Outlier Detection via Sampling Ensemble

Hongfu Liu[1], Yuchao Zhang[2], Bo Deng[2] and Yun Fu[1]
[1]*Northeastern University, Boston* [2]*Beijing Institute of System Engineering, Beijing*
liu.hongf@husky.neu.edu, dragonzyc@163.com, dengbo@gmail.com, yunfu@ece.neu.edu

***Abstract*—Outlier detection is a key technique in data ming and machine learning fields. The deviating characters of outliers make huge detrimental effects on the learning tasks. A lot of algorithms are therefore proposed to handle outliers from different perspectives, such as distance, density, angle and so on. Among these approaches, the density-based methods achieve better performance, but also suffer from huge time complexity. Recently, in order to accelerate the speed and improve the performance, the subsampling ensemble method attracts much attention, which has a reasonable theoretical interpretation and high performance. However, existing work only gives the partial picture of outlier detection via row-sampling, the effective portfolio of bi-sampling is still void. In light of this, we propose the general outlier detection framework via bi-sampling, Bi-Sampling Outlier Detection (BSOD) and provide the effective portfolios of the row and column-sampling ratios in a theoretical way. In addition, the benefits of BSOD are fully illustrated in terms of ensemble diversity and divide-and-conquer. Further we employ LOF within BSOD as BI-LOF to conduct extensive experiments. In general, on 30 synthetic and 17 real-world data sets we thoroughly explore the characteristics of BI-LOF with different numbers of instances, features, nearest neighbors, validate the theoretical analysis of BSOD condition on synthetic data sets, and show obvious advantages over other state-of-the-art algorithms in terms of low and high dimensional real-world data sets. And finally we use BI-LOF to conduct image outlier detection and show high quality and stableness of BI-LOF.**

***Keywords*-Outlier detection, Bi-sampling, Ensemble**

I. Introduction

Outlier detection or anomaly detection is a hot topic in data mining and machine learning areas [1, 13, 44], especially in the era of big data. Robustness analysis plays crucial roles for real-world applications [30–32], such as credit card fraud identification, network intrusion detection, valuable user mining. In these cases, we are more interested in outliers for their higher values. In other cases, outliers generate from noise disturbance or equipment fault, which should be removed first since the deviating characters of outliers make huge detrimental effects on the learning tasks.

To handle outliers, three strategies are proposed. (1) Some methods are designed to be naturally robust to outliers. These methods focus on specific tasks, such as image reconstruction [24, 34], robust clustering [11]. (2) Some methods apply outlier correction to modify outliers to good training instances [14, 16]. (3) Some methods remove outliers first then conduct the learning tasks [4, 12, 38–40]. The first two kinds belong to application-related or task-driven methods, and recent years have witnessed various outlier detection approaches of the third kind, including statistic-based methods [2], cluster-based methods [10, 11], density-based methods [9, 37] and angle-based methods [21, 36]. Among these methods, density-based methods attract a lot of increasing attention. Usually this kind of methods assigns a score to each instance and ranks the score for top K outlier candidates. Especially the methods from local perspective can handle multi-density data sets and identify the outliers located between different clusters. Local Outlier Factor (LOF) [8] is one of classic local outlier detection methods, which calculates the local density via averaging the density of its neighbors. Along this line, variants of the local outlier model include LoOP [19], LOCI [35], LDOF [41]. Although local outlier detection methods outperform other methods in terms of accuracy, the high time and space complexity precludes themselves to handle large-scale data sets.

Some scholars also apply ensemble method to improve the performance of outlier detection. With the success of ensemble methods [26–29], Lazarevic and Kumar [22] leveraged feature bagging to generate diversity sub data sets, conducted outlier detection on each sub data set and combines these scores via breadth-first approach. Further, Zimek et al. [43] used instance sampling method to detect outliers and first uncovered the theoretical foundation why the ensemble learning is conductive to outlier detection. However, such interpretation is misleading; as reported in [3], row-sampling can only increase the distance between inliers and outliers by the same rate, but it fails to enlarge the rank between them. Besides, all these existing studies only give the partial picture by row-sampling or column-sampling and the more general effective portfolio of bi-sampling is missing.

In order to accelerate the speed of local outlier methods and derive the theoretical analysis of bi-sampling for outlier detection, we propose the general outlier detection framework via bi-sampling, Bi-Sampling Outlier Detection (BSOD). First, dimension reduction is applied via column-sampling, then some instances are selected via row-sampling to build the neighborhood set. Next we build the distance matrix to find nearest neighbors. Different from traditional nearest neighbors which finds neighbors in all data except itself, here we only find neighbors in the selected instances. Next we apply LOF as the core outlier detection algorithm within BSOD and call it BI-LOF to assign a score to each instance. We repeat the above process several times to obtain

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

the set of basic results and then fuse these basic results into the final one. Based on the framework, we conduct a theoretical analysis of BSOD, derive the condition of BSOD and showcase the effective portfolios of row sampling and column sampling ratios for outlier detection. In addition, we find that column-sampling falls into infeasible area of BSOD condition so that it suffers from bad performance. Through extensive experiments, we systematically explore the impact factors of BI-LOF on 30 synthetic data sets such as the number of instances, features and nearest neighbors and row-and-column sampling ratios, validate the theoretical conclusion of BSOD condition, and demonstrate BI-LOF can generate competitive results with high efficiency compared to the state-of-the-art outlier detection algorithms on 17 real-world data sets. It is worth noting that (1) column-sampling suffers from bad performance on low dimensional data sets, however, it performs high results on high dimensional data sets, (2) for row-sampling, on the contrary, it achieves high quality results on low dimensional data sets and becomes struggled on high dimensional data sets, (3) our method keeps consistently high performance on both low and high dimensional data sets. Moreover, we also validate our framework in image outlier detection domain to illustrate the effectiveness and stableness of BI-LOF. Our contributions are highlighted in the following aspects:

- We propose Bi-Sampling Outlier Detection framework and derive the condition of BSOD in a theoretical way to illustrate the effective portfolios of row and column-sampling ratios to enlarge the gap between outliers and inliers.
- Benefits of BSOD are fully analysed in terms of ensemble diversity and divide-and-conquer. BSOD not only gains more meaningful density but also improves the performance with less time and space cost as well. At the same time, large-scale and high dimensional data sets can be decomposed into several small sub data sets which can be handled in a separate and independent way.
- Experimental results show BI-LOF is very effective and efficient compared to other state-of-the-art outlier detection methods even with only 10% row and column-sampling ratios and 10 ensemble members. This indicates that we can use less time and space resources to achieve superior performance compared to other outlier detection methods.

The rest of the paper is organized as follows. We discuss related work in Section II and derive the condition of BSOD and analyse the benefits of BSOD in Section III. Experimental results on both synthetic and real-world data sets are provided in Section IV, then we conclude the paper in Section V.

