MultiAspectSpotting: Spotting Anomalous Behavior within Count Data Using Tensor

Koji Maruhashi and Nobuhiro Yugami

Fujitsu Laboratories Ltd.
{maruhashi.koji,yugami}@jp.fujitsu.com

Abstract. Methods for finding *anomalous* behaviors are attracting much attention, especially for very large datasets with several attributes with tens of thousands of categorical values. For example, security engineers try to find *anomalous* behaviors, *i.e.*, remarkable attacks which greatly differ from the day's trend of attacks, on the basis of intrusion detection system logs with source IPs, destination IPs, port numbers, and additional information. However, there are large amount of *abnormal* records caused by noise, which can be repeated more *abnormally* than those caused by *anomalous* behaviors, and they are hard to be distinguished from each other. To tackle these difficulties, we propose a two-step anomaly detection. First, we detect *abnormal* records as individual anomalies by using a statistical anomaly detection, which can be improved by *Poisson Tensor Factorization*. Next, we gather the individual anomalies into groups of records with similar attribute values, which can be implemented by *CANDECOMP/PARAFAC (CP) Decomposition*. We conduct experiments using datasets added with synthesized anomalies and prove that our method can spot *anomalous* behaviors effectively. Moreover, our method can spot interesting patterns within some real world datasets such as IDS logs and web-access logs.

Keywords: anomaly detection, tensor decomposition.

1 Introduction

Our work is motivated by anomaly detection in datasets that have several attributes with tens of thousands of categorical values. We want to know the existence of *anomalous* behavior by finding *abnormal* records, *i.e.*, records strangely repeated or strangely less than expected. For example, an intrusion detection system (IDS) monitors network traffic for suspicious activity, and each record in IDS logs has attributes such as *srcIP*, *dstIP*, *port*, and *type* as shown in Table 1. A serious problem for analysts in charge of a company's security system is that IDS logs contain too many records to investigate all of them precisely. Therefore, it is important not only to determine *ordinary* behaviors, *i.e.*, the day's trend of attacks which changes rapidly day-to-day, but also to spot *anomalous* behaviors, *i.e.*, remarkable attacks which greatly differ from *ordinary* behaviors of the day, which are worth investigating. We can build a model of records caused by *ordinary* behavior under the assumption that the majority of records are caused by

V.S. Tseng et al. (Eds.): PAKDD 2014, Part II, LNAI 8444, pp. 474–485, 2014.

Table 1. Example of a dataset. (*a.a.a.a*, *b.b.b.b*, *80*, *type X*) were repeated two times.

source IP (srcIP)	*destination IP (dstIP)*	*port number (port)*	*possible attack type (type)*
a.a.a.a	b.b.b.b	80	type X (DOS attack)
a.a.a.a	b.b.b.b	80	type X (DOS attack)
c.c.c.c	d.d.d.d	12345	type Y (port scan)

ordinary behavior, and distinguish *anomalous* behaviors from them. One possible model is to assume the probability of an *ordinary* record which contains two attribute sets A and B as $P(A)P(B)$, *i.e.*, statistically independent, and to declare a record *anomalous* if their joint appearance $P(A, B)$ is much higher than $P(A)P(B)$. This is an intuition based on *suspicious coincidence* [1]. However, there are large amount of *abnormal* records caused by noise, *e.g.*, false positives in IDS logs [2], and they can be repeated more *abnormally* than those caused by *anomalous* behaviors, and they are hard to be distinguished. We can assume that an *anomalous* behavior can affect a group of records with similar attribute values, and can be distinguished from noise by gathering *abnormal* records into such a group. For example, many *abnormal* records with similar *srcIP*, *dstIP*, *port*, and *type* can be caused by a common remarkable attack, instead of false positives. However, a problem is that it becomes harder to detect *abnormal* records in such a group as the size of the group grow, because they become more likely to be *ordinary* behaviors, *e.g.*, $P(A)P(B)$ gets closer to $P(A, B)$.

To tackle these difficulties, we propose a two-step anomaly detection. In the first step, we detect *abnormal* records as individual anomalies with a statistical anomaly detection that models the distribution of the numbers of records caused by *ordinary* behaviors as Poisson distribution. By making a stronger assumption of the distribution for *ordinary* behaviors, we try to detect *abnormal* records in larger groups more effectively. This step can be improved by using *Poisson Tensor Factorization* (*PTF*) [3]. In the second step, we gather the individual anomalies into groups of records with similar attribute values. This step can be implemented by using *CANDECOMP/PARAFAC (CP) Decomposition* [4].

