Online Anomaly Detection via Incremental Tensor Decomposition

Ming Gao[1], Yunbing Zong[2], Rui Li[3], Chengzhi An[3], Qiang Duan[3], Qingshan Yin[3], and Xiangcheng Wang[2(✉)]

[1] Baidu Knows Business Department, Baidu, Beijing, China
[2] Department of Data and Enterprise Intelligence Products, Inspur Group, Jinan, China
wangxiangcheng@inspur.com
[3] Inspur AI Research Institute, Inspur Group, Science and Technology Park, No. 1036, Langchao Road, Jinan Hi-Tech Development Zone, Jinan, People's Republic of China

Abstract. Anomaly detection, though, is a common and intensively studied data mining problem in many applications, its online (incremental) algorithm is yet to be proposed and investigated, especially with respect to the use of tensor technique. As online (incremental) learning is becoming increasingly more important, we propose a novel online anomaly detection algorithm using incremental tensor decomposition. The online approach keeps updating the model while new data arrive, in contrast to the conventional approach that requests all data to rebuild the model. In addition, the online algorithm can not only track the trend in time evolving data, but also requests less memory since only the new data is necessary for model updating. The experimental results show that the presented algorithm has sound discriminative power that is essential for anomaly detection. In addition, the number of anomalies can be flexibly adjusted by the parameters in the algorithm, which is necessary in some real-world scenarios. The effects of these parameters are also consistent using two experimental datasets.

Keywords: Tensor decomposition · Online machine learning · Anomaly detection · Incremental · Streaming

1 Introduction

Anomaly detection [3] is a classic problem existing in a variety of areas, such as fraud detection in finance. It aims at finding patterns that behave differently from the rest of the main population. Anomaly detection and outlier detection are often used interchangeably, and it does not necessarily requires labelled data. In fact, unsupervised anomaly detection is of great practical interest, since

M. Gao, Y. Zong and R. Li—Both authors contributed equally to this study.

W. Ni et al. (Eds.): WISA 2019, LNCS 11817, pp. 3–14, 2019.
https://doi.org/10.1007/978-3-030-30952-7_1

labelled data are scare to obtain in many real-world applications. We reveal several challenges in anomaly detection: (a): The definition of normality is a hard task due to the absence of labelled data; (b): The noise and trend in data lend the problem even more difficult; (c): Abnormal patterns may evolve with time.

To date, time evolving data are becoming more prevalent. At the same time, different problems, such as changing patterns, arise due to the streaming/non-stationary data. As a result, online machine learning or similarly incremental machine learning draws much attention in recent years [4]. In contrast to the conventional machine learning approach, incremental machine learning keeps updating the model with respect to the underlying characteristics of the incoming data. The classic approach trains the model using the entire data, which may result in two drawbacks, namely neglecting the time-evolving phenomenon and costing too much time. An early example of incremental machine learning is the very fast decision tree (VFDT) proposed by Domingos [5], modifying the decision tree using Hoeffding's inequality as a statistics to handle the streaming data. Massive online analysis (MOA) [2] is particularly developed for analysis of incremental data with abundant methods devised for classification, regression etc.

Commonly, machine learning and data mining algorithms deal with matrix. In some scenarios, tensor may be more suitable to represent the data, which is an extension to matrix. In the movie recommendation application, a matrix can be formed by users and items, and a tensor can be shaped by adding time as the third dimension. The relational learning via tensor approach proposed by Nickel [13] stresses that the tensor factorization can capture the essential information via the decomposed latent factors. With all information reside in a tensor, the interrelationship between different dimensions can be better captured by tensor compared to matrix. Hence, decomposing the tensor uncovers useful patterns which may not be obtained by matrix factorization.

Having mentioned the usefulness of anomaly detection, online machine learning and tensor decomposition, we propose a new anomaly detection algorithm using incremental tensor decomposition, and demonstrate the results using two datasets. This work is a contribution to the anomaly detection community by introducing the tensor decomposition in the manner of incremental learning. The unsupervised algorithm is suitable for applications where the data are multivariate and vary with time, such as data collected by sensors used in IoT scenarios.

The paper is organized as follows: The introduction is followed by related work. The third section presents the preliminary knowledge on tensor, and the incremental tensor decomposition is introduced subsequently. The experimental results on two datasets are then shown in a later section, along with results discussion. Finally, conclusions are drawn in the last section.