II. Related Work

Generally speaking, outlier detection can be roughly generalized into three categories due to the availability of labels, supervised learning, semi-supervised learning and unsupervised learning. In supervised learning and semi-supervised learning, outlier detection is formulated into a classification problem. However, in most cases, we have to resort to unsupervised outlier detection due to lack of label information. Therefore we focus on unsupervised outlier detection in the following.

In unsupervised outlier detection, we calculate a score for each point and rank the scores or set a threshold for finding top K outlier candidates. Many algorithms have been proposed to measure the similarity among instances. The distance-based notion of outliers is the first database-oriented approach in the area of unsupervised outlier detection [17]. Along this line, model-based methods [2], density-based methods [9, 37], angle-based methods [21, 36], cluster-based methods [10, 11] are included to enrich this kind study. These methods are also known as $global$ methods in which the computed scores use the information of all other instances. Another kind of methods is $local$ methods. This kind of methods make uses of nearest neighbors to calculate the score for each instance. The most classic local outlier detection algorithm is LOF [8], which considers local density scores via averaging the density of k-nearest neighbors. Variants of the local outlier model includes LoOP [19], LOCI [35], LDOF [41]. Although these local outlier detection methods achieve better performances compared to global ones, it needs too much computational resources to calculate the similarity or dissimilarity matrix between instances, which precludes themselves to handle large scale data sets. More details can be found in the surveys [13, 44].

Besides, much research has aimed to improve the efficiency of unsupervised outlier detection by approximation or pruning techniques for mining top K outliers [5, 6, 15]. Sampling approach is another way to accelerate. Kollios et al. applied the biased sampling to generate subspace and transfer the problem into a small-size one [18]. Kriegel et al. proposed an axis-based method SOD to project the instances into each attribute and calculated the distance between each instance and the center point [20]. Recent years have witnessed the huge success of ensemble learning [25], especially in the classification and clustering domains. Lazarevic and Kumar combined sampling approach and ensemble method to conduct outlier detection [22]. Several basic results are obtained by employing outlier detection algorithm on the sub data generated by random feature selection and then they fused the basic results into the final one by averaging. Further, Zimek et al. used instance sampling methods to detect outliers and first uncovered the theoretical foundation why the ensemble learning is conductive to outlier detec-

tion [43]. However, such interpretation is misleading; as reported in [3], row-sampling can only increase the distance between inliers and outliers by the same rate, but it fails to enlarge the rank between them. In addition, all existing studies only show the partial picture by row-sampling or column-sampling, the theoretical analysis of bi-sampling outlier detection is heavily needed.

In addition, with the rich data collected from multi-sources, multi-view outlier detection catches increasing attentions, which aims to detect the instances exhibiting different behaviors in different views [23, 33, 42]. Multi-view outlier detection is not the problem we address here; thus, we do not include these studies in this paper.

In order to accelerate the speed of local outlier methods and derive the theoretical analysis of bi-sampling for outlier detection, we propose the general outlier detection framework via bi-sampling BSOD, and provide the condition for choosing effective portfolios of row and column-sampling ratios.

III. Bi-Sampling Outlier Detection Ensemble

In this section, we first give two assumptions of outlier detection based on expectation distance in order to enlarge the gap between outliers and inliers. During the analyses of the condition of BSOD, we cancel one assumption and derive the effectiveness of portfolios for row and column sampling ratios. Then the benefits of BSOD are fully illustrated in ensemble diversity and divide-and-conquer. Finally, we showcase the framework and methods of BSOD.

A. Condition of BSOD

Since outliers are far away from others, we expect to enlarge the gap between inliers and outliers via bi-sampling. To depict the distance, we introduce the definition of Expectation Distance [7].

Definition 1: **Expectation Distance.** Given a point and its sphere of radius r in a d-dimensional Euclidean space, containing n data points uniformly distributed within the sphere, the expected Euclidean distance from the point to its $k-$nearest neighbour $E\{d_k\}$ is given by:

$$E\{d_k\} = r\left(\frac{k}{n}\right)^{\frac{1}{d}}. \tag{1}$$

Based on Eq. 1, we analyse the expectation distance after sampling. Let $\lambda \in (0, 1]$ be the row-sampling ratio, or instance sample ratio, and $\phi \in (0, 1]$ be the column-sampling ratio, or the feature sample ratio. Then we have expectation distance of row-sampling $E_{rs}\{d_k\}$, expectation distance of column-sampling $E_{cs}\{d_k\}$ and expectation distance of bi-sampling $E_{bs}\{d_k\}$ as follows.

$$E_{rs}\{d_k\} = r\left(\frac{k}{\lambda n}\right)^{\frac{1}{d}} \text{ and } E_{cs}\{d_k\} = r\left(\frac{k}{n}\right)^{\frac{1}{\phi d}}. \tag{2}$$

$$E_{bs}\{d_k\} = r\left(\frac{k}{\lambda n}\right)^{\frac{1}{\phi d}}. \tag{3}$$

Based on the definition of $E_{rs}\{d_k\}$ and $E_{cs}\{d_k\}$, then we give the following Theorem 1 to demonstrate the change after row-sampling and column-sampling.

Theorem 1: Expectation distance would increase after row-sampling; on the contrary, expectation distance would decrease after column-sampling.

Proof. Due to

$$\frac{E\{d_k\}}{E_{rs}\{d_k\}} = \frac{r\left(\frac{k}{n}\right)^{\frac{1}{d}}}{r\left(\frac{k}{\lambda n}\right)^{\frac{1}{d}}} = \lambda^{\frac{1}{d}} < 1. \tag{4}$$

Since $0 \leq \lambda \leq 1$, we obtain $E\{d_k\} < E_{rs}\{d_k\}$. Similarly, we have the following equation for column-sampling

$$\frac{E\{d_k\}}{E_{cs}\{d_k\}} = \frac{r\left(\frac{k}{n}\right)^{\frac{1}{d}}}{r\left(\frac{k}{n}\right)^{\frac{1}{\phi d}}} = \left(\frac{k}{n}\right)^{\frac{1}{d}-\frac{1}{\phi d}} > 1. \tag{5}$$

That means $E\{d_k\} > E_{cs}\{d_k\}$ according to $0 \leq \phi \leq 1$. We complete the proof. □

Although it is easy to show that row-sampling increases expectation distance, on the contrary, column-sampling decreases expectation distance, we do not know the change of expectation distance after bi-sampling $E_{bs}\{d_k\}$. Furthermore, Theorem 1 gives the foundation of outlier detection via sampling and leads to the following assumptions to analysis the conditions for BSOD.

Assumptions of outlier detection via sampling: *When applying sampling to detect outliers, it should satisfy one of following conditions in order to enlarge the gap between inliers and outliers:*

- *The expectation distance of outliers should **rise faster** than the one of inliers if expectation distance **goes up**.*
- *The expectation distance of outliers should **drop slower** than the one of inliers if expectation distance **goes down**.*

In the assumptions of outlier detection via sampling, the key part is to measure the change rate of expectation distance. In addition, bi-sampling is a more general case of row-sampling and column-sampling. Thus, we we give the definition of Change Rate of Expectation Distance of bi-sampling in Definition 2.