Our main contributions are: (1) We propose a novel framework *MultiAspectSpotting* combining statistical anomaly detection with spotting groups of *abnormal* records. (2) By using datasets added with synthesized anomalies, we show our method can spot *anomalous* behaviors effectively. (3) We show our method can spot interesting patterns in real world datasets like IDS logs and web-access logs.

The remainder of this paper is organized as follows. We describe the related literature in Section 2 and introduce our method in Section 3. We describe the accuracy and scalability of our method in Section 4 and the experimental evaluation on real data in Section 5. In Section 6 we summarize our conclusions.

2 Related Work

2.1 Anomaly Detection in Categorical Datasets

Anomaly detection has attracted wide interest in many applications such as security, risk assessment, and fraud analysis [5]. Das et al. [6] proposed an anomaly

pattern detection in noisy categorical datasets based on a rule-based anomaly detection [7]. They searched through all possible one or two component rules and detected anomalies whose counts were significantly differed from the expected counts determined by the training dataset. They used the *conditional anomaly detection* [8,6] as a definition of anomalies which is an alternative of *suspicious coincidence* proposed by Barlow [1]. However, they tried to find groups of *abnormal* records which significantly differed from the training dataset, whereas our problem is to spot groups of *abnormal* records which are most remarkable among all records in the dataset.

2.2 Tensor Decomposition

Tensor decomposition is a basic technique that has been widely studied and applied to a wide range of disciplines and scenarios. *CP Decomposition* and *Tucker Decomposition* are two well-known approaches [4], and has been applied to study tensor streams [9]. Non-negative tensor factorizations have been proposed to retain the nonnegative characteristics of the original data [10], as natural expansions of non-negative matrix factorizations[11]. *PTF* is one such technique, that models *sparse count* data by describing the random variation via a Poisson distribution [3]. Our work is also related to the *Boolean Tensor Factorization* [12], which uses Boolean arithmetic, i.e., defining that $1 + 1 = 1$. The problems of *Boolean Tensor Factorization* were proved to be NP-hard, and heuristics for these problems were presented [12]. Some implementations of tensor decomposition algorithms have been made publicly available, such as MATLAB Tensor Toolbox [13]. We combine some of these tensor decompositions effectively to spot *anomalous* behaviors. Moreover, some works detected outliers in a low-dimensional space obtained by tensor decompositions [14], but outliers caused by *anomalous* behaviors were not distinguished from those caused by noise.

3 Proposed Method

3.1 Notation

A *tensor* can be represented as a multi-dimensional array of scalars, and we call each scalar an *entry*. Its *order* is the dimensionality of the array, while each dimension is known as one *mode*. A tensor is *rank one* if it can be written as the outer product of vectors. The *rank* of a tensor is defined as the smallest number of rank-one tensors that can generate the tensor as their sum, and we refer to each rank-one tensor as a *component*. Throughout, scalars are denoted by lowercase letters (a), vectors by boldface lowercase letters ($\mathbf{v}$), matrices by boldface capital letters ($\mathbf{A}$), and higher-order tensors by boldface Euler script letters ($\mathcal{X}$). The jth column of a matrix $\mathbf{A}$ is denoted by $\mathbf{a}_j$, and ith entry of a vector $\mathbf{v}$ is denoted by v_i. We use multi-index notation so that a boldface $\mathbf{i}$ represents the index $(i_1...i_M)$ of a tensor of order M. The size of nth mode is denoted as I_n. The notation $\|\cdot\|$ refers to the square root of the sum of the squares of the entries, analogous to the matrix Frobenius norm. The outer product is denoted by $\circ$, and the inner product is denoted by $\langle\cdot,\cdot\rangle$.

3.2 Problem Setting

Our problem can be defined as follows: Given a dataset in which each record $\mathbf{i}$ has M categorical attributes and repeated $x_{\mathbf{i}}$ times, how can we detect *abnormal* records repeated strangely more than or less than expected, caused by *anomalous* behaviors as distinguished from those caused by noise?

We make two assumptions. (1) The majority of records are caused by *ordinary* behavior, and we can build a model with minimal harm caused by *anomalous* behaviors and noise. (2) A group of *abnormal* records with similar attribute values is likely to be caused by a common *anomalous* behavior.

