Effective Dimension in Anomaly Detection: Its Application to Computer Systems

Tsuyoshi Idé and Hisashi Kashima

IBM Research, Tokyo Research Laboratory, 1623-14 Shimotsuruma, Yamato-shi, Kanagawa 242-8502, Japan,
{goodidea, hkashima}@jp.ibm.com

Abstract. We consider the issue of online anomaly detection from a time sequence of directional data (normalized vectors) in high dimensional systems. In spite of the practical importance, little is known about anomaly detection methods for directional data. Using a novel concept of the effective dimension of the system, we successfully formulated an anomaly detection method which is free from the "curse of dimensionality." In our method, we derive a probability distribution function (pdf) for an anomaly metric, and use a novel update algorithm for the parameters in the pdf, where the effective dimension is included as a fitting parameter. For directional data from a computer system, we demonstrate the utility of our algorithm in anomaly detection.

1 Introduction

A general approach in anomaly detection from vector sequences is to introduce a probability distribution function (pdf) of the collection of negative (or normal) examples. Using a pdf, a threshold value to identify positive (or anomalous) examples can be calculated in a consistent fashion. However, such probabilistic methods often fail for systems with higher dimensions. One major reason is that some of the degrees of freedom in such systems are inactive (almost constant) in many practical applications. A typical example can be found in online text classification, where the dimensions of the document vectors are often on the order of one million, but the dimensions that are effective in classification are known to be on the order of several hundreds [4].

In this paper, we formulate an online anomaly detection algorithm for directional data based on the von Mises-Fisher (vMF) distribution [10]. Although directional data often appears in many practical situations [10, 2, 8], little is known in the context of anomaly detection.

Explicitly, the vMF distribution is given by,

$$p(\boldsymbol{u}|\kappa,\boldsymbol{\mu}) = \frac{\kappa^{\frac{N}{2}-1}}{(2\pi)^{N/2} I_{\frac{N}{2}-1}(\kappa)} \exp\left(\kappa \boldsymbol{\mu}^T \boldsymbol{u}\right), \tag{1}$$

where $\boldsymbol{\mu}$ is a mean direction and $I_l(\cdot)$ represents the modified Bessel function of order l. The value $1/\kappa > 0$ is a constant parameter called the angular variance.

The intrinsic dimension of the directional data $\boldsymbol{u}$ is denoted as N. Intuitively, the vMF distribution describes fluctuations of $\boldsymbol{u}$ around the mean direction. The vMF distribution is the most natural distribution for directional data in that it can be derived using the maximum entropy principle under the conditions that (1) the total probability is unity and (2) that the average over $\boldsymbol{u}$ on the unit sphere is $\boldsymbol{\mu}$. Since the normal distribution is derived if the second condition is replaced with that of the average in the whole N-dimensional space, the vMF distribution can be regarded as the "normal" distribution for directional data.

To detect anomalies in an online fashion, we need to update the pdf in accordance with the data just given at the current time, t. One possible way is to perform maximum likelihood estimation (MLE) continuously for the new data. Banerjee *et al.* [2] employed a mixture of vMF distributions, and derived an approximated version of the MLE procedure. However, parameter estimation in this case is quite difficult due to the modified Bessel function. Especially, the parameter N often degrades the accuracy of the approximations of $I_l(\cdot)$ in many application areas if N is relatively large [1].

In this paper, we introduce a novel concept, effective dimensions, to overcome the curse of dimensionality. Starting from the vMF distribution, we derive a pdf for an anomaly metric based on the Fisher kernel [9]. The pdf contains two parameters that the angular variance and the effective dimension, n, instead of the intrinsic dimension of the directional data, N. Then we introduce a new online algorithm to update the pdf at each time step. To the best of our knowledge, this is the first attempt to overcome the curse of dimensionality using the notion of the effective dimension. Note that existing formulation using Gaussian mixtures [15, 14] are not appropriate in this case because of the degeneracies of the distributions due to the normalization condition and the existence of inactive variables.

We will experimentally show that the effective dimension is actually much less than the nominal dimension N for feature vectors extracted from a computer system. Also, we demonstrate that anomalies can be detected by comparing with a given critical probability that is independent of the details of the system.

The rest of this paper is organized as follows: In Section 2, we define the dependency matrix in computer systems and recapitulate our method of feature extraction [8]. In Section 3, we define an anomaly metric. In Section 4, a generative model for the anomaly metric is derived from the vMF distribution, and introduce the concept of effective dimensions. In Section 5, a novel incremental algorithm is proposed to estimate the parameter in the model. In Section 6, we report on experimental results in a benchmark system. In the final section, we summarize the major results in this paper.

2 Modeling computer systems

2.1 Dependency matrix

To model the behavior of Web-based computer systems at the application layer, where the interaction between servers is essential, we define a *service* as a quartet

of

$$(I_s, I_d, P, Q),$$

where I_s and I_d represent source and destination IP (Internet Protocol) addresses, respectively, and P denotes the port number of the destination application. We also use an attribute called the transaction type Q. Figure 1 illustrates a benchmark system. There are four server boxes in this system, and two server processes with port numbers p_1 and p_2 are installed on each of the boxes at i_2 and i_3.

