Change Detection From Heterogeneous Data Sources

Tsuyoshi Idé

Abstract Detecting the changes in business situations is an important technical challenge. This chapter focuses on *change detection* technologies, including outlier detection and change-point detection. In particular, we focus on how to handle the heterogeneous and dynamic features of the data in service businesses. We introduce a method of the singular spectrum transformation for change-point detection in heterogeneous data. We also introduce a technique for proximity-based outlier detection to handle dynamic data. Using real-world sensor data, we demonstrate the utility of the proposed methods.

1 Introduction

Recent advances in sensing and storage technologies are making it possible to collect and store real-valued time-series data in various domains. Examples of the data include POS (point-of-sales), biomarker, geospatio-temporal, etc. Unlike human-generated data such as call center text logs, analyzing real-valued time-series data is generally challenging, since the values of the sensors are not directly meaningful in general, and the amount of data is often intractably huge.

For industrial domains such as transportation, manufacturing, energy & utilities, etc., where optimized operations of physical systems are at the heart of successful business, the effective use of sensor data is critical. For example, early detection of a systematic occurrence of defective products is essential to avoid potential losses. We have recently witnessed rapid changes towards service businesses in various industries. Recent examples include system monitoring services for production systems and construction equipment. Another example where the analysis of sensor data is critical is location-based services, which are growing rapidly. To exploit geospatial

Tsuyoshi Idé
IBM Research – Tokyo, 1612-14 Shimo-tsuruma, Yamato-shi, 242-8502 Kanagawa, Japan, e-mail: goodidea@jp.ibm.com

data, real-time position information needs to be analyzed by combining it with certain data from individual products and consumers. All these examples clearly show the need for practical methods for analyzing sensor data in service businesses.

In spite of the growing awareness of sensor data analytics for service businesses, little about this topic appears in the literature. This is possibly due to the fact that knowledge discovery from noisy sensor data is quite difficult with traditional approaches, and the problem settings can be quite different from traditional situations. Figuratively, traditional methods are capable of handling only a small percentage of the data, leaving the rest unused. This also means that practical new technologies could lead to a major business advantage in the unexplored spaces, just as information retrieval techniques based on new disciplines of machine learning opened new doors to business on the Internet.

In this chapter, we focus on *anomaly detection* technologies, including the tasks of outlier detection and change-point detection. In particular, we focus on how to handle the heterogeneous and dynamic data that is often collected by service businesses. Toward this goal, we propose two new technologies. First, we introduce a change-point detection method called *singular spectrum transformation* (SST). Although traditional approaches to change-point detection consist of two separated steps, typically density estimation and scoring, SST unifies them to give a one-step algorithm with the aid of the mathematical theory called the Krylov subspace method. Thanks to the simplified structure of the algorithm, SST is quite robust to heterogeneities in the data.

Second, we propose a novel technique for proximity-based outlier detection. In this approach, we use a regularization technique to automatically discover modular structures of the system. In other words, for each variable, the algorithm automatically finds a set of variables that are in proximity to the variable. The size of the proximity sets is automatically determined in accordance with the strength of regularization. This feature is quite useful in heterogeneous systems, since how many neighbors each variable has depends on the nature of each variable. Based on the proximity analysis, we compute the degree of outlier-ness in a probabilistic fashion.

Here is the structure of the chapter. In Section 2, we first review previous approaches to anomaly detection, and then summarize the motivation behind our approach. In Section 3, we describe the practical change-point detection method called SST based on our previous work [13, 17]. In Section 4, we propose a novel outlier detection method based on sparse structure learning. In Section 5, we briefly present some experimental results that demonstrate the utility of our methods. Finally, in Section 6, we summarize our results.

2 Previous work and our motivation

As mentioned in the introduction, sensor data has different features from traditional data such as that used in statistics and data mining. Except for cases in which we have detailed knowledge of the internal structures of the systems, there are only a few options available. In practice, detecting signs of changes is perhaps the most important task. The first half of this section reviews existing anomaly detection techniques and their limitations. For a more complete survey, see [5].

There are many scenarios for anomaly detection, depending on the perspectives and the definitions of anomalies. In the data mining community, these scenarios appear in the literature:

- Outlier detection
- Change-point detection
- Discord discovery

We will give a brief description of each task in the following subsections. In what follows, we assume that we are given a sequence in an M-dimensional vector $\mathscr{D} \equiv \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(N)}\}$ up to a discrete time point N. By definition, $\boldsymbol{x}^{(n)} = ({x_1}^{(n)}, {x_2}^{(n)}, \cdots, {x_M}^{(n)})^\top$, and at each time point, which is assumed to be equi-interval, we observe M values from individual sensors. The superscript $^\top$ represents the transpose operation.

2.1 Outlier detection

Outlier detection looks at how much novelty a single sample reveals. Examples include temperature monitoring of a chemical plant, where an alert must be raised when an exceptionally high temperature is observed. In general, outlier detection consists of two steps: *density estimation* and *scoring*.

In the context of sensor data, density estimation is the same as creating a predictive model, and the goal of this step is to find a probability density $p(\boldsymbol{x}|\mathscr{D})$ that predicts the value of a newly observed sample, given the previous data $\mathscr{D}$. There are roughly two types of approaches for this step. One is based on density estimation techniques for i.i.d. (identically and independently distributed) samples, and the other is based on time-series prediction techniques. First we look at the i.i.d. models and time-series prediction methods are covered in the next subsection.

In statistics, a standard approach is to assume a Gaussian distribution:

$$\mathscr{N}(\boldsymbol{x}|\boldsymbol{\mu}, \Lambda^{-1}) = \frac{|\Lambda|^{\frac{1}{2}}}{(2\pi)^{\frac{M}{2}}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Lambda (\boldsymbol{x}-\boldsymbol{\mu})\right\}. \quad (1)$$

For the model parameters, the mean $\boldsymbol{\mu}$ and the precision matrix Λ, are typically determined using maximum likelihood. By using estimated parameters, $\hat{\boldsymbol{\mu}}$ and $\hat{\Lambda}$, in the model, we have $p(\boldsymbol{x}|\mathscr{D}) = \mathscr{N}(\boldsymbol{x}|\hat{\boldsymbol{\mu}}, \hat{\Lambda}^{-1})$. Although the Gaussian is the simplest model for multivariate data, accurately estimating Λ, which is defined as the inverse of the covariance matrix S, is challenging in practice, as explained below.

Based on this model, Hotelling's T^2-test is widely used as the standard technique of outlier detection [1]. The idea is to use the (squared) Mahalanobis distance

$$T^2(\boldsymbol{x}|\boldsymbol{\mu},\Lambda) = (\boldsymbol{x}-\boldsymbol{\mu})^\top \Lambda(\boldsymbol{x}-\boldsymbol{\mu})$$

as the measure of outlier-ness. Note that T^2 itself is a random variable given $\boldsymbol{x}$, since estimated values of $\boldsymbol{\mu}$ and Λ will not be perfect for finite training samples. In this definition, the precision matrix represents the effect of heterogeneities of different dimensions. Specifically, if one variable has a large variance (i.e. small precision), then the distribution should be stretched along the axis, giving an ellipsoidal distribution.

Although the Hotelling test is theoretically mature, in practice it is known to produce many false alerts. This is because the set of assumptions behind the theory are not fully satisfied. Problems that have been addressed to date in the literature include:

1. Non-stationarity. The distribution may change over time.
2. Multi-modality. The distribution consists of several clusters of densities.
3. Numerical instability. If the dimensionality is high ($M \gtrsim 30$) or collinear, then the rank-deficiency makes it hard to invert the covariance matrix.