3.3 *MultiAspectSpotting* Framework

In this paper, we focus on statistical anomaly detection based on the assumption "*Normal data instances occur in high probability regions of a stochastic model, while anomalies occur in the low probability regions of the stochastic model*" [5]. However, a simple statistical anomaly detection is insufficient to spot interesting anomalies effectively because we cannot distinguish *abnormal* records caused by *anomalous* behaviors from those caused by noise. To tackle this difficulty, we propose a novel framework *MultiAspectSpotting* that can spot *anomalous* behaviors by conducting two-step different tensor decompositions (Fig. 1):

1. Create a tensor $\mathcal{X}$ in which mth mode corresponds to mth attribute of a dataset and entries of $\mathcal{X}$ indicating the numbers of corresponding records. Then calculate anomaly score of each record by conducting *PTF*, and pick up records with larger anomaly scores than a threshold t as individual anomalies. We make a strong assumption that the distribution of the number of records caused by *ordinary* behaviors is a mixture of R Poisson distributions, to detect individual anomalies in larger groups effectively (see Section 3.4).
2. Create a binary tensor $\mathcal{B}$ in which 1s indicate individual anomalies, and spot groups of individual anomalies of the maximum number of S as *anomalous* behaviors by conducting *CP Decomposition* (see Section 3.5).

Deciding threshold t to pick up individual anomalies in the first step is very important. Our strategy is to set the *ratio of noise records* Z, and to decide threshold t so that the ratio Z of distinct records is picked up as individual anomalies. We assume that a specific ratio of records are caused by noise, and that the number of records caused by *anomalous* behavior is relatively small. If no groups are spotted in the second step, we conclude that the dataset is not affected by *anomalous* behaviors.

Now we do not have a clear strategy of the parameter settings of R, S, and Z, and there is a big room for improvement of our framework. However, in Section 4 we show we can achieve better results by using $R > 1$ or $S > 0$ than using $R = 1$ (assuming a single Poisson distribution) or $S = 0$ (without the second step). Moreover, we show the selection of Z does not dramatically affect the results of spotting *anomalous* behaviors.

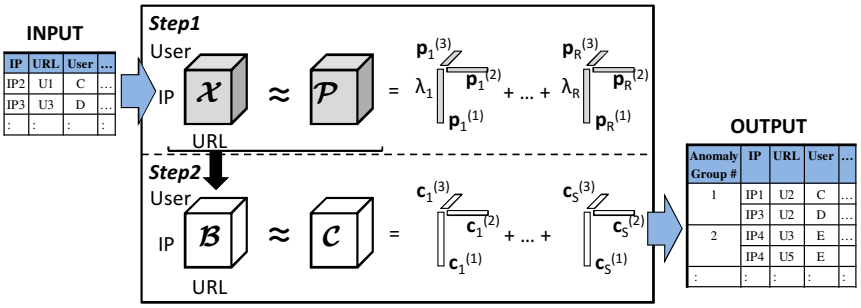

Fig. 1. The *MultiAspectSpotting* framework. Step 1: Conduct *Poisson Tensor Factorization*. Step 2: Create binary tensor indicating individual anomalies and conduct additional tensor decomposition.

3.4 A Statistical Anomaly Detection Approach

We describe details of the first step. The probability of the number of records of $\mathbf{i}$ to be $x_\mathbf{i}$ in a fixed interval of time can be modeled as the Poisson distribution in which the cumulative probability function is

$$F(x_\mathbf{i}, \mu_\mathbf{i}) = \sum_{k=0}^{x_\mathbf{i}} \frac{\mu_\mathbf{i}^k}{k!} e^{-\mu_\mathbf{i}}, \tag{1}$$

where $\mu_\mathbf{i}$ is the Poisson parameter equal to the expected number of the records of $\mathbf{i}$ caused by *ordinary* behaviors. Anomaly score is calculated as

$$anomaly\_score(x_\mathbf{i}, \mu_\mathbf{i}) = \begin{cases} (-1) * \log(F(x_\mathbf{i}, \mu_\mathbf{i})) & (x_\mathbf{i} \leq \mu_\mathbf{i}), \\ (-1) * \log(1 - F(x_\mathbf{i}, \mu_\mathbf{i})) & (x_\mathbf{i} > \mu_\mathbf{i}). \end{cases} \tag{2}$$