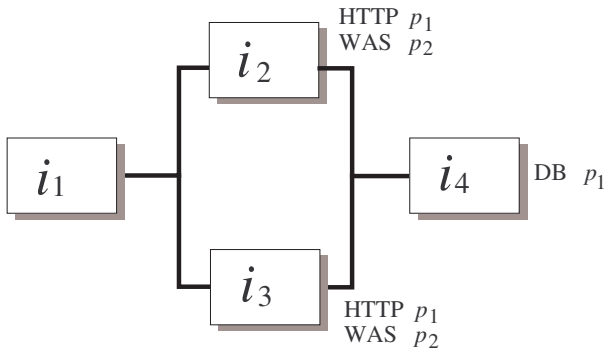

Fig. 1. Configuration of benchmark system. IP addresses and port numbers are denoted by i_k $(k = 1, .., 4)$ and p_j $(j = 1, 2, 3)$, respectively.

Consider a system with N different services, and imagine a graph each of whose nodes is one of the services. For the edge weights, we employ the following quantity [8]:

$$\mathsf{D}_{i,j} = [f(d_{i,j}) + f(d_{j,i})]\,(1 - \delta_{i,j}) + \alpha_i \delta_{i,j}, \tag{2}$$

where $\delta_{i,j}$ is Kroneker's delta function and the α_is are constants introduced to stabilize the numerical calculations. Considering the bursty nature of Web traffic, we use $f(\cdot) = \ln(1 + \cdot)$. In principle, the quantity $d_{i,j}$ can be measured through server logs or some estimation algorithm [13, 6]. By definition, D is a square non-negative matrix. Hereafter, we use a sans serif font to indicate matrices and use bold italic to indicate vectors. The norm of vectors is defined as the L_2-norm.

Figure 2 shows a subgraph of the dependency graph expected in the system depicted in Fig. 1. We drew links if $I_s = I_d$ holds between two services, and services involving only q_1 and q_2 are shown there. Generally, the dependency graph of a Web-based system is quite complicated even if the corresponding IP network is simple. For an instance of services, see Section 6.

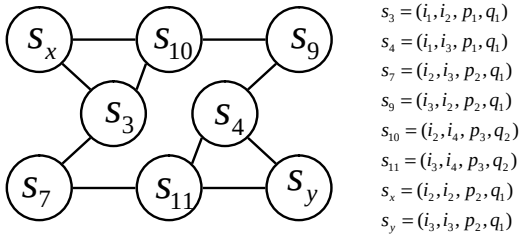

Fig. 2. A part of the dependency graph for the system in Fig. 1. Only services which have $Q = q_1$ or q_2 are shown. Graph edges are drawn if $I_s = I_d$ holds between two vertices.

2.2 Definition of feature vector

Let us assume that the data for the dependency matrix D is sequentially obtained at each time step t=1,2,... with a fixed interval, and that the dependency graph has a single connected component. We define the feature vector $\boldsymbol{u}$ of D as

$$\boldsymbol{u}(t) \equiv \arg\max_{\tilde{\boldsymbol{u}}} \left\{ \tilde{\boldsymbol{u}}^T \mathsf{D}(t) \tilde{\boldsymbol{u}} \right\} \tag{3}$$

subject to $\tilde{\boldsymbol{u}}^T \tilde{\boldsymbol{u}} = 1$, where T denotes transpose. Since D is a non-negative matrix, one can see that the maximum value is attained if the weight of $\boldsymbol{u}(t)$ is larger for services where $\mathsf{D}_{ij}(t)$ is larger. If a service i actively calls other services, $\boldsymbol{u}(t)$ has a large weight for the i-th element. Following this interpretation, we call this feature vector an *activity vector*.

By introducing a Lagrange multiplier λ, Eq. (3) can be rewritten as

$$\frac{d}{d\tilde{\boldsymbol{u}}} \left[\tilde{\boldsymbol{u}}^T \mathsf{D}(t) \tilde{\boldsymbol{u}} - \lambda \tilde{\boldsymbol{u}}^T \tilde{\boldsymbol{u}} \right] = 0, \tag{4}$$

so that

$$\mathsf{D}(t) \tilde{\boldsymbol{u}} = \lambda \tilde{\boldsymbol{u}}. \tag{5}$$

While this equation holds for any of the eigenvectors of $\mathsf{D}(t)$, the feature vector corresponding to Eq. (3) is defined as the principal eigenvector (the eigenvector whose eigenvalue is the largest). Since Eq. (5) is homogeneous in $\tilde{\boldsymbol{u}}$, the direction of the activity vector is invariant with respect to $\mathsf{D}(t) \to \eta \mathsf{D}(t)$ for any nonzero real number η. Thereby we can exclude overall traffic changes from analysis. It is the eigenvalue that is proportional to the global traffic volume. This is important to abstract a hidden structure from D.

