International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Anomaly Detection via Online Over-Sampling Principal Component Analysis

Pachunoori Naresh[1], Garine Bindu Madhavi[2]

[1]M.Tech student, Department of CSE, Anurag Group of Institutions, Hyderabad, India

[2]Assistant professor, Department of CSE,Anurag Group of Institutions, Hyderabad, India

Abstract: ***Anomaly detection has been an important research topic in data mining and machine learning. Many real-world applications for instance intrusion or credit card fraud detection require an effective and efficient framework to identify deviated data instances. Though, most anomaly detection methods are typically implemented in batch mode, and so cannot be easily extended to large-scale problems without sacrificing computation and memory requirements. In this article, we propose an online over-sampling principal component analysis (osPCA) algorithm to address this problem, and we plan at detecting the presence of outliers from a large amount of data via an online updating technique. Not like prior PCA based approaches, we do not store the whole data matrix or covariance matrix, and so our approach is especially of interest in online or large-scale problems. Through over-sampling the target instance and extracting the principal direction of the data, the proposed osPCA permit us to determine the anomaly of the target instance according to the variation of the resulting dominant eigenvector. While our osPCA need not perform eigen analysis explicitly, the proposed framework is privileged for online applications which have computation or memory limitations. Match up with the well-known power method for PCA and other popular anomaly detection algorithms.***

Keywords: Anomaly detection, online updating, least squares, over-sampling, principal component analysis.

1. Introduction

Anomaly (or outlier) detection aims to identify a small group of instances which deviate remarkably from the existing data. A well-known definition of "outlier" is given in [1]: "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism," which gives the general idea of an outlier and motivates many anomaly detection methods [1], [2], [3], [4], [5], [6], [7]. Almost, anomaly detection can be found in applications such as homeland security, intrusion and insider threat detection, credit card fraud detection, in cybersecurity, fault detection, or malignant diagnosis [3], [4],[6], [8], [9]. However, since only a limited amount of labeled data are available in the above real-world applications, how to decide anomaly of unseen data (or events) draws attention from the researchers in data mining and machine learning communities [1], [2], [3], [4], [5], [6], [7].

Despite the rareness of the deviated data, its occurrence might enormously affect the solution model such as the distribution or principal directions of the data. For ex., the calculation of data mean or the least squares solution of the associated linear regression model are both sensitive to outliers. As a outcome, anomaly detection needs to solve an unsupervised yet unbalanced data learning problem. Similarly, we observe that removing (or adding) an abnormal data instance will affect the principal direction of the resulting data than removing (or adding) a normal one does. Using the above "Leave One Out" (LOO) approach, we can calculate the principal direction of the dataset without the target instance present and that of the original dataset. Thus, the outlierness (or anomaly) of the data instance can be determined by the variation of the resulting principal directions. More exactly, the difference between these two eigenvectors will indicate the anomaly of the target instance. By ranking the variant scores of all data points, one can identify the outlier data by a pre-defined threshold or a pre-determined portion of the data.

We note that the above framework can be considered as a *decremental* PCA (dPCA) based approach for anomaly detection. As it works well for applications with moderate dataset size, the variation of principal directions might not be significant when the size of the dataset is large. In real-world anomaly detection problems dealing with a large amount of data, adding or removing one goal instance only produces negligible difference in the resulting eigenvectors, and one cannot simply affect the dPCA technique for anomaly detection. To deal with this practical problem, we advance the "over-sampling" strategy to duplicate the target instance, and we perform an over-sampling PCA (os- PCA) on such an over-sampled dataset. It is obvious that the effect of an outlier instance will be amplified due to its duplicates present in the PCA formulation, and this composes the detection of outlier data easier. Though, this LOO anomaly detection procedure with an over-sampling strategy will markedly increase the computational load. For every target instance, one always needs to create a dense covariance matrix and solves the associated PCA problem. This will forbid the use of our proposed framework for real-world large-scale applications. Even if the well known power method is able to produce approximated PCA solutions, it need the storage of the covariance matrix and cannot be easily extended to applications with streaming data or online settings. So, we present an online updating technique for our osPCA. This revised technique allows us to efficiently calculate the approximated dominant eigenvector without performing eigen analysis or storing the data covariance matrix. Match up to the power method or other popular anomaly detection algorithms, the necessary computational costs and memory requirements are significantly reduced, and so our method is especially preferable in online, streaming data, or largescale problems. Detailed derivations

and discussions of the osPCA with our proposed online updating technique will be presented

2. Literature Survey

Literature survey is the most important step in software development process. Before developing the tool it is essential to determine the time factor, economy n company power. Once these things r satisfied, ten next steps are to resolve which operating system and language can be used for developing the tool. Once the programmers start constructing the tool the programmers need lot of external support. This maintains can be obtained from senior programmers, from book or from websites. Earlier than building the system the above consideration r taken into account for developing the proposed system

3. Anomaly Detection Via Principal Component Analysis

We first briefly review the PCA algorithm in Section 3.1. Based on the leave-one-out (LOO) approach, Section 3.2 presents our study on the effect of outliers on the derived principal directions.