Definition 2: **Change Rate of Expectation Distance for bi-sampling.** $\Delta E\{d_k\}$ is the change rate of the expectation distance E_{d_k} for bi-sampling by

$$\Delta E\{d_k\} = \frac{E_{bs}\{d_k\}}{E\{d_k\}}. \tag{6}$$

Due to Definition 2, we derive the relationship between inliers and outliers on the change rate of expectation distance for bi-sampling in Theorem 2.

Theorem 2: Given one inlier and one outlier containing n_1 and n_2 points ($n_1 > n_2$) respectively in its corresponding sphere of radius r, we have $\Delta E^{in}\{d_k\} < \Delta E^{out}\{d_k\}$.

Proof. According to Definition 2, we have

$$\begin{aligned}\frac{\Delta E^{in}\{d_k\}}{\Delta E^{out}\{d_k\}} &= \frac{E^{in}_{bs}\{d_k\}}{E^{in}\{d_k\}} \cdot \frac{E^{out}\{d_k\}}{E^{out}_{bs}\{d_k\}} \\ &= \frac{E^{in}_{bs}\{d_k\}}{E^{out}_{bs}\{d_k\}} \cdot \frac{E^{out}\{d_k\}}{E^{in}\{d_k\}} \\ &= \left(\frac{n_2}{n_1}\right)^{\frac{1}{\phi d}} \cdot \left(\frac{n_1}{n_2}\right)^{\frac{1}{d}} = \left(\frac{n_2}{n_1}\right)^{\frac{1}{\phi d}-\frac{1}{d}}.\end{aligned} \tag{7}$$

According to Eq. 7, ϕ is less than 1 and given $n_1 > n_2$, we have $\Delta E^{in}\{d_k\} < \Delta E^{out}\{d_k\}$ for bi-sampling and finish the proof. □

Remark 1: In above analysis, for expectation distance, row-sampling makes it increase, column-sampling drops it and the change is uncertain for bi-sampling. However, for the change rate of expectation distance, the change rate of expectation distance of outliers is always larger than or equal the one of inliers no matter which sampling strategy is used.

Remark 2: In the existing literature, Zimek et al. [43] used row-sampling to detect outliers, which means $0 < \lambda < 1$ and $\phi = 1$, then $\Delta E^{in}\{d_k\} = \Delta E^{out}\{d_k\}$. That is to say, when applying row-sampling, the change of expectation distance of inliers and outliers can be kept in the same rate. Instead we can make the change rate of expectation distance of inliers and outliers different via bi-sampling. It is useful to enlarge the gap between inliers and outliers. Especially in this case, we make the change of outliers faster than the one of inliers. It is worthy to note that the change rate ratio between inliers and outliers has no relationship with the row-sampling ratio λ, instead the column-sampling ratio ϕ is the intrinsic factor to enlarge the gap ratio between inliers and outliers.

Theorem 2 indicates that the change rate of expectation distance of outliers is always larger than the one of inliers via bi-sampling. Therefore, we can cancel the second assumption of BSOD and focus on the first one. We will illustrate the concrete condition in Theorem 3.

Theorem 3: Given bi-sampling increases expectation distance so that $1 < \Delta E^{in}\{d_k\} < \Delta E^{out}\{d_k\}$, the gap between inliers and outlies after bi-sampling will be enhanced.

Proof. According to Eq. 6, we have $E_{bs}\{d_k\} = \Delta E\{d_k\}E\{d_k\}$. Then it follows that

$$\begin{aligned}&E^{out}_{bs}\{d_k\} - E^{in}_{bs}\{d_k\} \\ &= \Delta E^{out}\{d_k\}E^{out}\{d_k\} - \Delta E^{in}\{d_k\}E^{in}\{d_k\} \\ &= \Delta E^{in}(E^{out}\{d_k\} - E^{in}\{d_k\}) + (\Delta E^{out} - \Delta E^{in})E^{out}\{d_k\} \\ &> \Delta E^{in}(E^{out}\{d_k\} - E^{in}\{d_k\}) \\ &> E^{out}\{d_k\} - E^{in}\{d_k\}.\end{aligned} \tag{8}$$

We complete the proof. □

Remark 3: When $1 < \Delta E^{in}\{d_k\} < \Delta E^{out}\{d_k\}$, the gap between outliers and inliers will enlarge; however, under other conditions, like $\Delta E^{in}\{d_k\} < \Delta E^{out}\{d_k\} < 1$ or $\Delta E^{in}\{d_k\} < 1 < \Delta E^{out}\{d_k\}$, the enlarged gap cannot be always guaranteed. Thus, the first assumption is just the sufficient condition of BSOD. From Theorem 3, we cancel one assumption and finally have the following corollary.

Corollary 1: If bi-sampling makes the expectation distance of points increase, the gap between inliers and outliers will be enhanced to distinguish the outliers from inliers.

Therefore, how to make expectation distance of all instances increase via bi-sampling is the core part of BSOD. Theorem 4 gives the guidance of choosing effective portfolios of row-sampling and column-sampling ratios.

Theorem 4: Let the row-sampling ratio $\lambda \in (0, 1]$ and the column-sampling ratio $\phi \in (0, 1]$, if it satisfies that $k^{(1-\phi)} \geq \lambda n^{(1-\phi)}$, then $E_{bs}\{d_k\} \geq E\{d_k\}$ holds, where $E_{bs}\{d_k\}$ and $E\{d_k\}$ are the expectation distance of a point with bi-sampling or not, k is the k-nearest neighbor of the point, n is the number points within the sphere of radius r.

Proof. When $k^{(1-\phi)} \geq \lambda n^{(1-\phi)}$, we have

$$\begin{aligned}&k^{(1-\phi)} \geq \lambda n^{(1-\phi)} \\ &\Rightarrow \frac{1}{\phi d}\ln\left(\left(\frac{k}{n}\right)^{1-\phi} \cdot \frac{1}{\lambda}\right) \geq 0 \\ &\Rightarrow \frac{1}{\phi d}\ln\left(\frac{k}{\lambda n}\right) - \frac{1}{d}\ln\left(\frac{k}{n}\right) \geq 0 \\ &\Rightarrow \ln\left(r\left(\frac{k}{\lambda n}\right)^{\frac{1}{\phi d}}\right) - \ln\left(r\left(\frac{k}{n}\right)^{\frac{1}{d}}\right) \geq 0 \\ &\Rightarrow \ln\left(\frac{E_{bs}\{d_k\}}{E\{d_k\}}\right) \geq 0 \\ &\Rightarrow E_{bs}\{d_k\} \geq E\{d_k\}.\end{aligned} \tag{9}$$

We complete the proof. □

Remark 4: Through bi-sampling, the change direction of expectation distance is uncertain; however, Theorem 4 uncovers the relationship between λ and ϕ to make the expectation distance increase. Under the above circumstance, we can enlarge the gap between inliers and outliers via bi-sampling.