We consider a distinct record of $\mathbf{i}$ to be an individual anomaly if the anomaly score is higher than a threshold t. Also, $F(x_\mathbf{i}, \mu_\mathbf{i})$ can be easily computed with the incomplete gamma function. As $\mu_\mathbf{i}$ is the expected number of records of $\mathbf{i}$, we can estimate $\mu_\mathbf{i}$ as

$$\mu_\mathbf{i} = \lambda p_{i_1}^{(1)} \cdots p_{i_M}^{(M)}, \tag{3}$$

where λ is total number of records and $p_{i_m}^{(m)}$ is the probability of mth value to be i_m, under the assumption of independence among the attributes. $p_{i_m}^{(m)}$ can be estimated as $p_{i_m}^{(m)} = N_{i_m}^{(m)}/\lambda$, where $N_{i_m}^{(m)}$ is the number of records of which mth value is i_m. Alternatively, we can assume the distribution as the mixture of R Poisson distributions. The Poisson parameters can be estimated as

$$\mu_\mathbf{i} = \sum_{r=1}^{R} \lambda_r p_{ri_1}^{(1)} \cdots p_{ri_M}^{(M)}, \tag{4}$$

under the assumption of independence among the distributions, where λ_r is the expected total number of records emerged from rth distribution and $p_{ri_m}^{(m)}$ is the

probability of mth value to be i_m in rth distribution. We can estimate λ_r and $p^{(m)}_{ri_m}$ for all $\mathbf{i}$ and r by conducting *PTF*, which calculates each parameter so as to minimize the generalized Kullback-Leibler divergence, *i.e.*, $\sum_{\mathbf{i}} \mu_{\mathbf{i}} - x_{\mathbf{i}} log \mu_{\mathbf{i}}$ [3]. The details of *PTF* are outside the scope of this paper. Note that *PTF* of $R = 1$ is equivalent to calculating the parameters of equation (3).

3.5 Spotting Anomalous Behaviors by Tensor Decomposition

In the second step, we try to find S product sets $\mathcal{D}_s = \{(i_1, \ldots, i_M) | i_m \in \mathbf{d}^{(m)}_s \forall m = 1, \ldots, M\}$ where $\mathbf{d}^{(m)}_s$ is sth set of values of mth attribute ($s = 1, \ldots, S$), such that each product set contains as many individual anomalies as possible. We propose *DenseSpot*, a tensor decomposition approach (Algorithm 1). *DenseSpot* construct a binary tensor $\mathcal{B}$ of order M in which an entry $b_{\mathbf{i}}$ is

$$b_{\mathbf{i}} = \begin{cases} 1 & (anomaly\_score(x_{\mathbf{i}}, \mu_{\mathbf{i}}) > t), \\ 0 & (otherwise). \end{cases} \tag{5}$$

The aim of *DenseSpot* is to obtain a rank-S tensor

$$\mathcal{C} = \sum_{s=1}^{S} \mathbf{c}^{(1)}_s \circ \ldots \circ \mathbf{c}^{(M)}_s \tag{6}$$

which minimize $\|\mathcal{B} - \mathcal{C}\|$, where $\mathbf{c}^{(m)}_s$ are binary vectors. However, the decision version of this problem is a NP-hard problem similar to *Boolean Tensor Factorization* [12]. Thus, *DenseSpot* first obtains a rank-S tensor $\hat{\mathcal{C}} = \sum_{s=1}^{S} \hat{\mathbf{c}}^{(1)}_s \circ \ldots \circ \hat{\mathbf{c}}^{(M)}_s$ which minimize $\|\mathcal{B} - \hat{\mathcal{C}}\|$, where $\hat{\mathbf{c}}^{(m)}_s$ are real-value vectors. This is a relaxation problem of the above problem, and we can obtain a solution by conducting *CP Decomposition* [4]. After that, *DenseSpot* checks entries in sth component of $\hat{\mathcal{C}}$ corresponding to individual anomalies $(i_1, \ldots, i_M)$ and puts 1 on i_mth element of $\mathbf{c}^{(m)}_s$ if the entries are greater than a threshold h. Finally, *DenseSpot* selects h, which minimizes $\|\mathcal{B} - \mathcal{C}\|$ and returns those $\mathcal{C}$ calculated by the h. We can easily calculate $\|\mathcal{B} - \mathcal{C}\|^2$ as $\|\mathcal{B}\|^2 - 2\langle \mathcal{B}, \mathcal{C} \rangle + \|\mathcal{C}\|^2$. A set $\mathbf{d}^{(m)}_s$ can be created by selecting value 1 entries of $\mathbf{c}^{(m)}_s$.

Also, *Boolean Tensor Factorization* [12] might be a good solution for this. Even though this could improve the efficiency of our method, we explain how our simple heuristics can perform better than baseline methods in Section 4.