To understand the meaning of $\boldsymbol{u}$ further, one can relate $\boldsymbol{u}$ with a stationary state of a discrete-time linear dynamical system whose equation of motion is given by

$$\boldsymbol{x}(\tau + 1) = \mathsf{D}(t) \boldsymbol{x}(\tau), \tag{6}$$

where τ denotes a virtual time being independent of the actual time t, and $\boldsymbol{x}$ is associated with $\boldsymbol{u}$ by $\boldsymbol{u}$=$\boldsymbol{x}/||\boldsymbol{x}||$. Since $\mathsf{D}(t)$ is symmetric and of full-rank at least for $\alpha > 0$, all eigenvalues are real. Using the eigenvalues, $\boldsymbol{x}(0)$ can be expressed as a linear combination of the eigenvectors, so that

$$\boldsymbol{x}(\infty) = \lim_{n\to\infty} [\mathsf{D}(t)]^n \, \boldsymbol{x}(0) = \lim_{n\to\infty} \sum_{i=1}^{N} [\lambda_i(t)]^n c_i(t) \boldsymbol{u}_i(t),$$

where the eigenvalues and the normalized eigenvectors are denoted by $\lambda_i(t)$ and $\boldsymbol{u}_i(t)$ for i=1, 2, ..., N, respectively, and $c_i(t)$'s are coefficients of the linear combination. Evidently, the term of the maximum eigenvalue becomes dominant as $n \to \infty$. Thus, we have

$$\boldsymbol{u}(t) = \boldsymbol{x}(\infty)/||\boldsymbol{x}(\infty)||.$$

Specifically, the state vector approaches $\boldsymbol{u}$ after an infinite number of transitions. For computer systems, the stationary state can be interpreted as the distribution of the probability amplitude that a service is holding the control token of the system at a virtual time point of τ.

2.3 Activity vectors in disconnected systems

In real computer systems, the dependency graph is often disconnected. For such systems, a permutation matrix P exists such that

$$\mathsf{P}^T \mathsf{D} \mathsf{P} = \begin{bmatrix} \mathsf{D}_1 & & 0 \\ & \mathsf{D}_2 & \\ 0 & & \ddots \end{bmatrix},$$

where D_1, D_2, ... are square submatrices. To be concrete, consider the system shown in Fig. 3. Using

$$\mathsf{P} = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&0&0&0&1&0 \\ 0&0&1&0&0&0 \\ 0&0&0&0&0&1 \\ 0&1&0&0&0&0 \\ 0&0&0&1&0&0 \end{bmatrix},$$

the whole dependency matrix is decomposed into two square submatrices:

$$\mathsf{D}_1 = \begin{bmatrix} 0 & a_{15} & a_{13} & 0 \\ a_{15} & 0 & 0 & a_{56} \\ a_{13} & 0 & 0 & a_{36} \\ 0 & a_{56} & a_{36} & 0 \end{bmatrix}, \; \mathsf{D}_2 = \begin{bmatrix} 0 & a_{24} \\ a_{24} & 0 \end{bmatrix}. \tag{7}$$

Evidently, each submatrix corresponds to a single connected subgraph. Since the eigenvalue equation is invariant with respect to orthogonal transformations, the whole eigenvalue equation is written as

$$0 = \det\left|\mathsf{D}_1 - \lambda \mathsf{E}_{(4)}\right| \cdot \det\left|\mathsf{D}_2 - \lambda \mathsf{E}_{(2)}\right|, \tag{8}$$

where $\mathsf{E}_{(n)}$ represents the n-dimensional identity matrix. Consequently, the solution of the whole system can be obtained as the union of the solutions of each connected component. This fact allows us to analyze each subgraph separately.

For each connected component, the Perron-Frobenius theorem [3], which holds for non-negative irreducible matrices, guarantees that the principal eigenvector is positive, where an eigenvector is said to be positive if all the components of $\boldsymbol{u}$ or $-\boldsymbol{u}$ are positive and the corresponding eigenvalue is positive. This naturally supports the interpretation of the principal eigenvector as the activity vector, since the magnitude of the activities should be positive. In addition, the Perron-Frobenius theorem also guarantees that the principal eigenvalue is real [1] and has no degeneracy. From this, we understand that the activity vector is free from subtle problems due to level crossings of the eigenstates within a single connected component in the normal state of the system. If a level crossing easily occurs due to small fluctuations, the transition from one eigenstate to another eigenstate may be recognized as an outlier, resulting in a false alert. For more discussion on the stability of activity vectors, see [8].

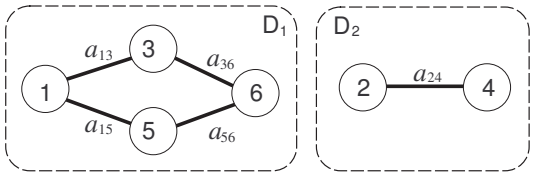

Fig. 3. Example of a disconnected graph.

