Anomaly Detection via Over-sampling Principal Component Analysis

Yi-Ren Yeh, Zheng-Yi Lee, and Yuh-Jye Lee

Abstract Outlier detection is an important issue in data mining and has been studied in different research areas. It can be used for detecting the small amount of deviated data. In this article, we use "Leave One Out" procedure to check each individual point the "with or without" effect on the variation of principal directions. Based on this idea, an over-sampling principal component analysis outlier detection method is proposed for emphasizing the influence of an abnormal instance (or an outlier). Except for identifying the suspicious outliers, we also design an on-line anomaly detection to detect the new arriving anomaly. In addition, we also study the quick updating of the principal directions for the effective computation and satisfying the on-line detecting demand. Numerical experiments show that our proposed method is effective in computation time and anomaly detection.

1 Introduction

Due to the reasons that only very few labeled data are available in real applications and the events that people are interested in are extremely rare or do not happen before, the outlier detection is getting people's attention more and more [3, 4, 7, 8, 9, 11]. Outlier detection can be used in many application domains such as homeland security, credit card fraud detection, intrusion and insider threat detection in cyber-security, fault detection and malignant diagnosis etc. [8, 11, 12, 13]. Thus, the outlier detection methods are designed for finding the rare instances or the deviated data. In other words, an outlier detection method can be applied to deal with extremely unbalanced data distribution problems, such as capturing the anomaly which exists in a small proportion of network traffic.

Yi-Ren Yeh, Zheng-Yi Lee, and Yuh-Jye Lee
Computer Science and Information Engineering, National Taiwan University of Science and Technology, No.43, Sec.4, Keelung Rd., Taipei, Taiwan 10607, e-mail: {D9515009, M9615018, yuh-jye}@mail.ntust.edu.tw

In the past, many outlier detection methods have been proposed [3, 7, 9]. One of the most popular outlier methods is using the density-based local outlier factor (LOF) to measure the outlierness for each instance [3]. The LOF uses the density of each individual instance's neighbors to define the degree of outlierness and concludes a suspicious ranking for all instances. The most important property of the LOF is considering the local data structure for estimating the density. This property makes the LOF discover the outliers which are sheltered under a global data structure. Besides, an angle-based outlier detection (ABOD) method has also been proposed recently [9]. The main concept of ABOD is using the variation of the angles between the each target instance and the rest instances. An outlier or deviated instance will generate a smaller variance among its associated angles. Based on this observation, the ABOD considers all the variance of the angles between the target instance and any pair of instances to detect outliers. However, the time complexity of ABOD is too high to deal with large datasets. In [9], the authors also proposed the fast ABOD which is an approximation of the original ABOD. The difference is that fast ABOD only considers the variance of the angles between the target instance and any pair of instances of target instance's k nearest neighbors. Even though, these methods mentioned above can not be scaled up to massive datasets because of the very expensive computational cost.

In this paper, we observe that removing (or adding) an abnormal instance (or outlier) will cause a lager effect on principal directions than removing (or adding) a normal one. From this observation, we apply the “Leave One Out” (LOO) procedure to check each individual point the “with or without” effect on the variation of principal directions. This will help us to remove the suspicious outliers in the dataset. Thus, it can be used for the data cleaning purpose. Once we have a clean dataset, we can extract the leading principal directions from it and use these directions to characterize the normal profile for the dataset. Similarly, we can evaluate the “with or without” effect of new arriving data point. That defines a suspicious score for the new arriving data point. If the score is greater than a certain threshold, we regard this point as an outlier. Based on this mechanism, we proposed an on-line anomaly detection method. Intuitively, the “with or without” effect on the principal direction will be diminished for a single data point even it is an outlier when the dataset is large. To overcome this problem, we employ the “over-sampling” scheme that will amplify the “with or without” influence made by an outlier. We also are aware of computation issues in the whole process. How to compute the principle directions efficiently when the mean and covariance matrix are changed slightly is also a key issue and the tricks for matrix computation will be included in this work as well.