4 Evaluation of Accuracy and Scalability

In this section, we present experimental results on the accuracy and scalability of our methods. The running example in this section comes from network traffic logs that consist of packet traces in an enterprise network (LBNL/ICSI Enterprise Tracing Project [1]). We abbreviate them as LBNL logs. Each trace in the logs is a

[1] `http://www.icir.org/enterprise-tracing/`

Algorithm 1. *DenseSpot*

Input: A binary tensor $\mathcal{B}$
Input: Maximum number S of anomalies to spot
Input: A set of thresholds $H = \{h_1, ..., h_d\}$
Output: Rank-S tensor $\mathcal{C} = \sum_{s=1}^{S} \mathbf{c}_s^{(1)} \circ ... \circ \mathbf{c}_s^{(M)}$ where $\mathbf{c}_s^{(m)}$ are binary vectors

$\hat{\mathcal{C}} \leftarrow \sum_{s=1}^{S} \hat{\mathbf{c}}_s^{(1)} \circ ... \circ \hat{\mathbf{c}}_s^{(M)}$ s.t. minimize $\|\mathcal{B} - \hat{\mathcal{C}}\|$ ▷*CP Decomposition*
for $j = 1$ **to** d **do**
  $\mathcal{C}^{(j)} \leftarrow \sum_{s=1}^{S} \mathbf{c}_s^{(1)} \circ ... \circ \mathbf{c}_s^{(M)}$ where $\mathbf{c}_s^{(m)}$ are I_m-length vectors of all 0
  forall the $(i_1...i_M)$ *of* 1 *entries in* $\mathcal{B}$ **do**
    for $s = 1$ **to** S **do**
      if $\hat{c}_{si_1}^{(1)} ... \hat{c}_{si_M}^{(M)} \geq h_j$ **then** $c_{si_1}^{(1)} \leftarrow 1, ..., c_{si_M}^{(M)} \leftarrow 1$
    end
  end
end
$j_{min} \leftarrow \arg\min_j \|\mathcal{B} - \mathcal{C}^{(j)}\|$
return $\mathcal{C}^{(j_{min})}$

triplet of {*source IPs (srcIP)*, *destination IPs (dstIP)*, and *port number (port)*}, which can be represented as a 3-mode tensor. First, we evaluate the accuracy of spotting *anomalous* behaviors by using 10 largest LBNL logs added with synthesized anomalies. Then we evaluate the scalability by using many LBNL logs of various numbers of records.

MultiAspectSpotting is implemented in the MATLAB language, and we use implementations of *PTF* (*cp_apr*) and *CP Decomposition* (*cp_als*), publicly available in MATLAB Tensor Toolbox [13]. All the experiments are performed on a 64-bit Windows XP machine with four 2.8GHz cores and 8GB of memory.

4.1 Putting Synthesized Anomalies on Datasets

We create some synthesized anomalies and add into 10 largest LBNL logs, and evaluate how effectively our method can spot these anomalies. These LBNL logs have about 900,000 to 9,000,000 records and 15,000 to 50,000 distinct records, with 1,400 to 4,500 srcIPs, 1,400 to 4,800 dstIPs and 5,400 to 24,000 ports. Each distinct record is repeated about 50 to 350 times in average, and the standard deviation is about 1,000 to 22,000.

Given parameters of *volume* V, *density* D and *maximum number* P, we create N groups of *abnormal* records as follows: (1) For each group, we randomly select three values s,d,p between 0 and 1, and decide the number of srcIPs and dstIPs and ports in accordance with the ratio of three selected values, so that sdp is not lower than V, *e.g.*, the number of srcIPs is $\lceil s(V/(sdp))^{1/3} \rceil$ where $\lceil \cdot \rceil$ is the ceiling function. (2) $\lceil VD \rceil$ distinct records are randomly selected for each group, and (3) the number of each record is decided randomly between 1 and P. We test for $V = 50$, $D = 0.1, 0.3, 0.5, 0.7, 0.9$, $P = 500$, and $N = 10$.

4.2 Methods Compared

We compare the accuracies in spotting synthesized anomalies among the following methods:

MASP-Multi *MultiAspectSpotting* with $R = 10$ and $S = 20$.

MASP-Single *MultiAspectSpotting* with $R = 1$ and $S = 20$, which is equivalent to modeling the probabilities of the numbers of the records caused by *ordinary* behaviors as a single Poisson distribution.

DS-Only Conducting just *DenseSpot* of $S = 20$ by picking up all distinct records as individual anomalies.

SC-DS Using a measure of *suspicious coincidence* proposed by Barlow [1]. For each record, we calculate the ratio $r = P(A, B)/(P(A)P(B))$ where $P(A)$ and $P(B)$ are probabilities of a record having attribute sets A and B (*e.g.*, {*srcIP*}, {*dstIP*, *port*}), and $P(A, B)$ is the joint probability. The anomaly score of the record is defined as the minimum value of r among those of all possible combinations of A and B. We pick up individual anomalies and conduct *DenseSpot* as the same as *MultiAspectSpotting*.