For the first and second problems, Yamanishi et al. [32] proposed using a sequential parameter estimation algorithm for Gaussian mixtures. Although in theory this approach seems useful to handle the problems of non-stationarity and multi-modality, their algorithm is known to be numerically unstable in practice. This because estimation of a covariance matrix is much harder than expected especially for high dimensional systems. Breunig proposed a simpler and numerically stable approach with a metric for outlier-ness called LOF (local outlier factor) [4]. Although LOF was first introduced in an intuitive fashion, this metric amounts to an approximation for the density estimation step, where a k-nearest neighbor (k-NN) heuristic is used in place of full density estimation. Thanks to the locality of k-NN, LOF can in principle handle multi-modality. However, again, the intuitive notion of LOF does not necessarily work with many dimensions, where, due to the curse of dimensionality, all of the samples are necessarily very close to each other. Also, choosing an optimal k requires a heuristic approach. In addition, the k-NN approach is memory intensive since all of the previous samples must be available.

For the third problem, numerical instability, which is essentially due to the gap between the nominal and intrinsic dimensionalities, there are at least two approaches. The first approach is *dimensionality reduction*, which focuses on a subspace where the redundant dimensions are ignored. One of the earliest work in this direction in the context of anomaly detection is [14], where the use of PCA (principal component analysis) was proposed to detect anomalies in computer networks. The second approach involves the use of *regularization*. In Section 4, we explain how useful it is in outlier detection.

2.2 Change-point detection

Change-point detection is the problem of detecting structural changes in the data generation mechanism behind observed data. For example, one might want to raise an alert when the system starts producing unusual vibrations even if the variables are within standard ranges.

Unlike outliers, change-points can take various forms, such as cusps, steps, changes in frequency, etc. A general-purpose approach is to learn a generative model for the data based on previous recordings, and compute the degree of goodness of fit for the model with the present data. If the goodness of the model is sufficient for the present data, we determine that a change is occurring in the system [3].

In this procedure, there are two steps in change-point detection: *density estimation* and *scoring*. In the first step, we try to find a generative model based on recently observed data. Let w be the size of window along the time axis, and let $\mathscr{D}_w^{(t)}$ be a notation for $\{\boldsymbol{x}^{(t-w+1)}, \boldsymbol{x}^{(t-w+2)}, ..., \boldsymbol{x}^{(t)}\}$. Our first step is to find the probability function that best fits the recent data $\mathscr{D}_w^{(t)}$ for the model $p(\boldsymbol{x}|\mathscr{D})$. For the next scoring step, the likelihood ratio is a basic metric for scoring the degree of change:

$$z(t) \equiv \sum_{\boldsymbol{x} \in \mathscr{D}_w^{(t)}} \ln \frac{p(\boldsymbol{x}|\mathscr{D}_w^{(t)})}{q(\boldsymbol{x})}, \tag{2}$$

where $q(\cdot)$ represents a baseline distribution. In practical scenarios, $q(\cdot)$ is often thought of as the distribution under the normal situation. In this case, the likelihood ratio is a metric of the faultiness of the system.

For a single variable that is Gaussian-distributed around a constant value, a method called CUSUM (cumulated summation) is well-known as a baseline method for change-point detection [3]. If the Gaussian assumption is allowed, the likelihood ratio has a number of desirable properties with Chi-squared distribution and Neyman-Pearson optimality [1]. However, as expected, in most cases in sensor data analytics, its utility is quite limited due to the dynamic and non-stationary nature of the systems.

To tackle the problems in traditional approaches, there are three prominent and recent approaches to change-detection in the data mining community. The first approach, which is perhaps the most similar to the traditional statistical analysis, is based on direct estimation of the density ratio [20]. In this approach, rather than separately estimating the densities of the numerator and denominator, the likelihood ratio is directly modeled and estimated using a kernel method. For details, see [28].

In the second approach, a time-series prediction model is estimated rather than using i.i.d. models to handle the dynamic nature of the data. One of the earliest descriptions includes [31], where a sequential update algorithm is proposed for fitting an AR (auto-regressive) model. We can focus on the simplest case of $M = 1$ for simplicity. The AR model of order m is defined as

$$p(x^{(t)}|\boldsymbol{a}, b) = x^{(t-1)}a_1 + x^{(t-2)}a_2 + \cdots + x^{(t-m)}a_m + b,$$

where $\boldsymbol{a} \in \mathbb{R}^m$ and $b \in \mathbb{R}^1$ are parameters to be estimated from the data in a sequential fashion. As is well-known, the AR model assumes a specific periodicity through m, the order of the AR model. This means that the AR model is not capable of handling non-stationary dynamics. One approach to this problem is to introduce a latent state into the model. The earliest work in this direction includes the method of SST [13, 17], and its theoretical analysis was given in [21], which shows a clear relationship between SST and system identification of state-space models. In a later section, we will revisit this point.

Finally, in the third but maturing approach, the task of change-point detection is treated as a model selection problem [30, 29]. This is a new and interesting research area, where the practical requirements interact with deep theoretical analysis.

2.3 Discord discovery

So far we have looked at approaches explicitly based on probabilistic methods. In addition, algorithmic approaches are also popular in the data mining community. One of the typical tasks in the present context is *discord discovery*. In this task, the time series data is first transformed into a set of subsequences, and then each subsequence is checked to see if it is far from the average behavior. This type of approach is practically useful in some applications. For example, if an unusual pulse pattern is found in the time-series data of an ECG (electro-cardiogram), it may be an indication of a heart attack [22].

Let us consider the simplest case of $M = 1$ for simplicity. Let w be the size of the sliding windows. Using w, we transform the data into a set of subsequences $\{\boldsymbol{s}^{(w)}, \boldsymbol{s}^{(w+1)}, \ldots, \boldsymbol{s}^{(N)}\}$, where

$$\boldsymbol{s}^{(t)} \equiv \left(x^{(t-w+1)},\, x^{(t-w+2)},\, \ldots,\, x^{(t)}\right)^\top \tag{3}$$

is a subsequence represented as a vector in a w-dimensional space. A *discord* is defined as an outlier in the set of subsequences. As metrics of the outlier-ness, the mean and median of the k-NN distances are often used. In this approach, one needs to compute the k-NNs for each sample, which is computationally expensive and memory intensive. To address these limitations, several heuristics have been proposed [33].

In the data mining community, the task of discord discovery (and closely related task of motif discovery) is often handled with a technique called SAX [22]. SAX is a data compression method that converts real-valued time series into discrete symbols. After the conversion, a number of useful techniques in discrete mathematics such as dynamic programming can be used. However, the optimality of the symbolic representations has not been deeply addressed in the literature to date. This is an interesting research topic, which calls for a combination with probabilistic approaches [11].

One subtle problem in the sliding window approach is that the overlap between neighboring windows may cause pathological phenomena such as sinusoidal effects in the subsequence time-series clustering [24, 12]. How to avoid such effects is another interesting research topic [8].

2.4 Goal of this chapter

As mentioned, outlier detection and change-point detection are traditional problem settings in statistics for anomaly detection. However, methods developed in statistics are known to be of limited effectiveness in practice in many cases. A typical example is asymptotic theories. In modern sensor data that can be dynamic and noisy, the number of samples is almost infinite along the time axis. Therefore it is sometimes the case that the confidence interval derived from an asymptotic distribution can be too narrow to produce a reasonable false positive rate. These types of difficulties are well-known in such tasks as FDC (fault detection and classification) in semiconductor manufacturing processes. Therefore recent research focus has been on newly developed approaches in the data mining and machine learning communities.

This chapter covers these new approaches to anomaly detection from two perspectives. First we look at SST for change-point detection. As mentioned in the introduction, SST has the unique feature that the density estimation and scoring steps are unified. As a result, we can avoid numerically unstable parameter estimation. Although SST relies on SVD (singular value decomposition) that is usually computationally expensive, we will show our algorithm based on the Krylov subspace method allows overcoming this issue.

