to most existing approaches to outlier detection, which are strictly local. Our experiments performed on several simulated and real life data sets demonstrate superior performance of the proposed approach. Moreover, we also demonstrate the ability to detect unusual shapes.

1 Introduction

Outlier detection refers to the problem of finding patterns in data that do not conform to expected behavior. These non-conforming patterns are often referred to as outliers or anomalies. Detection of outliers (or 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 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. Unlike supervised learning methods that typically require labeled data (the training set) to classify rare events [1], unsupervised techniques detect outliers (rare events) as data points that are very different from the normal (majority) data based on some prespecified measure. These methods are typically called outlier/anomaly detection techniques, and their success depends on the choice of similarity measures, feature selection, weighting, and most importantly on an approach used to detect outliers.

Outlier detection techniques can be categorized into several groups: (1) statistical or distribution-based approaches; (2) geometric-based approaches; (3) profiling methods; and (4) model-based approaches. In statistical techniques [2, 3], the data points are typically modeled using a data distribution, and points are labeled as outliers depending on their relationship with the distributional model. Geometric-based approaches detect outliers by (i) computing distances among points using all the available features [4, 5] or only feature projections [6]; (ii) computing densities of local neighborhoods [7, 8]; (iii) identifying side products of the clustering algorithms (as points that do not belong to clusters) [9] or as clusters that are significantly smaller than others. 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. Finally, model-based approaches usually first characterize the normal behavior using some predictive models (e.g., replicator neural networks [10] or unsupervised support vector machines [11]), and then detect outliers as deviations from the learned model.

One of the main challenges of outlier detection algorithms are data sets with non-homogeneous densities. Clustering-based outlier detection algorithms cannot properly detect the outliers in case of noisy data and unless the number of clusters is known in advance. In this paper, we propose a novel approach that successfully solves these challenges. First we fit a Gaussian Mixture Model (GMM) with Gaussians centered at each data point to a given data set. Since we only

*Supported by NSF Grant IIS-0812118 in the Robust Intelligence Cluster and by the DOE Grant DE-FG52-06NA27508.
CIS Dept., Temple University, Philadelphia, PA 19122, USA, {xingwei, latecki}@temple.edu.

†CIS Dept. and AMRC, Delaware State University, Dover DE 19901, dpokraja@desu.edu

estimate the mixture proportions, the estimation in the EM framework leads to a convex optimization with a unique globally optimal solution (e.g., see [12]).

Intuitively, each mixture proportion represents the degree to which the point is a cluster center. The higher the mixture proportion, the more likely it is a cluster center, which means it has higher influence on other points. Reversely, the lower the mixture proportion is, the less likely the point is a cluster center, which means it has lower influence on other points. The outlier factor at each data point is then defined as a weighted sum of the mixture proportions with weights representing the similarities to other data points. The main advantage of the proposed approach is that it brings a global information to each data point. Thus, each outlier decision is made in global context. In contrast, the existing density based approaches usually only consider the local neighborhoods of points, which make them unable to consider global information in the computation of outliers.

The remainder of this paper is organized as follows. In Section 2, we briefly review some well-known geometric-based outlier detection algorithms. The details of the proposed approach is given in section 3. Section 4 gives the experimental results to show the advantage of the proposed approach. The time complexity of the proposed approach is analysed in Section 5. Conclusion and discussion are given in Section 6.

2 Related Work

Breunig et al. [7] assign an outlier score to any given data point, known as Local Outlier Factor (LOF), depending on distances in its local neighborhood. LOF as well as all other outlier detection methods based on local neighbors have difficulty to identify outliers in data sets with varying densities. A simple example of such data sets is shown in Fig. 1. There are four outliers and two clusters with different densities. As we show in Section 4, LOF is unable to detect the four outliers for any size of the local neighborhood. Besides, some distance-based outlier detection work has been introduced recently [36, 34, 35].

Tang et. al [13]introduced the Connectivity-based outlier factor (COF) algorithm, dual to the LOF algorithm. Analog to LOF, COF algorithm identifies as outliers points where a quantity called average chaining distance is larger than the average chaining distance at their k-nearest neighborhood. Instead of comparing the density of the point to its neighbors densities to calculate the outlier factor as in LOF, COF algorithm considers the Set Based Nearest path (SBN), which is basically a minimal spanning tree with k nodes, starting from the point in question. Outlier factor of a point

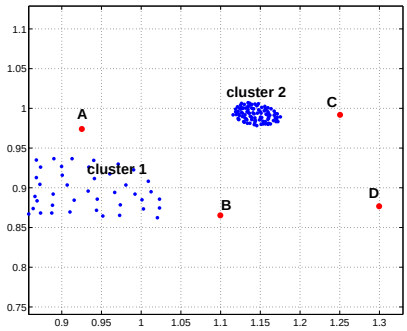

Figure 1: A simulated data set with two clusters of different densities and four outliers A, B, C and D.

is calculated using the SBN for that particular point and that of its neighbors. Both LOF and COF are very similar in nature, and are based on local neighborhoods.