Note that we have tried several methods similar to *SC-DS*, such as those using the maximum value of r, or those considering records with lower r as anomalous, or those using the ratio $r = P(A, B, C)/(P(A)P(B)P(C))$ where $P(A)$,$P(B)$,$P(C)$ and $P(A, B, C)$ correspond to attributes A, B and C, but these variations have obtained far worse results than *SC-DS* (not shown).

4.3 Accuracy of Spotting Synthesized Anomalies

We apply the above methods to LBNL logs added with synthesized anomalies and compare a group of records spotted by each method with a group of synthesized anomalies. We conduct chi-square tests of independence, which assess whether these two groups are independent of each other. In short, given these two group, we calculate $\chi^2 = n(a(n-e-g+a)-(e-a)(g-a))^2/(e(n-e)g(n-g))$ where n is the total number of distinct records, a is the number of common distinct records between two groups, e and g are the numbers of distinct records of two groups. If χ^2 is greater than a value of p-value at 0.05 of the chi-squared distribution for 1 degree of freedom, we conclude that the method has successfully spotted the synthesized anomalous group.

Fig. 2 is the number of groups spotted by each method. *MASP-Multi* and *MASP-Single* can spot many more groups than *DS-Only*, which suggests the statistical anomaly detection in the first step works efficiently. However, *SC-DS* is worse than *DS-Only*, which suggests the measure of *suspicious coincidence* is not good at detecting the anomalies we consider in this paper. Moreover, *MASP-Multi* is better than *MASP-Single*, which indicates we can model *ordinary* behaviors better by using a mixture of Poisson distributions. Overall, the more *density* grows, the better *MASP-Multi* and *MASP-Single* can spot than *DS-Only* and *SC-DS*. Moreover, the results of *MASP-Multi* and *MASP-Single* do not dramatically differ between $Z = 0.01$ (Fig. 2 left) and $Z = 0.1$ (Fig. 2 right), especially for higher *density* such as $P = 0.7, 0.9$.

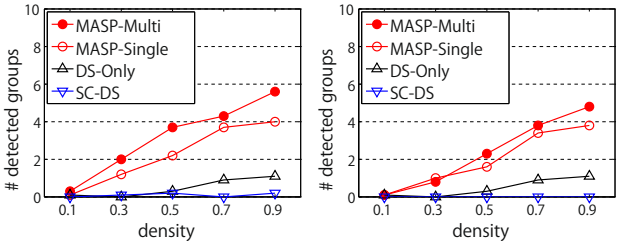

Fig. 2. Vertical axis: average number of spotted groups. Horizontal axis: *density* of groups of synthesized anomalies. (left) $Z = 0.01$. (right) $Z = 0.1$.

4.4 Details of Accuracy

We detail the effectiveness of each step. For the first step, we show the area under the ROC curve (AUC) of each method in detecting distinct synthesized anomalies out of all distinct records at various settings of Z (Fig. 3 left). Overall, AUCs of *MASP-Multi* and *MASP-Single* are better than those of *SC-DS*, and do not become much worse as *density* grow, whereas AUCs of *SC-DS* become much worse. These indicate that this statistical anomaly detection is a better strategy at least in detecting anomalies described here. There are no significant differences between *MASP-Multi* and *MASP-Single* in view of AUCs. For the second step, we select as many records with the highest anomaly scores as the total number of individual anomalies *DenseSpot* has detected, which we call *TopRecords*. We compare the *precision* of detecting synthesized anomalies out of individual anomalies between *DenseSpot* and *TopRecords*. The *precision* of *DenseSpot* and *TopRecords* are calculated as p/k and q/k, where p is the total number of distinct synthesized anomalies that *DenseSpot* has detected, q is the number of those *TopRecords* has selected, and k is the total number of individual anomalies that *DenseSpot* has detected. Fig. 3 (right) shows *precisions* on each method ($Z = 0.01$). The *precisions* of *DenseSpot* are much better than those of *TopRecords* on *MASP-Multi* and *MASP-Single* for higher *density*. This means the synthesized anomalies do not have very high anomaly scores among individual anomalies, whereas *DenseSpot* can pick up these synthesized anomalies, especially for higher density. Additionally, the *precisions* of *DenseSpot* on *MASP-Multi* are better than those on *MASP-Single*, which indicates that the difference in the number of Poisson distributions in the first step strongly affects the second step, even though differences in AUCs are very small. In addition, the *precisions* of *DenseSpot* are worse than those of *TopRecords* on *SC-DS*, possibly due to the poor accuracy of the *suspicious coincidence* in the first step.