Next we propose a novel method for outlier detection, which is based on sparse structure learning of the graphical Gaussian model (GGM). Our method has a number of advantages over existing methods. First, our algorithm is numerically stable thanks to a regularization technique. Second, the sparse structure learning provides insights into the system. Identifying a sparse structure between variables amounts to looking at an essential relationship between those variables. More importantly, thanks to the sparseness, we can automatically find modular or cluster structures within the system. Finally, based on the modular structure, our algorithm is capable of doing *anomaly localization* [16, 15, 18]. This means that, for M-dimensional time series, our output is M anomaly scores for a single sample, rather than a single scalar. This is a very important feature in practice, since we can easily come up with a response for a detected anomaly once we know which variables are responsible for the fault.

3 Change-point detection

Before getting into the details of the algorithm, let us first look at an motivating example of change-point detection in heterogeneous systems [13]. The essence of our idea is illustrated in Fig. 1, where two artificially generated data sets and their SSTs are shown. While it is difficult to infer any relationship between the two original variables, SST clearly reveals a hidden relationship involving the synchronization of their change points. Note that the results in Fig. 1 (b) and (d) were obtained using a common algorithm and a shared parameter set, so we see that, by performing SST, the problem of data mining in heterogeneous systems can be reduced to those of homogeneous problems without using any detailed knowledge about the behavior of data. The notion of *change-point correlation* [13] is indeed a key idea for knowledge discovery from dynamic systems that are strongly-correlated and heterogeneous.

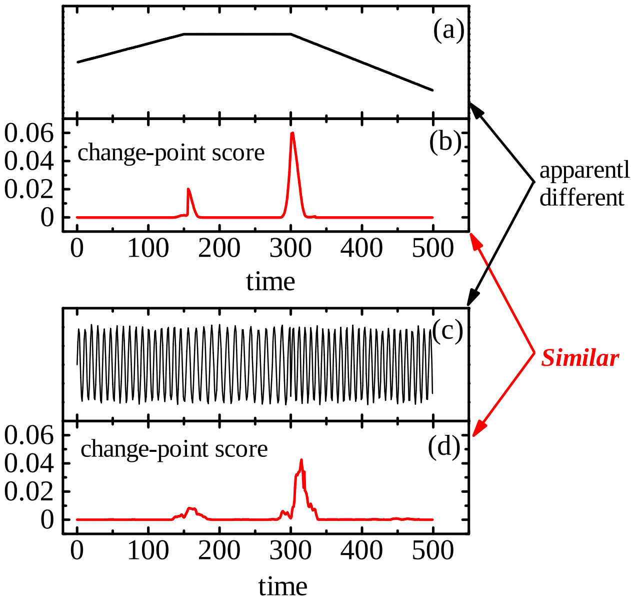

Fig. 1 Example of SST in a heterogeneous system. The original time-series in (a) and (c) are transformed into change-point scores in (b) and (d), revealing a hidden similarity. Clear synchronization of the two change points suggests a causal relationship between the two variables.

3.1 Overview of the SST algorithm

For clarity in the presentation, let us consider a one-dimensional time-series $\{x^{(t)} \in \mathbb{R} \mid t = 1,2,...\}$. We are given a subsequence of length w as Eq. (3) (We assume that

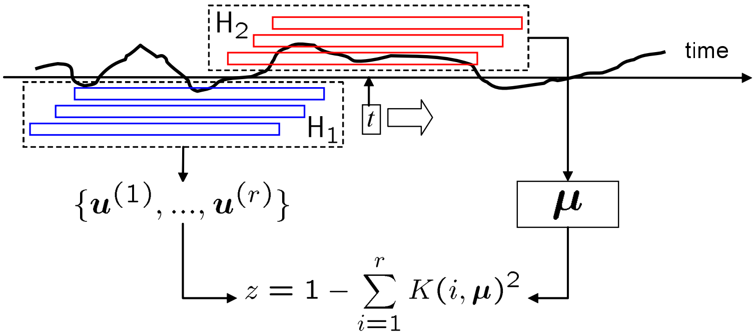
H2
time
H1
t
$\{\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(r)}\}$
$\boldsymbol{\mu}$
$z = 1 - \sum_{i=1}^{r} K(i, \boldsymbol{\mu})^2$

Fig. 2 Overview of SST.

the data points are collected at constant intervals). At each time t, let H_1 and H_2 be matrices containing n subsequences defined as

$$\mathsf{H}_1(t) \equiv [\boldsymbol{s}(t-n), \ldots, \boldsymbol{s}(t-2), \boldsymbol{s}(t-1)]$$
$$\mathsf{H}_2(t) \equiv [\boldsymbol{s}(t-n+\gamma), \ldots, \boldsymbol{s}(t-1+\gamma)],$$

where γ is a positive integer. Fig. 2 shows an example where three subsequences are taken both in the vicinity of the present time and in the past.

The column space of $\mathsf{H}_1(t)$, the space spanned by the column vectors, should contain the information about the patterns appearing in the past domain of the time series. The SST uses the principal components as typical representative patterns of the column space: Find the $r\ (< w, n)$ top left singular vectors of $\mathsf{H}_1(t)$, $\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \ldots, \boldsymbol{u}^{(r)}$. We assume these are orthonormal. Hereafter, we omit the argument t unless confusion is likely. Let the subspace spanned by these vectors be

$$\mathscr{H}_r \equiv \mathrm{span}\{\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \ldots, \boldsymbol{u}^{(r)}\}. \tag{4}$$

Similarly, we can get the representative patterns around the present time t by performing the SVD of H_2. We use the top principal component $\boldsymbol{\mu}$ of H_2 as the representative pattern.

We define the change-point (CP) score z at time t as

$$z \equiv 1 - \sum_{i=1}^{r} K(i, \boldsymbol{\mu})^2, \tag{5}$$

which can be interpreted as the distance between the subspaces. Since it is empirically true that the score is not very sensitive to the choices of n and γ [13], we set $n = w$ and $\gamma = w/2$. For w, a value less than 100 typically works well. An appropriate preprocess (e.g. down-sampling) can be used to adjust w to this range. Empirically, a value of three or four works well for r even when w is on the order of 100.

3.2 Introducing Krylov subspace

By definition, the singular vectors $\boldsymbol{u}^{(i)}$ are in the column space of H_1. Instead of using the full column space, we attempt to use a k-dimensional subspace $\mathscr{V}_k$, and thus reduce the original eigen problem to a $k \times k$ matrix problem (assuming $r < k < w$). Notice that what we want is not singular vectors themselves but the inner product w.r.t. a given vector $\boldsymbol{\mu}$.