2.4 Remark

Our feature extraction technique provides a natural way to summarize the information contained in D. The eigencluster decomposition allows us to analyze each single eigencluster separately, and the activity vector extraction technique allows us to further reduce the degrees of freedom. When the set of all services is unknown, it is practically possible to find the activity vectors by choosing positive vectors from a set of eigenvectors [11]. Thus, we expect that the degrees of freedom of each of subproblems are still moderate even when the whole degrees of freedom are very large. In addition, the feature vector has a clear

[1] In this case, all of eigenvalues are real since Eq. (2) makes D real and symmetric.

interpretation that is comprehensible to system administrators. Understanding what is happening is as essential as detection itself in practical situations. These are advantages over naive approaches such as defining a feature vector simply by connecting all of the column vectors, where the scalability cannot be achieved and interpretation of results is often unclear.

For the numerical calculations, an extremely fast and simple algorithm called the power method [12] is known to find the principal eigenvector. While the activity vector must be calculated online whenever D is updated in the given time interval Δt, typically on the order of a few tens of seconds, our experience shows that the time to convergence is far less than Δt even for N on the order of 10^3.

3 Anomaly metric

3.1 Definition

Now we consider how to detect anomalous changes from the sequence of activity vectors $\{\boldsymbol{u}^{(t)}\}$ for $t = 1, 2, ...$ Since $\boldsymbol{u}^{(t)}$ is normalized, this is a time sequence of *directional data*.

To define the anomaly measure, recall the fact that the Fisher kernel function [9] defines a natural affinity between observables in terms of Fisher's information matrix. For the vMF distribution, the Fisher kernel function is given as

$$K(\boldsymbol{u}_i, \boldsymbol{u}_j) = \kappa^{-2} \left[\frac{\partial \ln p(\boldsymbol{u}_i|\kappa, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}} \right]^T \frac{\partial \ln p(\boldsymbol{u}_j|\kappa, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = \boldsymbol{u}_i^T \boldsymbol{u}_j.$$

This is nothing but the cosine similarity. Since it takes a value within $[0, 1]$, we define the anomaly (or dissimilarity) measure $z(t)$ as

$$z(t) \equiv 1 - K\left(\boldsymbol{r}(t), \boldsymbol{u}(t)\right), \tag{9}$$

where $\boldsymbol{r}(t)$ denotes the past typical pattern defined at t. The value of $z(t)$ is unity if the present activity vector is orthogonal to the typical pattern, and zero if the present activity vector is identical to the typical pattern. In the present context, if $z(t)$ is greater than a given threshold, we infer that an anomalous situation is occurring in the system.

3.2 Extraction of typical activity pattern

We define a matrix $\mathsf{U}(t)$ by

$$\mathsf{U}(t) = \left[\boldsymbol{u}(t-1), \boldsymbol{u}(t-2), \cdots, \boldsymbol{u}(t-W)\right], \tag{10}$$

where W is a window size. Clearly, $\mathsf{U}(t)$ is an $N \times W$ matrix. We suppose that the typical pattern is a linear combination of the column vectors:

$$\boldsymbol{r}(t) = c \sum_{i=1}^{W} v_i \boldsymbol{u}(t-i), \tag{11}$$

where c is the normalization constant to satisfy $\boldsymbol{r}^T\boldsymbol{r} = 1$ under the condition of $\sum_{i=1}^{W} v_i^2 = 1$. The easiest way to obtain $\boldsymbol{r}(t)$ is to assume that the v_is are independent of i. In that case, $\boldsymbol{r}(t)$ is parallel to the mean vector, $\overline{\boldsymbol{r}}(t)$. Practically, a good way to reduce the unwanted effects of noisy fluctuations is to optimize the coefficients $\boldsymbol{v}^T = (v_1, v_2, ..., v_W)$ based on

$$\boldsymbol{v}(t) \equiv \arg\max_{\tilde{\boldsymbol{v}}} \left|\left| \sum_{i=1}^{W} \tilde{v}_i \boldsymbol{u}(t-i) \right|\right|^2 = \arg\max_{\tilde{\boldsymbol{v}}} \left\{ \tilde{\boldsymbol{v}}^T U(t)^T U(t) \tilde{\boldsymbol{v}} \right\} \tag{12}$$

subject to $\tilde{\boldsymbol{v}}^T\tilde{\boldsymbol{v}} = 1$. It is well-known in the field of pattern recognition that the solution of this equation is given by the Karhunen-Loève decomposition [5]. Specifically, $\boldsymbol{v}(t)$ is a right singular vector of $\mathsf{U}(t)$, and c is the inverse of the corresponding singular value. So, we conclude that $\boldsymbol{r}(t)$ is the principal left singular vector of $\mathsf{U}(t)$, where a singular vector is said to be principal if it corresponds to the largest singular value. Again, the power method [12] is a good way to perform the singular value decomposition (SVD).

4 Generative model for anomaly metric

4.1 Marginal distribution over z

A conventional method to detect anomalies in a time sequence of multivariate vectors is to find outliers using a generative model that describes the distribution of the multivariate vectors [15, 14, 2]. However, as discussed in Introduction, such approaches have difficulties for high-dimensional data. Instead, we consider a pdf of the anomaly measure itself, assuming that the distribution of $\boldsymbol{u}$ basically obeys the vMF distribution given in Eq. (1).