3.1 Principal Component Analysis

PCA is a well known unsupervised dimension reduction method, and which determines the principal directions of the data distribution. To get these principal directions, one needs to create the data covariance matrix and calculate its dominant eigenvectors. These eigenvectors will be the *majority informative* among the vectors in the original data space, and are hence considered as the principal directions. $\mathbf{A} = [\mathbf{x}_1^\top; \mathbf{x}_2^\top; \cdots; \mathbf{x}_n^\top] \in \mathbb{R}^{n \times p}$, Where each row $\mathbf{x}_i$ represents a data instance in a p dimensional space, and n is the number of the instances. in general, PCA is formulated as the following optimization Problem

$$\max_{\mathbf{U} \in \mathbb{R}^{p \times k}, \|\mathbf{U}\| = \mathbf{I}} \sum_{i=1}^{n} \mathbf{U}^\top (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top \mathbf{U}, \qquad (1)$$

where **U** is a matrix consisting of k dominant eigenvectors. From this formulation, one is able to see that the standard PCA can be viewed as a task of determining a subspace where the projected data has the largest variation. Alternatively, one can approach the PCA problem as minimizing the data reconstruction error, that is.

$$\min_{\mathbf{U} \in \mathbb{R}^{p \times k}, \|\mathbf{U}\| = \mathbf{I}} J(\mathbf{U}) = \sum_{i=1}^{n} \|(\mathbf{x}_i - \boldsymbol{\mu}) - \mathbf{U}\mathbf{U}^\top (\mathbf{x}_i - \boldsymbol{\mu})\|^2, \quad (2)$$

where $\mathbf{U}^\top (\mathbf{x}_i - \mu)$ determines the optimal coefficients to weight each principal directions when reconstructing the approximated version of $(\mathbf{x}_i - \mu)$. Generally, the problem in either (1) or (2) can be solved by deriving an eigenvalue decomposition problem of the covariance data matrix, that is.

$$\boldsymbol{\Sigma}_\mathbf{A} \mathbf{U} = \mathbf{U} \boldsymbol{\Lambda}, \qquad (3)$$

Where

$$\boldsymbol{\Sigma}_\mathbf{A} = \frac{1}{n} \sum_{i=i}^{n} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top \qquad (4)$$

is the covariance matrix, $\boldsymbol{\mu}$ is the global mean. Every column of **U** represents an eigenvector of $\boldsymbol{\Sigma}\mathbf{A}$, and the corresponding diagonal entry in $\boldsymbol{\Lambda}$ is the associated eigenvalue. For the use of dimension reduction, the last few eigenvectors will be discarded due to their negligible contribution to the data distribution.

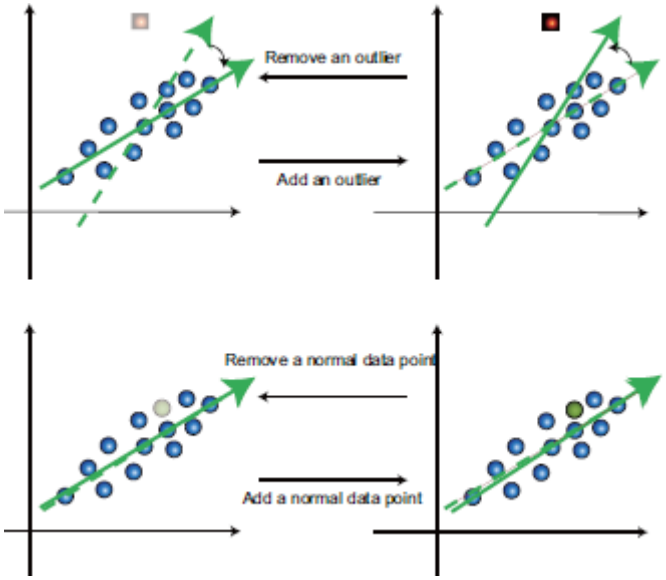

Figure 1: The effects of adding/ removing an outlier r a normal data instance on the principal directions

While PCA requires the calculation of global mean and data covariance matrix, we initiate that both of them are sensitive to the presence of outliers. As shown in [19], if there are outliers present in the data, dominant eigenvectors formed by PCA will be remarkably affected by them, and therefore this will produce a significant variation of the resulting principal directions. We will further discuss this issue in the following subsections, and explain how we advance this property for anomaly detection.

3.2 The Use of PCA for Anomaly Detection

In this section, we study the variation of principal directions when we remove or add a data instance, and how we make use of this property to determine the outlierness of the target data point. We use Figure 1 to illustrate the above observation. We note down that the clustered blue circles in Figure 1 represent normal data intances, the red square denotes an outlier, and also the green arrow is the dominant principal direction. From Figure 1, we see that the principal direction is deviated when an outlier instance is added. additionally, the presence of such an outlier instance produces a large angle between the resulting and the original principal directions. Then again, this angle will be small when a normal data point is added. So, we will use this property to determine the outlierness of the target data point using the LOO strategy. We now present the idea of combining PCA and the LOO strategy for anomaly detection. Given a data set **A** with n data instances, we first take out the dominant principal direction **u** from it. If the target instance is $\mathbf{x}_t$, we next calculate the leading principal direction $\tilde{\mathbf{u}}_t$ without $\mathbf{x}_t$ present. To identify the outliers in a

dataset, we simply repeat this procedure n times with the LOO strategy (one for each target instance):

$$\Sigma_{\tilde{\mathbf{A}}}\tilde{\mathbf{u}}_t = \lambda\tilde{\mathbf{u}}_t, \quad (5)$$

where $\tilde{\mathbf{A}} = \mathbf{A} \setminus \{\mathbf{x}_t\}$. We note that $\tilde{\mu}$ is the mean of $\tilde{\mathbf{A}}$, and thus

$$\Sigma_{\tilde{\mathbf{A}}} = \frac{1}{n-1}\sum_{\mathbf{x}_t \in \mathbf{A}/\{\mathbf{x}_t\}} (\mathbf{x}_i - \tilde{\mu})(\mathbf{x}_i - \tilde{\mu})^\top. \quad (6)$$