Imagine that we have k orthonormal bases representing such a subspace: $\{\boldsymbol{q}_1, ..., \boldsymbol{q}_k\}$, and we approximate each of the singular vector $\boldsymbol{u}^{(i)}$ as

$$\boldsymbol{u}^{(i)} \simeq \sum_{\alpha=1}^{k} b_{\alpha}^{(i)} \boldsymbol{q}_{\alpha}, \tag{6}$$

where $b_{\alpha}^{(i)}$ is a coefficient that is assumed to be unknown at this point. Since our goal is to compute the overlap with $\boldsymbol{\mu}$ and there is arbitrariness in the choice of the bases in the subspace, we assume

$$\boldsymbol{q}_1 = \boldsymbol{\mu},$$

which is always possible. Because of the orthogonality of the $\boldsymbol{q}_\alpha$s and the fact that $\boldsymbol{q}_1 = \boldsymbol{\mu}$, computing $\boldsymbol{\mu}^\top \boldsymbol{u}^{(i)}$ can be reduced to taking the first element of the $\boldsymbol{b}^{(i)}$s. Explicitly, the kernel function is approximated by

$$K(i, \boldsymbol{\mu}) \simeq \sum_{\alpha=1}^{k} b_{\alpha}^{(i)} \boldsymbol{\mu}^\top \boldsymbol{q}_{\alpha} = b_1^{(i)}, \tag{7}$$

which means that *the inner product can be computed directly from the k-dimensional vectors $\boldsymbol{b}^{(i)}$s without explicitly using the $\boldsymbol{u}^{(i)}$s.*

Now our remaining problem is how to find $\mathscr{V}_k$ and the coefficient vectors $\boldsymbol{b}^{(i)}$. To find $\mathscr{V}_k$, let us consider this problem:
Given an s-dimensional subspace $\mathscr{V}_s \subset \mathbb{R}^w$, construct a subspace $\mathscr{V}_{s+1}$ by adding a vector to $\mathscr{V}_s$ so that the increase of the overlap between $\mathscr{V}_{s+1}$ and $\{\boldsymbol{u}^{(1)}, ..., \boldsymbol{u}^{(s)}\}$ is maximized.

Let us start with $\mathscr{V}_1$ spanned by $\boldsymbol{\mu}$. Recall that finding the left singular vector for H_1 is equivalent to the eigen problem of $\mathsf{C} \equiv \mathsf{H}_1 \mathsf{H}_1{}^\top$, and the eigen equation for C is equivalent to the maximization problem of the Rayleigh quotient [9], which is defined by

$$R(\boldsymbol{u}) = \frac{\boldsymbol{u}^\top \mathsf{C} \boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{u}}.$$

To satisfy the requirement, when we construct $\mathscr{V}_2 = \mathrm{span}\{\boldsymbol{\mu}, \boldsymbol{\Delta}\}$ by adding $\boldsymbol{\Delta} \in \mathbb{R}^w$, the added vector should contain the steepest ascent direction of R given by

$$\left.\frac{d}{d\boldsymbol{u}} R(\boldsymbol{u})\right|_{\boldsymbol{u}=\boldsymbol{\mu}} = \frac{-2}{\boldsymbol{\mu}^\top \boldsymbol{\mu}} \left[R(\boldsymbol{\mu}) \boldsymbol{\mu} - \mathsf{C} \boldsymbol{\mu} \right].$$

Thus, if we choose $\mathsf{C}\boldsymbol{\mu}$ as $\boldsymbol{\Delta}$, $\mathrm{span}\{\boldsymbol{\mu}, \mathsf{C}\boldsymbol{\mu}\}$ contains this steepest direction.

Continuing this procedure, we see that a k-dimensional space

$$\mathscr{V}_k(\boldsymbol{\mu}, \mathsf{C}) \equiv \mathrm{span}\{\boldsymbol{\mu}, \mathsf{C}\boldsymbol{\mu}, ..., \mathsf{C}^{k-1}\boldsymbol{\mu}\}$$

is the best k-dimensional subspace in terms of maximization of R, given $\boldsymbol{\mu}$. In other words, there are many choices of a k-dimensional subspace over the entire column space of H_1, but among all of the choices, the subspace that has the largest weight of $\boldsymbol{u}^{(1)}, ..., \boldsymbol{u}^{(r)}$ is $\mathscr{V}_k(\boldsymbol{\mu}, \mathsf{C})$, under the constraint that $\boldsymbol{\mu}$ is the starting base. In mathematics, $\mathscr{V}_k(\boldsymbol{\mu}, \mathsf{C})$ is called the *Krylov subspace* induced by $\boldsymbol{\mu}$ and C [9]. Alternatively, one may say that $\boldsymbol{\mu}$ is the *seed* of the Krylov subspace.

3.3 Fast computation of z

Let us consider our next question: how to find the coefficient vectors $\boldsymbol{b}^{(i)}$. Before directly considering $\boldsymbol{b}^{(i)}$, let us consider how to find the orthonormal set $\{\boldsymbol{q}_1, ..., \boldsymbol{q}_k\}$. This is an easy task, since, given $\mathscr{V}_k(\boldsymbol{\mu}, \mathsf{C}) \equiv \mathrm{span}\{\boldsymbol{\mu}, \mathsf{C}\boldsymbol{\mu}, ..., \mathsf{C}^{k-1}\boldsymbol{\mu}\}$, we can use Gram-Schmidt orthogonalization starting from $\boldsymbol{\mu}$ to produce the orthonormal set. Note that the Gram-Schmidt orthogonalization is essentially equivalent to the QR factorization of

$$\mathsf{V}_k(\boldsymbol{\mu}, \mathsf{C}) \equiv \left[\boldsymbol{\mu}, \mathsf{C}\boldsymbol{\mu}, ..., \mathsf{C}^{k-1}\boldsymbol{\mu}\right],$$

which is called the Krylov matrix. Fortunately, in the QR factorization of the Krylov matrix, a special and helpful property holds (for proof, see [9]):

Theorem 1 *The orthogonal matrix* $\mathsf{Q}_k \equiv [\boldsymbol{q}_1, ..., \boldsymbol{q}_k] \in \mathbb{R}^{w \times k}$ *given by the QR factorization of* $\mathsf{V}_k(\boldsymbol{\mu}, \mathsf{C})$ *tridiagonalizes* C.

This theorem says that $\mathsf{Q}_k^\top \mathsf{C} \mathsf{Q}_k$ is a tridiagonal matrix. What is this matrix? To see it, note that Eq. (6) can be written as

$$\boldsymbol{u}^{(i)} \simeq \sum_{\alpha=1}^{k} b_\alpha^{(i)} \boldsymbol{q}_\alpha = \mathsf{Q}_k \boldsymbol{b}^{(i)}.$$

Then the eigen equation for C, which is equivalent to SVD of H_1, is rewritten as

$$\mathsf{Q}_k^\top \mathsf{C} \mathsf{Q}_k \boldsymbol{b} = \lambda \boldsymbol{b}. \tag{8}$$

This means that we can directly find the coefficient vectors $\{\boldsymbol{b}^{(1)}, ..., \boldsymbol{b}^{(r)}\}$ by diagonalizing a tridiagonal matrix $\mathsf{T}_k \equiv \mathsf{Q}_k^\top \mathsf{C} \mathsf{Q}_k$.

Let $\alpha_1, ..., \alpha_k$ and $\beta_1, ..., \beta_{k-1}$ be the diagonal and subdiagonal elements of T_k. If we consider the s-th column of the equation $\mathsf{C}_k \mathsf{Q}_k = \mathsf{Q}_k \mathsf{T}_k$, it follows that

$$\mathsf{C}\boldsymbol{q}_s = \alpha_s \boldsymbol{q}_s + \beta_{s-1} \boldsymbol{q}_{s-1} + \beta_s \boldsymbol{q}_{s+1},$$

where $\boldsymbol{q}_s$ is the s-th column vector of Q_k. Using the orthogonal relation $\boldsymbol{q}_i^\top \boldsymbol{q}_j = \delta_{i,j}$, we immediately have $\alpha_s = \boldsymbol{q}_s{}^\top \mathsf{C} \boldsymbol{q}_s$. In this way, it is easy to construct an algorithm to find α_s and β_s sequentially from this recurrent equation:

Subroutine 1 `Lanczos(C,μ,k)` *Input* $\mathsf{C} \in \mathbb{R}^{w \times w}$, $\boldsymbol{\mu} \in \mathbb{R}^w$, *and a positive integer* k *(*$< w$*). Initialize as* $\boldsymbol{r}_0 = \boldsymbol{\mu}$, $\beta_0 = 1$, $\boldsymbol{q}_0 = 0$, *and* $s = 0$. *Repeat*

```
    $\boldsymbol{q}_{s+1} = \boldsymbol{r}_s / \beta_s$
    $s \leftarrow s + 1$
    $\alpha_s = \boldsymbol{q}_s{}^\top \mathsf{C} \boldsymbol{q}_s$
    $\boldsymbol{r}_s = \mathsf{C}\boldsymbol{q}_s - \alpha_s \boldsymbol{q}_s - \beta_{s-1} \boldsymbol{q}_{s-1}$
    $\beta_s = ||\boldsymbol{r}_s||$
until $s = k$. Return $\{\alpha_1, .., \alpha_k\}$ and $\{\beta_1, .., \beta_{k-1}\}$.
```

By running this procedure up to $k < w$, we obtain T_k $(= \mathsf{Q}_k^\top \mathsf{C} \mathsf{Q}_k)$ directly. Notice that we do *not* need to explicitly compute $\boldsymbol{q}_1, ..., \boldsymbol{q}_k$. This tridiagonalization procedure is called the Lanczos algorithm.