2 Over-sampling Principal Component Analysis

In this section, we first introduce the classical dimension reduction method PCA briefly. The study on the influence of the variation of principal directions via LOO procedure is also be exhibited. Finally, we introduce the over-sampling scheme in

PCA to emphasize the influence of an abnormal instance. In addition, an effective computation for computing the covariance matrix and estimating principal directions in LOO procedure is also proposed.

2.1 Principal Component Analysis

PCA is an unsupervised dimension reduction method. It can retain those characteristics of the data set that contribute most to its variance by keeping lower-order principal components. These few components often contain the “most important” aspects of the data. Let $A \in \mathbb{R}^{p \times n}$ be the data matrix and each column, $x_i \in \mathbb{R}^p$, represents an instance. PCA involves the eigenvalue decomposition in the covariance matrix of the data. Its formulation is solving an eigenvalue problem as follows:

$$\Sigma_A \Gamma = \lambda \Gamma, \tag{1}$$

where $\Sigma_A = \frac{1}{n} \sum_{i=i}^{n} (x_i - \mu)(x_i - \mu)^\top$ is the covariance matrix, μ is the grand mean, and the resulting Γ is the eigenvector set. In practical, some eigenvalues have little contribution to variance and can be discarded. It means that we only need to keep few components to represent the data. In addition, PCA explains variance and is sensitive to outliers. A few points distant from the center would have a large influence on variance and its principal directions. In other words, these first few principal directions will be influenced seriously if our data contain some outliers.

2.2 The Influence of An outlier on Principal Directions

Based on the concept that we mentioned in the previous section, PCA is sensitive to outliers and we only need few principal components to represent the main data structure. That is, an outlier or a deviated instance will cause a larger effect on these principal directions. Hence, we explore the variation of principal directions when removing or adding an instance. This concept is illustrated in Fig. 1 where the clustered blue circles represent the normal data, the red square represents an outlier, and the green arrow is the first principal direction. From the right panel to the left panel in Fig. 1, we can see that the first principal direction is affected when we remove an outlier. The first principal direction is changed and forms a larger angle between the old one and itself. In this case, the first principal direction will not be affected and only form an extremely small angle between the old first principal direction and the new one if we remove a normal instance. Via this observation, we use LOO procedure to check each individual point the “with or without” effect. On the other hand, we might have the pure normal data in hand. In this case, we use the same concept in LOO setting but with incremental strategy. That is, adding an instance to see the variation of the principal directions. Similarly, adding a normal

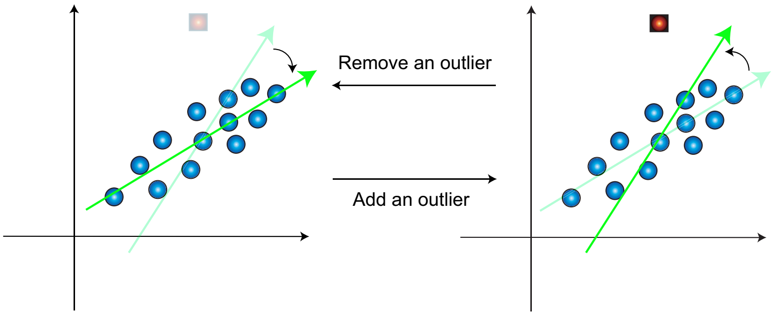


Fig. 1 The illustration for the effect of an outlier on the first principal direction.

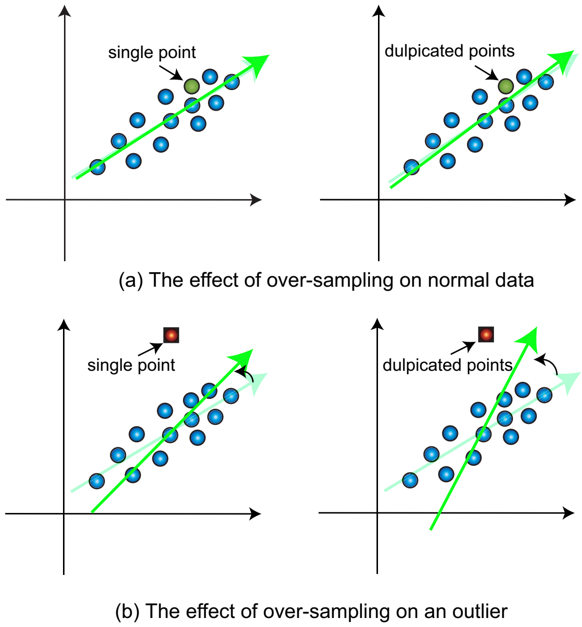
data point will create a smaller angle between the old one and itself while it will form a larger angle with adding an outlier (from the left panel to the right panel in Fig. 1). We check the variation of the principal directions for each new arriving instance and regard it as an outlier if the variation of the principal directions is significant.