Before plunging into the detail, we summarize our anomaly detection procedure in Fig. 4, where we denote the angle between $\boldsymbol{r}(t)$ and $\boldsymbol{u}(t)$ as θ. As shown, the basic procedure is to extract a typical pattern from the past activity vectors, and to calculate the dissimilarity of the present activity vector from this typical one. We believe that this is reasonable approach if the typical pattern is relatively stable, and it is the case at the application layer of Web-based computer systems.

Since θ has a one-to-one correspondence to z as $z = 1 - \cos\theta$, one can derive the pdf over z through the marginalized distribution with respect to θ, starting from Eq. (1). We perform a transformation of the variables from $\boldsymbol{u}$ to angular variables $\{\theta, \theta_2, ..., \theta_{N-1}\}$ of the N-dimensional spherical coordinates. By using

$$d^{N-1}\Omega = d\theta d\theta_2 \cdots d\theta_{N-1} \sin^{N-2}\theta \sin^{N-3}\theta_2 \cdots \sin\theta_{N-2},$$

where $d^{N-1}\Omega$ is the area element on the unit sphere in an N-dimensional Euclidean space, the marginalized distribution for θ is written as

$$\int d\theta_2 d\theta_3 \cdots d\theta_{N-1} \left[p(\boldsymbol{u}|\kappa, \boldsymbol{\mu}) \sin^{N-2}\theta \sin^{N-3}\theta_2 \cdots \sin\theta_{N-2} \right]. \tag{13}$$

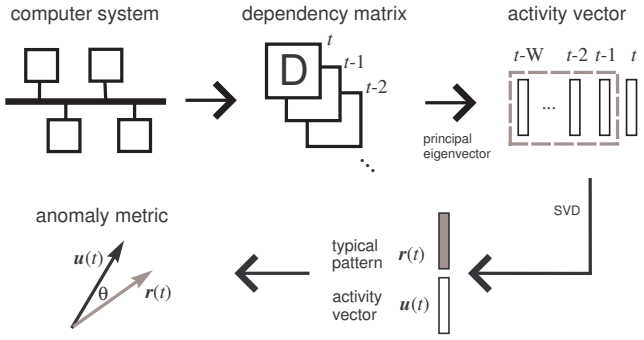

Fig. 4. Summary of our anomaly detection procedure.

Since

$$z(t) \simeq \frac{\theta^2}{2},\ \cos\theta \simeq 1 - \frac{\theta^2}{2},\ \sin\theta \simeq \theta$$

hold for $|\theta| \ll 1$, we see that the distribution for z is given by

$$q(z) \propto \exp\left[-\frac{z}{2\Sigma}\right] z^{\frac{N-1}{2}-1}, \tag{14}$$

where we used $\theta d\theta = dz$ and set $1/2\kappa$ to be Σ. Apart from a prefactor and the scaling factor Σ, this is the same as the χ^2-distribution with $N-1$ degrees of freedom.

We have derived this generative model from the vMF distribution of $\boldsymbol{u}$, which is the most natural assumption as long as the fluctuations around the mean direction is relatively small. However, our empirical study shows that the above model is not consistent with the experimental distribution at all. One reason can be found in the fact that some of the degrees of freedom happen to be inactive over some duration of time. In the derivation of the vMF distribution, an implicit assumption is that all of the degrees of freedom are equally active. These observations lead us to the concept of the effective dimension.

4.2 Effective dimension

One of the most important steps in our formulation is to replace N in Eq. (14) with a parameter n, and regard it as a fitting parameter. We call n the *effective dimension* of the system. If properly estimated, the effective dimension represents the active degrees of freedom of the system. We expect that n is much smaller than N in many application domains. For Web-based systems, the activities of some of services are much lower than those of others, so that n is much smaller than N, as shown in Section 6.

Since the function $q(z)$ rapidly decreases as $z \to \infty$ for a moderate value of the degrees of freedom, the normalization constant can be evaluated by integrating

over $[0, \infty)$. Using the definition of the gamma function $\Gamma(\cdot)$, we have

$$q(z|n, \Sigma) = \frac{1}{(2\Sigma)^{\frac{n-1}{2}}\Gamma(\frac{n-1}{2})} e^{-z/(2\Sigma)} z^{\frac{n-1}{2}-1}, \tag{15}$$

where we use the notation of $q(z|n, \Sigma)$ instead of $q(z)$ to emphasize the effective dimension as a fitting parameter. We see that the distribution of $z \in [0, 1]$ can be written as the χ^2-distribution with $n - 1$ degrees of freedom.