Finally, the CP score is computed using Eqs. (5) and (7) as

$$z \simeq 1 - \sum_{i=1}^{r} x^{(i)^2}. \tag{9}$$

Notice that we do not at all need to explicitly compute either the $\boldsymbol{u}^{(i)}$s or the inner product. We call this implicit kernel calculation based on Krylov subspace learning the *implicit Krylov approximation* (IKA).

Our fast SST algorithm is summarized as:

Algorithm 1 (IKA-SST) *At each* t, *do*

1. *Compute* $\boldsymbol{\mu}$ *as the SVD of* H_2.
2. $\alpha_1, .., \alpha_r, \beta_1, .., \beta_{k-1} \leftarrow$ `Lanczos(C,μ,k)`.
3. *Compute the* r *top eigenvectors of the tridiagonal matrix* T_k.
4. *Compute the CP score using Eq. (9).*

For the dimension of the Krylov subspace $\mathscr{K}_k(\boldsymbol{\mu}, \mathsf{C})$, one reasonable choice is

$$k = \begin{cases} 2r & r \in \text{even} \\ 2r - 1 & r \in \text{odd} \end{cases}. \tag{10}$$

The rationale of this rule is that the Krylov subspace is also the best subspace for the smallest eigenstates as well as for the largest eigenstates [9], so k should be about twice r. Note that the IKA is independent of the choice of the SST-native parameters n and γ.

3.4 Relationship to subspace identification method

The SVD approach for the SST is similar to the subspace identification method in control theory [25]. This is indeed the case, and the equivalence between them was theoretically explored by Kawahara [20]. This work showed that SST can be thought of as an approximated version of the subspace identification algorithm to determine the system parameters of a state-space model. This suggests that SST has a unique feature that unifies the previous sequential AR model approach [31] into a single, computationally efficient framework. Due to space limitations, however, we will discuss this in a separate paper.

4 Proximity-based outlier detection

This section proposes a new outlier detection method based on sparse structure learning. We first consider how to learn a sparse structure from the data. We assume that $\mathcal{D}$ has been standardized to have zero mean and unit variance. Then the sample covariance matrix S is given by

$$\mathsf{S}_{i,j} \equiv \frac{1}{N}\sum_{n=1}^{N} x_i^{(n)} x_j^{(n)}, \tag{11}$$

which is the same as the correlation coefficient matrix for this data.

The use of sparse structure learning in anomaly detection was first proposed in [15]. While the work [15] addresses an anomaly scoring problem in a setting similar to two-sample test in statistics, we extend their framework to include outlier detection by considering a conditional probability function.

4.1 *Penalized maximum likelihood*

In the GGM, structure learning is reduced to finding a precision matrix $\mathsf{\Lambda}$ for the multivariate Gaussian (Eq. (1)). If we do not consider any regularization for now, we can get $\mathsf{\Lambda}$ by maximizing the log-likelihood

$$\ln \prod_{t=1}^{N} \mathcal{N}(\boldsymbol{x}^{(t)}|\mathbf{0}, \mathsf{\Lambda}^{-1}) = \text{const.} + \frac{N}{2}\left\{\ln\det(\mathsf{\Lambda}) - \mathrm{tr}(\mathsf{S}\mathsf{\Lambda})\right\},$$

where tr represents the matrix trace (sum over the diagonal elements), and we used the well-known identity $\boldsymbol{x}^{(t)\top}\boldsymbol{x}^{(t)} = \mathrm{tr}(\boldsymbol{x}^{(t)}\boldsymbol{x}^{(t)\top})$ and Eq. (11). If we use the well-known formulas for matrix derivatives

$$\frac{\partial}{\partial \mathsf{\Lambda}}\ln\det(\mathsf{\Lambda}) = \mathsf{\Lambda}^{-1}, \quad \frac{\partial}{\partial \mathsf{\Lambda}}\mathrm{tr}(\mathsf{S}\mathsf{\Lambda}) = \mathsf{S}, \tag{12}$$

then we readily obtain the formal solution $\mathsf{\Lambda} = \mathsf{S}^{-1}$. However, as mentioned before, this produces less practical information on the structure of the system, since the sample covariance matrix is often rank deficient and the resulting precision matrix will not be sparse in general.

Therefore, instead of the standard maximum likelihood estimation, we solve an L_1-regularized version of the maximum likelihood:

$$\Lambda^* = \arg\max_{\Lambda} f(\Lambda; \mathsf{S}, \rho), \tag{13}$$

$$f(\Lambda; \mathsf{S}, \rho) \equiv \ln\det\Lambda - \mathrm{tr}(\mathsf{S}\Lambda) - \rho||\Lambda||_1, \tag{14}$$

where $||\Lambda||_1$ is defined as $\sum_{i,j=1}^{M} |\Lambda_{i,j}|$. Thanks to the penalty term, many of the entries in Λ will be exactly zero. The penalty weight ρ is an input parameter, which works as a threshold below which correlation coefficients are thought of as zero, as discussed later.

4.2 Graphical lasso algorithm

Since Eq. (13) is a convex optimization problem [2], one can use subgradient methods to solve it. Recently, Friedman, Hastie, and Tibshirani [7] proposed an efficient subgradient algorithm named graphical lasso. We recapitulate it in this subsection.

The graphical lasso algorithm first reduces the problem Eq. (13) to a series of related L_1-regularized regression problems by utilizing a block coordinate descent technique [2, 6]. Using the formula Eq. (12), we see that the gradient of Eq. (13) is given by

$$\frac{\partial f}{\partial \Lambda} = \Lambda^{-1} - \mathsf{S} - \rho \,\mathrm{sign}(\Lambda), \tag{15}$$

where the sign function is defined so that the (i, j) element of the matrix $\mathrm{sign}(\Lambda)$ is given by $\mathrm{sign}(\Lambda_{i,j})$ for $\Lambda_{i,j} \neq 0$, and a value $\in [-1, 1]$ for $\Lambda_{i,j} = 0$.