In summary, we find that the principal directions will be affected with removing an outlier while the variation of the principal direction will be smaller with removing a normal instance. This concept can be used for identifying the anomaly or outliers in our data. On the contrary, adding an outlier will also cause a larger influence on the principal directions while the variation of the principal directions will be smaller with adding a normal one. It means that we can use the incremental strategy to detect the new arriving abnormal data or outliers. In other words, we explore the variation of the principal directions with removing or adding a data point and use this information to identify outliers and detect new arriving deviated data.

2.3 Over-sampling Principal Components Analysis

As we mentioned in Section 2.2, we identify outliers in our data and detect the new arriving outliers through the variation of the principal directions. However, the effect of “with or without” a particular data may be diminished when the size of the data is large. On the other hand, the computation in estimating the principal directions will be heavy because we need to recompute the principal directions many times in LOO scenario.

In order to overcome the first problem, we employ “over-sampling” scheme to amplify the outlierness on each data point. For identifying an outlier via LOO strategy, we duplicate the target instance instead of removing it. That is, we duplicate the target instance many times (10% of the whole data in our experiments) and observe how much variation do the principal directions vary. With this over-sampling scheme, the principal directions and mean of the data will only be affected slightly if the target instance is a normal data point (see Fig. 2(a)). On the contrary, the vari-



(a) The effect of over-sampling on normal data

(b) The effect of over-sampling on an outlier

Fig. 2 The effect of over-sampling on an outlier and a normal instance.

ation will be enlarged if we duplicate an outlier (see Fig. 2(b)). On the other hand, we also can apply over-sampling scheme in the LOO procedure with incremental case. The main idea is to enlarge the difference of the effect between a normal data point and an outlier. Based on the over-sampling PCA, we make the idea discussed in Section 2.2 more practical.

For computation issue, we need to recompute the principal directions many times in the LOO scenario. In order to avoid this heavy loading, we also proposed two strategies to accelerate the procedure in estimating principal directions. The first one is the fast updating for the covariance matrix. The another one is the solving the eigenvalue problem via the power method [6]. As (1) shows, the formulation of PCA is solving an eigenvalue decomposition on the covariance matrix of the data. However, it is unnecessary to completely re-compute the covariance matrix in the LOO procedure. The difference of covariance matrix can be easily adjusted while we only duplicate one instance. Hence, we consider a light updating of covariance matrix for fast computation [5]. Let $Q = \frac{AA^\top}{n}$ be the pre-computed scaled outer-product matrix. We use the following updating for the adjusted mean vector $\tilde{\mu}$ and covariance matrix $\tilde{\Sigma}$:

$$\tilde{\mu} = \frac{\mu + r \cdot x_t}{1+r} \tag{2}$$

and

$$\tilde{\Sigma} = \frac{1}{1+r}Q + \frac{r}{1+r}x_t x_t^\top - \tilde{\mu}\tilde{\mu}^\top, \tag{3}$$