Papadimitriou et al. [8] computes the neighborhood size (the number of neighbors) for each point and identifies as outliers points whose neighborhood size significantly vary with respect to the neighborhood size of their neighbors. More precisely, they propose the use of Multi-granularity Deviation Factor (MDEF). MDEF measures to which extent the density in a point varies from the average density in its neighborhood. Note that here the density is measured as the number of neighbors within a specified distance (and not as the average distance of the specific number of neighbors as it is basically performed in LOF). Instead of assigning an outlier score to a test point, the LOCI algorithm uses a richer LOCI plot which contains information such as inter-cluster distance, cluster diameter, etc. Recent extensions of this algorithm also use density estimates. In [14] a kernel based distribution density estimate is used in context of LOCI-based on-line outlier detection [8]. This approach uses Epanechnikov kernels with fixed bandwidths throughout the whole data set. Lazarevic and Kumar [15] introduce a novel feature bagging approach to detection outliers. It combines results from multiple outlier detection algorithms that are applied using different set of features.

Many of the earlier clustering-based outlier detection techniques find outliers as by-product of a clustering algorithm [16, 17]. There, the data point which does not belongs to any cluster is considered as an outlier. For example, Jiang et. al [18] proposed a variant of k-means algorithm to detect outliers. Since the main aim is to find clusters, such techniques are not optimized to find outliers. However, several clustering based techniques focus on detecting outliers, instead of generating

clusters. The CLAD algorithm [19] derives the width from the data by taking a random sample and calculating the average distance between the closest points. All the clusters whose density is lower than a threshold are declared as 'local' outliers while all those clusters which are far away from other clusters are declared as 'global' outliers. FindCBLOF [20] uses squeezer [21] to determine the Cluster-Based Local Outlier Factor (CBLOF) for each data point. Similar to it, Barbara et. al [22] introduced an approach which is based on Transductive Confidence Machines. One of the main problems of DBSCAN [17] is that it cannot deal with clusters of different densities. A recent clustering algorithm called OPTICS [23] can however perform well in the presence of unknown number of clusters with various densities. It is extended for outlier detection [24]. As reported in [25], it is very similar to LOF [7]. The main problem of clustering-based outlier detection techniques is that the outlier factor is a direct outcome of clustering. Consequently, the quality of outlier detection is directly linked to the quality of clustering. Although the proposed approach is related to clustering in that we estimate the degree of being cluster center for each data point, we actually do not perform any clustering. Therefore, the quality of the proposed outlier factor is not directly related to any clustering outcome.

3 Convex Optimization and Outlier Detection

Given a set of data points $\mathcal{X} = \{x_1, \ldots, x_n\}$, a standard Gaussian mixture model clustering seeks to maximize the scaled log-likelihood function

$$l(\pi_{1:m},\ \mu_{1:m}, \lambda\,; \mathcal{X}) = \frac{1}{n}\sum_{i=1}^{n} \log[\sum_{j=1}^{m} \pi_j p(x_i|\mu_j, \lambda)] \tag{3.1}$$

where m is the number of model components, $\pi_j = p(\omega_j|\lambda)$ represents the strength of jth component ω_j with $\sum_{i=1}^{m} \pi_i = 1$, and $\pi_{1:m}$ is a vector composed of π_j for $j = 1, \ldots, m$. The probability $p(x_i|\mu_j, \lambda)$ is a Gaussian and λ is a vector of parameters specified below.

In the standard mixture model μ_j is the unknown mean vector for jth component and is estimated together with other parameters using an EM algorithm. Since our goal is not clustering but an estimation of an outlier factor at every data point, we assume that each data point is a cluster center. Thus, in our setting, $m = n$ and $\mu_j = x_j$ for $j = 1, \ldots, n$. This way, the mixture proportion π_j represents the likelihood of point x_j to be a cluster center. We obtain a simplified version of Eq. (3.1):

$$l(\pi_{1:m}\,; \mathcal{X}) = \frac{1}{n}\sum_{i=1}^{n} \log[\sum_{j=1}^{n} \pi_j p(x_i|x_j, \lambda)]. \tag{3.2}$$

We obtain a particularly simple version of the EM algorithm in which at tth iteration, we only need to estimate the vector $\lambda_t = \{\pi_1(t),\ \cdots\ ,\ \pi_m(t)\}$. As it is the case in every EM algorithm, we iterate the following two steps.

In E-step we compute for each class $i = 1, \ldots, n$ and for each data point $k = 1, \ldots, n$:

$$p(x_i|x_k, \lambda_t) = \frac{p(x_k|x_i, \lambda_t)p(x_i|\lambda_t)}{p(x_k|\lambda_t)} \tag{3.3}$$

$$= \frac{p(x_k|x_i)\pi_i(t)}{\sum_{j=1}^{n} p(x_k|x_j)\pi_j(t)}, \tag{3.4}$$

since $p(x_i|\lambda_t) = p(\omega_i|\lambda) = \pi_i$.

Our M-step is particularly simple, since we only need to update the mixture proportions:

$$\pi_i(t+1) = \frac{1}{n}\sum_{k=1}^{n} p(x_i|x_k, \lambda_t). \tag{3.5}$$

Then, plugging Eq. (3.4) into Eq. (3.5) gives

$$\pi_i(t+1) = \frac{1}{n}\sum_{k=1}^{n} \frac{p(x_k|x_i)\pi_i(t)}{\sum_{j=1}^{n} p(x_k|x_j)\pi_j(t)} \tag{3.6}$$

where

$$p(x_k|x_j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(d(x_k,x_j))^2}{2\sigma^2}}$$

and $d(x_k, x_j)$ is a distance between data points x_k and x_j. The scale factor σ is the only parameter of the proposed approach.