To use a block coordinate descent algorithm for solving $\partial f/\partial\Lambda = 0$, we focus on a particular single variable x_i, and partition Λ and its inverse as

$$\Lambda = \begin{pmatrix} L & \boldsymbol{l} \\ \boldsymbol{l}^\top & \lambda \end{pmatrix}, \quad \Sigma \equiv \Lambda^{-1} = \begin{pmatrix} W & \boldsymbol{w} \\ \boldsymbol{w}^\top & \sigma \end{pmatrix}, \tag{16}$$

where we assume that rows and columns are always arranged so that the x_i-related entries are located in the last row and column. In these expressions, $W, L \in \mathbb{R}^{(M-1)\times(M-1)}$, $\lambda, \sigma \in \mathbb{R}$, and $\boldsymbol{w}, \boldsymbol{l} \in \mathbb{R}^{M-1}$. Corresponding to this x_i-based partition, we also partition the sample covariance matrix S in the same way, and write it as

$$\mathsf{S} = \begin{pmatrix} S^{\backslash i} & \boldsymbol{s} \\ \boldsymbol{s}^\top & s_{i,i} \end{pmatrix}. \tag{17}$$

Now let us find the solution of the equation $\partial f/\partial\Lambda = 0$. Since Λ must be positive definite, the diagonal elements must be strictly positive. Thus, for the diagonal elements, the condition of the vanishing gradient leads to

$$\sigma = s_{i,i} + \rho. \tag{18}$$

For the off-diagonal entries represented by $\boldsymbol{w}$ and $\boldsymbol{l}$, the optimal solution under which all the other variables are hold constant is obtained by solving

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} ||W^{\frac{1}{2}} \boldsymbol{\beta} - \boldsymbol{b}||^2 + \rho \; ||\boldsymbol{\beta}||_1 \right\} = 0, \tag{19}$$

where $\boldsymbol{\beta} \equiv W^{-1}\boldsymbol{w}$, $\boldsymbol{b} \equiv W^{-1/2}\boldsymbol{s}$, and $||\boldsymbol{\beta}||_1 \equiv \sum_l |\beta_l|$. For the proof, see [15]. This is an L_1-regularized quadratic programming problem, and again can be solved efficiently with a coordinate-wise subgradient method [7].

Now to obtain the final solution Λ^*, we repeatedly solve Eq. (19) for $x_1, x_2, ..., x_M, x_1, ...$ until convergence. Note that the matrix W is full rank due to Eq. (18). This suggests the algorithm is numerically stable. In fact, as shown later, it gives a stable and reasonable solution even when some of the variables are highly correlated.

4.3 Connection to Lasso

The coordinate-wise optimization problem (Eq. (19)) derived by the graphical lasso algorithm has a clear similarity to the lasso-based structure learning algorithm. The algorithm of Ref. [26] solves separate lasso regression problems for each x_i:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} ||\mathsf{Z}_i \boldsymbol{\beta} - \boldsymbol{y}_i||^2 + \mu ||\boldsymbol{\beta}||_1 \right\}, \tag{20}$$

where we defined $\boldsymbol{y}_i \equiv (x_i^{(1)}, ..., x_i^{(N)})^\top$ and a data matrix $\mathsf{Z}_i \equiv [\boldsymbol{z}_i^{(1)}, ..., \boldsymbol{z}_i^{(N)}]^\top$ with

$$\boldsymbol{z}_i^{(n)} \equiv (x_1^{(n)}, .., x_{i-1}^{(n)}, x_{i+1}^{(n)}, ..., x_M^{(n)})^\top \in \mathbb{R}^{M-1}.$$

Using the definition of S (Eq. (11)), it is easy to see that this problem is equivalent to Eq. (19), when

$$W = S^{\backslash i} \quad \text{and} \quad \rho \propto \mu \tag{21}$$

are satisfied. Since W is a principal submatrix of Λ^{-1}, we see that there is a correspondence between W and $S^{\backslash i}$ when ρ is small. It will never be satisfied for $\rho > 0$, however. In this sense, the graphical lasso algorithm solves an optimization problem similar to but different from the one in [26]. This fact motivates us to empirically study the difference between the two algorithms as shown in the next section.

4.4 Choosing ρ

So far we have treated the penalty parameter ρ as a given constant. In many regularization-based machine learning methods, how to choose the penalty parameter is a subtle issue. In the present context, however, ρ should be treated as an input parameter since our goal is not to find the "true" structure but to reasonably select the neighborhood.

To gain insight on how to relate ρ with the neighborhood size, we note this result:

Proposition 1 *If we consider a* 2×2 *problem defined only by two variables* x_i *and* x_j *(*$i \neq j$*), the off-diagonal element of the optimal* Λ *as the solution to Eq.* (13) *is given by*

$$\Lambda_{i,j} = \begin{cases} -\frac{\mathrm{sign}(r)(|r|-\rho)}{(1+\rho)^2-(|r|-\rho)^2} & \text{for } |r| > \rho \\ 0 & \text{for } |r| \leq \rho, \end{cases}$$

where r *is the correlation coefficient between the two variables.*

For the proof, see [15].

Although this is not the solution to the full system, it gives us a useful guide about how to choose ρ. For example, if a user wishes to think of any dependencies corresponding to absolute correlation coefficients less than 0.5 as noise, then the input ρ should be less than the intended threshold, and possibly a value around $\rho = 0.3$ would work. If ρ is close to 1, the resulting neighborhood graphs will be very small, while a value close to 0 leads to an almost complete graph where all of the variables are thought of as being connected.

We should also note that sparse structure learning allows us to conduct neighborhood selection in an adaptive manner. If a variable is isolated with almost no dependencies on other variables, then the number of selected neighbors will be zero. Also, we naturally expect that variables in a tightly-connected cluster would select the cluster members as their neighbors. We will see, however, that the situations when there are highly correlated variables are much trickier than they seem.

4.5 Outlier score

Now that a complete probabilistic model has been defined, let us proceed to the next step. Here we define the anomaly score for the i-th variable as

$$z_i(\boldsymbol{x}|\Lambda) \equiv -\ln p(x_i|x_1,..,x_{i-1},x_{i+1},...,x_M,\Lambda). \tag{22}$$

Note that we have M scores, corresponding to individual variables, for a single observation $\boldsymbol{x}$. The definition tells us the discrepancy between the value of the i-th variable and its expected value given surrounding variables. Thanks to the sparseness, the surrounding variables should be in the same module or cluster of the i-th variable.

Since the right hand side of Eq. (22) is Gaussian, we can analytically write down the expression. For example, for the first variable, the conditional distribution is

$$p(x_1|x_2,\cdots,x_M) = \mathcal{N}\left(x_1 \middle| -\frac{1}{\Lambda_{1,1}}\sum_{i=2}^{M}\Lambda_{1,i}x_i\,,\ \frac{1}{\Lambda_{1,1}}\right),$$

and the score is given as

$$s_1 \equiv \frac{1}{2}\ln\frac{2\pi}{\Lambda_{1,1}} + \frac{1}{2\Lambda_{1,1}}\left(\sum_{i=1}^{M}\Lambda_{1,i}x_i\right)^2. \tag{23}$$

Putting together the M scores into a single vectorial expression, we get the final result of the outlier scores as

$$\boldsymbol{s} \equiv \boldsymbol{s}_0 + \frac{1}{2}\mathrm{diag}(\Lambda\boldsymbol{x}\mathrm{D}^{-1}\boldsymbol{x}^\top\Lambda),$$

where $D \equiv \mathrm{diag}^2(\Lambda)$ and

$$(\boldsymbol{s}_0)_i \equiv \frac{1}{2}\ln\frac{2\pi}{\Lambda_{i,i}}.$$

5 Experiment

This section presents experimental results for the two anomaly detection methods introduced in the previous sections: IKA-SST for change-point detection, and the proximity-based outlier detection.

5.1 Parameter dependence of SST

An example of SST was already shown in Fig. 1. The time series (a) was generated using three linear functions with slopes of 1/300, 0, and $-1/200$. The other time-series (c) was generated using a sine function $x(t) = \sin(2\pi t/\lambda)$, for $\lambda = \sqrt{80}$, $\sqrt{120}$, and $\sqrt{70}$. In (c), we also added random fluctuations to the amplitude and the periods of up to ±7.5% and ±0.5%, respectively, to simulate fluctuations in realistic observations. For both data sets, the change points are located at $t = 150$ and 300. The results of SST in Figs. 1 (b) and (d) were calculated with $w = 20$ and $r = 3$. No IKA approximation was used. In spite of the apparent differences in the original data, we see that SST strikingly reveals the similarities without any ad hoc tuning for individual time series. Existing methods such as differentiation [10] and wavelet-based approaches [19] fail to detect the change points if a common parameter set is used for both sets.