Remark 5: Zimek et al. [43] used row-sampling to detect outliers, which means $0 < \lambda < 1$ and $\phi = 1$, Theorem 4 degenerates into $\lambda \leq 1$, which always holds for row-sampling. In this paper, we give the more general framework via bi-sampling, which shows that the row-sampling is only a special case of BSOD. However, the column-sampling is not an effective strategy for outlier detection. We also verify this point in the experimental section.

Based on Theorem 2, 3 and 4, we finally give the sufficient condition of BSOD for outlier detection.

Corollary 2: If bi-sampling is conductive to outlier detection, the portfolios of row-sampling ratio λ and column-sampling ratio ϕ satisfy that $k^{(1-\phi)} \geq \lambda n^{(1-\phi)}$.

Table I
TIME COMPLEXITY ANALYSIS

Strategy	Build Distance Matrix	Find Nearest Neighbors	Calculate Density	Total Time Complexity
Basic	$O(dn^2)$	$O(n^2)$	$O(n)$	$O(dn^2+n^2+n)$
Row-sampling	$O(\lambda dn^2)$	$O(\lambda n^2)$	$O(n)$	$O(t(\lambda dn^2+\lambda n^2+n))$
Column-sampling	$O(\phi dn^2)$	$O(n^2)$	$O(n)$	$O(t(\phi dn^2+n^2+n))$
Bi-sampling	$O(\phi\lambda dn^2)$	$O(\lambda n^2)$	$O(n)$	$O(t(\phi\lambda dn^2+\lambda n^2+n))$

B. Benefits of BSOD

We continue to analysis the benefits of BSOD in ensemble diversity and divide-and-conquer.

Ensemble diversity. BSOD also enjoys the benefit from ensemble diversity. Let $d(x)$ be the observed $k-$nearest neighbors distance of x, $\hat{d(x)}$ be the true $k-$nearest neighbors distance of x and $v(x)$ be the residual between the observed and true distance, we have

$$\begin{aligned} & d(x_{in}) - d(x_{out}) \\ &= \underbrace{\hat{d(x_{in})} - \hat{d(x_{out})}}_{\alpha} + \underbrace{v(x_{in}) - v(x_{out})}_{\beta}. \end{aligned} \tag{10}$$

Although $\alpha > 0$, it is unknown about the signal of $\alpha+\beta$ due to the residual. It might inverse the density ranking between inliers and outliers. When it comes to BSOD, we run the whole process several times and then ensemble these results by average. Thus, the diversity of ensemble helps to alleviate the problem. Let x' be the new data produced by BSOD, we have

$$\begin{aligned} & E[d(x'_{in}) - d(x'_{out})] \\ &= E[\hat{d(x'_{in})} - \hat{d(x'_{out})} + v(x'_{in}) - v(x'_{out})] \\ &= E[\hat{d(x'_{in})} - \hat{d(x'_{out})}] + E[v(x'_{in})] - E[v(x'_{out})]. \end{aligned} \tag{11}$$

If $v(x)$ obeys a $N(0,\epsilon)$ gaussian distribution, we have the above equation less than zero. By making use of the diversity of different results, the negative effect of residual might be alleviated so that the ensemble process will enhance the robustness and accuracy of outlier detection. Note that the number of ensemble members is a core factor to control the stability. From the experimental conclusion, 10 ensemble members are enough to pursue a satisfactory and stable result. It is worthy to note that if all the data points are scaled by 2, there is no change to the traditional outlier detection methods. However, the gap between inliers and outliers increases, which leads to a better chance to pursue higher performance via diverse ensemble members.

Indeed increasing the expectation distance will also increase the variance of the model; however, this is not the main problem addressed in this paper. Readers who are interested in this issue, please refer to [3], which provides some techniques to lower the variance.

Divide-and-conquer. One of our motivations is to accelerate the speed of the density-based outlier detection methods. Although this kind of methods achieves better performance than others, it needs the similarity or distance matrix to find nearest neighbors, which is very time-consuming for large-scale data sets. Therefore, how to decompose a large data set into several small pieces and keep high quality at the same time is very appealing.

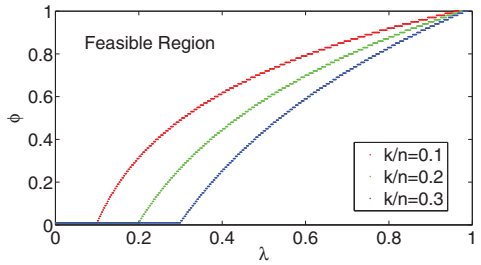


Figure 1. Relationship between λ and ϕ for BSOD condition.

Fortunately, BSOD shows a promising candidate for outlier detection on large scale data sets. Corollary 2 gives effective portfolios for row and column-sampling ratio to conduct outlier detection via bi-sampling. We draw the effective portfolios of λ and ϕ in Figure 1. The part above the cure is the feasible region. As Figure 1 shows, row-sampling strategy is represented by the horizontal line with $\phi = 1$, which validates the effectiveness of [43] in a theoretical way and column-sampling strategy is denoted by the vertical line with $\lambda = 1$, which falls into the infeasible area and suffers from worse performance. In later experimental part, we also verify this point on low dimensional data sets.

By taking a closer look at Figure 1, there exist some portfolios of λ and ϕ are both small to satisfies the BSOD condition to conduct divide-and-conquer strategy, even when the value of k/n is very small, like 0.1. This indicates that BSOD can decompose a large and high dimensional data set into several small and low dimensional sub data sets via bi-sampling, and then the outlier detection process can be conducted separately and independently.

C. Framework and Method

Here we illustrate the framework of BSOD and provide its corresponding analyses. Let X be the data matrix with n instances and d features, and $\lambda \in (0,1]$ be the row-sampling ratio, $\phi \in (0,1]$ be the column-sampling ratio. The overall framework consists of four phases as follows:

- *Phase i-Column sampling*: We first use column sampling to conduct dimension reduction and obtain a sub data X_c with n instances and ϕd features.

- *Phase ii-Row sampling*: On X_c, we employ row sampling to select some instances for the neighborhood set and obtain a sub data X_{cr} with λn instances and ϕd features.
- *Phase iii-Outlier detection*: If we apply some outlier detection methods on the sub data sets, only some selected instances are assigned a score and the majority of the instances are discard. Here we expect to calculate a score for each instance. Therefore, we build a $n \times \lambda n$ distance matrix between X_c and X_{cr}, i.e., we find the nearest neighbors for each data instance in the sub data X_{cr}, rather than the all data set X_c. And then outlier detection method is used to score each data instance.
- *Phase iv-Ensemble results*: The above process is repeated by t times, then we calculate the average of several results via multi bi-sampling to obtain the consensus result.