A very important property of our approach is the fact that the recursive estimation in Eq. (3.6) has a globally unique solution, since we perform convex optimization that is guaranteed to converge to the global optimum if distance d is a Bergman divergence [12]. A particular instance of a Bergman divergence is the Euclidean distance [26].

In order to provide a more detailed explanation of Eq. (3.6), we denote $s_{kj} = p(x_k|x_j)$. We observe that $S = (s_{kj})$ is the affinity matrix of the graph spanned by the data set $\mathcal{X}$, where s_{kj} represents the strength of the connection between points x_k and x_j. Consequently, we obtain the following formulation of Eq. (3.6)

$$\pi_i(t+1) = \frac{1}{n}\sum_{k=1}^{n} \frac{s_{ki}\pi_i(t)}{\sum_{j=1}^{n} s_{kj}\pi_j(t)}, \tag{3.7}$$

which can be expressed as:

$$\pi_i(t+1) = \frac{1}{n}\sum_{k=1}^{n} \frac{s_{ki}\pi_i(t)}{z_k(t)} \tag{3.8}$$

where

$$z_k(t) = \sum_{j=1}^{n} s_{kj}\pi_j(t). \tag{3.9}$$

From the view of point x_k, Eq. (3.9) represents how all the other points influence it. In particular, the term $s_{kj}\pi_j(t)$ represents how much point x_k is influenced by point x_j with s_{kj} being the strength of the connection and $\pi_j(t)$ measuring the importance of point j. This motivates the proposed definition of the ***outlier factor*** at point x_k as

$$F_k = z_k(t_h) = \sum_{j=1}^{n} s_{kj}\pi_j(t_h), \tag{3.10}$$

where t_h represents the final iteration step of Eq. (3.8). In other words, we iterate Eq. (3.8) until convergence, and then define the outlier factor by Eq. (3.10).

According to Eq. (3.10), the smaller F_k, the more likely is data point x_k to be an outlier. However, usually an outlier factor is defined so that the larger it is, the more likely a given data point is an outlier (e.g., [13, 7]). Therefore, we will use the reciprocal of Eq. (3.10) as our definition of ***outlier factor*** in the rest of this paper, and will denote it with

$$OF_k = \frac{1}{F_k}. \tag{3.11}$$

As we demonstrate in the next section, the globally optimal procedure used to define the proposed outlier factor leads to robust outlier detection results.

4 Experimental Results

Our experiments were performed on several synthetic and real life data sets. In all 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. Besides, in order to show the improvement of the proposed approach, we compare to three state of the art outlier detection algorithms: COF [13], LOF [7], and LOCI [8].

4.1 Synthetic data sets In Fig. 2, a synthetic data set is used to illustrate the advantage of the proposed approach compared to LOF [7] and COF algorithms [13]. The data set contains 41 points in the sparse *cluster*1, 104 points in the dense *cluster*2, and four outstanding outliers A, B, C and D (marked with red dots).

LOF and COF methods were not able to detect points A and B as outliers for any value of its parameter k. Fig. 2(a,b,c,d) illustrates this fact for two values of parameter k ($k = 10$ and $k = 20$). Unlike LOF and COF, the proposed approach is able to clearly identify all four outliers (Fig.2(e)).

As our second synthetic data set on the plane, we use an elongated data set shown in Fig. 3(a). The *Curvepoints* data set is obtained by digitalization and thresholding of a silicon wafer micrography image. The goal is to detect and eliminate outliers (image artifacts) in order to obtain a proper parametric approximation of the depicted curve. The data set consists of 868 two-dimensional points positioned along a curved line. It contains three outstanding outliers with coordinates $[217, 855]$, $[706, 714]$ and $[707, 716]$, which are marked with a, b, c in Fig. 3(a). Due to the image resolution, the last two outliers are shown as a single point in Fig. 3(a), since they are very close to each other.

As can be seen in Fig. 3(b,c) neither LOF nor COF can correctly detect the three outliers, even though COF is designed for finding outliers in elongated data sets. Here we take a different approach from COF to deal with elongated data sets. We propose to learn new distances between data points so that we can apply our outlier detection method without any modifications. The new distances are learned with Path-Based Similarity Measure [27, 28]. The idea of this measure is to compute new distances between points by considering certain paths through the data set. First a weighted complete graph $G = (\mathcal{X}, E, d)$ is constructed with edge weights $d_{ij} = d(x_i, x_j)$ being the original distance between points x_i, x_j. Let P_{ij} represent all paths from point x_i to point x_j. Then the new distance between points x_i and x_j is defined as:

$$d'_{ij} = min_{p \in P_{ij}}\{max_{1 \le h \le |p|-1} d(p[h], p[h+1])\} \tag{4.12}$$

where $p[h]$ represents the hth point in the path p. As shown in [27, 28], d' can be efficiently computed with a liner algorithm with respect to the the number of data points n.

For the *Curvepoints* data set, the distance between points are learned according to Eq. (4.12). Then, the proposed outlier detection algorithm is used and the result is shown in Fig. 3(d). To summarize, we propose a two step process for outlier detection in elongated data sets: first learning new distances and then detecting the outliers.

We have also tested LOCI on both data sets (two cluster and *Curvepoints*), and it was unable to correctly detect the outliers. We did not include any figures with LOCI results, since it only returns binary labels for outliers; thus, making the visualization as shown in Figs. 2 and 3 impossible.