The dependence on w is of particular interest in SST. We calculated SST as a function of w for $r = 3$. The results are shown in Fig. 3. It is surprising that the essential features remain unchanged over a very wide range of w, $6 \lesssim w \lesssim 40$, while the widths of the major features become broader as w increases. This robustness is quite advantageous for heterogeneous systems.

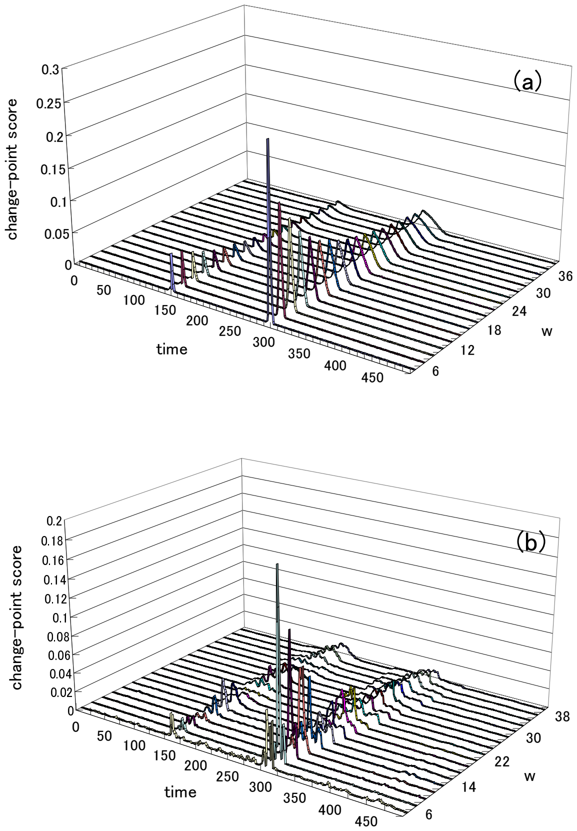

Fig. 3 The dependence of SST on w for (a) the linear function and for (b) the oscillatory function shown in Fig. 1 (a) and (c), respectively.

Table 1 Tested methods.

#	symbol	$\boldsymbol{\mu}$	feedback	$\{\boldsymbol{u}^{(i)}\}$	kernel
1	OI	`power`	no	`OI`	explicit
2	EM	`EMPCA`	no	`EMPCA`	explicit
3	OI_FB	`power`	yes	`OI`	explicit
4	EM_FB	`EMPCA`	yes	`EMPCA`	explicit
5	IKA	`power`	yes	-	implicit

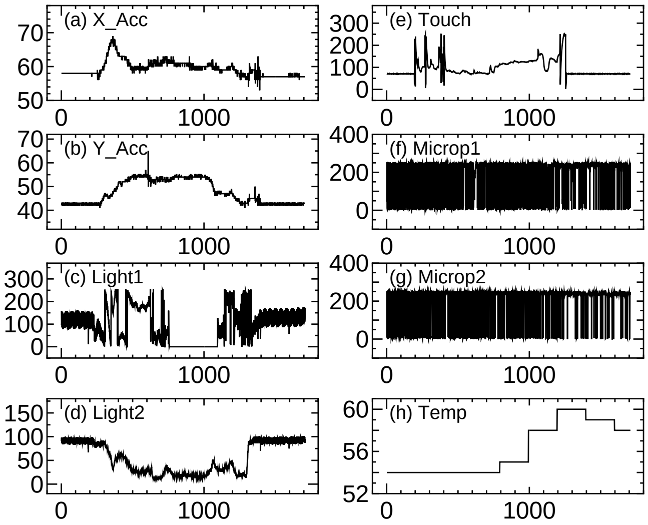

Fig. 4 The phone1 data.

5.2 Accuracy of IKA-SST

We implemented five different types of SST algorithms in Java as shown in Table 1. The first four explicitly compute the singular vectors using different routines: `power` (the power method), `OI` (orthogonal iteration [9]), and the `EMPCA` (EMPCA algorithm [27]). These were compared to our IKA-based SST algorithm. All of the calculations were done in a Java 1.4.2 virtual machine on an older workstation (Pentium 4, 2.0 GHz, 1 GB of memory). In the iterative algorithms, the convergence threshold was set to be 10^{-5} for the norm of the residual vectors.

The data used was the *phone1* data (Fig. 4) containing eight time series of various types measured by embedded sensors in a mobile phone [23]. Each of the variables consists of 1,708 data points, but information about the sampling rate is not given. From the title attached to the data file, it seems that the data represents the actions of picking up the phone and putting it down.

We measured the computational times of these five SST algorithms. As a preprocess, the original signals were scaled to have unit variance and a mean of three.

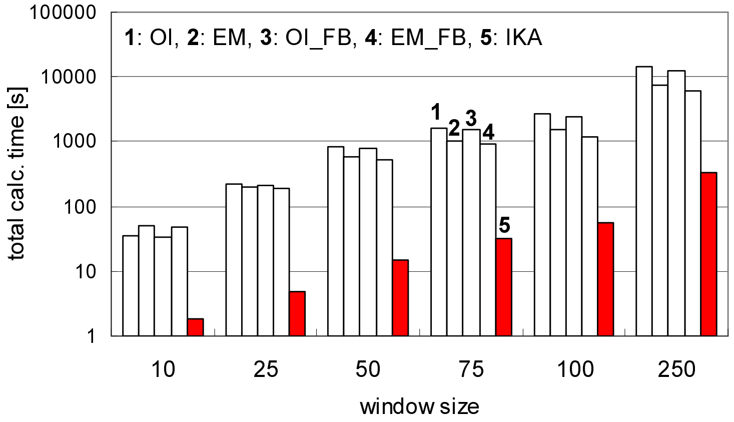

Fig. 5 Total computation time of SST.

We imposed a periodic boundary condition on the data in performing SST. This is was keep the number of data points the same over different values of w. We used $(r,k) = (3,5)$.

Figure 5 compares the computational times of the different algorithms on a logarithmic scale, averaged over five trials. We see that the improvement with the IKA-SST is drastic. It is about 50 times faster than the conventional SST methods for each w.

Notice that this was accomplished with no significant approximation error. To show this, Fig. 6 compares the CP scores between EM and IKA for $w = 50$. As shown, the overall fit between the EM and IKA results is very good, although there are a few peaks which are not reproduced by IKA as indicated in Figs. 6 (b) and (g). Again, it is surprising that the IKA almost perfectly reproduces the results of EM, since IKA solves only 5×5 problems while EM performs the complete SVD for 50×50 matrices.

5.3 Outlier detection: hot box detection

We used the proximity-based anomaly detection method with a real problem in the rail road industry. The task is often called hot box detection, where the goal is to detect anomalously behaving wheel axles based on temperature recordings. Under normal operations, the temperature of an axle is expected to be highly correlated with the temperature of the other axles. Thus the proximity-based outlier detection is useful in this application.