We also analysis the time complexity of different sampling strategies in Table I. The differences among different sampling strategies lie in building distance matrix and finding nearest neighbors. Column-sampling decreases the time cost in the phase of calculating distance matrix, while row-sampling and bi-sampling drop the time complexity in both building distance matrix and finding nearest neighbors. Especially for bi-sampling the time saving is very appealing, for instance bi-sampling runs 100 faster than no sampling when $\lambda = \phi = 10\%$. Compared with the no-sampling strategy, these sampling methods are easy to run in parallel. It is worthy to note that ϕ and λ can be both small to make the framework effective. Although the row-sampling and bi-sampling have similar time complexity, the advantage of bi-sampling will outstand in terms of high dimensional data.

IV. Experimental Results

In this part, we systematically explore the impact factors of BI-LOF on synthetic data sets, such as the number of instances, the number of features, the number of nearest neighbors and row-and-column sampling ratio, validate the theoretical condition of BSOD. On real-world data sets, we demonstrate BI-LOF can generate competitive results with superior efficiency compared to the state-of-the-art outlier detection algorithms. Finally, we apply BI-LOF on image outlier detection and show the effectiveness and stableness of BI-LOF.

A. Experimental Settings

Data. For synthetic data sets, we simulate 30 data sets with different the number of instances and features for statistical assessment. The number of instances n varies from 1000, 2000, 10000 to 50000 and the number of feature d varies from 20, 100, 200, 500, 1000 to 2000. Each synthetic data set consists of c clusters (c is randomly selected from 5 to 10) and each instance D^c obeys a Gaussian model $D^c \sim N(\vec{\mu_c}, \Sigma_c)$ with $\vec{\mu_c} = (\mu_c^1, \mu_c^2, \cdots, \mu_c^d)$ and $\Sigma_c = (\sigma_c^{ij})^{d \times d}$, in which $\vec{\mu_c} \sim U(-10, 10)$ and $\sigma_c^{ij} \sim U(0.1, 1)$. Then we calculate the Mahalanobis distance with $\vec{\mu_c}$ and Σ_c between the data instance D^c and its corresponding cluster center $D_c^M = \sqrt{(D_c - \vec{\mu_c})^\top \Sigma_c^{-1} (D_c - \vec{\mu_c})}$, which obeys $D_c^M \sim \chi^2(d)$. Thus, we label the instances beyond 0.975 fractile as outliers. By these means, we generate 2.5% outliers for each data sets. The settings is the same with [43], except that we use larger scale synthetic data sets.

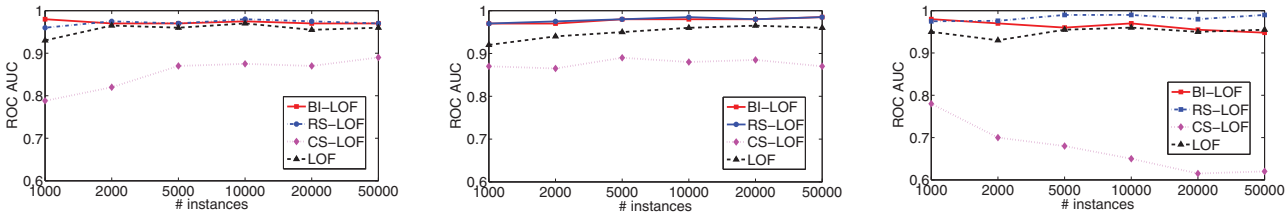
Table II
Experimental real-world data sets

Data set	#Instances	#Features	#Classes	#MinClass	#MaxClass
$bupa$	345	6	2	145	200
cmc	1473	9	3	333	629
$diabetes$	768	20	8	50	120
$iris$	150	4	3	50	50
$letter$	20000	16	26	734	813
$pageblock$	5445	10	5	28	4913
$pendigits$	10992	16	10	1055	1144
$satimage$	4436	36	6	415	1072
$yeast$	1484	8	10	5	463
$cacmcisi$	4663	14409	2	1460	3203
$cranmed$	2432	41681	2	1033	1398
$classic$	7094	41681	4	1033	3203
mm	2521	126373	2	1133	1388
$reivews$	4069	126373	5	137	1388
$sports$	8580	126373	7	145	3412
$Stanford\ Dogs$	12000	12000	120	100	100

For real-world data sets, we choose 10 low dimensional data sets from UCI machine learning repository[1], 6 high dimensional text data sets from CLUTO[2] and "Stanford Dogs" image data sets also is included to validate the effectiveness of BSOD in different domain[3]. For each data sets, we use the largest cluster as inliers. For outliers we do not use the smallest cluster as outliers, because the smallest cluster might also present density cluster structure, which should not be regarded as outliers. We neither randomly select instances from the rest as outliers, because that kind of outliers is easily to be detected. Instead we randomly select 10 instances in the smallest cluster as outliers. Note that since the MinClass of $yeast$ is only 5, we choose these 5 instances as outliers. Table II shows some important characteristics of 17 real-world data sets.

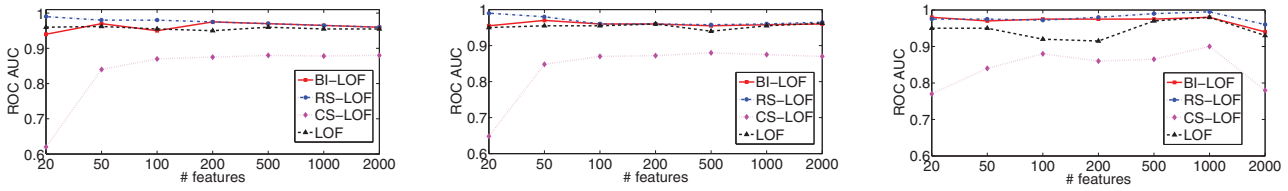
Tools. In this paper, we aim to uncover the effective portfolio of bi-sampling for density-based outlier detection; therefore, we include several density-based outlier detection for comparison. As [43] reported, LOF [8], LoOP [19] and LDOF [41] presented consistently conclusions when employed to validate the effectiveness of row-sampling. Therefore, we only select Local Outlier Factor (LOF) [8] as core outlier detection algorithm, other outlier detection algorithms based on local density are also suitable for bi-sampling outlier detection framework. Then we apply different sampling strategies via LOF, there come the LOF via bi-sampling (BI-LOF), the LOF via row-sampling (RS-

[1] http://archive.ics.uci.edu/ml/.
[2] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview.
[3] http://vision.stanford.edu/aditya86/ImageNetDogs/.



