Finding Outliers at Multiple Scales

Tianming Hu*and Sam Yuan Sung
Department of Computer Science, National University of Singapore
Singapore 117543

Abstract: Outlier detection targets those exceptional data whose pattern is rare and lie in low density regions. In this paper, under the assumption of complete spatial randomness inside clusters, we propose an MDV (Multi-scale Deviation of the Volume) approach to identifying outliers. In addition to assigning an outlier score for each object, it directly outputs a crisp outlier set. It also offers a plot showing the data structure in every object's vicinity, which is useful in explaining why it may be outlying. Finally, the effectiveness of MDV is demonstrated with both artificial and real datasets.

Keywords: outlier detection, clustering, complete spatial randomness, knowledge discovery

1 Introduction

In contrast to clustering that aims to find general pattern for the majority of data, outlier detection targets the finding of the rare data whose behavior is very exceptional compared to other data. A well-known definition of outlier was given by Hawkins [1] who defined it as an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism. Similar definition also appeared in Barnett and Lewis's book [2] which stated an outlier is an observation or subset of observation which appears to be inconsistent with the remainder of that set of data.

Although outliers are often treated as noise or error in many operations, such as clustering, and discarded, they may have potential causes and bear useful information that cannot be mined from other data that reside deeply inside clusters. It is not unusual that one man's noise is another one's signal. After identifying those outliers, we may go further to study the underlying reason why they happen and these knowledge may be profitable. For instance, outliers may be produced by an incorrect assumption of distribution. In such situations, further investigation for outliers can lead to a more appropriate statistical model, which, in turn, leads to a more appropriate statistical inference. So in a way, finding outliers is at least as important as finding clustering structure. Outlier detection has already found application including deviation detection in large databases [3], discovering network intrusion [4, 5], detecting cellular fraud [6], etc. There are many similar problems in other fields. For instance, in association rule mining, an outlier is an interesting rule and the outlier factor is the interestingness. The rule's interestingness can be measured in terms of its unexpectedness, i.e., how much it changes the current belief of the whole system of all mined rules so far [7, 8].

1.1 Problem Formulation

Now we give the formal formulation of outlier detection problem. Given a dataset partitioned as outliers and non-outliers, the problem of detecting outliers is essentially an unsupervised binary classification.

- Given: A dataset $X = \{\mathbf{x}_i \in \Re^d\}_{i=1}^n$, i.e., each object is represented as a point in d-D(dimensional) space. Associated with each object, there is a class label $\omega(\mathbf{x}_i) \in \{\omega_o(\text{outlier}), \omega_n(\text{non-outlier})\}$. Class labels are unknown to the learner.
- Find: A mapping function $f : X \to \Re^+$, i.e., f maps each object to a non-negative value called outlier factor describing the degree of outlierness. An alternative is just to output a subset $O \subset X$ regarded as outliers.

*Corresponding author. Email: hutianmi@comp.nus.edu.sg

- Objective: $\forall \mathbf{x}_i, \mathbf{x}_j, \omega(\mathbf{x}_i) = \omega_o \bigwedge \omega(\mathbf{x}_j) = \omega_n \Rightarrow f(\mathbf{x}_i) > f(\mathbf{x}_j)$, i.e., any outlier's factor must be greater than all non-outliers' factors. For the alternative above, the objective is to maximize the similarity between the predicted outlier set O and the true outlier set $\{\mathbf{x} : \omega(\mathbf{x}) = \omega_o\}$.

1.2 Our Contribution

In this paper, we present a novel outlier detection approach called MDV (Multi-scale Deviation of Volume). Our approach provides the following advantages over other methods.

- Most methods output a soft set of pairs, i.e., they assign for each data point a value called outlier factor, describing how outlying it is. The user needs to determine a threshold to divide the dataset into two crisp sets, outliers and non-outliers. This threshold is difficult to determine. Like other methods, our approach does provide such a soft set, but it is only for experts who wish to probe for more information. Our approach automatically outputs a crisp set of predicted outliers. This feature is especially useful to most common users.