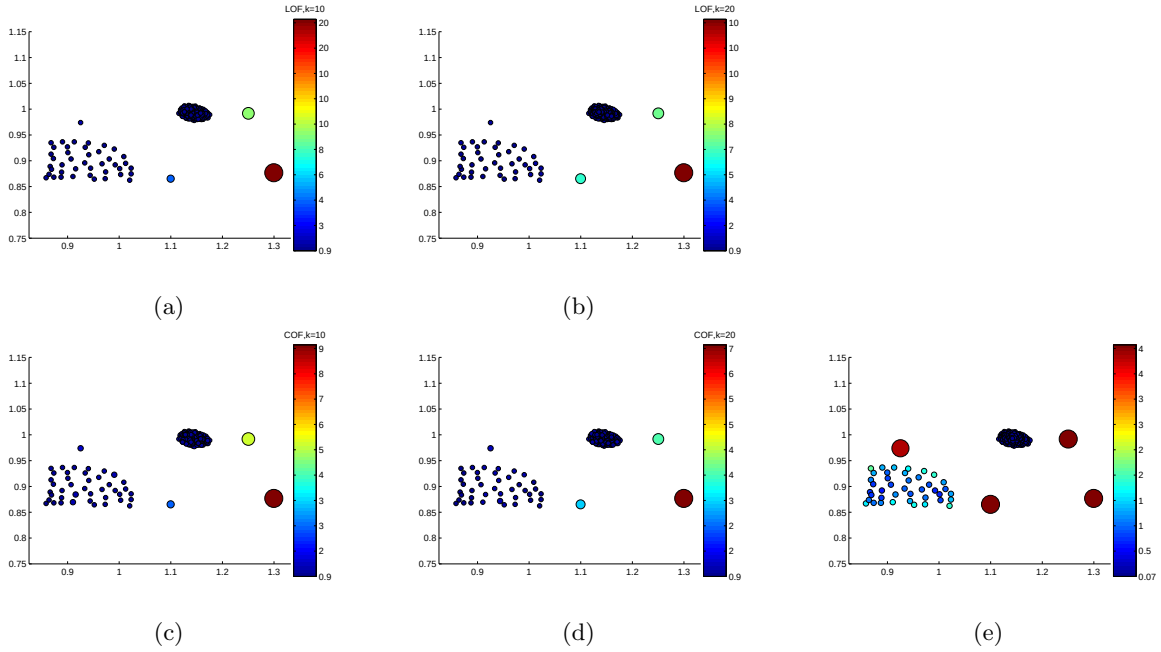

Figure 2: Results on the data set in Fig. 1 for LOF in (a) k=10 and (b) k=20, for COF in (c) k=10 and (d) k=20 The result of the proposed approach is shown in (e).

4.2 Performance Evaluation Before we describe our experimental results on real data sets in Section 4.3, we briefly describe here a standard performance measure. Outlier detection algorithms are typically evaluated using the detection rate, the false alarm rate, and the ROC curves [33]. 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:

$$\begin{aligned} &\text{Detection Rate} = \text{TP}/(\text{TP}+\text{FN}) \\ &\text{False alarm rate} = \text{FP}/(\text{FP}+\text{TN}) \end{aligned} \tag{4.13}$$

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, 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. 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. The quality of a specific outlier detection algorithm can be measured by computing the surface area under the ROC curve (AUC). The AUC for the ideal ROC curve is 1, while AUCs of "less than perfect" outlier detection algorithms are less than 1.

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 Positive (TP)	False Negative (FN)
Actual Normal Class NC	False Positive (FP)	True Negative (TN)

4.3 Real data sets The real life data sets used in our experiments have been used earlier by other researchers for the outlier detection [32, 15]. Since rare class analysis is conceptually the same problem as the outlier detection, we employed those data sets for the purpose of outlier detection, where we detected rare class as