Algorithm 1 Over-sampling Principal Component Analysis Outlier Detection for Data Cleaning

```
Input: a data matrix $A \in \mathbb{R}^{p\times n}$ and the ratio $r$
Output: the suspicious outlier ranking for the data
1. Compute outer-product $Q = \frac{AA^\top}{n}$, the mean $\mu$, and the first principal direction $v$
2. Using LOO strategy to duplicate the target instance $x_t$ and compute the adjusted mean vector
    $\tilde{\mu}$ and covariance matrix $\tilde{\Sigma}$:
    $\tilde{\mu} = \frac{\mu + r\cdot x_t}{1+r}$
    $\tilde{\Sigma} = \frac{1}{1+r}Q + \frac{r}{1+r}x_t x_t^\top - \tilde{\mu}\tilde{\mu}^\top$
3. Extract the adjusted first principal direction $\tilde{v}$ and compute the cosine similarity of $v$ and $\tilde{v}$
4. Repeat step 2 and 3 until scanning all the data
5. Ranking all instances according to their suspicious outlier scores (1− | cosine similarity |)
```

where $A \in \mathbb{R}^{p\times n}$ is the data matrix, x_t is the target instance and r is the parameter of the proportion of the whole data in duplicating x_t. From (3), it shows that we can keep the matrix Q in advance and need not to recompute it completely in LOO procedure.

In extracting the first principal direction, we also apply the power method for fast computation. Power method [6] is an eigenvalue algorithm for computing the greatest eigenvalue and the corresponding eigenvector. Given a matrix M, this method starts with an initial normalized vector u_0, which could be an approximation to the dominant eigenvector or a nonzero random vector, then iteratively computes the u_{k+1} as follows:

$$u_{k+1} = \frac{Mu_k}{\|Mu_k\|}. \tag{4}$$

The sequence $\{u_k\}$ converges on the assumption that there exists an largest eigenvalue of M in absolute value. From (4), we can see that power method does not compute a matrix decomposition but only uses the matrix multiplication. Based on this property, the power method can converge rapidly and make our LOO procedure faster. On the other hand, if we want to find the remaining eigenvectors, we could use deflation process [6]. Note that we only use the first principal component in our experiments so we only apply the power method in estimating the first principal direction.

3 Data Cleaning and On-line Anomaly Detection

In this section, we present the framework of our data analysis. There are two phases in our framework, data cleaning and on-line anomaly detection. In the data cleaning phase, the goal is to identify the suspicious outliers. First, we over-sample each instance with LOO strategy to see the variation of the first principal direction. Here we use the absolute value of cosine similarity to measure the difference of the first principal direction and define “one minus the absolute value of cosine similarity”

Algorithm 2 Over-sampling Principal Component Analysis for On-line Anomaly Detection

```
Input: the scaled outer-product matrix $Q = \frac{AA^\top}{n}$, the mean vector $\mu$ and the first principal direction $v$ of the normal data, the ratio $r$, and threshold $h$, and the new arriving instance $x$
Output: $x$ is an outlier or not
1. Compute the updated mean vector $\tilde{\mu}$ and covariance matrix $\tilde{\Sigma}$:
    $\tilde{\mu} = \frac{\mu + r \cdot x}{1+r}$
    $\tilde{\Sigma} = \frac{1}{1+r} Q + \frac{r}{1+r} xx^\top - \tilde{\mu}\tilde{\mu}^\top$
2. Extract the updated first principal direction $\tilde{v}$ and compute the cosine similarity of $v$ and $\tilde{v}$
3. Check the cosine similarity of $v$ and $\tilde{v}$ and see if it is higher than the specified threshold $h$
```

as the suspicious outlier scores. A higher suspicious outlier score implies the higher probability of being an outlier. Once we have the suspicious outlier scores for each instance, we can rank the instances and filter out the outliers in the given data according to the ranking. The over-sampling principal component analysis outlier detection algorithm (OPCAOD) for data cleaning is described in Algorithm 1.

After filtering the suspicious points, we can get the pure normal data and apply the on-line anomaly detection which is not suitable for LOF and ABOD. Nevertheless, the quick updating of the principal directions in our proposed method can satisfy the on-line detecting demand. In this phase, the goal is to identify the new arriving abnormal instance. Similarly, we also apply over-sampling PCA for the new arriving instance to check the variation of the principal directions. However, how to determine the threshold for identifying an abnormal instance is a problem. In order to overcome this problem, we use some statistics to set the threshold. The idea is calculating the mean and standard deviation of the suspicious scores which are computed from all normal data points. Once we have the mean and standard deviation, a new arriving instance will be marked if its suspicious score is higher than the mean plus a specified multiple of the standard deviation. The over-sampling principal component analysis for on-line anomaly detection (OPCAAD) is also described in Algorithm 2.