4.5 Scalability

We conduct experiments for scalability on 123 different LBNL logs with various numbers of records, from less than 100 to more than $9,000,000$. As shown in Fig. 4, computation time of *PTF* (left) and *DenseSpot* (right) increases linearly

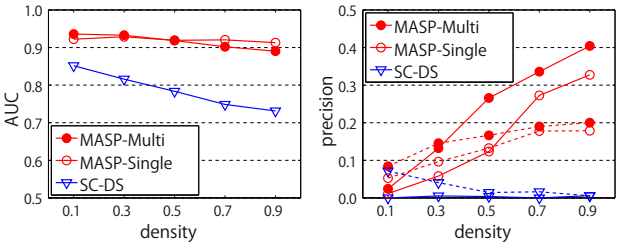

Fig. 3. (left) Average AUC of each method. Vertical axis: average AUC. Horizontal axis: *density* of groups of anomalous records. (right) The *precision* of *DenseSpot* (solid lines) and *TopRecords* (dotted lines) as described in Section 4 ($Z = 0.01$). Vertical axis: average of the *precision*. Horizontal axis: *density* of groups of anomalous records.

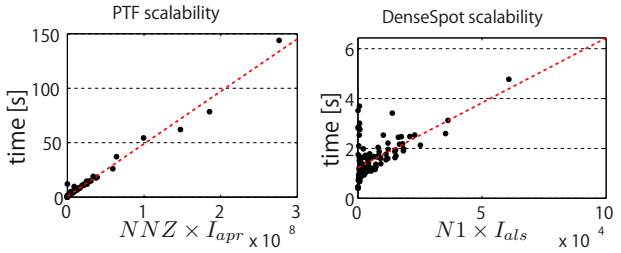

Fig. 4. Computation time have high correlation with $NNZ \times I_{apr}$ on *PTF* of $R = 10$ (left) and $N1 \times I_{als}$ on *DenseSpot* of $S = 20$ (right), where NNZ is the number of non-zeros (NNZ) in $\mathcal{X}$, I_{apr} is the number of inner iterations of *cp_apr*, $N1$ is the number of 1s in $\mathcal{B}$, and I_{als} is number of iterations of *cp_als*. Vertical axis: computation time (s). Horizontal axis: $NNZ \times I_{apr}$ (left), $N1 \times I_{als}$ (right).

along with the number of non-zero entries in the tensor (entries of 1 for *DenseSpot*) multiplied by the number of iterations of *cp_apr* and *cp_als*. These results are consistent with these alternating optimization algorithms implemented for sparse tensor [4,3] and suggest that these two steps of our framework scale linearly along with the number of distinct records in the dataset.

5 Empirical Results on Real Data

We present our experimental results on two sets of real world data: intrusion detection system logs and web-access logs ($R = 20$, $S = 20$, and $Z = 0.1$). We cannot mention the names of the companies from whom we have obtained these datasets because of the business relationship. Table 2 summarizes these datasets.

5.1 Intrusion Detection System Logs

We apply our method to IDS logs of a crowd system of an IT company. We analyze inbound logs on Dec 2011, that is, suspicious packets sent from outside

Table 2. Summary of Datasets

	mode#1	mode#2	mode#3	mode#4	# of records
IDS logs	22,733 srcIPs	3,171 dstIPs	11,310 ports	362 types	228,852
web-access logs	24 hours	462 IPs	40,465 URLs	51,960 UserIDs	462,271

Table 3. Summary of the groups of individual anomalies spotted by *MultiAspectSpotting*. We show the unique number of values of each attribute along with the total number of distinct records (#records), the average of repeated times of each distinct record (ave. num.), the average of Poisson parameters (eve. exp.), and description.

(a) IDS logs

No.	#srcIP	#dstIP	#port	#type	#records	ave. num.	ave. exp.	description
1	1	60	1	1	60	22.52	2.65	*FingerPrint*
2	53	1	1	1	53	5.26	0.00	*FTP Login Fail*
3	1	1	45	1	45	1.00	0.00	*Malicious Javascript*
4	1	34	1	1	34	20.41	2.17	*TCP Invalid flags*
5	5	1	1	7	33	262.42	41.23	*Scanning to Web Server*

(b) web-access logs

No.	#hour	#IP	#URL	#UserID	#records	ave. num.	eve. exp.	description
1	5	1	3	1	15	4.07	0.00	*weather-checking*
2	2	1	5	2	13	43.23	1.85	*point-gathering*
3	1	13	1	1	13	7.92	0.02	*photo-uploading*
4	6	2	1	1	12	408.83	66.48	*photo-uploading*
5	1	1	1	12	12	6.08	0.04	*advertisement-viewing*

the crowd system. Each record represents a report which has attributes of {*source IP (srcIP)*, *destination IP (dstIP)*, *port number (port)*, and *attack type (type)*}.