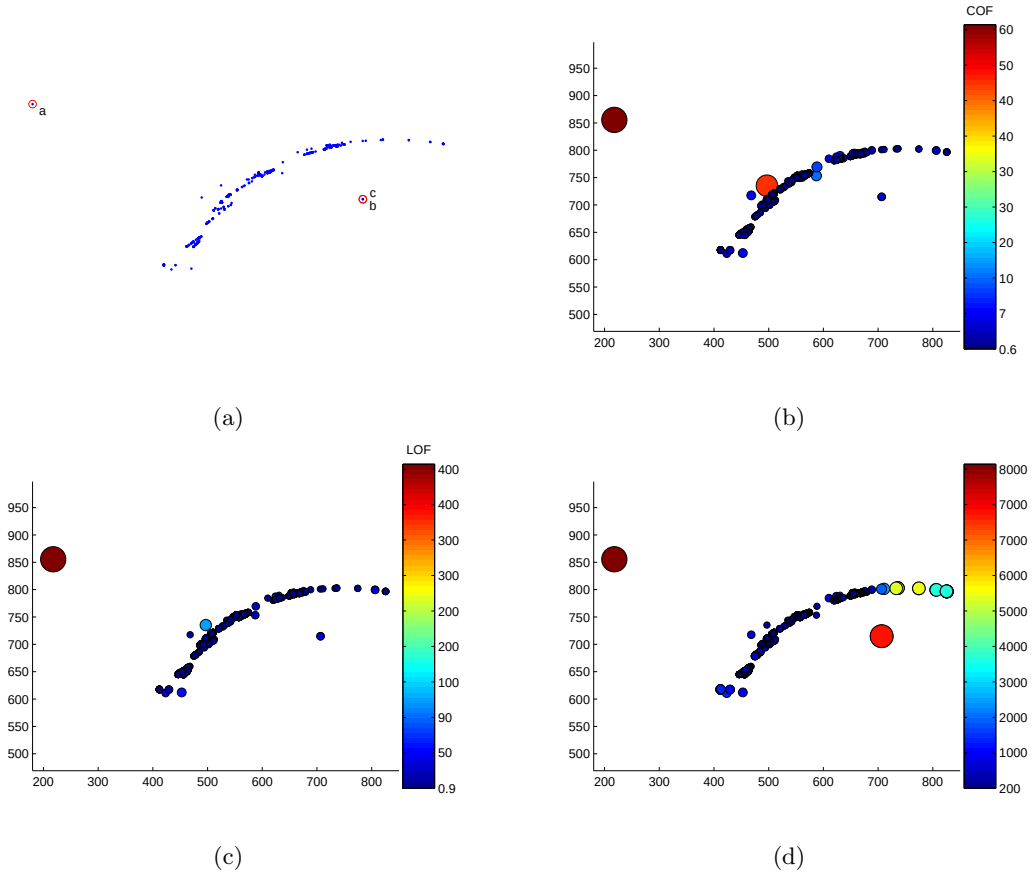

Figure 3: *Curvepoints* are shown in (a) and the outlier detection result of COF in (b), LOF in (c), and of the proposed approach in (d).

outliers. In all our experiments reported in this section, we use the original, Euclidean distances as used by other researches on the test data sets. In particular, we did not use Eq. (4.12) to modify the distances.

The COIL 200 data set consists of two classes and the size of the data set is 5822 with 348 outliers. Similar to COIL 200, Mammography also has two classes, in which the larger one contains 10923 instances and the smaller one contains 260 instances. The Rooftop data set contains 17829 data points with 9 continuous features, where 781 data points (4.38% of entire distribution) correspond to the rare class (outliers). For the Satimage data set we choose the smallest class as the minority class and collapsed the remaining classes into one class as was done in [32]. This procedure gives us a skewed 2-class data set, with 5809 majority class examples and 626 minority class outliers.

The results of the proposed approach compared to other approaches are shown by ROC curves in Fig. 4 and AUC values in Table 4.3. It can be observed that the proposed approach outperforms all of the state of art methods [7, 13]. The improvement in the detection performance for the COIL 200 is very obvious, but it is still under 60% (Fig. 4 and Table 4.3). The main reason may be the large number of attributes in the data set, which has 85 attributes. Similar to COIL200, Satimage data set has 36 features for each object. Therefore, the proposed approach has significant improvement for Satimage data set (Fig. 4(b) and Table 4.3), but it still needs further improvement. The greatest enhancements in outlier detection are achieved for the mammography and Rooftop data sets (Fig. 4(c),(d) and Table 4.3).

4.4 Unusual Shapes Among the visual features of multimedia content, shape is of particular interest because humans can often recognize objects solely based on shape. There has been a great amount of research on shape analysis, focusing mostly on shape indexing, clustering and classification. Recently [29] introduced a new problem of finding shape discords, which are the