5 Online estimation of parameters

5.1 Moment-based estimation scheme

To estimate the two parameters n and Σ in Eq. (15), we employ a moment-based estimation scheme. Note that MLE for the χ^2-distribution is intractable due to the gamma function. Fortunately, analytic formulas about the first (m_1) and second (m_2) moments are available for the χ^2-distribution. By using the definition of the gamma function, it is easy to show

$$m_1 \equiv \int_0^\infty dz\, q(z|n, \Sigma) z = (n-1)\Sigma$$

and

$$m_2 \equiv \int_0^\infty dz\, q(z|n, \Sigma) z^2 = (n^2-1)\Sigma^2,$$

where we again extended the domain of integration to $[0, \infty)$. This can be also justified within the approximation made above. Solving these equation with respect to n and Σ, we have

$$n = 1 + \frac{2{m_1}^2}{{m_2} - {m_1}^2} \quad \text{and} \quad \Sigma = \frac{{m_2} - {m_1}^2}{2m_1}. \tag{16}$$

These relations provide a way to evaluate the parameters n and Σ experimentally. For example, if we have T samples $\{z(t)|t = 1, 2, ..., T\}$, the experimental first and second moments can be calculated as $\hat{m}_1 = (1/T)\sum_{t=1}^T z(t)$ and $\hat{m}_2 = (1/T)\sum_{t=1}^T z(t)^2$, respectively.

5.2 Incremental algorithm

These formulas can be extended to their incremental versions. Using an identity

$$\frac{1}{t}\sum_{i=1}^{t} z(t) = \left(1 - \frac{1}{t}\right)\frac{1}{t-1}\sum_{i=1}^{t-1} z(i) + \frac{1}{t} z(t).$$

and setting $1/t$ as β, we have

$$\hat{m}_1(t) = (1-\beta)\cdot \hat{m}_1(t-1) + \beta z(t). \tag{17}$$

Similarly for the second moment, we have

$$\hat{m}_2(t) = (1-\beta)\cdot \hat{m}_2(t-1) + \beta z(t)^2. \tag{18}$$

Naturally, β satisfies $0 < \beta < 1$ and is called the discounting factor.

Since $1/\beta$ can be associated with the number of data points, a rough estimate of β may be $\beta \sim \Delta t/L$, where L denotes the time scale we are interested in. Similarly, W can be estimated as $W \sim L/\Delta t$. In our benchmark system, we empirically take L on the order of 10 minutes.

Based on the above discussion, we have an online algorithm to calculate a threshold value to judge whether it is anomalous or not:

1. Give a critical boundary $0 < p_c < 1$.
2. Calculate $\hat{m}_1$ and $\hat{m}_2$ at t using Eqs. (17) and (18).
3. Calculate n and Σ using Eq. (16).
4. Find z_{th} numerically such that $\int_{z_{\mathrm{th}}}^{\infty} dz\, q(z|n,\Sigma) = p_c$.
5. Emit an alert if $z(t) > z_{\mathrm{th}}$.

The above algorithm includes three parameters, p_c, β, and W. Since β and W can be easily estimated with L and Δt, the only parameter we must specify is substantially p_c, which is totally independent of the details of the system.

6 Experiment

6.1 Experimental settings

The configuration of our benchmark system is illustrated in Fig. 1. As shown, the HTTP servers and WASs are doubly redundant. On the WASs, two applications, "Trade" and "Plants," are running. Trade is a standard benchmark application called Trade 3 [7], and Plants is a sample application bundled with IBM WebSphere Application Server V5.0 and simulates an online store dealing with plants and gardening tools. For both, the number of clients was fixed to be 16 and the think time was randomly chosen from 0 to 4 seconds.

We generated a matrix D every 20 seconds using a method that evaluates $d_{i,j}$ from captured IP packets. Loopback packets were ignored in the experiments, so that the services s_x and s_y in Fig. 2 are not observed for $i_1 = 192.168.0.53$ and $i_2 = 192.168.0.54$. The principal eigencluster is defined in Table 1, and small perturbations affecting it were ignored. In Table 1, the zeroth service was introduced to describe the situation where an optimal pair between callee and caller could not be identified. For example, services triggered by those outside the intranet will be associated with the zeroth service.

Apart from these, there are other service types, "DB2" and "JMS," in Table 1. DB2 denotes a request for the DB server, and JMS is for communications related to the Java Messaging Service.

Table 1. Services appearing in the principal eigencluster

Index	I_s	I_d	P	Q
0	0.0.0.0	0.0.0.0	0	(none)
1	192.168.0.19	192.168.0.53	80	Plants
2	192.168.0.19	192.168.0.54	80	Plants
3	192.168.0.19	192.168.0.53	80	Trade
4	192.168.0.19	192.168.0.54	80	Trade
5	192.168.0.54	192.168.0.53	5558	JMS
6	192.168.0.53	192.168.0.54	9081	Plants
7	192.168.0.53	192.168.0.54	9081	Trade
8	192.168.0.54	192.168.0.53	9081	Plants
9	192.168.0.54	192.168.0.53	9081	Trade
10	192.168.0.53	192.168.0.52	50000	DB2
11	192.168.0.54	192.168.0.52	50000	DB2

6.2 Statistical properties in the normal state

We calculated $\boldsymbol{u}$ and z online over a period when the system exhibited no failures. The dependency matrix was generated over 52.7 minutes, so we had 158 matrices. The α_i values were taken as small random numbers on the order of 0.01. To see the fluctuation in D, we show in Fig. 5 the time dependence of $d_{9,11}$ as an example. We see that there are approximately 500 calls within 20 seconds under these experimental conditions and that the amplitude of fluctuation of $d_{9,11}$ is almost of the same order as the average. Hence, it makes little sense to place a threshold value on an isolated $d_{i,j}$.