4 Experimental Results

In our experiments, we evaluate our methods in three datasets. For the outlier detection, we first generate a 2-D synthetic data for testing our method. The synthetic data is consisting of 200 normal instances (blue circle in Fig. 3) and 10 deviated instances (red stars in Fig. 3). The normal data points are generated for the normal distribution with zero mean and standard deviation 1. On the other hand, the deviated data points are generated from the normal distribution with zero mean and standard deviation 15. Note that the clustering algorithm is not useful here because the outliers do not belong to a certain cluster. In this 2-D synthetic data, we apply our over-sampling principal component analysis outlier detection (OPCAOD) on it and filter 5% of the

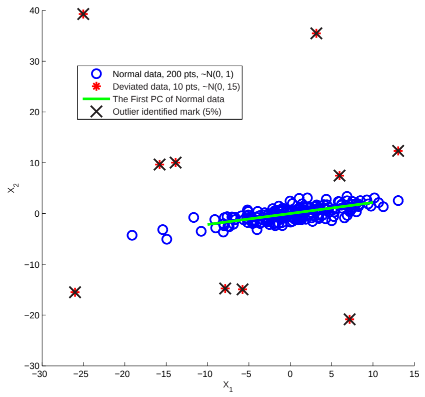


Fig. 3 The result of identifying outliers in the 2-D synthetic data.

whole data (10 points) as outliers (with black crosses). The result is shown in the Fig. 3 and we can see the effectiveness of our proposed because of catching all the outliers in this synthetic contaminated data. Except for the 2-D synthetic data, we also evaluate our outlier detection method on `pendigits` dataset which can be obtained from UCI Repository of machine learning data archive [1]. We fixed the digit “0” as the normal data (780 instances) and set up 9 different combination via other digits “1” to “9” (20 data points for each) to evaluate our method in outlier detection. In this dataset, we compare our methods PCAOD (only removing one instance in LOO) and OPCAOD with LOF and fast ABOD. We use the area under the ROC curve (AUC) [2] to evaluate the suspicious outlier ranking. The results are shown in Table 1 and Table 2. Here r is the ratio of the duplicated points relative to the whole data and k is the number of nearest neighbors which is needed to be given in LOF and fast ABOD. In our experiments, we have tried several parameters for these methods and used the best parameter for each method respectively. These results show that our method is comparable with LOF and fast ABOD in detecting the outliers. Nevertheless, our method is faster than LOF and fast ABOD. On the other hand, we also can see the effect of over-sampling strategy from Table 1. The AUC score of over-sampling PCA is much better than that without over-sampling.

For the on-line anomaly detection phase, we evaluate our method with KDD cup 99 dataset [10]. In our experiments, we focus on the 10% training subset under the tcp protocol. We extract 2000 normal instances points as the training set and also re-sample another 2000 normal instances and different size of attacks as our testing set. The details are recorded in Table 3. In the beginning, we apply the data cleaning phase to filter 100 points (5%) in the normal data to avoid the deviated data. After that, we extract the normal pattern (the first principal direction) and use the on-line anomaly detection to detect the new arriving attack. Note that we use the training set and re-sample other attacks to determine the threshold. The results are shown in Table 3. From Table 3, we can see the good performance of our proposed method

Scenario	PCAOD	OPCAOD ($r = 0.1$)	LOF ($k = 100$)	Fast ABOD ($k = 40$)
`0 vs. 1`	0.9098	0.9994	0.9942	0.9541
`0 vs. 2`	0.9235	0.9999	0.9962	0.9502
`0 vs. 3`	0.4677	0.9978	0.9972	0.9232
`0 vs. 4`	0.8765	0.9533	0.9859	0.9201
`0 vs. 5`	0.8421	0.9515	0.9981	0.9472
`0 vs. 6`	0.9865	0.9939	0.9785	0.9386
`0 vs. 7`	0.9227	0.9984	0.9966	0.9297
`0 vs. 8`	0.8343	0.9556	0.9947	0.9246
`0 vs. 9`	0.7881	0.9985	0.9944	0.9641

Table 1 The AUC scores of PCAOD, OPCAOD, LOF, and fast ABOD in 9 different combinations of `pendigits` dataset.