2 Related Work

Recently, tensor factorization has gain popularity due to its ability in finding interesting patterns in data. A survey [14] offers an overview of a range of applications based on tensor decomposition. They show that the tensor decomposition

is able to extract hidden correlations and meaningful structure in the data mining field. Furthermore, a few open challenges in tensor decomposition are highlighted for future research direction. Kolda and Bader [9] study several methods of higher-order tensor decompositions and suggest relevant applications as well as available software. CANDECOMP/PARAFAC (CP) [8] and Tucker decompositions [17] are the primary focused methods. Moreover, they mention that a flourish of research focuses on more efficient and better methods and a range of applications is developed. A similar work [16] introduces an incremental tensor analysis (ITA) that efficiently computes a summary for high-order data and finds latent correlations. They propose three related methods that are dynamic, streaming and window-based, showing significant gain in performance. Another work conducted by Sun *et al.* [15] further shows experiments on anomaly detection and multi-way latent semantic indexing using two real-world large datasets. The result shows that the dynamic tensor analysis and streaming tensor analysis are effective and efficient. They can find interesting pattern, such as outliers. Most recently, Gujral *et al.* [7] introduce a sample-based batch incremental tensor decomposition algorithm (SamBasTen). This algorithm can update the existing decomposition without recomputing the entire decomposition by summarizing the existing tensor and incoming updates. Then, it updates the reduced summary space. This method can be regarded as an approach to decomposing the tensor incrementally, rather than an approach devised for anomaly detection.

A research work [6] suggests an online self-organizing map (SOM) to model the large number of behavior patterns in the crowed. The proposed online learning method has been proved to efficiently reduce the false alarms while still keep the ability to detect most of the anomalies. In 2011, Li *et al.* [12] presents a tensor-based incremental learning algorithm for anomaly detection in background modeling. Compared to the other vector-based methods, this algorithm is a robust tensor subspace learning algorithm that has ability to fit the variation of appearance model via adaptively updating the tensor subspace. Their experimental results show the robustness of tensor subspace-based incremental learning for anomaly detection. An unsupervised incremental learning method [18] was developed to detect maritime traffic patterns. It can automatically derive knowledge of maritime traffic in an unsupervised way without any Automatic Identification System (AIS) description. In 2014, Rikard Laxhammar and Göran Falkma [10] propose an improved online learning and sequential anomaly detection algorithm based on their previous works. The new algorithm is called Sequential Hausdorff Nearest-Neighbor Conformal Anomaly Detector (SHNN-CAD) that is a kind of parameter-light anomaly detection algorithm. It has good performance on unsupervised online learning and sequential anomaly detection in trajectories. The biggest advantage of SHNN-CAD is the ability to achieve competitive classification performance by utilizing minimum parameter tuning. A review [19] introduces the challenges, current techniques and applications with respect to the online aggregation.

3 Methods

Prior to introducing the proposed method, we first briefly reveal the related knowledge as background information.

3.1 Background Knowledge

Tensor is a natural extension to matrix, with N distinct dimensions (also known as orders, modes, ways). The data used in this study can be denoted as an order-3 tensor $\mathcal{T} \in \mathbb{R}^{N \times P \times Q}$, where N, P, Q are the dimensions. n, p, q can take on the specific value in the N, P and Q respectively. An order-3 tensor $\mathcal{T}$ can be factorized (decomposed) into three components bases (factor matrices A, B, C) with a pre-defined rank R. The CANDECOMP/PARAFAC (CP) [8] decomposition[1] is employed in this work. In contrast to the tensor decomposition, principal component analysis (PCA) is a classic unsupervised dimensionality reduction technique, being applied to numerous applications such as face recognition. PCA assumes that the low-dimensional manifold is an affine space, transforming the data into another linear space. The original data $X \in \mathbb{R}^{N \times M}$ can be reduced to new data $X \in \mathbb{R}^{N \times R}$ ($R \ll M$) by the projection matrix $U \in \mathbb{R}^{R \times M}$. Figure 1 (a) shows the basic concepts relevant to tensor, while sub-figures (b) and (c) demonstrate how the tensor data are constructed in our experiments.

$$\mathcal{T} \approx \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r. \tag{1}$$

Equation 1 suggests that the tensor $\mathcal{T}$ can be computed by the outer product ("$\circ$") of $\mathbf{a}_r \in \mathbb{R}^N$, $\mathbf{b}_r \in \mathbb{R}^P$ and $\mathbf{c}_r \in \mathbb{R}^Q$. Given the rank R (dimension after decomposition), the factor matrices are in the form of $A = [\mathbf{a}_1\ \mathbf{a}_2 \cdots \mathbf{a}_R] \in \mathbb{R}^{N \times R}$, $B = [\mathbf{b}_1\ \mathbf{b}_2 \cdots \mathbf{b}_R] \in \mathbb{R}^{P \times R}$ and $C = [\mathbf{c}_1\ \mathbf{c}_2 \cdots \mathbf{c}_R] \in \mathbb{R}^{Q \times R}$ respectively. Generally, the A, B, C are normalized by a scalar λ_r, representing some latent structure in the data such that they can be used to perform data mining task [14].