To experimentally validate the pdf of z, we plotted the frequency distribution of z in Fig. 6 (a), where the χ^2 pdf is also shown. The parameters of the χ^2 pdf were calculated using all of the 158 data points with no discounting. The result was

$$n = 4.62 \quad \text{and} \quad \Sigma = 6.79 \times 10^{-5}.$$

It is noteworthy that the calculated effective dimension is much less than $N = 12$. In spite of the limitation of the number of data points, the frequency distribution is a good fit to the χ^2 pdf. We also drew a quantile-quantile plot in Fig. 6 (b). As shown, the experimental data is well placed on the 45 degree line. These results clearly support our formulation.[2]

6.3 Detection of an application fault

Next, we performed a more realistic experiment: A bug in one of the applications ("Plants") only on 192.168.0.54 causes a malfunction of the service of 11 at a time point. The server process itself continues running, so the network communication

[2] We rounded the $n - 1$ value to be 4 to fit the χ^2 pdf because of the limitation of the numerical library we used. This is the main reason the deviation of the χ^2 pdf from the experimental frequency.

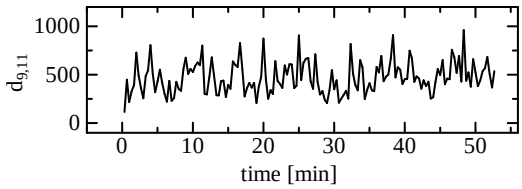

Fig. 5. Time dependence of $d_{9,11}$.

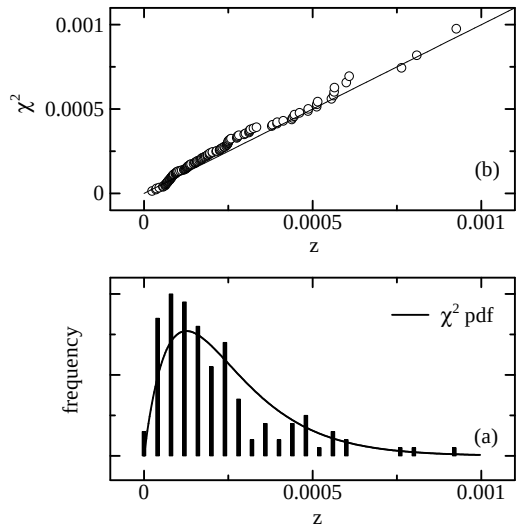

Fig. 6. Statistics of z in the normal state. (a) Comparison of the experimental frequency and the χ^2 pdf. (b) The quantile-quantile plot.

is normal at the IP layer or below. Since two Web servers are working on the system, a client may feel no change in response time as long as the overall traffic is sufficiently small. Although this defect occurs within a single service, it can cause a massive change in D. In fact, the dependencies of the services directly related with the service 11 will be considerably changed. What we would like to detect is a transition of this kind.

Figure 7 shows the generated time-series of the activity vector. We see that a sudden change in activities is observed at t_A and t_B, which correspond to the malfunction of the service 11 and its recovery. From the figure, the activities of the services 2, 6, and 11 are clearly decreased during this period. This result demonstrates that the service activity vector actually expresses the activity of services, and suggests a way to visualize the whole system.

To detect this fault automatically, we calculated z and its threshold value, following the algorithm explained in Section 5. In Fig. 8, we depicted the z values with vertical bars and the threshold values with thick gray curves for W =5, 25, and 50. The discounting factor and the critical boundary were taken as $\beta = 0.005$ and $p_c = 0.5\%$, respectively. While the result is considerably affected by the choice of W, we observe clear features at t =35.0 and 45.7 minutes, which correspond to t_A and t_B in Fig. 7, respectively. These time points are highlighted with dashed vertical lines in Fig. 8. Note that the feature at t_B (recovery from the malfunction) demonstrates the learnability for gradual changes of the environment. The dependence on W is an inevitable consequence of the choice of the applications. Since the benchmark applications simulate human behavior, they must have a characteristic time scale. Comparing Fig. 7 with Fig. 8, we conclude that an appropriate value of W is about 25 (8.3 minutes). We see that this value of W allows us to pinpoint the time points t_A and t_B.

The curves plotted with thin lines ("not SVD") in Fig. 8 represent the result using the simple mean vector $\overline{\boldsymbol{r}}$ instead of $\boldsymbol{r}$. The trend of z is similar to that of the SVD-based method, but is blurred out by the noise. This result demonstrates the effectiveness of the SVD-based pattern extraction technique.