	PCAOD	OPCAOD ($r = 0.1$)	LOF ($k = 100$)	Fast ABOD ($k = 40$)
cpu time (sec.)	0.2671	0.2878	3.221	18.772

Table 2 The average cpu times of PCAOD, OPCAOD, LOF, and fast ABOD in `pendigits` dataset.

Attack	Testing data size		TP	FP	Error
type	normal	attack	Rate	Rate	Rate
Dos	2000	100	0.940	0.073	0.073
Probe	2000	100	0.980	0.022	0.023
R2L	2000	100	0.900	0.071	0.072
U2R	2000	49	0.816	0.038	0.038

Table 3 The true positive (TP) rate, false positive (FP) rate, and error rate of KDD Cup 99 dataset. TP rate is the percentage of attacks detected; FP rate is the percentage of normal connections falsely classified as attacks.

because of the high true positive rates and low false negative rates. On the other hand, our proposed method also work well in detecting the rare attacks, like $U2R$. It shows that an outlier detection method is suitable for the extremely unbalanced data distribution. In summary, our proposed method and framework not only can detect the outliers in the given data but also can be applied to predict the abnormal behavior.

5 Conclusion and Future Work

We have explored the variation of principal directions in the leave one out scenario. From the experimental results, we demonstrated that the variation of principal directions caused by outliers indeed can help us to detect the anomaly. We also proposed the over-sampling PCA to enlarge the outlierness of an outlier. In addition, an effective computation for computing the covariance matrix and estimating principal directions in LOO is also proposed for reducing the computational loading and satisfying the on-line detecting demand which is not suitable for LOF and ABOD. On the other hand, our proposed PCA based anomaly detection is suitable for the extremely unbalanced data distribution (such as network security problems). In the future, we will also study how to speed up the procedure via online learning techniques (ie., develop a quick adjusting for the principal directions directly).

References

1. A. Asuncion and D. J. Newman, UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/mlrepository.html,” 2007.
2. A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition, 30, 1145-1159,1997.
3. M. M. Breunig, H.-P. Kriegel, R. Ng, J. Sander, LOF: Identifying density-based local outliers. *Proc. of the 2000 ACM SIGMOD Int. Conf. on Management of Data*, Dallas, Texas, 2000.
4. V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey. *ACM Computing Surveys*, 2009.
5. D. Erdogmus, Y. Rao, H. Peddaneni, A. Hegde, J. C Principe, Recursive principal components analysis using eigenvector matrix perturbation. *Jornal of Applied Signal Process.* 13, 2034-2041, 2004.
6. G. H. Golub, C. F. Van Loan, *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, 1983.
7. D. Hawkins, *Identification of Outliers*. Chapman and Hall, London, 1980
8. L. Huang, X. Nguyen, M. Garofalakis, M. I. Jordan, A. Joseph, N. Taft, In-network pca and anomaly detection. *Advances in Neural Information Processing Systems 19*, pp. 617-624. MIT Press, Cambridge, MA, 2007.
9. H.-P. Kriegel, M. Schubert, A. Zimek, Angle-based outlier detection. *Proc. of 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Las Vegas, NV, 2008.
10. KDD Cup 1999 Data, Available on : http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, August 2003.
11. A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, J. Srivastava, A comparative study of anomaly detection schemes in network intrusion detection. *Proc. of the Third SIAM Conference on Data Mining*, 2003.
12. S. Rawat, V. P. Gulati, A. K. Pujari, On the use of singular value decomposition for a fast intrusion detection system. *Electronic Notes in Theoretical Computer Science*, 142, 215-28, 2006.
13. W. Wang, X. Guan, X. Zhang, A novel intrusion detection method based on principal component analysis in computer security. *Proceedings of the International Symposium on Neural Networks*, Dalian, China, pp. 657-662, 2004.