3.2 Incremental Tensor Decomposition

According to the work proposed by Kolda [9], the CP decomposition tries to approximate the tensor $\mathcal{T}$ with R components, i.e.,

$$\min_{\hat{\mathcal{T}}} \|\mathcal{T} - \hat{\mathcal{T}}\| \quad \text{with} \quad \hat{\mathcal{T}} = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r. \tag{2}$$

The solution of Eq. 2 can be reduced to a least-squares problem by fixing all but one matrix using the alternating least squares (ALS) approach. As suggested in the work [9], $\mathcal{T}_1^{A,BC}$ is roughly equal to $A(C \odot B)^T$. $\mathcal{T}_1^{A,BC} \in \mathbb{R}^{N \times PQ}$ represents that the data $\mathcal{T}_1$ is unfolded by concatenating the second and third

[1] Tensor decomposition and tensor factorization (TF) are often used interchangeably.

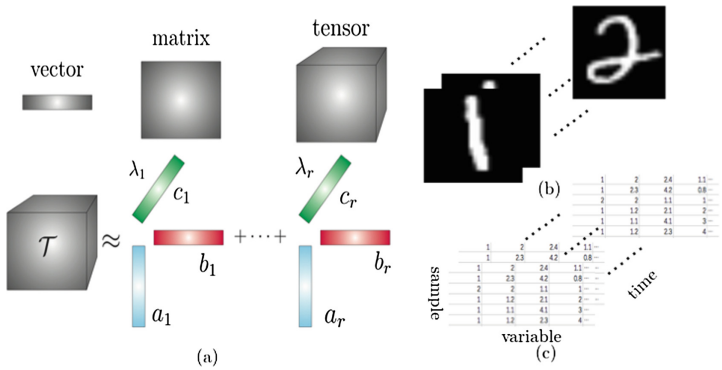

Fig. 1. (a): Illustration of vector, matrix, tensor and CP decomposition. (b): The construction of tensor data using MNIST dataset. (c): The construction of tensor data using the real-world (in-house) dataset.

mode (order) into a long matrix, while keeping the first mode intact[2]. Similarly, $\mathcal{T}_1^{B,AC} \in \mathbb{R}^{P \times NQ}$ is achieved by unfolding the first and third mode while keeping the second mode unchanged. According to the Eq. 2, the optimal solution is then boiled down to $\hat{A} = \mathcal{T}_1^{A,BC}[(C \odot B)^T]^+$. One of the core parts in our algorithm is to keep updating the matrix $\hat{A}$ (similarly $\hat{B}$ and $\hat{C}$) to perform online anomaly detection. To simplify the computation, we give the detailed following proof showing that $\hat{A}$ can be calculated straightforwardly[3].

$$\hat{A} = \mathcal{T}_1^{A,BC}[(C \odot B)^T]^+ \tag{3}$$

$$= \mathcal{T}_1^{A,BC}[((C^T C) \circ (B^T B)) * (C \odot B)^+]^+ \tag{4}$$

$$\text{due to } (C \odot B)^T (C \odot B) = (C^T C) \circ (B^T B), \text{cf. [11]} \tag{5}$$

$$\text{hence } (C \odot B)^T = ((C^T C) \circ (B^T B)) * (C \odot B)^+ \tag{6}$$

$$= \mathcal{T}_1^{A,BC}(C \odot B)^+((C^T C) \circ (B^T B)) \tag{7}$$

The detailed algorithm description and a graphical illustration are presented in Algorithm 1 and Fig. 2 respectively. It is essential to explain some key parameters in the algorithm before elaborating the approach.

1. S: mini-batch, a chunk of streaming data that are supposed to be processed.
2. T: reference interval length, every T data points are selected as reference (non-anomaly) points for similarity calculation.