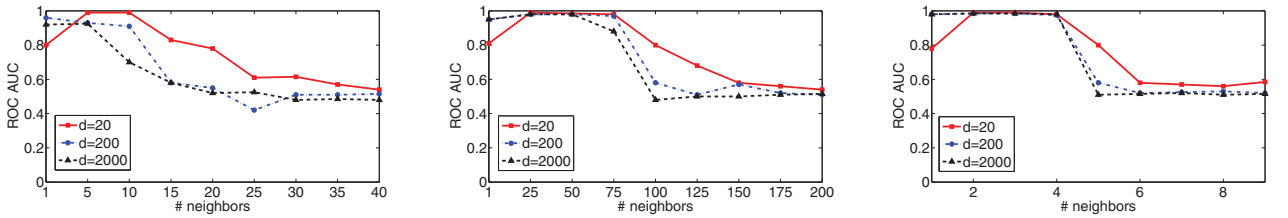
(a) $d = 20$ (b) $d = 200$ (c) $d = 2000$

Figure 2. Performance with different number of instances on synthetic data by AUC.



(a) $n = 1000$ (b) $n = 10000$ (c) $n = 50000$

Figure 3. Performance with different number of features on synthetic data by AUC.



(a) $n = 1000$ (b) $n = 5000$ (c) $n = 10000$

Figure 4. BI-LOF performance with different number of neighbors on synthetic data by AUC.

LOF) [43], the LOF via column-sampling (CS-LOF) [22] and LOF itself as baseline. The default settings are as follows: the row-sampling ratio $\lambda = 10\%$ and the column-sampling ratio $\phi = 10\%$ for BI-LOF, the row-sampling ratio $\lambda = 10\%$ for RS-LOF and the column-sampling ratio $\phi = 10\%$ for CS-LOF. The repeated times for BI-LOF, RS-LOF and CS-LOF are all 10 and we set 3-nearest neighbors for these methods as recommended in [43]. Note that BI-LOF, RS-LOF and CS-LOF run 10 times and return the average result, while LOF only runs one time due to its certainty.

Metric. Since we have the label information, here we use ROC-AUC, which is widely used in outlier detection.

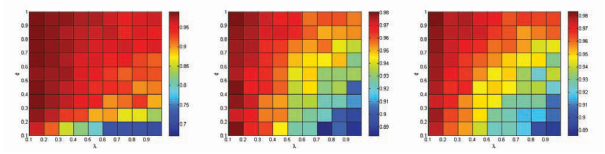
Environment. All the experiments were run on a Windows Server Standard platform of 64-bit edition, which has two Intel Xeon x7550 2.0GHz*8 CPUs and 32GB RAM.

B. Performance on Synthetic Data sets

In this part, we first use synthetic data to explore the characteristics of BI-LOF, especially in terms of the performance with different number of instances, features and nearest neighbors. Then we validate the theoretical analysis of BSOD condition in Corollary 2.

1) Impact of the number of instances: First, we investigate the impact of the number of instances. Figure 2 shows the performance of these four algorithms with different number of instances and fixed feature number 20, 200 and 2000 respectively. We can see that BI-LOF, RS-LOF and LOF have good performance on synthetic data sets, which validates the effectiveness of bi-sampling and row-sampling strategies. By taking a closer look at Figure 2, both BI-LOF and RS-LOF slightly outperform LOF; however, CS-LOF obviously performs worse compared with other methods, especially when the number of instances is huge in Figure 2(c). Recall that the BSOD condition in Corollary 2, we can find that RS-LOF with $\lambda = 10\%$ is in the feasible area, instead CS-LOF with $\phi = 10\%$ is not a feasible solution so that it suffers from detrimental performance. For these outliers which LOF can easily detect, BI-LOF and RS-LOF can also accomplish the same tasks where BI-LOF requires less computational resources than RS-LOF.

2) Impact of the number of features: Next, we explore the impact of the number of features. Figure 3 shows the performance of these four algorithms with different number of features and fixed instance number 1000, 10000 and 50000 respectively. Figure 3 has the similar trends with Figure 2. It can be seen that BI-LOF, RS-LOF and LOF still have good ability to detect outliers when the number of features increase; however, the performance of CS-LOF is much worse in the circumstance of few features, and goes up with the increasing features, which is still worse than other algorithms. This is because more features enhance the discriminative ability for CS-LOF. Generally speaking,



(a) $d = 20$ (b) $d = 200$ (c) $d = 2000$

Figure 5. Performance of BI-LOF with different sampling ratio on synthetic data with $n = 10000$ by AUC.

Figure 3 also shows the consistency with the condition of BSOD. For BI-LOF and RS-LOF, the sampling ratios of λ and ϕ are both in the feasible region; however, the sampling ratios of CS-LOF fall into the infeasible region. That is the reason that BI-LOF and RS-LOF outperform LOF, instead CS-LOF has worse results than LOF. So far, it seems that CS-LOF always returns the worst results among these four algorithms. However, when it comes to high dimensional data sets, the performance of CS-LOF has dramatic improvement, which will be illustrated later.

3) Impact of the number of neighbors: In the following, we study the factor of the number of neighbors of BI-LOF. Figure 4 shows the BI-LOF performance with different number of nearest neighbors on synthetic data sets. On all these data sets with different number of instances and features, the performance keeps high and stable within the number of neighbors less than certain threshold and goes down beyond the threshold. Such phenomena also occur on other methods. This is reasonable when the number of neighbors is too large, the radius of the data points would include too many false neighbors to harm the detection performance. It is worthy to note that the performance of BI-LOF is satisfactory when the number of neighbors is very small. Thus, we set 3-nearest neighbors as default setting, which has two purposes, one is to have the same setting for comparison, the other is to save the computational cost.

4) Impact of the portfolio of sampling ratio: Finally, we use synthetic data to validate the correctness of the condition of BSOD. Figure 5 shows the heat maps of the performance with different portfolios of sampling ratios λ and ϕ on three data sets, where the red parts mean better results and the blue parts indicate worse results. The top right points denote the results by LOF. It is easy to see that these figures have validated Theorem 4. It is worthy to note that Theorem 4 gives the sufficient condition of BSOD, which guarantees the effectiveness of bi-sampling. From Figure 5, we can see that the practical boundaries are much wider than the sufficient condition. In addition, BI-LOF with $\lambda = 10\%$ and $\phi = 10\%$ has almost equal results with RS-LOF with $\lambda = 10\%$ and $\phi = 100\%$; that is to say, although the result generated by RS-LOF has substantial improvement than LOF, we can use less information and less computation cost and achieve matchable results via BI-LOF.