- On the input part, most methods require the user to input some parameters such as size or diameter of neighborhood, on which their performance depends a lot. However, such parameters are generally hard to determine, considering the final user is the domain expert, not the outlier detection expert. Our approach also has a few parameters but they have default values to make usage easier.

- As a by-product, our approach offers information about the data in the vicinity of outliers. For instance, in contrast to single outlier that lies far away from the rest of the data, several outliers may lie together and hence form a micro-cluster. Our approach can tell the size of such clusters and the distance to the nearby big clusters of non-outliers.

The rest of the paper is organized as follows. Related work is reviewed in Section 2. In Section 3, after introducing complete spatial randomness, which is assumed to be the data structure inside clusters, we propose our MDV approach to detecting outliers at multiple scales. Section 4 gives experimental evaluation of MDV, in comparison with local outlier factor approach, on both synthetic and real datasets. Section 5 concludes this paper with a summary and some discussion on future work.

2 Related Work

Most outlier detection techniques handle outliers where all attributes of the object are treated equally, i.e., each object with d continuous attributes is regarded as a point in $\Re^d$. In the rest of this paper, we will sometimes use word like object, data point and event interchangeably, provided no ambiguity occurs. Generally speaking, outlier detection techniques can be divided into the following categories: distribution-based, depth-based, distance-based density-based, etc.

Distribution-based methods often handle one dimensional data and are mainly developed in the statistical field [2]. They assume a statistical distribution such as Gaussian and try to fit the data to the model by estimating the parameters such as mean and variance from the data. They vary in terms of type of distribution, number of outliers to be identified and type of outliers. Then they employ a test based on the distribution property to identify outliers w.r.t. this distribution. In reality, prior knowledge about the distribution of the dataset is not always available. Furthermore, it is hard to justify model selection in advance, e.g., Gaussian over exponential.

Depth-based approaches [9, 10] employ computational geometry to compute different layers of convex hulls and declare those objects in the outer layer as outliers. However, they suffer from the dimensionality curse and cannot cope with large dimension [11].

The remaining two categories are capable of dealing with multi-dimensional data and are mainly developed in the database community recently. These techniques are closely related to the corresponding clustering algorithms that try to find the general pattern followed by the majority of points. In fact, given a clustering algorithm with a function to measure its clustering quality, a naive algorithm for calculating

outlier factor can assign each point a value which equals the absolute difference between the original clustering quality and the new clustering quality after removing that point.

Distance-based techniques distinguish points which are likely to be outliers from others based on the number of points in their neighborhood. They do not assume any prior distribution of the data and limit the counting of points to the neighborhood of each point. Corresponding to clustering algorithms that find convex clusters [12, 13], one such technique is the well-known DB(p, d)-outlier [14], where a point in a dataset T is an outlier if at least p fraction of points in T lie greater than distance d from it. A special case of DB(p, d)-outlier is proposed in [15], where the distance to the k-th nearest neighbor is used to rank the outlierness. The strength of this definition includes simplicity and capture of the basic meaning of Hawkins' definition. However, it cannot handle data with different local densities and hence can only find global outliers. Besides, they need the user to input parameters like p, d, k, which are hard to determine beforehand.

Because we mainly compare our approach against local outlier factor approach, we introduce it here in some detail. Density-based approaches focus on the local density comparison only with the immediate neighbors. Corresponding to clustering algorithms capable of finding arbitrary shape clusters consisting of points with similar local densities [16, 17], the notion of local outlier factor (LOF) is proposed in [18], which measures the degree of outlierness, based on the difference in the local density of a point and its k nearest neighbors. Comparatively, DB(p, d)-outlier cannot detect local outliers w.r.t. a neighboring dense cluster in the presence of another very sparse cluster. LOF solves this problem by thinking locally, i.e., comparing local density of the outlier only with those of its neighboring points. If points inside the cluster are approximately uniformly distributed, their LOF will be close to one. For an outlier lies away from the cluster, its LOF will be higher than one. The weakness of LOF is that it still needs input parameter k, on which its performance depends a lot. Suppose there is a micro-cluster of outliers, if k is chosen to be less than the cluster size, then their outlier factors will also be close to one and it is impossible for LOF to identify them.