[2] We use A, B, C on the superscripts to represent the respective modes in the tensor $\mathcal{T}$.
[3] "$\odot$" denotes the Khatri–Rao product, "$\circ$" denotes the vector outer product, "$*$" denotes the Hadamard product that is the elementwise matrix product, the superscript "$^+$" denotes pseudo inverse.

Algorithm 1. Online Anomaly Detection via Incremental Tensor Decomposition

Data: Input: $\mathcal{T}_1$: the historical data used for tensor decomposition, $\mathcal{T}_2$: the streaming data. R: the dimension after decomposition. S: the mini-batch streaming data. T: the length of the interval for selecting reference training points. δ: threshold used for anomaly detection.

Result: Output: detected anomaly time point OAD_t of a certain instance OAD_i.

The first phase infers the decomposed matrices A, B, C, and detects all anomaly points using historical data.

$A, B, C \leftarrow$ decomposes the historical data $\mathcal{T}_1$ using CP decomposition.

$\hat{A} \leftarrow \mathcal{T}_1^{A,BC} \times (C \odot B) * (B^T B * C^T C)^+$.

$\mathcal{T}_1^T \leftarrow$ sampled training (reference) data points in the $\mathcal{T}_1$ according to the defined interval T.

for $i = 1 : |\mathcal{T}_1|$, *"$|\bullet|$" denotes the cardinality of a set* **do**

  for $j = 1 : |\mathcal{T}_1^T|$ **do**

    $\text{similarity}(i, j) = \sqrt{\sum(\hat{A}(i,:) - \hat{A}(\mathcal{T}_1^T(j),:))^2}$

  end

end

$d \in \mathbb{R}^{N \times 1} \leftarrow \min(\text{similarity}(i, :))$, the smallest distance of each point to sampled data

$d_{\max} \leftarrow \max(\max(\text{similarity}(i, :)))$, the greatest distance used for distance normalization

$d \leftarrow \frac{d}{d_{\max}}$

anomalies $\leftarrow$ for all the $d > \delta$

The second phase detects anomalies on the incoming streaming data.

while $\mathcal{T}_2$ *is not empty* **do**

  $i = i + 1$, new instances in $\mathcal{T}_2$ go to $\mathcal{T}_1$ for following similarity computation

  if $(i - 1 \mod T) = 0$ **then**

    append index i to the $\mathcal{T}_1^T$

  end

  $\hat{A}(i,:) \leftarrow \mathcal{T}_2^{A,BC} \times (C \odot B) * (B^T B * C^T C)^+$.

  for $i = 1 : |\mathcal{T}_1|$ **do**

    for $j = 1 : |\mathcal{T}_1^T|$ **do**

      $\text{similarity}(i, j) = \sqrt{\sum(\hat{A}(i,:) - \hat{A}(\mathcal{T}_1^T(j),:))^2}$

    end

  end

  $d_{\text{score}} \leftarrow \min(\text{similarity}(i, :))$

  $d^s_{\max} \leftarrow \max(\max(\text{similarity}(i, :)))$, the s in $d^s_{\max}$ denotes the distance in the streaming data

  $d_{\text{score}} \leftarrow \frac{d_{\text{score}}}{d^s_{\max}}$

end

for $j = 1 : |S|$ **do**

  $distance = \sqrt{\frac{\sum(\mathcal{T}_2(i,j,:) - \mathcal{T}_2(i-1,j,:))^2}{\sum \mathcal{T}_2(i,j,:)}}$

  if $distance > \delta$ **then**

    collect anomaly time point and the instance as results

  end

end

$\mathcal{T}_1 \leftarrow \mathcal{T}_1 + \mathcal{T}_2$

$\hat{A} \leftarrow \mathcal{T}_1^{A,BC} \times (C \odot B) * (B^T B * C^T C)^+$

$\hat{B} \leftarrow \mathcal{T}_1^{B,AC} \times (C \odot A) * (A^T A * C^T C)^+$.

$\hat{C} \leftarrow \mathcal{T}_1^{C,AB} \times (B \odot A) * (A^T A * B^T B)^+$.

Go to the while loop to continue the process.