In contrast with obvious faults that are easily detected by a temperature threshold, detecting subtle signs of correlation anomalies is generally challenging. This

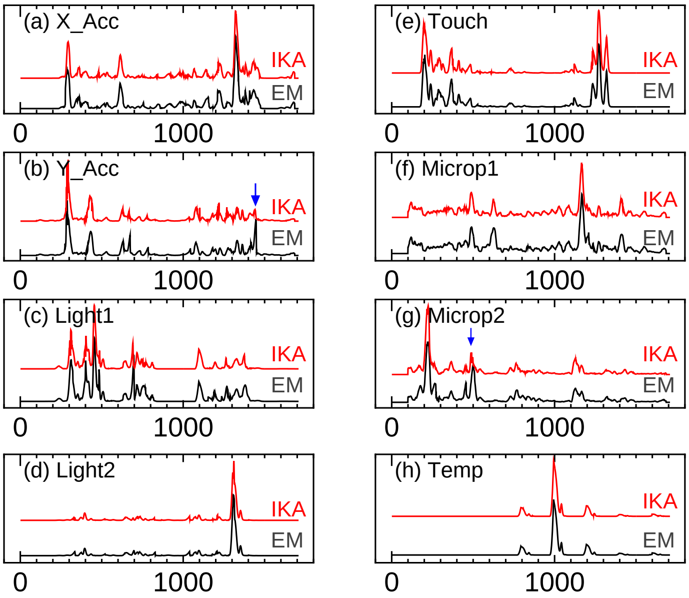

Fig. 6 CP score of the phone1 data ($w = 50, r = 3$).

is mostly because the temperature measurements are quite sensitive to external whether conditions. For example, the temperatures on rainy days are more than 10 degrees lower than those on sunny days. Also, the temperatures of the first and the last cars exhibit considerably different behaviors from the other cars.

We tested our outlier detection method, and compared the performance with a state-of-the-art method created by domain experts using extensive domain knowledge. The results were quite encouraging. Our method was several times better in a detection power, which is defined for a truly faulty axle i as

$$\frac{1}{\sigma_i}[s_i(\boldsymbol{x}) - \langle s_i \rangle].$$

Here $\langle s_i \rangle$ and σ_i are the mean and the standard deviation of the i-th outlier score over all of the samples, while $s_i(\boldsymbol{x})$ is the outlier score of the faulty sample.

6 Summary

We have discussed approaches to anomaly detection for sensor data. We first reviewed existing methods and their limitations. We then described two new approaches to anomaly detection that are capable of handling heterogeneous variables.

First we gave a fast change-point detection method called IKA-SST. Thanks to the robustness of SVD, this approach has a remarkable feature that no parameter tuning is needed to handle heterogeneities of the variables. Second, we presented a proximity-based outlier detection method, which has a very useful feature for automatic discovery of the modular structure of a system. Finally, we showed some experimental results for these methods.

References

1. T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley-Interscience, 3rd. edition, 2003.
2. O. Banerjee, L. E. Ghaoui, and G. Natsoulis. Convex optimization techniques for fitting sparse Gaussian graphical models. In *Proc. Intl. Conf. Machine Learning*, pp. 89–96. Press, 2006.
3. M. Basseville and I. Nikiforov. *Detection of Abrupt Changes*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.
4. M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: identifying density-based local outliers. *ACM SIGMOD Record*, 29(2):93–104, 2000.
5. V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Survey*, 41(3):1–58, 2009.
6. J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302–332, 2007.
7. J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.
8. R. Fujimaki, S. Hirose, and T. Nakata. Theorectical analysis of subsequence time-series clustering from a frequency-analysis viewpoint. In *Proc. SIAM Intl. Conf. Data Mining*, pp. 506–517, 2008.
9. G. H. Golub and C. F. V. Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, 1996.
10. S. Hirano and S. Tsumoto. Mining similar temporal patterns in long time-series data and its application to medicine. In *Proc. 2002 IEEE International Conference on Data Mining*, pp. 219–226, 2002.
11. B. Hu, T. Rakthanmanon, Y. Hao, S. Evans, S. Lonardi, and E. Keogh. Discovering the intrinsic cardinality and dimensionality of time series using mdl. In *Proc. of the 11th IEEE Intl. Conf. on Data Mining (ICDM 11)*, 2011.
12. T. Idé. Why does subsequence time-series clustering produce sine waves? In *Proc. 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, (PKDD 06)*, pp. 211–222, 2006.
13. T. Idé and K. Inoue. Knowledge discovery from heterogeneous dynamic systems using change-point correlations. In *Proc. 2005 SIAM Intl. Conf. Data Mining (SDM 05)*, pp. 571–575, 2005.
14. T. Idé and H. Kashima. Eigenspace-based anomaly detection in computer systems. In *Proc. ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, pp. 440–449, 2004.
15. T. Idé, A. C. Lozano, N. Abe, and Y. Liu. Proximity-based anomaly detection using sparse structure learning. In *Proc. of 2009 SIAM International Conference on Data Mining (SDM 09)*, pp. 97–108.
16. T. Idé, S. Papadimitriou, and M. Vlachos. Computing correlation anomaly scores using stochastic nearest neighbors. In *Proc. of IEEE Intl. Conf. Data Mining (ICDM 07)*, pp. 523–528, 2007.
17. T. Idé and K. Tsuda. Change-point detection using krylov subspace learning. In *Proc. 2007 SIAM Intl. Conf. Data Mining (SDM 07)*, pp. 515–520, 2007.

18. R. Jiang, H. Fei, and J. Huan. Anomaly localization for network data streams with graph joint sparse PCA. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 886–894, 2011.
19. S. Kadambe and G. Boudreaux-Bartels. Application of the wavelet transform for pitch detection of speech signals. *IEEE Trans. Information Theory*, 38:917–924, 1992.
20. Y. Kawahara and M. Sugiyama. Change-point detection in time-series data by direct density-ratio estimation. In *Proc. of 2009 SIAM Intl. Conf. on Data Mining (SDM 09)*, 2009.
21. Y. Kawahara, T. Yairi, and K. Machida. Change-point detection in time-series data based on subspace identification. In *Proc. of the 7th IEEE Intl. Conf. on Data Mining (ICDM 07)*, 2007.
22. E. J. Keogh, J. Lin, and A. W.-C. Fu. HOT SAX: Efficiently finding the most unusual time series subsequence. In *Proc. of the 5th IEEE Intl. Conf. Data Mining (ICDM 05)*, pp. 226–233, 2005.
23. E. Keogh and T. Folias. The UCR time series data mining archive [`http://www.cs.ucr.edu/~eamonn/TSDMA/index.html`]. 2002.
24. E. Keogh, J. Lin, and W. Truppel. Clustering of time series subsequences is meaningless: Implications for previous and future research. In *Proc. IEEE Intl. Conf. on Data Mining*, pp. 115–122. IEEE, 2003.
25. L. Ljung. *System Identification – Theory For the User*. PTR Prentice Hall, 2nd. edition, 1999.
26. N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.
27. S. Roweis. EM algorithms for PCA and SPCA. In M. I. Jordan, M. J. Kearns, and S. A. Solla eds., *Advances in Neural Information Processing Systems*, Vol. 10. The MIT Press, 1998.
28. M. Sugiyama, T. Suzuki, and T. Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, 1st. edition, 2012.
29. Y. Urabe, K. Yamanishi, R. Tomioka, and H. Iwai. Real-time change-point detection using sequentially discounting normalized maximum likelihood coding. In *Proc. of the 15th Pacific-Asia Conference Conf. on Knowledge Discovery and Data Mining (PAKDD 11)*, 2011.
30. X. Xuan and K. Murphy. Modeling changing dependency structure in multivariate time series. In *Proc. the 24th Intl. Conf. Machine Learning*, pp. 1055–1062, 2007.
31. K. Yamanishi and J. Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. In *Proc. the Eighth ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining (KDD 02)*, pp. 676–681, 2002.
32. K. Yamanishi, J. Takeuchi, G. Williams, and P. Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In *Proc. the Sixth ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pp. 320–324, 2000.
33. D. Yankov, E. J. Keogh, and U. Rebbapragada. Disk aware discord discovery: Finding unusual time series in terabyte sized datasets. In *Proc. of the 7th IEEE Intl. Conf. on Data Mining (ICDM 07)*, 2007.