3 Outlier Detection with Multi-scale Deviation of Volume

3.1 Complete Spatial Randomness

Complete spatial randomness (csr) [19] refers to a lack of structure (pattern) in the spatial point process, where events (points regarded as a realization of events) are uniformly distributed in the study region $A \subset \Re^d$. For any sub-region $B \subset A$, the probability that there is at least one event within it is equal to the ratio of its volume over the total volume, i.e., $|B|/|A|$, where $|\cdot|$ denotes volume. This probability is independent from B's location and shape. This kind of spatial point process is also called a homogeneous Poisson process because $N(B)$, the number of events in B, follows a Poisson distribution with mean $\lambda|B|$, where λ denotes the constant intensity in the study region.

Let V_j denote the random variable of the hyper-sphere volume centered at a randomly chosen point in $B \subset \Re^d$ with radius R_j, the distance to its j-th nearest neighboring object. Note that it does not matter whether or not there is an event (object) happening at that point location. By assuming the distribution of the objects follows csr (homogeneous Poisson process) with constant intensity λ, we observe that the random variable V_j actually follows a gamma distribution with parameter (j, λ). If the randomly chosen point above is replaced by a randomly chosen object, the distribution of the corresponding random variable is the same, with its expectation and variance in Eq. (1).

$$E(V_j) = \frac{j}{\lambda}, \ \mathrm{Var}(V_j) = \frac{j}{\lambda^2} \tag{1}$$

3.2 MDV Approach

Whenever we devise an approach to detecting local outliers defined as objects different from their neighbors, there are two basic questions we need to answer. Let $d(x, y)$ and $r_k(x)$ denote the distance between two points x and y, and x's distance to its k-th nearest neighbor. The first question is, for each data point

x, how to choose neighbors $N_k(x) = \{y : d(x, y) \leq r_k(x)\}$ (hence $x \in N_k(x)$), i.e., how to choose k, the sampling neighborhood size. The other is how to measure difference. Different answers determine the essence of different algorithms. Our approach is determined by the following answers.

As for size of the sampling neighborhood against which we compare each object, most approaches leave it to the user. However, a data point which is outlying (in a low density region) at one scale may not be outlying at another scale. So our reply is to choose a wide range of scales between $k_{\min}$ and $k_{\max}$. $k_{\min}$ cannot be too small to avoid statistical error. In the meantime, we need to ensure that for each point, especially those forming micro-cluster of outliers, it does try a neighborhood of size $k \leq k_{\max}$, which is large enough to include much more non-outliers than outliers.

As for the difference, we still try to discover those points with lower densities. In detail, we employ normalized deviation of sphere volume V about the mean, which is also called trimmed z-score. For a set of data $\{x\}$, the original form of z-score is defined as $z(x) \equiv (x - \hat{\mu})/\hat{\sigma}$, where $\hat{\mu}, \hat{\sigma}$ are sample mean and sample standard deviation. A popular test labels x outlier if $|z(x)| > 3$ and it is especially useful for Gaussian distributed data. For $x \sim N(\mu, \sigma^2)$, $(x - \mu)/\sigma \sim N(0, 1)$. If we assume true mean (deviation) can be approximated with sample mean (deviation), then $P(z(x) > 3) = 0.0013$. Our difference measure is based on the observation that, if at a particular sampling scale k, a data point is outlying in a low density region while most of its k neighbors lie in a nearby high density cluster, then its distance to its j-th ($j(k)$ treated as a function of k) nearest neighbor R_j must be larger than most of those k neighbors'. Consequently, its volume V_j must also be much larger. If j is relatively large, gamma distribution can be approximated by Gaussian distribution. Hence we can try z-score of sphere volume at various scale k and output the maximum as the final outlier factor. In detail, for each data point x_i, compute $z(x_i) \equiv (V_j(x_i) - \overline{V_j}(x_i))/\sqrt{S_j^2(x_i)}$, where $\overline{V_j}(x_i) \equiv \sum_{x \in N_k(x_i)} V_j(x)/(k+1), S_j^2(x_i) \equiv \sum_{x \in N_k(x_i)} (V_j(x) - \overline{V_j}(x))^2/k$. Besides, we identify those data with greater than 3 score to form a predicted outlier set.