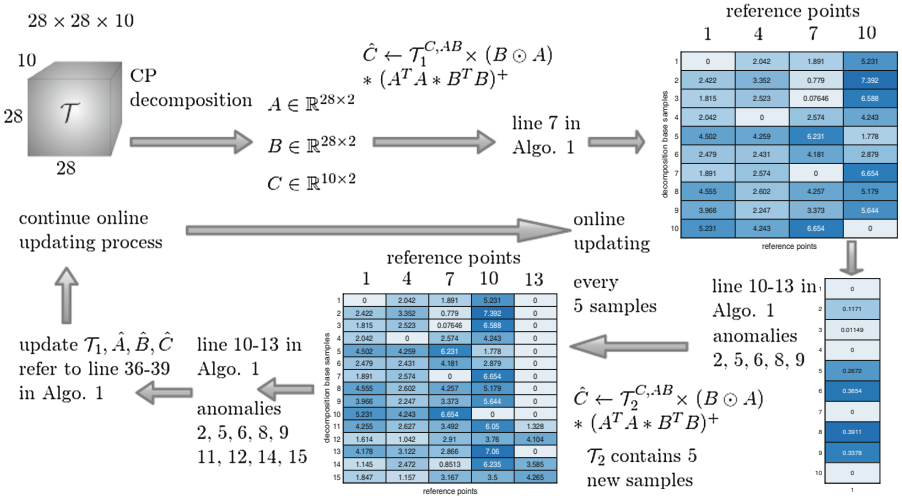

Fig. 2. An illustrative workflow of proposed online anomaly detection algorithm.

3. δ: threshold, a data point is abnormal if the similarity is greater than the threshold.
4. R: dimension after CP decomposition.
5. $\mathcal{D}_1$: the historical data (or base data) used for CP decomposition.
6. $\mathcal{D}_2$: the streaming data to be processed.

We take the MNIST dataset (cf. Sect. 4.1) as a running example (cf . Fig. 2), setting parameters $S = 5$, $T = 3$, $\delta = 0.1$, $R = 2$, with 10 examples (images) in $\mathcal{T}_1$, i.e., $\mathcal{T}_1 \in \mathbb{R}^{28\times 28\times 10}$. We format the tensor data by adding images on the third dimension, with an aim to detect images that are not "1". Therefore, the matrix $\hat{C}$ should be used and updated to compute the abnormal level (dissimilarity). Alternatively, we would then compute the matrix $\hat{A}$ if we form the tensor as $\mathcal{T}_1 \in \mathbb{R}^{10\times 28\times 28}$. After CP decomposition, we obtain the decomposed matrices A and B that can further approximate the $\hat{C}$. It is worthy mentioning that C and $\hat{C}$ are numerically very close. In the subsequent procedures, $\hat{C}$ is constantly updated so that we also apply $\hat{C}$ in the very beginning. $\mathcal{T}_1 \in \mathbb{R}^{28\times 28\times 10}$ is transformed into a matrix $\mathcal{T}_1^{C,AB} \in \mathbb{R}^{10\times 784}$ along the first and second modes ($28 \times 28 = 784$), while the third mode is remained. Within the first ten samples in $\mathcal{T}_1$, sample points $1, 4, 7, 10$ are chosen as the reference (training, normal) points because the reference interval length $T = 3$. Thus, the distance between each data point to the reference point can be readily calculated by the Euclidean distance based on $\hat{A}$, cf. line 7 in Algorithm 1. Taking the smallest distance values for each line in the similarity matrix (line 10 in Algorithm 1), we receive a vector that can be directly used for anomaly judgement using the δ. The anomaly point is determined if its minimum distance to all the reference points is greater than the pre-defined threshold, which suggests that it is dissimilar to normal points and hence should be noted as anomaly. Calculation of the Euclidean distance

using decomposed factor matrices directly suggests the similarity between data points, since the decomposed matrices inherently capture some latent structural information in the data. In the newly coming data (every $S = 5$ samples), the $13th$ data point is picked as a reference point based on the previous 10 base points. New anomalies are found given an updated similarity matrix. Finally, the key matrices $\hat{A}, \hat{B}, \hat{C}$ are recalculated in the presence of new data.

4 Experiments

4.1 Datasets and Baseline Methods

To demonstrate the effectiveness of the proposed method, we use one benchmark and an in-house dataset, which can be formed in the shape of tensor. The use of MNIST dataset is motivated by its feasibility in transforming images into tensor data as well as with known labels. Due to the simple fact that it is hard to obtain publicly free tensor datasets which contain ground truth abnormal labels, we, therefore, employ the in-house dataset to demonstrate the experimental results.