Once these eigenvectors $\tilde{\mathbf{u}}_t$ are obtained, we use the complete value of cosine similarity to measure the variation of the principal directions, that is.

$$s_t = 1 - \left|\frac{\langle \tilde{\mathbf{u}}_t, \mathbf{u}\rangle}{\|\tilde{\mathbf{u}}_t\|\|\mathbf{u}\|}\right|. \quad (7)$$

This s_t can be considered as a "score of outlierness", which designate the anomaly of the target instance $\mathbf{x}_t$. We note that s_t can be also viewed as the influence of the target instance in the resulting principal direction, and a higher s_t score (closer to 1) means that the target instance is more likely to be an outlier. For a target instance, if its s_t is above some threshold, we then recognize this instance as an outlier.We refer to this procedure as a *decremental PCA* with LOO scheme for anomaly detection.

In contrast with decremental PCA with the LOO strategy, we also regard as the use of adding/duplicating a data instance of interest when applying PCA for outlier detection. This setting is particularly practical for streaming data anomaly detection problems. To be more exact, when receiving a new target instance $\mathbf{x}_t$, we solve the following PCA problem:

$$\Sigma_{\tilde{\mathbf{A}}}\tilde{\mathbf{u}}_t = \lambda\tilde{\mathbf{u}}_t, \quad (8)$$

where $\tilde{\mathbf{A}} = \mathbf{A} \cup \{\mathbf{x}_t\}$.Again, $\tilde{\mu}$ is the mean of $\tilde{\mathbf{A}}$, and the covariance matrix can be calculated as

$$\Sigma_{\tilde{\mathbf{A}}} = \frac{1}{n+1}\sum_{\mathbf{x}_t \in \mathbf{A}} (\mathbf{x}_i - \tilde{\mu})(\mathbf{x}_i - \tilde{\mu})^\top + \frac{1}{n+1}(\mathbf{x}_t - \tilde{\mu})(\mathbf{x}_t - \tilde{\mu})^\top. \quad (9)$$

After deriving the principal direction $\tilde{\mathbf{u}}_t$ of $\tilde{\mathbf{A}}$, we apply (7) and calculate the score s_t, and the outlierness of that target instance can be determined accordingly. This plan is also preferable for online anomaly detection applications, in which we need to decide whether a newly received data instance (viewed as a target instance) is an outlier. If the newly received data points are usual ones, adding such instances will not significantly affect the principal directions (and vice versa). As one might argue that it might not be sufficient to simply use the variation of the principal direction to evaluate the anomaly of the data, we will give details in the next section why our over-sampling PCA alleviates this problem and makes the online anomaly detection problem solvable.

It is worth noting that if an outlier instance is far away from the data cloud (of normal data instances) but along the direction of its dominant eigenvector, our technique will not be able to identify such anomaly. It is value pointing out that, such an outlier in fact indicates the anomaly in most (if not all) of the feature attributes. This is that, most of the feature attributes of this instance are way beyond the normal range/distribution (in the same scale) of each feature variable. As a consequence, the anomaly of such a data input can be easily detected by simple outlier detection methods such as single feature variable thresholding. For ex, one can compute the mean and standard deviation of the normal data instances projected onto the dominant eigenvector. For a contribution data point, if its projected coefficient onto this eigenvector is beyond two or three times of the standard deviation (i.e., away from 95.45% or 99.73% of normal data instances), it will be standard as an outlier.

We would also like to point out that, such an outlier instance might not be open in practical outlier detection scenarios due to the violation of system limitations. Enchanting network traffic/behavior anomaly detection for example, one might take power, bandwidth, capacity (data rates), and other parameters of a router/switch as the features to be experiential. If a data instance is far away from the normal data cloud but along its principal direction, we contain most of these router parameters simultaneously above their normal ranges, as some of them might even exceed their physical limitations. So, the anomaly of this input will be easily detected by system designs and does not require a more advanced outlier detection method like ours.

4. Over-Sampling Pca For Anomaly Detection

For practical anomaly detection problems, the size of the data set is usually large, and so it might not be easy to observe the variation of principal directions caused by the presence of a single outlier. Also, in the above PCA framework for anomaly detection, we need to do n PCA analysis for a data set with n data instances in a p-dimensional space, which is not computationally possible for large-scale and online problems. Our proposed over-sampling PCA (osPCA) together with an online updating strategy will address the above issues.

4.1 Over-Sampling Principal Components Analysis (osPCA)

As mentioned earlier, when the size of the dataset is large, adding (or removing) a single outlier instance will not significantly affect the resulting principal direction of the data. Therefore, we advance the over-sampling strategy and present an over-sampling PCA (osPCA) algorithm for large-scale anomaly detection problems.

The proposed osPCA scheme will duplicate the target instance multiple times, and the idea is to intensify the effect of outlier rather than that of normal data. As it might not be sufficient to perform anomaly detection simply based on the most dominant eigenvector and ignore the remaining ones, our online osPCA method plan to efficiently determine the anomaly of each target instance without sacrificing

computation and memory efficiency. More specifically, if the target instance is an outlier, this over sampling scheme allows us to overemphasize its effect on the most dominant eigenvector, and so we can focus on extracting and approximating the dominant principal direction in an online fashion, as an alternative of calculating multiple eigenvectors carefully.