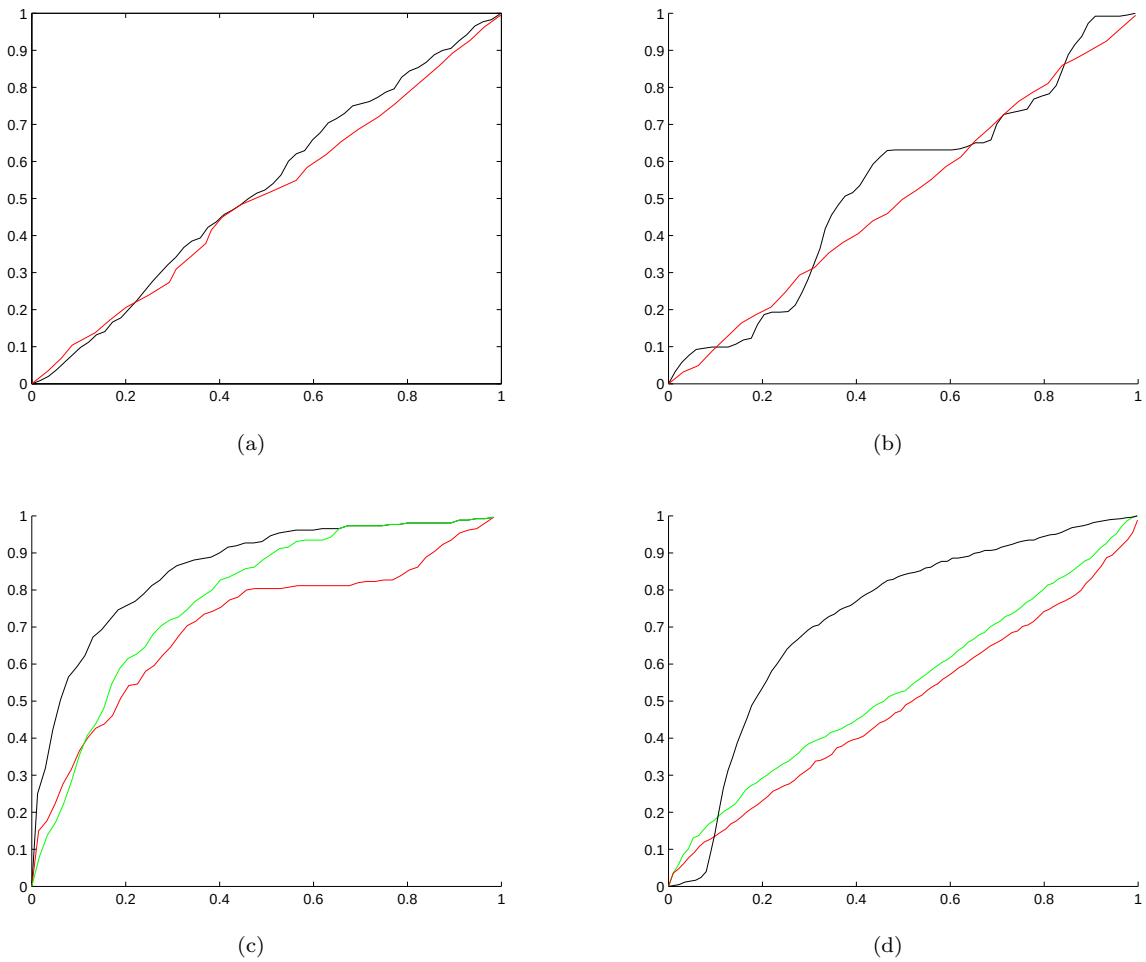

Figure 4: ROC curves for LOF (red), COF (green), and the proposed approach (black) for the data sets: (a) COIL 200, (b) Satimage, (c) Mammography, and (d) Rooftop.

Table 2: AUC (areas under the curves).

	AUC		
Data sets	LOF	COF	Proposed Approach
COIL200	0.499	0.505	0.529
Satimage	0.497	0.503	0.533
Mamography	0.710	0.780	0.862
rooftop	0.538	0.498	0.722

most unusual shapes in a collection. Thus, shape discords are simply outliers in shape data sets. In [29], the shape discord is defined as follows. Given a collection of shapes S, shape D is the discord of S if D has the largest distance to its nearest neighbor, i.e., $\forall$ shape C in S, the nearest neighbor M_C of C, and the nearest match M_D of D, $Dist(D, M_D) > Dist(C, M_C)$. This is a particularly simple definition of outliers. The main advantage of the approach in [29] is the fact that it is three to four orders of magnitude faster than the brute force algorithm for finding so defined shape discords. However, with the increasing speed, the accuracy of the approach is reduced. The first nearest neighbor is a too local criterion for detecting unusual shapes. For example, in Fig. 5, there are 35 shapes in the shape database, two bones, ten hearts, twenty deers and three horses. It

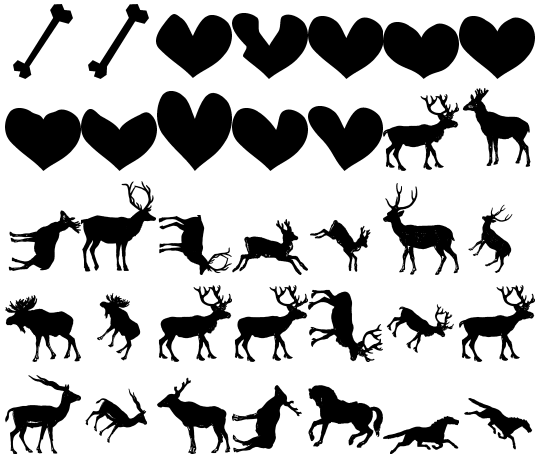

Figure 5: The shape database for unusual shape detection

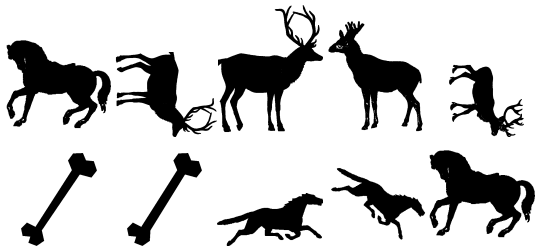

Figure 6: From left to right is the first to fifth most unusual shapes for the data set in Fig. 5. (first row) The unusual shape detection results of [29]. (second row) The unusual shape detection results of proposed approach

.

is obvious that the five outliers in the data set should be the two bones and three horses. The difference between two shapes is calculated by the shape dissimilarity measure introduced in [30], called Inner Distance Shape Context (IDSC). As demonstrated in [30], IDSC provides very good shape retrieval results. However, IDSC is not an Euclidean distance, and it is not a metric, since it violates the triangle inequality. Therefore, our results shown below demonstrate the ability of the proposed approach to work with distances measures that are not metrics.

According to the definition of [29], the first five shape discords are shown in Fig. 6 (first row). Thus, the approach in [29] identified only one horse as outlier and it missed two most obvious outliers, which are the two bones. The too local criterion in [29] is only able to find the correct unusual shapes if they are very different from all other shapes. However, it is not sufficient for solving the unusual shape problem. As shapes are very complex and there are different distance distributions in different classes, a global context information is needed. In contrast, the proposed outlier detection algorithm considers the whole data set to determine the unusual shapes. As shown in Fig. 6 (second row), it correctly identified all five outlier shapes.