Introduction to MNIST Dataset: The MNIST dataset is composed of handwritten digits, containing 60,000 training images with each of dimension 28×28. It has been intensively used for image classification task, in particular for deep learning. The image data can be represented as a two-dimensional matrix, and a tensor is formed when we stack the images on top of each other. In the experimental setting, we regard digit "1" as the normal class with 6742 instances, and all the rest as the abnormal class. As we know the true labels of the data, we therefore employ the F1-Score $= 2((precision \times recall)/(precision + recall))$ to compare the results.

Introduction to In-house Dataset: Different from the MNIST data, the in-house data come from a real-world scenario that contains five same type equipments running in long period of time. Ten variables represent the equipments' running condition. Therefore, time (2039 recordings), equipments (five) and variables (ten) constitute the tensor data of size $2039 \times 5 \times 10$ as depicted in Fig. 1 (c). The anomaly can first be detected by the time. Further, we are able to know which equipment is running abnormally by calculating the distance (using variables) to the neighbouring time points. However, the dataset is in shortage of ground truth anomaly labels, although there may be some abnormal states in the data. Hence, it is a completely unsupervised anomaly detection task.

Introduction to Compared Methods: We compare the methods with different updating batch sizes, namely Online Anomaly Detection (OAD) with mini-batch of size 10, i.e., OAD-10 and so on. Regarding the MNIST, we apply the well-known k-means clustering as a baseline comparison, since both k-means and the presented approach are unsupervised algorithms. We used the Matlab tensor toolbox [1] implementation in the experiments. To the best of our knowledge, similar algorithms using tensor for online anomaly detection are very rare. Thus, k-means is the only chosen compared method.

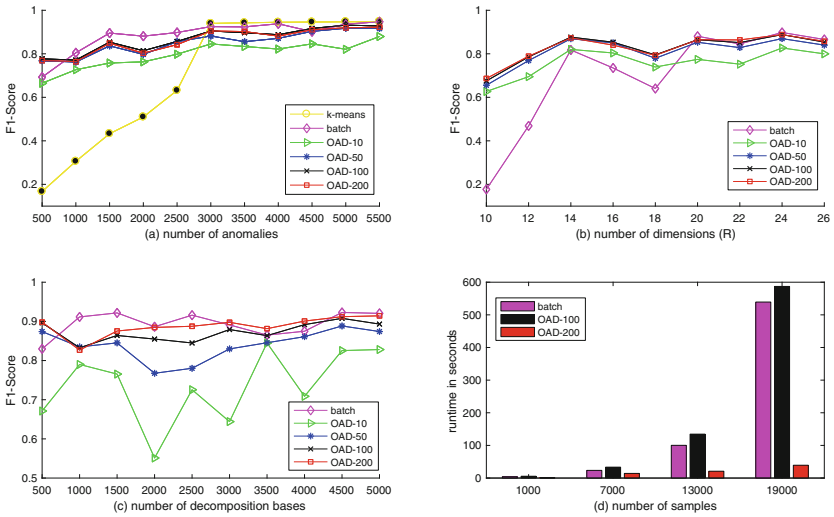

Fig. 3. MNIST dataset results. (a): The F1-Score affected by number of anomalies, $R = 20$, $T = 20$, $\delta = 0.1$. (b): The F1-Score affected by the number of dimensions (R), $T = 20$, $\delta = 0.1$, $\mathcal{T}_1$ contains 5000 samples as initial base, $\mathcal{T}_2$ contains 6742 anomalies and 1742 normal points. (c): The F1-Score affected by the size of initial decomposition bases, $T = 20$, $\delta = 0.1$, decomposition base $\mathcal{T}_1$ and $\mathcal{T}_2$ contain 13484 samples as a whole. (d): Runtime comparison.

4.2 Experimental Results

In Fig. 3 (a), the number of anomalies is randomly sampled from all digits except "1", i.e., the rest numbers are treated as abnormal ones. As the number of anomalies increases, the k-means ($k = 2$) performs gradually better which means that k-means has trouble in separating abnormal points when the two clusters are too imbalanced. The batch method, without online updating, seems to behave the best. We see better performance when the size of mini-batches increases. Figure 3 (b) reveals that we may obtain poor results if the R is set to a low value, especially for the batch method. Figure 3 (c) illustrates that the performance tend to be poor when the mini-batch size is small. In terms of the running time, the time drops dramatically when the mini-batch reaches 200. In fact, the smaller the mini-batch the longer running time.