We now give the detailed formulation of the osPCA. Suppose that we over-sample the target instance ˜*n* times, the associated PCA can be formulated as follows

$$\Sigma_{\tilde{\mathbf{A}}}\tilde{\mathbf{u}}_t = \lambda\tilde{\mathbf{u}}_t, \qquad (10)$$

where $\tilde{\mathbf{A}} = \mathbf{A} \cup \{\mathbf{x}_t, \ldots, \mathbf{x}_t\} \in \mathbb{R}^{(n+\tilde{n})\times p}$. The mean of $\tilde{\mathbf{A}}$ is $\tilde{\mu}$, and thus

$$\begin{aligned}\Sigma_{\tilde{\mathbf{A}}} &= \frac{1}{n+\tilde{n}}\sum_{\mathbf{x}_i\in\mathbf{A}}\mathbf{x}_i\mathbf{x}_i^\top + \frac{1}{n+\tilde{n}}\sum_{i=1}^{n}\mathbf{x}_t\mathbf{x}_t^\top - \tilde{\mu}\tilde{\mu}^\top. \\ &= \frac{1}{1+r}\frac{\mathbf{A}\mathbf{A}^\top}{n} + \frac{r}{1+r}\mathbf{x}_t\mathbf{x}_t^\top - \tilde{\mu}\tilde{\mu}^\top. \qquad (11)\end{aligned}$$

In this osPCA framework, we will duplicate the target instance ˜*n* times (e.g. 10% of the size of the original data set), and we will calculate the score of outlierness *st* of that target instance, as defined in (7). If this score is above some predetermined threshold, we will reflect on this instance as an outlier. With this over-sampling strategy, if the target instance is a normal data (see Fig. 2a for example), we will observe negligible changes in the principal directions and the mean of the data. The case of over-sampling an abnormal instance is shown in Fig. 2b. It is worth noting that the use of osPCA not only determines outliers from the existing data, it can be relate to anomaly detection problems with streaming data or those with online requirements.

Clearly, the major concern is the computation cost of calculating or updating the principal directions in largescale problems. We will discuss this issue and propose our solutions in the following sections.

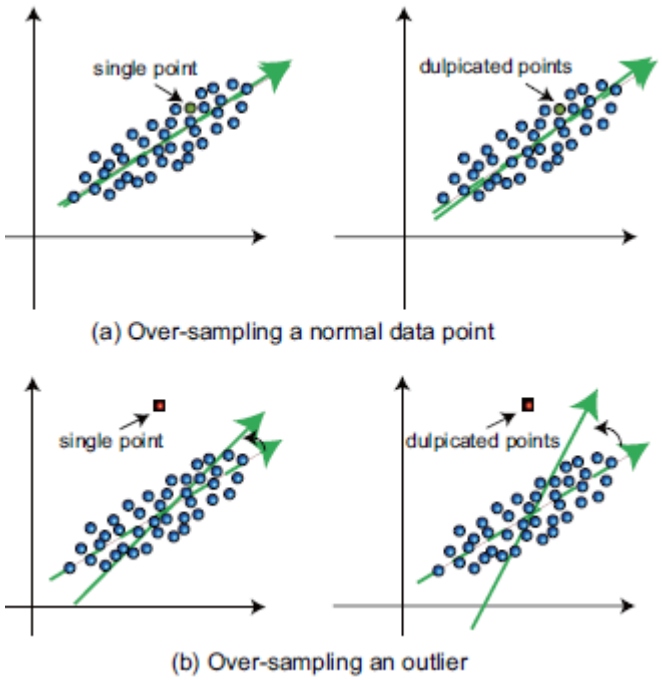

Figure 2: The effect of an over-sampled normal data or outlier instance on the principal direction

4.2 Effects of the Over-sampling Ratio on osPCA

Using the proposed osPCA for anomaly detection, the over-sampling ratio *r* in (11) will be the parameter for the user to be determined. Since there is no training or validation data for practical anomaly detection problems, one cannot make cross-validation or similar strategies to determine this parameter in advance.

When applying our osPCA to detect the presence of outliers, estimate the principal direction of the updated data matrix (with over-sampled data introduced) can be considered as the task of eigenvalue decomposition of the perturbed covariance matrix. Hypothetically, the degree of perturbation is dependent on the oversampling ratio *r*, and the sensitivity of deriving the associated dominant eigenvector can be studied as follows.

To discuss such perturbation effects, let $\mathbf{A} = [\mathbf{x}_1^\top; \mathbf{x}_2^\top; \cdots; \mathbf{x}_n^\top] \in \mathbb{R}^{n\times p}$ as the data matrix, where each row represents a data instance in a *p* dimensional space, and *n* is the number of the instances. For at target instance $\mathbf{x}_t$ over-sampled $\tilde{n}$ times, we can derive the resulting covariance matrix. Let $\epsilon = \frac{\tilde{n}}{n+\tilde{n}'}$, we calculate the perturbed data covariance matrix $\tilde{\Sigma}'_\epsilon$ as

$$\begin{aligned}\Sigma_\epsilon &= \frac{1}{n+\tilde{n}}\{\sum_{i=1}^{n}(\mathbf{x}_i-\boldsymbol{\mu}_\epsilon)(\mathbf{x}_i-\boldsymbol{\mu}_\epsilon)^\top \\ &\quad +\sum_{i=1}^{\tilde{n}}(\mathbf{x}_t-\boldsymbol{\mu}_\epsilon)(\mathbf{x}_t-\boldsymbol{\mu}_\epsilon)^\top\} \\ &= \frac{1}{n+\tilde{n}}\{\sum_{i=1}^{n}(\mathbf{x}_i-\boldsymbol{\mu})(\mathbf{x}_i-\boldsymbol{\mu})^\top \\ &\quad +\sum_{i=1}^{\tilde{n}}(\mathbf{x}_t-\boldsymbol{\mu})(\mathbf{x}_t-\boldsymbol{\mu})^\top\}+O(\epsilon^2) \\ &= (1-\epsilon)\Sigma+\epsilon\Sigma_{\mathbf{x}_t}+O(\epsilon^2). \qquad (12)\end{aligned}$$