Table III
TIME EXECUTION ON HIGH DIMENSIONAL DATA SETS (BY SECOND)

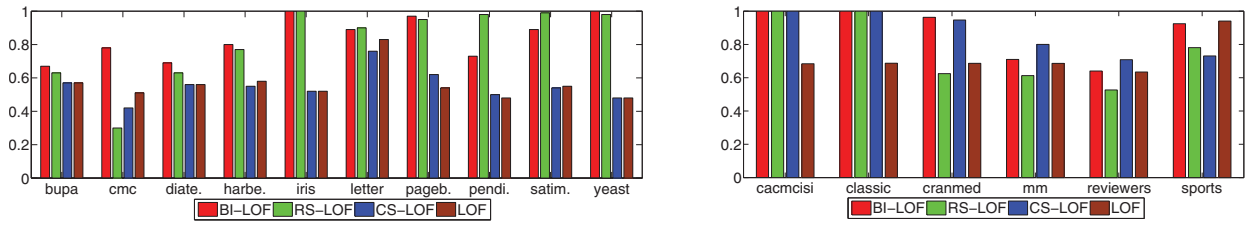
Time	BI-LOF 10 times	RS-LOF 10 times	CS-LOF 10 times	LOF 1 times
$cacmcisi$	7.86	16.62	67.15	8.15
$classic$	11.09	24.15	73.19	14.43
$cranmed$	8.01	26.25	104.66	24.43
mm	15.03	87.76	294.44	63.22
$reviews$	14.54	72.72	297.75	64.37
$sports$	81.41	544.13	1016.62	1002.66

C. Performance on Real-world Data sets

Figure 6(a) shows the results on 10 low dimensional real-world data sets. Generally speaking, BI-LOF and RS-LOF still have better performance than CS-LOF and LOF in all 10 data sets. BI-LOF achieves the best results on 6 data sets and draws a tie on $iris$ data set with RS-LOF. RS-LOF gets the best results 3 times, which indicates the effectiveness of BI-LOF on low dimensional real-world data sets. Admittedly RS-LOF achieves better results than BI-LOF on *pendigits* by a large margin, the major goal of this paper is to derive the effect portfolios for bi-sampling to improve the original outlier detection algorithm LOF, rather than to beat other outlier detection methods. Note that RS-LOF is also a special case of the BSOD condition. From this view, we can see that BI-LOF outperforms LOF on all 10 data sets, which demonstrates the great effectiveness of BI-LOF.

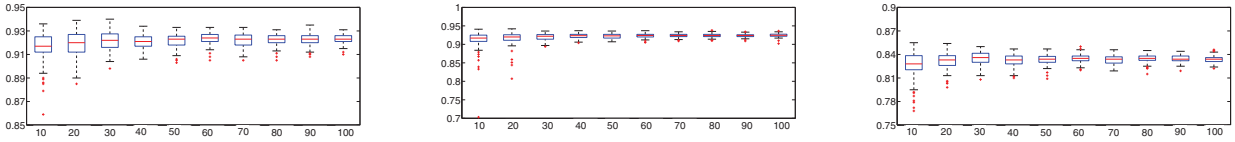
CS-LOF does not perform well on the above experiments; however, the advantage of CS-LOF is obviously observed on high dimensional data sets. Figure 6(b) shows the results of these four algorithms on 6 high dimensional real-world data sets. On $cacmcisi$ and $classic$ data sets, BI-LOF, RS-LOF and CS-LOF detect all the outliers; on the other data sets, we can see the performance of RS-LOF drops sharply compared with its performance on low dimensional data sets and the performance of CS-LOF goes up. Due to high dimensionality, the distance between any data pair trends to be equal, which heavily harms the performance of RS-LOF. Instead, the column-sampling of CS-LOF can be regarded as a dimension reduction technique, which helps to alleviate such difficulty. As for BI-LOF, high performance can be guaranteed no matter on low dimensional or high dimensional data sets, thanks to the inheritance of good properties from both row and column-sampling.

Moreover, the advantage of BI-LOF in terms of time execution outstands when it comes to high dimensional data sets. Table III gives the time execution of these four algorithms on these 6 data sets. Although BI-LOF runs 10 times, it is still the fastest one in these four algorithms. Especially, the time saving of BI-LOF is huge on $sports$ data sets compared to others. BI-LOF is almost 6 times faster than RS-LOF and 12 times faster than CS-LOF and LOF. Since each run is independent, we can easily run BI-LOF in parallel to further speed up.



(a) low dimensional data (b) high dimensional data

Figure 6. Performance on real-world data by AUC.



(a) *Sussex spaniel/Borzo* (b) *Kerry blue terrier/Cairn* (c) *Australian terrier/Otterhound*

Figure 7. Performance of BI-LOF with different repeated times on image data by AUC.

D. BSOD for Image Outlier Detection

Finally, we apply BI-LOF in the image outlier detection to validate its performance on different domain. The "Stanford Dogs" image data set contains 120 kinds of different kinds of dogs and each kind of dogs has 100 images. We select all images from one kind of dogs as inliers and 10 images from other kind as outliers. As can be seen in Table IV, BI-LOF and RS-LOF have obvious advantages than other algorithms. CS-LOF do not work well partly because the dimension of the image data sets is not as high as text data sets. And BI-LOF outperforms RS-LOF on all data sets except the last one. Despite the fact that BI-LOF only has subtle improvement over RS-LOF, taking the efficiency into account, it is very appealing in real-world applications.

On *Chew/Walker hound* data set, BI-LOF hits 5 true outliers in top 10 candidates and the images with two dogs and different colors result in the wrong decision of BI-LOF. These conditions can also be regarded as outliers to some extent since other inliers only contain one dog in one image; and on *Norfolk terrier/Giant schnauzer* data set, BI-LOF hits 7 true outliers in top 10 candidates. If we enlarge the number of candidates to 20, BI-LOF achieves 80% and 100% accuracy on these two data sets respectively.

So far, we set the number of repeated times of BI-LOF as 10, which aims to compare with RS-LOF and CS-LOF in a fair way. Here we explore the impact of repeated times of BI-LOF. Figure 7 shows the performance of BI-LOF with different repeated times. As can be seen, the performance goes slight up and the violation becomes narrow with the increase of repeated times. The phonomania are consistent on all these three data sets. This indicates two points: the repeated time is a key factor to control the stability of BI-LOF, and even at few repeated times such as 10, the performance of BI-LOF is still satisfactory.

Table IV
PERFORMANCE ON IMAGE DATA SETS BY AUC

Inliers / Outliers	BI-LOF 10 times	RS-LOF 10 times	CS-LOF 10 times	LOF 1 times
Chew / Walker hound	**0.9097**	0.9010	0.5130	0.5800
Sussex spaniel / Borzoi	**0.9239**	0.9220	0.6070	0.6250
Kerry blue terrier / Cairn	**0.9181**	0.8810	0.4650	0.4620
Australian terrier / Otterhound	**0.8262**	0.8010	0.6180	0.6120
Norfolk terrier / Giant schnauzer	0.9765	**0.9800**	0.6010	0.6600

V. CONCLUSION

In this paper, we apply bi-sampling on outlier detection, derive the condition of BSOD in a theoretical way, and analyse the benefits of BSOD in terms of ensemble diversity and divide-and-conquer. The effective portfolios of low row and column-sampling ratios are demonstrated to enlarge the gap between outliers and inliers so that a large-scale and high dimensional data set can be decomposed into several sub data sets separately and independently. Further we employ LOF within BSOD as BI-LOF to conduct extensive experiments. In general, we thoroughly explore the characteristics of BI-LOF with different number of instances, features, nearest neighbors on synthetic data sets, validate the theoretical analysis of BSOD condition, and show obvious advantages over other algorithms in terms of low and high dimensional real-world data sets. And finally we use BI-LOF to conduct image outlier detection and show high quality and stableness of BI-LOF.