Regarding the in-house real-world dataset, we investigate the effects of several parameters to offer an insight into parameters. From Fig. 4 (a), we see that smaller mini-batch finds more anomalies than greater ones. In Fig. 4 (b), we observe more anomalies as the interval becomes larger, since fewer data points are selected as reference (normal) points in larger interval. In Fig. 4 (c), the goodness of fit (definition is referred to [7]) is a measure of how well the decomposed model explains the underlying data, which were tested using the remaining data points after excluding the base data. Figure 4 (d) clearly shows that smaller mini-batches cost more time.

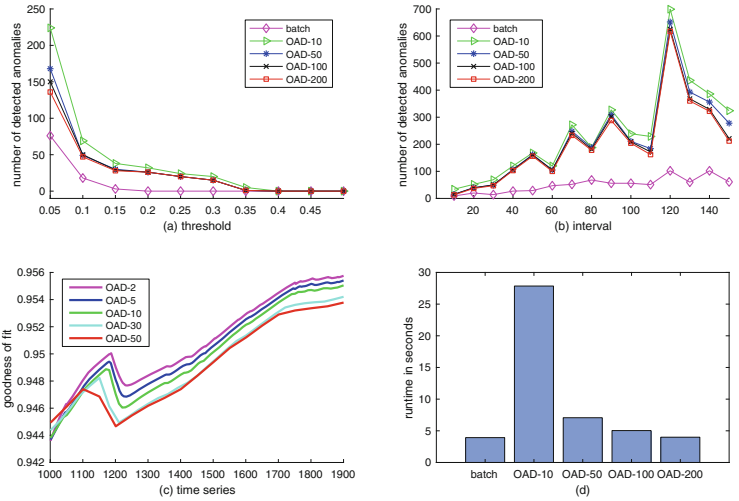

Fig. 4. In-house dataset results, $R = 5$, $\mathcal{T}_1$ contains 1000 samples as initial base. (a): The number of anomalies affected by the thresholds, $T = 30$. (b): The number of anomalies affected by the interval length (c): The goodness of fit varies with the time series. (d): Runtime comparison.

In Fig. 5, we can conclude three trends, (1): The number of anomalies declines as we increase the decomposition bases. (2): The number of anomalies is inverse proportional to the value of R. (3): Small mini-bathes detect more anomalies, which is consistent with the observation in MNIST data.

4.3 Discussion of Experimental Results

From above experiments, we see that the mini-batch size S and reference interval length T play essential role in finding the number of anomalies. The introduction of T may lead to some biased results, because the selected reference points (depends on value T) can be actually anomalies whereas they are regarded as normal training data for model updating. As we frequently update the model, more anomalies can be detected. A larger S also means less computational time. The number of decomposition base (historical data for initial model) is not as critical as S and T. The parameter threshold δ is certainly an important factor influencing the number of anomalies, which can be tuned according to domain specific knowledge or by a training dataset.

From the algorithmic point of view, the algorithm can still be further studied in following aspects. A forgetting factor can be introduced to serve as a memory mechanism to remember or forget the historical data. In addition, we may consider excluding the anomaly points prior to updating the model, which would yield different results and may be beneficial in some scenarios. Furthermore, we may use more benchmark datasets to validate the approach. Regarding the

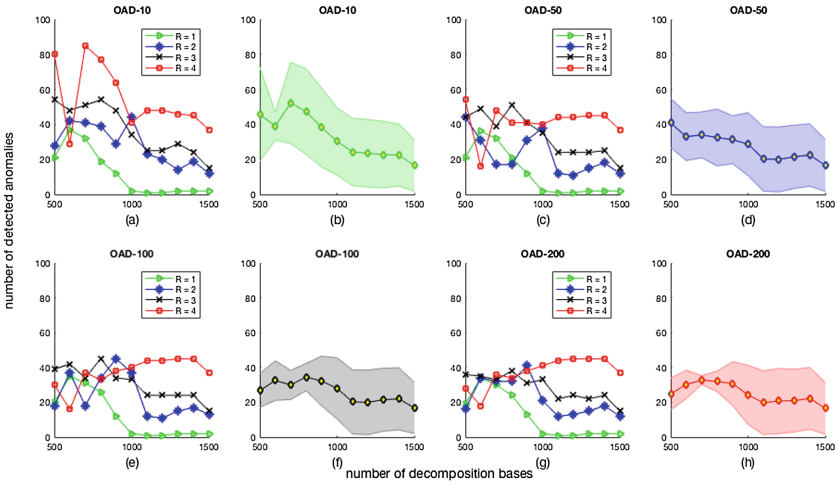

Fig. 5. The number of detected anomalies affected by the size of initial decomposition bases using various parameter settings. The green line in subplot (b) is the average value calculated from the four lines in subplot (a), and the shaded area represents the standard deviation of these lines. (Color figure online)

algorithm, extra efforts need to be paid to study the higher order (mode) incremental tensor learning, which is beyond the scope of this work.