In order to further demonstrate the ability of the proposed approach for detecting unusual shapes, it is tested on the widely used MPEG-7 shape data set for detecting the unusual shapes in each class. MPEG-7 consists of 70 classes with 20 shapes in each class. Since this data set is designed to test shape similarity retrieval, there are large in-class variances of shapes [31]. Nevertheless, as demonstrated by our results in Fig. 7 for three example shape classes, the detected outlier shapes can be easily justified. For each class, the two shapes in the first row are the outliers and the rest of the 18 shapes in the class are shown in rows 2 to 4. For the class in Fig. 7 (a), the two 'comma' outliers are distorted a lot in comparison to the other shapes. Similarly, the detected unusual hearts in Fig. 7 (b) are very different compared to other hearts. For the elephants in Fig. 7 (c), the two elephants have different viewpoints and their pose is different from the others.

5 Time Complexity Analysis

Since Eq. (3.9) is computed for each data point, and it requires computing Eq. (3.8), the complexity of a single iteration is $O(n^3)$, where n is the number of data points. For N iterations, the total time complexity is $O(N\ n^3)$. Empirically, we set N to 5000 in all our experiments. Thus, the proposed method has cubic time complexity with rather large constant factor N.

Compared to the proposed approach, LOF algorithm requires a constant number of nearest neighbor searches per each point in the dataset. Assuming that there is available indexing structure to support fast nearest neighbor search, the computational complexity of static LOF is $O(n \log n)$ where n is the number of data points. The COF algorithms requires computation of k-nearest neighbor queries per each example and computation of the average chain distance. Under the same assumption as above, the computational complexity of COF is $O(n \log n)$.

6 Conclusions

We have presented a novel technique to detect outliers based on a globally optimal variant of EM. The proposed approach does not make any assumption about the data distributions and it is unsupervised. It only

requires one parameter, the σ of a Gaussian kernel. We have shown that it provides excellent results for synthetic and widely used real data sets. In particular, it significantly outperforms the approaches in [7, 13]. Though it also outperforms the approach in [29] for detecting unusual shapes in the shape database, as it considers global context information, the complexity is much higher than [29]. Therefore, reducing the complexity will be the main goal in future.

References

[1] M. Joshi, R. Agarwal, V. Kumar and P. Nrule, *Mining Needles in a Haystack: Classifying Rare Classes via Two-Phase Rule Indection*, Proceedings of the ACM SIGMOD Conference, 2001.

[2] N. Billor, A. Hadi and P. Velleman, *BACON: Blocked Adaptive Computationally-Efficient Outlier Nominators*, Computational Statistics and Data Analysis., 34 (2000), pp. 279–298.

[3] E. Eskin, *Anomaly Detection over Noisy Data using Learned Probability Distributions*,ICML, 2000.

[4] S. Ramaswamy, R. Rastogi and K. Shim, *Efficient algorithms for mining outliers from large data sets*, Proceedings of the ACM SIGMOD Conference, 2000.

[5] E. Knorr and R. Ng, *Algorithms for Mining Distance based Outliers in Large Data Sets*, VLDB, 1998.

[6] C. C. Aggarwal and P. Yu , *Outlier detection for high dimensional data*, Proceedings of the ACM SIGMOD Conference, 2001.

[7] M. M. Breunig, H. P. Kriegel, R. T. Ng and J. Sander, *LOF: Identifying Density Based Local Outliers*, Proceedings of the ACM SIGMOD Conference, 2000.

[8] S. Papadimitriou, H. Kitagawa, P. B. Gibbons and C. Faloutsos, *LOCI: Fast Outlier Detection Using the Local Correlation Integral*, ICDE, 2003.

[9] D. Yu, G. Sheikholeslami and A. Zhang, *FindOut: Finding Outliers in Very Large Datasets*, The Knowledge and Information Systems (KAIS), 4 (2002), pp. 387–412.

[10] S. Hawkins, H. He, G. Williams and R. Baxter, *Outlier Detection Using Replicator Neural Networks*, Proc. of the 4th Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK02), 2002.

[11] A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava and V. Kumar, *A comparative study of anomaly detection schemes in network intrusion detection*, SDM, 2003.

[12] D. Lashkari and P. Golland, *Convex Clustering with Exemplar-Based Models*, Advances in Neural Information Processing Systems, 2007.

[13] J. Tang, Z. Chen, A. Fu and D. Cheung, *Enhancing Effectiveness of Outlier Detections for Low Density Patterns*, PAKDD, 2002.

[14] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki and D. Gunopulos, *Online Outlier Detection in Sensor Data Using Non-Parametric Models*, VLDB, 2006.

[15] A. Lazarevic and V. Kumar, *Feature Bagging for Outlier Detection*, KDD, 2005.

[16] D. M. J. TAX, *One-class classification: concept-learning in the absence of counter-examples*, PHD Thesis, 2001.

[17] M. Ester, H.-P Kriegel, J. Sander and X. Xu, *A density-based algorithm for discovering clusters in large spatial databases with noise*, KDD, 1996

[18] M. F. Jaing, S. S. Tseng and C. M. Su, *Two-phase clustering process for outliers detection*, Pattern Recognition Letters, 22 (2001), pp. 691–700.

[19] M. V. Mahoney and P. K. Chan, *Learning rules for anomaly detection of hostile network traffic*, ICDM, 2003.

[20] Z. He, X. Xu and S. Deng, *Discovering cluster-based local outliers*, Pattern Recognition Letters, 24 (2003), pp. 1641–1650.