Assuming those data points in $N_k(x_i)$ are uniformly distributed in the same cluster with intensity λ and hence their volumes V_j are independently and identically distributed (iid) gamma random variables, we can see that $\mathrm{Var}(\overline{V_j}) = j/(\lambda^2(k+1))$. For large j, if we approximate gamma distribution with Gaussian distribution, then $k\lambda^2 S_j^2/j$ is a chi-squared random variable with k degrees of freedom [20]. Therefore, $\mathrm{Var}(S_j^2) \approx 2j^2/(k\lambda^4)$. For accurate estimation of true mean and variance, their estimators' variances are preferred small, which means j cannot be too large. On the other hand, large sampling neighborhood size k is always desirable to give low bias estimates.

The above inference is based on the assumption that all volumes are iid, but it no longer holds in the presence of outliers. For robust estimation to accommodate outliers, we employ trimmed mean to estimate true mean and variance and that is why we call the resulting score trimmed z-score. Trimmed mean has been proven to be more efficient in estimating sample location than sample mean in the presence of outliers [21]. In detail, for each point x_i's k-th neighborhood, the set of $k+1$ volume values (the subscript j is omitted) $\{v(x) : x \in N_k(x_i)\}$ is first sorted in ascending order $\{v^1 \leq v^2 \leq, ..., \leq v^{k+1}\}$. Let $r = \lfloor pn \rfloor$, then the $100p\%$ trimmed mean, defined in Eq. (2), is just the sample mean over the remaining middle $k + 1 - 2r$ values after excluding the highest and lowest r values. Similarly, the corresponding trimmed variance and z-score are given in Eq. (3, 4).

$$\overline{V}(x_i, p) \equiv \frac{\sum_{i=r+1}^{k+1-r} v^i}{k+1-2r} \tag{2}$$

$$S^2(x_i, p) \equiv \frac{\sum_{i=r+1}^{k+1-r} (v^i - \overline{V}(x_i, p))^2}{k-2r} \tag{3}$$

$$z(x_i, p) \equiv \frac{V(x_i) - \overline{V}(x_i, p)}{\sqrt{S^2(x_i, p)}} \tag{4}$$

Now we give our MDV approach as follows with some parameters.

1. Initialize $k := k_{\min}$. Array $mdv[1, ..., n]$ is used to store the current outlier scores (maximum so far) and is initialized to $-\infty$, negative infinity.

2. For each data point $x_i, i = 1, ..., n$, retrieve its k-th neighborhood $N_k(x_i)$. For each data point $x \in N_k(x_i)$, compute $v_j(x) = \pi^{d/2} r_j(x)^d / \Gamma(1 + d/2)$, the volume of hyper-sphere in d-D with radius $r_j(x)$, the distance to its j-th nearest neighbor. Compute the trimmed z-score $z(x_i, p)$ for x_i. If $z(x_i, p) > mdv[i]$, then $mdv[i] := z(x_i, p)$.

3. $k := k + \Delta k$. If $k > k_{\max}$, stop, otherwise go back to step 2.

MDV outputs array $mdv[1, ..., n]$ that contains the final outlier factors for all data. Besides, those data with greater than 3 score are also output as a crisp outlier set denoted by MDV3. MDV has a few parameters but they have default values to make usage easier. Empirical studies suggest the following values: (1) $p = 0.15$, a commonly recommended value in trimmed mean. (2) The radius $j = k/2$, for certain tradeoff between low variance of estimators and locality in density. (3) $\Delta k = 2$, $k_{\min} = 20, k_{\max} = \lfloor n/2 \rfloor - (\lfloor n/2 \rfloor \bmod 2)$ (make $k_{\max}$ even and hence $j = k/2$ integer).