Note that $\|\boldsymbol{\mu}_\epsilon-\boldsymbol{\mu}\|=\epsilon\|\mathbf{x}_t-\boldsymbol{\mu}\|=O(\epsilon)$ and $\|(\boldsymbol{\mu}_\epsilon-\boldsymbol{\mu})(\boldsymbol{\mu}_\epsilon-\boldsymbol{\mu})^\top\|=O(\epsilon^2)$. Based on (12), we can observe that a normal data instance (i.e., close to μ) would make $\epsilon\to 0$ and $\|\Sigma_{\mathbf{x}_t}\|\to 0$, and thus the perturbed covariance matrix Σ_ϵ will not be remarkably different from the original one Σ. On the other hand, if an outlier instance (i.e., far away from μ) is a target input to be over-sampled, Σ_ϵ will be significantly affected by $\Sigma_{\mathbf{x}_t}$ (due to a larger ϵ), and thus the derived principal direction will also be remarkably different from the one without noteworthy perturbation. More information of this study, which focuses on the effects of the perturbed data on the resulting covariance matrix, can be found in [21] (see Lemma 2.1 in [21]).

The above theoretical analysis supports our use of the variation of the dominant eigenvector for anomaly detection. Using (12), while we can theoretically estimate the perturbed eigenvector u_ϵ with a residual for an oversampled target instance, such an opinion is associated with the residual term $O(\epsilon^2)$, and ϵ is a function of $\tilde{n}$ (and thus a function of the over-sampling ratio *r*). Based on (12), while a larger *r* values will more significantly affect the resulting principal direction, the occurrence of the residual term prevents us from performing further theoretical evaluation or comparisons (e.g., threshold determination). Nevertheless, one can suppose to detect an outlier instance using the above strategy. No matter how larger the over-sampling ratio *r* is, the presence of outlier data will affect more on the dominant eigenvector than a normal instance does. In perform, we also find that our anomaly detection performance is *not* sensitive to the choice of the over-sampling ratio *r*.

4.3 The Power Method for osPCA

Typically, the solution to PCA is determined by solving an eigenvalue decomposition problem. In the LOO situation, one will need to solve the PCA and to calculate the principal directions *n* times for a data set with *n* instances. This is very computationally expensive, and prohibits the sensible use of such a framework for anomaly detection.

It can be observed that, in the PCA formulation with the LOO setting, it is not necessary to re-compute the covariance matrices for each PCA. This is since when we duplicate a data point of importance, the difference between the updated covariance matrix and the original one can be easily determined. Let $\mathbf{Q}=\frac{\mathbf{A}\mathbf{A}^\top}{n}$ be the outer product matrix and $\mathbf{x}_t$ be the target instance (to be oversampled), we use the following method to update the mean $\tilde{\boldsymbol{\mu}}$ and the covariance matrix $\Sigma_{\tilde{\mathbf{A}}}$:

$$\tilde{\boldsymbol{\mu}}=\frac{\boldsymbol{\mu}+r\cdot\mathbf{x}_t}{1+r} \qquad (13)$$

And

$$\Sigma_{\tilde{\mathbf{A}}}=\frac{1}{1+r}\mathbf{Q}+\frac{r}{1+r}\mathbf{x}_t\mathbf{x}_t^\top-\tilde{\boldsymbol{\mu}}\tilde{\boldsymbol{\mu}}^\top, \qquad (14)$$

where $r<1$ is the parameter controlling the size when over-sampling $\mathbf{x}_t$. From (14), we can see that one only needs to keep the matrix **Q** when calculating $\Sigma_{\tilde{\mathbf{A}}}$, and there is no need to re-compute the entire covariance matrix in this LOO framework.

Once the update covariance matrix $\Sigma_{\tilde{\mathbf{A}}}$ is obtained, the principal directions can be get by solving the eigenvalue decomposition problem of each of the matrices $\Sigma_{\tilde{\mathbf{A}}}$. In order to alleviate this computation load, we apply the well-known *power method* [20], which is a simple iterative algorithm and does not compute matrix decomposition. This technique starts with an initial normalized vector $\mathbf{u}_0$, which could be an approximation of the dominant eigenvector or a nonzero random vector. Next, the new $\mathbf{u}_{k+1}$ (a better approximated version of the dominant eigenvector) is updated by

$$\mathbf{u}_{k+1}=\frac{\Sigma_{\tilde{\mathbf{A}}}\mathbf{u}_k}{\|\Sigma_{\tilde{\mathbf{A}}}\mathbf{u}_k\|}. \qquad (15)$$

The sequence $\{\mathbf{u}_k\}$ converges under the assumption that the dominant eigenvalue of $\Sigma_{\tilde{\mathbf{A}}}$ is markedly larger than others. From (15), it is clear that the power method only requires matrix multiplications, not decompositions; so, the use of the power method can alleviate the computation cost in calculating the dominant principal direction.