[21] Z. He, X. Xu and S. Deng, *Squeezer: An Efficient Algorithm for Clustering Categorical Data*, Journal of Computer Science and Technology, 17 (2003), pp. 611–624.

[22] D. Barbara, C. Domeniconi and J. P. Rogers, *Detecting Outliers using Transduction and Statistical Testing*, KDD, 2006.

[23] M. Ankerst, M. M. Breunig, H. Kriegel and J. Sander, *OPTICS: Ordering Points To Identify the Clustering Structure*, SIGMOD, 1999.

[24] M. M. Breunig, H. P. Kriegel, R. T. Ng and J. Sander, *Optics-of:Identifying local outliers*, Proceedings of the third European Conference on Principles of Data Mining and Knowledge Discovery, 1999.

[25] V. Chandola, A. Banerjee and V. Kumar, *Outlier Detection- A Survey*, ACM Computing Surveys, to appear.

[26] A. Banerjee, S. Merugu, I. S. Dhillon and J. Ghosh, *Clustering with Bregman Divergences*, Journal of Machine Learning Research, 6 (2005), pp. 1705–1749.

[27] B. Fischer and J. M. Buhmann, *Path-based clustering for grouping of smooth curves and texture segmentation*, IEEE. PAMI, 25 (2003), pp. 513–518.

[28] B. Fischer, V. Roth and J. M. Buhmann, *Clustering with the connectivity kernel*, Advances in Neural Information Processing Systems, 2004.

[29] L. Wei, E. Keogh and X. Xi, *SAXually Explicit Images: Finding Unusual Shapes*, ICDM, 2006.

[30] H. Ling and D.W. Jacobs, *Shape Classification Using the Inner-Distance*, IEEE. PAMI, 29 (2007), pp. 286–299.

[31] L. J. Latecki, R. Lakämper and U. Eckhardt, *Shape Descriptors for Non-rigid Shapes with a Single Closed Contour*, CVPR, 2000.

[32] N. Chawla, A. Lazarevic, L. Hall and K. Bowyer, *SMOTEBoost: Improving the Prediction of Minority Class in Boosting*, PKDD, 2003.

[33] F. Provost and T. Fawcett, *Robust Classification for Imprecise Environments*, Machine Learning, 42 (2001), pp. 203–231.

[34] S. D. Bay and M. Schwabacher. *Mining Distance-Based*

Outliers in Near Linear Time with Randomization and a Simple Pruning Rule, KDD, 2003.

[35] A. Ghoting, S. Parthasarathy and M. E. Otey *Fast mining of distance-based outliers in high-dimensional datasets*, Data Mining and Knowledge Discovery, 16 (2008), pp. 349-364.

[36] F. Angiulli and F. Fassetti. *Very efficient mining of disance-based outliers*, Proc. of 16th ACM Conf. on Information and Knowledge Management, 2007.

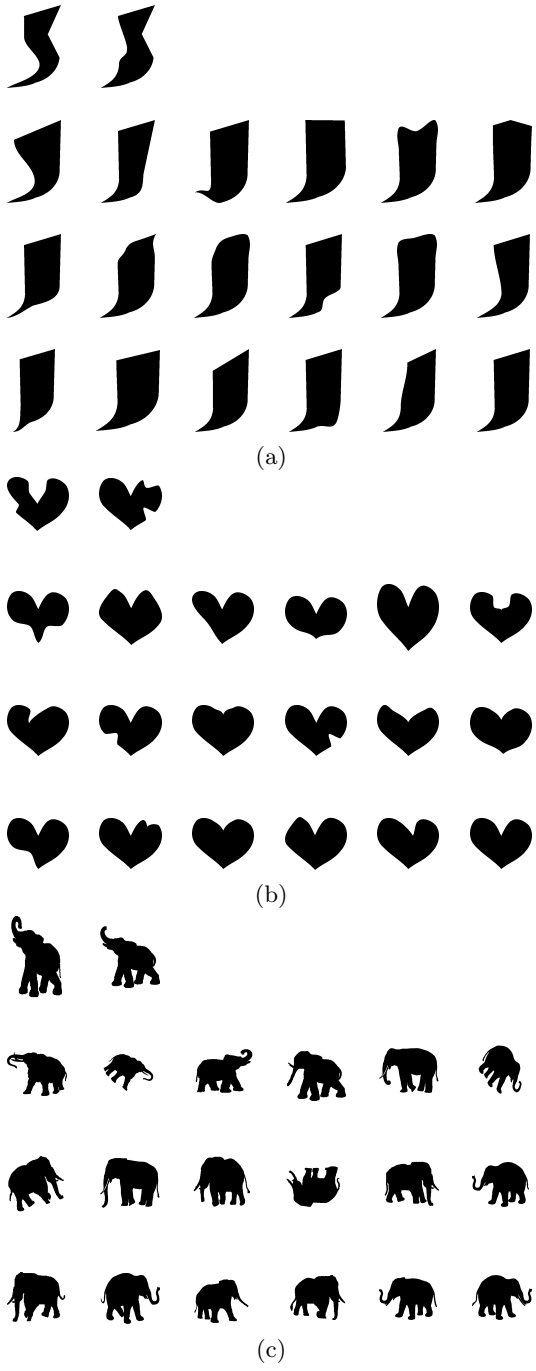

Figure 7: The unusual shape detection results on three example shape classes of the MPEG-7 data set. The two most unusual shapes for each class are shown in top rows.