4 Experimental Evaluation

We use precision and recall [22] to evaluate the effectiveness of MDV in comparison with LOF. In detail, let O and P denote the true outlier set and predicted outlier set respectively, then precision $\equiv |O \cap P|/|P|$, recall $\equiv |O \cap P|/|O|$. Because both methods output an outlier score for each object, we can select those with top p scores for $P(p)$ and observe their performance by varying p.

4.1 Synthetic Data

A dataset is illustrated in Fig. 1(a), consisting of a big cluster, a micro-cluster of outliers of size 15 and a single outlier. Data are uniformly distributed inside both clusters. By thresholding the MDV outlier score with 3, the precision and recall are 0.59 and 1. There are two main reasons why we cannot achieve high precision. One is the inherent randomness of data distribution inside clusters. The other is that the j-th (especially for large j) nearest neighbor distances of points on the cluster border are generally larger than those inside the cluster. This is also called edge effect, which must be overcome for robust clustering.

Fig. 1(b) gives the single outlier's volume V at all scales k, together with the corresponding trimmed mean μ and deviation σ of its neighborhood. Consistently its $V > \mu + 3\sigma$, which means its trimmed z-score is much larger than 3. This indicates it is a single outlier that lies far away from all other data. Fig. 1(c) illustrates the plot for an outlier (denoted as x) in the micro-cluster. Before the sampling size $k = 30$(volume radius $j = k/2 = 15$), although its sampling neighborhood includes some points not belonging to the micro-cluster, those points are mostly trimmed off in computation of mean and deviation. The points involved in the computation are mainly from the same micro-cluster, together with their j-th nearest neighbors, the distance to which are used to compute the volume. So their Vs still follow the gamma distribution and $V \approx \mu$. At $k = 30$, because x's $j = 15$-th nearest neighbor is no longer in the same micro-cluster but lies far away in another big cluster, there is an abrupt increase in volume, mean and deviation. This tells us that the micro-cluster's size is 15 and the distance to the nearby cluster is approximately x's distance to its 15-th nearest neighbor, provided the micro-cluster's diameter is small enough to be ignored. As k grows, more points from the big cluster join the sampling neighborhood. As a result, more points from the micro-cluster are trimmed off and finally x's volume is greater than $\mu + 3\sigma$. Remember that we choose $p = 0.15$ trimmed mean and it explains why σ stops sharply decreasing and begins smoothly increasing again around $k = 100$, when all volumes of points from the micro-cluster become the highest 15% values and hence all get trimmed off. Consequently, the computation of mean and deviation only involves the points in the big cluster. Finally, Fig. 1(d) shows the plot for a point in the big cluster. As expected, $V \approx \mu$. μ is approximately linear in k, since $\mu \approx j/\lambda = k/(2\lambda)$.

Because our assumption is uniform distribution inside clusters, a natural question arises how it performs when the assumption fails, e.g., Gaussian cluster. For this purpose, we randomly draw a sample from $N(0, 1)$ of size 120. Five values are drawn from $N(0, 5^2)$ as true outliers. We choose $k = 10$ for LOF so that non-outliers will make the majority among 10 nearest neighbors of outliers. Since both MDV and LOF output outlier factors, we select some top p (around 5) outliers. Note that in practice, we do not know the number of true outliers. This experiment is repeated 100 times and the average precision and