For the limitations of our approach, first, the probability of false alarms will be finite even if W is set to be the optimal value. As understood from Fig. 8, there is small finite probability of having outliers beyond a threshold value. Second, since the basic assumption of our approach is the stability of the direction of the activity vector, our approach is not appropriate for anomaly detection of rarely invoked services. Finally, there is much room for improvement in the calculations of the threshold values since the numerical library we used handles only integer degrees of freedom in the χ^2 pdf.

7 Summary

We have proposed a new framework of statistical anomaly detection for a time-sequence of directional data. First, we defined an anomaly metric, z, based on the Fisher kernel function of the von Mises-Fisher distribution, and derived its probability distribution as the χ^2 distribution in an approximated manner.

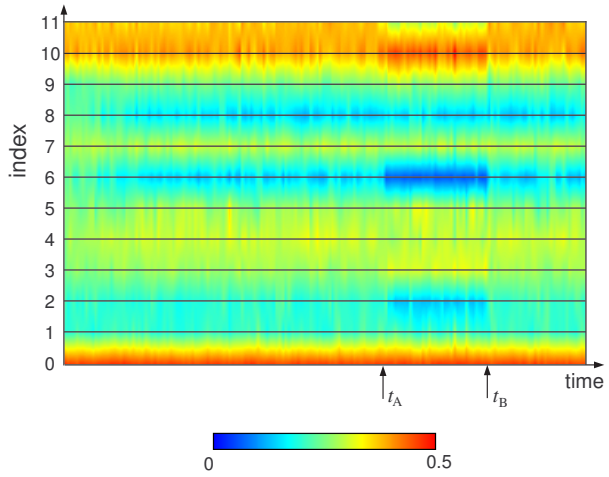

Fig. 7. Time dependence of the activity vector. The failure duration starts at t_A and ends at t_B, as shown by arrows. The definition of the service indices are shown in Table 1

Second, we proposed a new concept of the effective dimension, n, and gave its online estimation algorithm based on the method of moments. Our generative model of z is the χ^2 distribution with $n-1$ degrees of freedom. Third, we derived an online algorithm to calculate threshold values of z. Only a value of the critical probability p_c is needed to determine the threshold. Finally, we demonstrated the utility of our method in a fault detection task in a benchmark computer system.

References

1. A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra. Expectation maximization for clustering on hyperspheres. *Technical Report*, TR-03-07, 2003. Department of Computer Sciences, University of Texas at Austin.
2. A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra. Generative model-based clustering of directional data. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 19–28, 2003.
3. A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*, volume 9 of *Classics in applied mathematics*. SIAM, 1994.
4. S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
5. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2000.
6. M. Gupta, A. Neogi, M. K. Agarwal, and G. Kar. Discovering dynamic dependencies in enterprise environments for problem determination. In *Proceedings of*

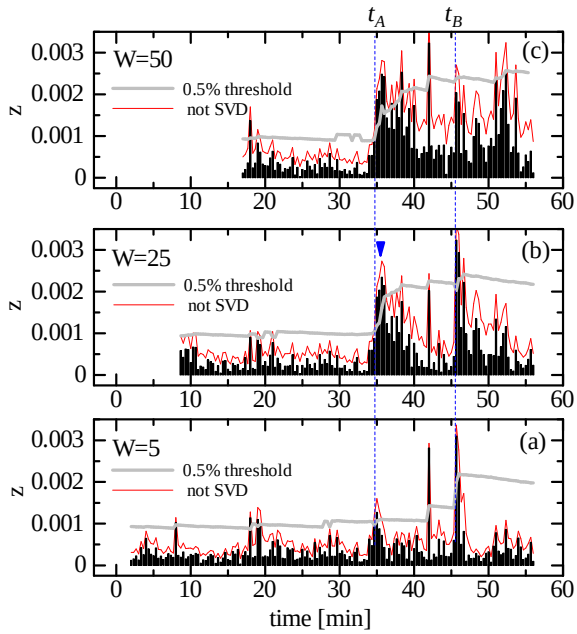

Fig. 8. The dependence of z for W = (a) 5, (b) 25, and (c) 50. The 0.5% threshold is denoted by gray curves.

14th IFIP/IEEE Workshop on Distributed Systems: Operations and Management, pages 221–233, 2003.
7. IBM. Trade3; http://www-306.ibm.com/software/webservers/appserv/benchmark3.html.
8. T. Idé and H. Kashima. Eigenspace-based anomaly detection in computer systems. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.
9. T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems*, volume 11, pages 487–493, 1999.
10. K. Mardia. *Multivariate Analysis.* Academic Press, 1980.
11. S. Sarkar and K. Boyer. Quantitative measures for change based on feature organization: Eigenvalues and eigenvectors. *Computer Vision and Image Understanding*, 71:110–136, 1998.
12. G. Strang. *Linear Algebra and its Applications.* Academic Press, 1976.
13. The Open Group. Application response measurement — ARM; http://www.opengroup.org/tech/management/arm/.
14. K. Yamanishi and J. Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 676–681, 2002.
15. K. Yamanishi, J. Takeuchi, G. Williams, and P. Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 320–324, 2000.