5 Conclusion

The present study proposed a new online anomaly detection algorithm using incremental tensor decomposition. The updating process is achieved by refreshing the factor matrices, and the anomalies are detected by computing a distance based on the information captured in these matrices. Since anomaly detection is often a hard task by itself due to the fact of lacking true anomaly labels, we therefore used MNIST dataset to show the discriminative effectiveness. The other real-world in-house dataset demonstrates the effects of various parameters influencing the results. Our proposed algorithm can be a suitable approach for applications that involve large amount of data, because online algorithm does not request large memory space, whereas tensor decomposition on a large matrix can be computationally very expensive. The online algorithm can be a considered candidate for applications in which the data is hard to fit in memory and change with time, because the updating algorithm and the tunable parameters allow the model to track the changes.

References

1. Brett, W.B., Tamara, G.K, et al.: MATLAB tensor toolbox version 2.6, February 2015

2. Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: Moa: massive online analysis. J. Mach. Learn. Res. **11**, 1601–1604 (2010)
3. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. **41**(3), 15:1–15:58 (2009)
4. Cai, D., Hou, D., Qi, Y., Yan, J., Lu, Y.: A distributed rule engine for streaming big data. In: Meng, X., Li, R., Wang, K., Niu, B., Wang, X., Zhao, G. (eds.) WISA 2018. LNCS, vol. 11242, pp. 123–130. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02934-0_12
5. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2000, pp. 71–80. ACM, New York, NY, USA (2000)
6. Feng, J., Zhang, C., Hao, P.: Online learning with self-organizing maps for anomaly detection in crowd scenes. In: 2010 20th International Conference on Pattern Recognition, pp. 3599–3602 (2010)
7. Gujral, E., Pasricha, R., Papalexakis, E.: Sambaten: Sampling-based batch incremental tensor decomposition, pp. 387–395 (2018)
8. Hitchcock, F.L.: The expression of a tensor or a polyadic as a sum of products. J. Math. Phys. **6**(1), 164–189 (1927)
9. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
10. Laxhammar, R., Falkman, G.: Online learning and sequential anomaly detection in trajectories. IEEE Trans. Pattern Anal. Mach. Intell. **36**(6), 1158–1173 (2014)
11. Lev-ari, H.: Efficient solution of linear matrix equations with application to multistatic antenna array processing. Commun. Inf. Syst. **5**, 123–130 (2005)
12. Li, J., Han, G., Wen, J., Gao, X.: Robust tensor subspace learning for anomaly detection. Int. J. Mach. Learn. Cybern. **2**(2), 89–98 (2011)
13. Nickel, M., Tresp, V.: Tensor factorization for multi-relational learning. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) ECML PKDD 2013. LNCS (LNAI), vol. 8190, pp. 617–621. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40994-3_40
14. Papalexakis, E.E., Faloutsos, C., Sidiropoulos, N.D.: Tensors for data mining and data fusion: models, applications, and scalable algorithms. ACM Trans. Intell. Syst. Technol. **8**(2), 16:1–16:44 (2016)
15. Sun, J., Tao, D., Faloutsos, C.: Beyond streams and graphs: Dynamic tensor analysis. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2006, pp. 374–383. ACM, New York, NY, USA (2006)
16. Sun, J., Tao, D., Papadimitriou, S., Yu, P.S., Faloutsos, C.: Incremental tensor analysis: theory and applications. ACM Trans. Knowl. Discov. Data **2**(3), 11:1–11:37 (2008)
17. Tucker, L.R.: Some mathematical notes on three-mode factor analysis. JPsychometrika **31**(3), 279–311 (1966)
18. Vespe, M., Visentini, I., Bryan, K., Braca, P.: Unsupervised learning of maritime traffic patterns for anomaly detection. In: 9th IET Data Fusion & Target Tracking Conference (DF & TT 2012): Algorithms & Applications, p. 14. IET (2012)
19. Li, Y., Wen, Y., Yuan, X.: Online aggregation: a review. In: Meng, X., Li, R., Wang, K., Niu, B., Wang, X., Zhao, G. (eds.) WISA 2018. LNCS, vol. 11242, pp. 103–114. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02934-0_10