Table 3(a) summarizes the five largest groups of anomalous records spotted by our method. The descriptions are characteristics of these groups guessed by a specialist knowledgeable about the IDS of this crowd system. These include several kinds of attacks: attacks from many srcIPs including suspicious FTP login trials(#2), attacks on many dstIPs (#1,#4), attacks on many port numbers (#3), and attacks from several srcIPs of various attack types (#5). For example, the group #3 indicates that an outside IP has attacked many port numbers of an inside IP with a specific attack type, and that these attacks are remarkable because they are very rare events. Moreover, it is hard for analysts to notice the existence of this group of attacks because the number of records of this group is almost 0.02% of the total number of records within this dataset.

5.2 Web Access Logs

We also apply our method to web-access logs of a web-service company on Jan 10, 2013. Each record has attributes of {*hour*, *IP*, *URL*, and *UserID*} which means an access on the *URL* by the *UserID* from the *IP* at the *hour* of a day. The engineers at this company want to find any strange accesses within web-access logs and surprising or illegal usage of their web pages.

Table 3(b) summarizes the five largest groups of anomalous records spotted by our method, with descriptions of characteristics of these groups guessed by a specialist knowldgeable about the web site. For example, the group #2 is a *point-gathering* group, in which two users have strangely accessed a set of URLs

many times from a IP continuously from 16 pm to 17 pm of this day. By accessing these URLs, users can obtain *points* that can be exchanged for some gifts, so the user who accesses just for gathering *points* illegally is suspicious.

6 Conclusion

We proposed a novel framework *MultiAspectSpotting* that can effectively spot *anomalous* behaviors by leveraging a two-step approach of a different kind of tensor decomposition. Experimental results of synthesized anomalies show our method can spot groups of individual anomalies more effectively than some baseline methods and can be improved by using *PTF*. The effectiveness of our method is achieved thanks to the combination of the accuracy of statistical anomaly detection in the first step and the ability of gathering individual anomalies in the second step, even though it might become harder for our method to model *ordinary* behaviors as the number of the attributes grows, *i.e.*, the dataset becomes sparser. Moreover, experimental results on real world data proved that our method could spot interesting patterns within IDS logs and web-access logs.

References

1. Barlow, H.B.: Unsupervised learning. Neural Computation 1, 295–311 (1989)
2. Julisch, K., Dacier, M.: Mining intrusion detection alarms for actionable knowledge. In: KDD, pp. 366–375 (2002)
3. Chi, E.C., Kolda, T.G.: On tensors, sparsity, and nonnegative factorizations. SIAM J. Matrix Analysis Applications 33(4), 1272–1299 (2012)
4. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Review 51(3), 455–500 (2009)
5. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Comput. Surv. 41(3) (2009)
6. Das, K., Schneider, J.G., Neill, D.B.: Anomaly pattern detection in categorical datasets. In: KDD, pp. 169–176 (2008)
7. Wong, W.K., Moore, A.W., Cooper, G.F., Wagner, M.M.: Rule-based anomaly pattern detection for detecting disease outbreaks. In: AAAI/IAAI, pp. 217–223 (2002)
8. Das, K., Schneider, J.G.: Detecting anomalous records in categorical datasets. In: KDD, pp. 220–229 (2007)
9. Sun, J., Tao, D., Papadimitriou, S., Yu, P.S., Faloutsos, C.: Incremental tensor analysis: Theory and applications. TKDD 2(3) (2008)
10. Shashua, A., Hazan, T.: Non-negative tensor factorization with applications to statistics and computer vision. In: ICML, pp. 792–799 (2005)
11. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: NIPS, pp. 556–562 (2000)
12. Miettinen, P.: Boolean tensor factorizations. In: ICDM, pp. 447–456 (2011)
13. Bader, B.W., Kolda, T.G., et al.: Matlab tensor toolbox version 2.5. Available online (January 2012), `http://www.sandia.gov/~tgkolda/TensorToolbox/`
14. Hayashi, K., Takenouchi, T., Shibata, T., Kamiya, Y., Kato, D., Kunieda, K., Yamada, K., Ikeda, K.: Exponential family tensor factorization for missing-values prediction and anomaly detection. In: ICDM, pp. 216–225 (2010)