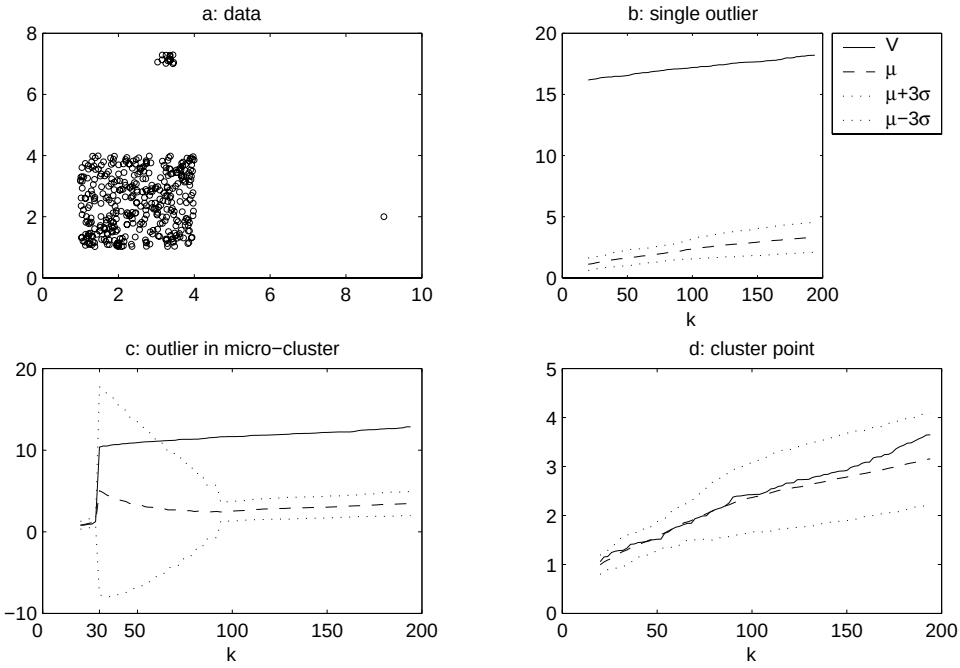

Figure 1: (a) shows a dataset with multiple types of outliers. The volume and neighborhood's mean and deviation are illustrated for three types of data, single outlier (b), outlier from the micro-cluster (c), and point from the big cluster (d).

Table 1: MDV vs LOF on Gaussian cluster.

p	5.08	5		6		7		8	
	MDV3	MDV	LOF	MDV	LOF	MDV	LOF	MDV	LOF
precision	0.59	0.55	0.56	0.50	0.48	0.43	0.43	0.39	0.38
recall	0.60	0.55	0.56	0.60	0.58	0.60	0.60	0.62	0.61

recall is shown in Table 1, together with the results of MDV3. We can see that at all p, MDV gives a comparable result to LOF. The average output size by MDV3 is 5.08, which is very close to 5, the number of true outliers. Besides, its precision is significantly higher than MDV and LOF.

4.2 Real Data

As mentioned in [23], in practice the performance of outlier detection approaches can be evaluated based on real data to recover data from rare classes. We choose from the UCI repository [24] two datasets, ionosphere and Wisconsin diagnostic breast cancer, both of which have two classes. All data from the majority class are treated as non-outliers. Then we randomly draw a few data from the minority class as outliers such that they make 10% of the resulting final dataset.

In ionosphere data, instances are described by the complex values (two attributes for each of 17 pulse numbers), returned by the autocorrelation function whose arguments are the time of a pulse and the pulse number, resulting from the complex electromagnetic signal. The targets are good (majority) or bad (minority), where good radar returns are those showing evidence of some type of structure in the ionosphere, and bad returns are those that do not. For LOF, we try $k = 5, 10, 20$ and get the best result at 10. The results over top $100p\%$ outliers is shown in Figs. 2(a,b), which shows that MDV is comparable to LOF at all p. In particular, we concentrate more on small p(e.g., $p < 0.5$), because it is the common practice in reality that we usually select some top predicted outliers for further investigation. Furthermore, the smaller p, the more important the results. Detailed results are given in Table 2. As for MDV3, its precision and recall are 0.55 and 0.84. Its output constitutes 14% of the total dataset, a fraction close to