VI. ACKNOWLEDGEMENT

This work is supported in part by the NSF IIS Award 1651902, NSF CNS Award 1314484, National Natural Science Foundation of China (61271252, 71471009). We thank anonymous reviewers and SPC for their constructive comments, which help to improve this work to a new level in the final revision.

REFERENCES

[1] C. Aggarwal. *Outlier analysis*. Springer Science and Business Media, 2013.

[2] C. Aggarwal. Probabilistic and statistical models for outlier detection. *Outlier Analysis*, pages 41–74, 2013.

[3] C. Aggarwal and S. Sathe. Theoretical foundations and algorithms for outlier ensembles. *ACM SIGKDD Explorations Newsletter*, 17(1):24–47, 2015.

[4] C. Aggarwal, Y. Zhao, and P. Yu. Outlier detection in graph streams. *In ICDE*, 2011.

[5] F. Angiulli and F. Fassetti. Dolphin: An efficient algorithm for mining distance-based outliers in very large datasets. *ACM Transactions on Knowledge Discovery from Data*, 3(1), 2009.

[6] S. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. *In KDD*, 2003.

[7] M. Breunig, H. Kriegel, P. Krger, and J. Sander. Data bubbles: Quality preserving performance boosting for hierarchical clustering. *In SIGMOD*, 2001.

[8] M. Breunig, H. Kriegel, R. Ng, , and J. Sander. Lof: identifying density-based local outliers. *In SIGMOD*, 2000.

[9] Y. Chen and L. Tu. Density-based clustering for real-time stream data. *In KDD*, 2007.

[10] L. Duan, L. Xu, Y. Liu, and J. Lee. Cluster-based outlier detection. *Annals of Operations Research*, 168(1):151–168, 2009.

[11] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *In KDD*, 96(34):226–231, 1996.

[12] M. Gupta, J. Gao, Y. Sun, and J. Han. Integrating community matching and outlier detection for mining evolutionary community outliers. *In KDD*, 2012.

[13] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[14] D. Huynh, R. Hartley, and A. Heyden. Outlier correction in image sequences for the affine camera. *In ICCV*, 2003.

[15] W. Jin, A. Tung, J. Han, and W. Wang. Ranking outliers using symmetric neighborhood relationship. *In PKDD*, 2006.

[16] J. Kim and J. Han. Outlier correction from uncalibrated image sequence using the triangulation method. *Pattern recognition*, 39(3), 2006.

[17] E. Knorr and R. Ng. A unified notion of outliers: Properties and computation. *In KDD*, 1997.

[18] G. Kollios, D. Gunopulos, N. Koudas, and S. Berchtold. Efficient biased sampling for approximate clustering and outlier detection in large data sets. *IEEE Transactions on Knowledge and Data Engineering*, 15(5):1170–1187, 2003.

[19] H. Kriegel, P. Krger, E. Schubert, and A. Zimek. Loop: local outlier probabilities. *In CIKM*, 2009.

[20] H. Kriegel, P. Krger, E. Schubert, and A. Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. *In KDD*, 2009.

[21] H. Kriegel and A. Zimek. Angle-based outlier detection in high-dimensional data. *In KDD*, 2008.

[22] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. *In KDD*, 2005.

[23] S. Li, M. Shao, and Y. Fu. Multi-view low-rank analysis for outlier detection. *In SDM*, 2015.

[24] X. Li. 3d orthographic reconstruction based on robust factorization method with outliers. *In ICCV*, 2004.

[25] F. Liu, K. Ting, and H. Zhou. Isolation forest. *In ICDM*, 2008.

[26] H. Liu, T. Liu, J. Wu, D. Tao, and Y. Fu. Spectral ensemble clustering. *In KDD*, 2015.

[27] H. Liu, M. Shao, and Y. Fu. Consensus guided unsupervised feature selection. *In AAAI*, 2016.

[28] H. Liu, M. Shao, S. Li, and Y. Fu. Infinite ensemble for image clustering. *In KDD*, 2016.

[29] H. Liu, J. Wu, D. Tao, Y. Zhang, and Y. Fu. Dias: A disassemble-assemble framework for highly sparse text clustering. *In SDM*, 2015.

[30] T. Liu and D. Tao. Classification with noisy labels by importance reweighting. *In ICIST*, 2014.

[31] T. Liu and D. Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2016.

[32] T. Liu and D. Tao. On the performance of manhattan nonnegative matrix factorization. *IEEE Transactions on Neural Nentworks and Learning Systems*, 27(9):1851–1863, 2016.

[33] A. Marcos, M. Yamada, A. Kimura, and T. Iwata. Clustering-based anomaly detection in multi-view data. *In CIKM*, 2013.

[34] D. Martinec and T. Pajdla. Consistent multi-view reconstruction from epipolar geometries with outliers. *Image Analysis*, pages 493–500, 2003.

[35] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos. Loci: Fast outlier detection using the local correlation integral. *In ICDE*, 2003.

[36] N. Pham and R. Pagh. A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data. *In KDD*, 2012.

[37] D. Ren, B. Wang, and W. Perrizo. Rdf: A density-based outlier detection method using vertical data representation. *In ICDM*, 2004.

[38] E. Schubert, A. Zimek, and H. Kriegel. Generalized outlier detection with flexible kernel density estimates. *In SDM*, 2014.

[39] E. Schubert, A. Zimek, and H. Kriegel. Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. *Data Mining and Knowledge Discovery*, 28(1):190–237, 2014.

[40] S. Wu and S. Wang. Information-theoretic outlier detection for large-scale categorical data. *IEEE Transactions on Knowledge and Data Engineering*, 25(3):589–602, 2013.

[41] K. Zhang, M. Hutterand, and H. Jin. A new local distance-based outlier detection approach for scattered real-world data. *In PKDD*, 2008.

[42] H. Zhao and Y. Fu. Dual-regularized multi-view outlier detection. *In IJCAI*, 2015.

[43] A. Zimek, M. Gaudet, R. Campello, and J. Sander. Subsampling for efficient and effective unsupervised outlier detection ensembles. *In KDD*, 2013.

[44] A. Zimek, E. Schubert, and H. Kriegel. A survey on unsupervised outlier detection in high dimensional numerical data. *Statistical Analysis and Data Mining*, 5(5):363–387, 2012.