We note that, in order to avoid keeping the data covariance matrix $\Sigma_{\tilde{\mathbf{A}}}\in\mathbb{R}^{p\times p}$ during the entire updating process, we can first compute $\mathbf{y}=\mathbf{A}\mathbf{u}_{k-1}$ and then calculate $\mathbf{u}_k=\mathbf{y}^\top$. As a result, when applying this technique for the power method, we do not need to calculate and store the covariance matrix. Though, as can be seen from the above process, we still need to keep the data matrix **A** (with the memory cost $O(n\times p)$) for the matrix-vector multiplication. Also, this multiplication needs to be performed for each iteration of the power method.

In our anomaly detection framework, we only regard as the first principal component and evaluate its variation in computing the score of outlierness of each sample. One could use the deflation process [20] if other principal directions besides the dominant one need to be determined.

4.4 Least Squares Approximation and Online Updating for osPCA

In the previous subsection, we apply a matrix update technique in (14) and the power method to solve our over-sampling PCA for outlier detection. Though, the major concern of the power method is that it does not guarantee a fast convergence, still if we use prior principal directions as its initial solutions. In addition, the use of power method still requires the user to keep the entire covariance matrix, which prohibits the troubles with high dimensional data or with limited memory resources. Inspired by [22], [23], we propose an online updating algorithm to calculate the dominant eigenvector when over-sampling a target instance. We now talk about the details of our proposed algorithm. Recall that, in Section 3, PCA can be considered as a problem to minimize the reconstruction error

$$\min_{\mathbf{U}\in\mathbb{R}^{p\times k},\mathbf{U}^\top\mathbf{U}=\mathbf{I}} J(\mathbf{U}) = \sum_{i=1}^{n} \|\bar{\mathbf{x}}_i - \mathbf{U}\mathbf{U}^\top\bar{\mathbf{x}}_i\|^2, \qquad (16)$$

where $\bar{\mathbf{x}}_i$ is $(\mathbf{x}_i - \mu)$, $\mathbf{U}$ is the matrix consisting of k dominant eigenvectors, and $\mathbf{U}\mathbf{U}^\top\bar{\mathbf{x}}_i$ is the reconstructed version of $\bar{\mathbf{x}}_i$ using the eigenvectors in $\mathbf{U}$. The above reconstruction error function can be further approximated by a least squares form [24]:

$$\begin{aligned}\min_{\mathbf{U}\in\mathbb{R}^{p\times k},\mathbf{U}^\top\mathbf{U}=\mathbf{I}} J_{ls}(\mathbf{U}) &= \sum_{i=1}^{n} \|\bar{\mathbf{x}}_i - \mathbf{U}\mathbf{U}'^\top\bar{\mathbf{x}}_i\|^2 \quad (17)\\ &= \sum_{i=1}^{n} \|\bar{\mathbf{x}}_i - \mathbf{U}\mathbf{y}_i\|^2,\end{aligned}$$

where $\mathbf{U}'$ is the approximation of $\mathbf{U}$, and thus $\mathbf{y}_i = \mathbf{U}'^\top\bar{\mathbf{x}}_i \in \mathbb{R}^k$ is the approximation of the projected data $\mathbf{U}^\top\bar{\mathbf{x}}_i$ in the lower k dimensional space. Based on this method, the reconstruction error has a quadratic form and is a function of $\mathbf{U}$, which can be computed by solving a least squares problem. The deception for this least squares problem is the approximation of $\mathbf{U}^\top\bar{\mathbf{x}}_i$ by $\mathbf{y}_i = \mathbf{U}'^\top\bar{\mathbf{x}}_i$. In an online setting, we approximate each $\mathbf{U}_i^\top\bar{\mathbf{x}}_i$ by its previous solution $\mathbf{U}_{i-1}^\top\bar{\mathbf{x}}_i$ as follows

$$\min_{\mathbf{U}_t\in\mathbb{R}^{p\times k},\mathbf{U}^\top\mathbf{U}=\mathbf{I}} J_{ls}(\mathbf{U}_t) = \sum_{i=1}^{t} \|\bar{\mathbf{x}}_i - \mathbf{U}_i\mathbf{y}_i\|^2, \qquad (18)$$

where $\mathbf{y}_i = \mathbf{U}_{i-1}^\top\bar{\mathbf{x}}_i$. This projection approximation provides a fast calculation of principle directions in our over-sampling PCA. Linking this least squares form to our online over-sampling plan, we have

$$\min_{\tilde{\mathbf{U}}\in\mathbb{R}^{p\times k},\mathbf{U}^\top\mathbf{U}=\mathbf{I}} J_{ls}(\tilde{\mathbf{U}}) \approx \sum_{i=1}^{n} \|\bar{\mathbf{x}}_i - \tilde{\mathbf{U}}\mathbf{y}_i\|^2 + \|\bar{\mathbf{x}}_t - \tilde{\mathbf{U}}\mathbf{y}_t\|^2. \qquad (19)$$

In (19), $\mathbf{y}_i$ and $\mathbf{y}_t$ are approximated by $\mathbf{U}^\top\bar{\mathbf{x}}_i$ and $\mathbf{U}^\top\bar{\mathbf{x}}_t$ in that order, where $\mathbf{U}$ is the solution of the original PCA (which can be calculated in advance), and $\bar{\mathbf{x}}_t$ is the target instance. While over-sampling the target instance $\tilde{n}$ times, we approximate the solution $\tilde{\mathbf{U}}$ by solving the following optimization problem

$$\min_{\tilde{\mathbf{U}}\in\mathbb{R}^{p\times k},\mathbf{U}^\top\mathbf{U}=\mathbf{I}} J_{ls}(\tilde{\mathbf{U}}) \approx \sum_{i=1}^{n} \|\bar{\mathbf{x}}_i - \tilde{\mathbf{U}}\mathbf{y}_i\|^2 + \tilde{n}\|\bar{\mathbf{x}}_t - \tilde{\mathbf{U}}\mathbf{y}_t\|^2. \qquad (20)$$