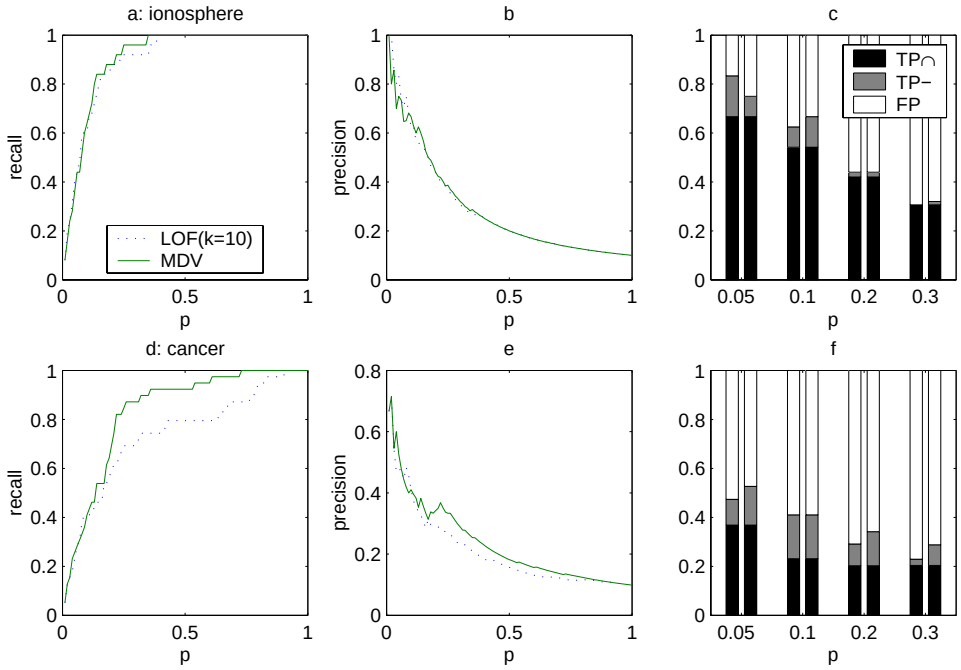

Figure 2: (a) and (b) depict the recall and precision on ionosphere data, respectively. (c) shows the composition of prediction by LOF (left bar) and MDV (right bar). $TP\cap$ denotes intersection of true positive. $TP-$ denotes difference of true positive. FP denotes false positive. The corresponding values for cancer data are given in the second row.

Table 2: MDV vs LOF on real data. $^+$ denotes the fraction output by MDV3.

	recall						precision					
iono: p	0.1	0.14^+	0.2	0.3	0.4	0.5	0.1	0.14^+	0.2	0.3	0.4	0.5
MDV	0.64	0.84	0.88	0.96	1	1	0.67	0.54	0.44	0.32	0.25	0.2
LOF	0.60	0.76	0.88	0.92	1	1	0.63	0.55	0.44	0.31	0.25	0.2
canc: p	0.07^+	0.1	0.2	0.3	0.4	0.5	0.07^+	0.1	0.2	0.3	0.4	0.5
MDV	0.31	0.41	0.69	0.87	0.92	0.92	0.46	0.41	0.34	0.29	0.23	0.18
LOF	0.31	0.41	0.59	0.69	0.74	0.79	0.44	0.41	0.29	0.23	0.18	0.16

10%.

To further compare the prediction by MDV and LOF, we divide the prediction set P into three subsets: intersection of true positive ($P \cap O$) between LOF and MDV, difference of true positive, and false positive ($P - O$). The fractions of these subsets are given in Figs. 2(c). Both methods share a lot in true positive at small p. As p increases to 0.3, however, all true outliers predicted by LOF are recovered by MDV.

In cancer data, 30 real-valued features are computed from a digitized image of a fine needle aspirate of a breast mass. They describe characteristics of the cell nuclei present in the image to distinguish malignant (minority) from benign (majority) breast cancer. LOF is still better at $k = 10$ than at $k = 5, 20$. The results are shown in Fig. 2(d-e). This dataset does not fit our outlier assumption (outlier from rare class lie in a low density region at some scale) as good as the ionosphere data, since the recall is generally lower and achieves 100% at a much larger p. Nevertheless, the lead of MDV over LOF seems more obvious. Detailed values are also given in Table 2, where 7% (close to 10% again) data are chosen by MDV3 with precision 0.46 and recall 0.31. Fig. 2(f) illustrates the composition of the prediction sets. At nearly all p, MDV has a larger fraction ($TP-$) of correctly predicted outliers that LOF fails to detect, which explains why MDV's precision is almost consistently higher than LOF's.