Equivalently, we convert the above problem into the following form

$$\min_{\tilde{\mathbf{U}}\in\mathbb{R}^{p\times k},\mathbf{U}^\top\mathbf{U}=\mathbf{I}} J_{ls}(\tilde{\mathbf{U}}) \approx \beta\Big(\sum_{i=1}^{n} \|\bar{\mathbf{x}}_i - \tilde{\mathbf{U}}\mathbf{y}_i\|^2\Big) + \|\bar{\mathbf{x}}_t - \tilde{\mathbf{U}}\mathbf{y}_t\|^2, \qquad (21)$$

where β can be regarded as a weighting factor to suppress the information from existing data. Note that the relation between β and the over-sampled number

Algorithm 1 Anomaly Detection via Online Over-sampling PCA

Require: The data matrix $\mathbf{A} = [\mathbf{x}_1^\top; \mathbf{x}_2^\top; \cdots; \mathbf{x}_n^\top]$ and the weight β.
Ensure: Score of outlierness $\mathbf{s} = [s_1 s_2 \cdots s_n]$. If s_i is higher than a threshold, $\mathbf{x}_i$ is an outlier.

Compute first principal direction $\mathbf{u}$ by using (18);
Keep $\bar{\mathbf{x}}_{proj} = \sum_{j=1}^{n} y_j\bar{\mathbf{x}}_j$ and $y = \sum_{j=1}^{n} y_j^2$ in (22);
for $i = 1$ **to** n **do**
  $\tilde{\mathbf{u}} \leftarrow \frac{\beta\bar{\mathbf{x}}_{proj} + y_i\bar{\mathbf{x}}_i}{\beta y + y_i^2}$ by (18)
  $s_i \leftarrow 1 - |\frac{\langle\mathbf{w},\mathbf{w}\rangle}{\|\tilde{\mathbf{u}}\|\|\mathbf{u}\|}|$ by (7)
end for

$\tilde{n}$ is $\beta = \frac{1}{\tilde{n}} = \frac{1}{nr}$, where r is the ratio of the oversampled number over the size of the original dataset. To improve the convergence rate, we use the resolution of the original PCA (without over-sampling data) as the initial projection matrix in (21). If only the dominant principal direction (i.e. k = 1) is of concern, we calculate the solution of $\tilde{\mathbf{u}}$ by taking the derivative of (21) with respect to $\tilde{\mathbf{u}}$, and thus we have

$$\tilde{\mathbf{u}} = \frac{\beta(\sum_{i=1}^{n} y_i\bar{\mathbf{x}}_i) + y_t\bar{\mathbf{x}}_t}{\beta(\sum_{i=1}^{n} y_i^2) + y_t^2}. \qquad (22)$$

Compared with (10) and (15), (22) provides an effective and efficient updating technique for osPCA, which allows us to decide the principal direction of the data. This modernized process makes anomaly detection in online or streaming data settings feasible. More prominently, since we only need to calculate the solution of the original PCA offline, we do *not* need to keep the whole covariance or outer matrix in the entire updating process. Formerly the final principal direction is determined, we use the cosine similarity to find out the difference between the current solution and the original one (without oversampling), and so the score of

outlierness for the target instance can be determined accordingly (as discussed in Section 3.2). The pseudo code of our *online osPCA* with the LOO strategy for outlier detection is described in Algorithm 1. It is value noting that we only need to compute $\mathbf{x}_{proj}$ and y once in Algorithm 1, and so we can further reduce the computation time when calculating $\tilde{\mathbf{u}}$.

Table 1 compares computation complexity and memory requirements of several anomaly detection methods, including fast ABOD [5], LOF [2], our previous osPCA using power method [19], and the proposed online osPCA. In this table, we list calculation and memory costs of each method when determining the outlierness of a newly received data instance (i.e., in a streaming data fashion). In favor of ABOD and LOF, the memory requirements are both $O(np)$ since they need to store the entire data matrix for the k nearest neighbor search (recall that n and p are the size and dimensionality of the data, in that order).

TABLE 1
Comparisons of our proposed osPCA (with power method and the proposed online updating technique), fast ABOD, and LOF for online anomaly detection in terms of computational complexity and memory requirements. Note that n and p are the size and dimensionality of data, respectively. The power method requires the number of iterations m, and the number of nearest neighbors k is used in both ABOD and LOF.

	osPCA [19] (power method)	Online osPCA	Fast ABOD [5]	LOF [2]
Computation complexity	$O(mp^2)$(or $O(mnp)$)	$O(p)$	$O(n^2p+k^2p)$	$O(n^2p+k)$
Memory requirement	$O(p^2)$(or $O(np)$)	$O(p)$	$O(np)$	$O(np)$

The time complexities for ABOD and LOF are $O(n^2p+k^2p)$ and $O(n^2p+k)$, in which $O(n^2p)$ is required for finding k nearest neighbors and thus is the bottleneck of the computation complexity. As for the power method, it needs to perform (15) iteratively with m times, and its time complexity in the online detection procedure for outlier detection is $O(np^2+mp^2)$ (we have $O(np^2)$ for deriving the updated covariance matrix, and $O(mp^2)$ for the implementation of the power method). Almost, we reduce the above complexity to $O(mp^2)$ by applying the covariance update trick in (14). As discussed in Section 4.3, the time complexity might be $O(mnp)$ if we choose to store the data matrix instead of keeping the covariance matrix during the updating process. As a result, the related memory requirement will be reduced from $O(p^2)$ to $O(np)$. Finally, when using our online updating technique for osPCA, we simply changing the principal direction by (22) and result in $O(p)$ for both computation complexity and memory requirement, correspondingly.

5. Conclusion

In this paper, we proposed an online anomaly detection method based on over-sample PCA. We showed that the osPCA with LOO strategy will amplify the effect of outliers, and so we can successfully use the variation of the dominant principal direction to identify the presence of rare but irregular data. When oversampling a data instance, our proposed online updating method enables the osPCA to efficiently update the principal direction without solving eigenvalue decomposition problems. Also, our method does not need to keep the entire covariance or data matrices during the online detection process. So, compared with other anomaly detection methods, our approach is able to attain satisfactory results while significantly reducing computational costs and memory requirements. Therefore, our online osPCA is preferable for online large-scale or streaming data problems.

Future research will be directed to the following anomaly detection scenarios: normal data with multiclustering structure, and data in a very high dimensional space. For the previous case, it is typically not easy to use linear models such as PCA to estimate the data distribution if there exists multiple data clusters. Also, many learning algorithms encounter the "curse of dimensionality" problem in a extremely high dimensional space. In our proposed method, even if we are able to handle high dimensional data since we do not need to compute or to keep the covariance matrix, PCA may not be preferable in estimating the principal directions for such kind of data. So, we will pursue the study of these issues in our future work.

References

[1] D. M. Hawkins, Identification of Outliers. Chapman and Hall,1980.
[2] M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in Proceeding of the 2000 ACM SIGMOD International Conference on Management of Data, 2000.
[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, no. 3, pp. 15:1–58, 2009.
[4] L. Huang, X. Nguyen, M. Garofalakis, M. Jordan, A. D. Joseph, and N. Taft, "In-network pca and anomaly detection," in Proceeding of Advances in Neural Information Processing Systems 19, 2007.
[5] H.-P. Kriegel, M. Schubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008.
[6] A. Lazarevic, L. Ert¨oz, V. Kumar, A. Ozgur, and J. Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection," in Proceedings of the Third SIAM International Conference on Data Mining, 2003.
[7] X. Song, M.Wu, and C. J. andSanjay Ranka, "Conditional anomaly detection," IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 5, pp. 631–645, 2007.

[8] S. Rawat, A. K. Pujari, and V. P. Gulati, "On the use of singular value decomposition for a fast intrusion detection system," Electronic Notes in Theoretical Computer Science, vol. 142, no. 3, pp. 215–228, 2006.
[9] W. Wang, X. Guan, and X. Zhang, "A novel intrusion detection method based on principal component analysis in computer security," in Proceeding of the International Symposium on Neural Networks, 2004.
[10] F. Angiulli, S. Basta, and C. Pizzuti, "Distance-based detection and prediction of outliers," IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 2, pp. 145–160, 2006.
[11] V. Barnett and T. Lewis, Outliers in statistical data. John Wiley& Sons, 1994.
[12] W. Jin, A. K. H. Tung, J. Han, and W. Wang, "Ranking outliers using symmetric neighborhood relationship," in Proceeding of Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2006.
[13] N. L. D. Khoa and S. Chawla, "Robust outlier detection using commute time and eigenspace embedding," in Proceeding of Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2010.
[14] E. M. Knox and R. T. Ng, "Algorithms for mining distancebased outliers in large datasets," in Proceedings of the International Conference on Very Large Data Bases, 1998.
[15] H.-P. Kriegel, P. Kr¨oger, E. Schubert, and A. Zimek, "Outlier detection in axis-parallel subspaces of high dimensional data," in Proceeding of Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2009.
[16] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," in Proceeding of ACM SIGMOD international conference on Management of data, 2001.
[17] D. Pokrajac, A. Lazarevic, and L. Latecki, "Incremental local outlier detection for data streams," in Proceeding of IEEE Symposium on Computational Intelligence and Data Mining, 2007.
[18] T. Ahmed, "Online anomaly detection using KDE," in Proceedings of IEEE conference on Global telecommunications, 2009.
[19] Y.-R. Yeh, Z.-Y. Lee, and Y.-J. Lee, "Anomaly detection via oversampling principal component analysis," in Proceeding of the First KES International Symposium on Intelligent Decision Technologies, 2009, pp. 449–458.
[20] G. H. Golub and C. F. V. Loan, Matrix Computations. Johns Hopkins University Press, 1983.
[21] R. Sibson, "Studies in the robustness of multidimensional scaling: perturbational analysis of classical scaling," Journal of the Royal Statistical Society B, vol. 41, pp. 217–229, 1979.
[22] B. Yang, "Projection approximation subspace tracking," IEEE Transaction on Signal Processing, vol. 43, pp. 95–107, 1995.

Author Profile

Garine Bindu Madhavi M.TECH(P.HD) working as assistant professor in Computer Science Engineering from CVSR College of Engineering from ANURAG GROUP OF INSTITUTIONS Venkatapur (V), Ghatkesar(M), Ranga Reddy District, Hyderabad-500088, Telangana State, India

Pachunoori Naresh received the B.Tech degree in CSE from JNTU Hyderabad 2012 and pursuing M.tech. degree in Computer science and Engineering from Anurag Group of Institutions, JNTU Hyderabad University, India