5 Concluding Remarks

In this paper, under the assumption of complete spatial randomness inside clusters, we proposed an MDV approach to identifying outliers, using multi-scale deviation of the volume of the hyper-sphere centered at each data point. In addition to assigning an outlier score for each data point, it directly outputs a crisp outlier set. Compared to other approaches, its strength is that it has default values for parameters, which are hard to determine for average users but are critical for performance. For each data point, MDV also provides a plot showing the information about the data in its vicinity, which is useful in explaining why some data are outlying. Finally, the effectiveness of MDV was demonstrated with both artificial and real datasets, which showed MDV is at least comparable to LOF in terms of precision and recall of true outliers.

As for time complexity, LOF takes $O(n(k\text{NN} + k))$ time, where kNN denotes the time for a k nearest neighbors query. Without any optimization or approximation, MDV's complexity is much larger. To exhaust all scales, we may need the huge $n \times n$ proximity matrix that stores sorted distances for every pair of data. Its construction costs $O(n(n + \log n))$. Then the worst case complexity is $O(n^2 n_k)$ to compute deviation about the trimmed mean at n_k scales. Obviously such complexity is infeasible for large datasets. To improve efficiency, it is worth studying how to automatically determine the optimal sampling size k and volume radius j. Ideally we should sample at some finite n_k scales that is independent of n, which may be achieved by dividing data into cells. We can exploit the property that nearby data points share a lot of nearest neighbors, which causes much overlap between computation of their trimmed means. Besides, we may not need to compute deviation for all data or at all scales, if the users only want the MDV3 output or, say, we can infer that outlier scores at subsequent larger scales cannot be greater than the current candidate. In a word, much work can be done to improve MDV.

Acknowledgements
Special thanks to the referees for their helpful comments.

References

[1] D. M. Hawkins. *Identification of Outliers*. Chapman and Hall, 1980.

[2] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley & Sons, 3 edition, 1994.

[3] A. Arning, R. Agrawal, and P. Raghavan. A linear method for deviation detection in large databases. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 164–169, 1996.

[4] W. Dumouchel and M. Schonlau. A fast computer intrusion detection algorithm based on hypothesis testing of command transition probabilities. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 189–193, 1998.

[5] W. Lee, S. J. Stolfo, and K. W. Mok. Mining in a data-flow environment: Experiences in network intrusion detection. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 114–124, 1999.

[6] T. Fawcett and F. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery Journal*, 1(3):291–316, 1997.

[7] B. Liu, W. Hsu, L. Mun, and H. Lee. Finding interesting patterns using user expectations. *IEEE Transactions on Knowledge and Data Engineering*, 11(6):817–832, 1999.

[8] A. Silberschatz and A. Tuchilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):970–974, 1996.

[9] I. Ruts and P. Rousseeuw. Computing depth contours of bivariate point clouds. *Computational Statistics and Data Analysis*, 23:153–168, 1996.

[10] T. Johnson, I. Kwok, and R.T. Ng. Fast computation of 2-dimensional depth contours. In *Proceedings of the 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 224–228, 1998.

[11] H. Edelsbrunner. *Algorithms in Computational Geometry*. Springer-Verlag, 1987.

[12] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[13] R. Ng and J. Han. CLARANS: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, 14(5):1003–1016, 2002.

[14] E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: Algorithms and applications. *The Very Large Data Bases Journal*, 8(3):237–253, 2000.

[15] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 427–438, 2000.

[16] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[17] A. Hinneburg and D. A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 58–65, 1998.

[18] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 93–104, 2000.

[19] N. A. Cressie. *Statistics for Spatial Data*. Wiley, 1993.

[20] S. Ross. *A First Course in Probability*. Prentice Hall, 5 edition, 1998.

[21] B. Iglewicz and D. C. Hoaglin. *How to Detect and Handle Outliers*, volume 16. Quality Press, 1993.

[22] G. Salton and M. Mcgill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[23] C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, pages 37–46, 2001.

[24] P. M. Murphy and D. W. Aha. *UCI Repository of Machine Learning Databases*. Department of Information and Computer Science, University of California at Irvine, 1994. http://www.ics.uci.edu/ mlearn/MLRepository.html.