Change-Point Detection in a Sequence of Bags-of-Data

Kensuke Koshijima, Hideitsu Hino, *Member, IEEE,* and Noboru Murata

Abstract—In this paper, the limitation that is prominent in most existing works of change-point detection methods is addressed by proposing a nonparametric, computationally efficient method. The limitation is that most works assume that each data point observed at each time step is a *single* multi-dimensional vector. However, there are many situations where this does not hold. Therefore, a setting where each observation is a collection of random variables, which we call a *bag* of data, is considered. After estimating the underlying distribution behind each bag of data and embedding those distributions in a metric space, the change-point score is derived by evaluating how the sequence of distributions is fluctuating in the metric space using a distance-based information estimator. Also, a procedure that adaptively determines when to raise alerts is incorporated by calculating the confidence interval of the change-point score at each time step. This avoids raising false alarms in highly noisy situations and enables detecting changes of various magnitudes. A number of experimental studies and numerical examples are provided to demonstrate the generality and the effectiveness of our approach with both synthetic and real datasets.

Index Terms—Change-point detection, Entropy Estimator, Earth Mover's Distance, Anomaly detection.

1 Introduction

Significant events occurring in the real world often trigger changes in the data that one can acquire from sources related to the event, and conversely, changes in time-series data are often signs of important events happening at that time. Therefore, detecting changes in time-series data has long been a problem of great interest for researchers from various areas [1], [2]. This technique, which is often referred to as *change-point detection*, can be directly applied to various situations such as intrusion detection in computer networks [3], fault detection in machines [4], and fraud detection in credit card use [5]. It could also be employed to preprocess and segment time-series data. For time-series prediction, dramatic changes in the data would be detrimental to the performance of the prediction model, and therefore, the data should be segmented beforehand using a change-point detection technique. Segmenting time-series data can also be used for signal processing [6].

One approach to the problem of change-point detection is to fit a stochastic model to the sequence of data and determine when the data deviates from the built model [2], [7], [8]. For example, in [8], auto-regressive models are employed to construct a sequence of probability density functions which describe the underlying structure of the time-series data, and the deviation is evaluated by logarithmic loss. These methods rely on parametric models, and their applicability is frequently limited.

Therefore, in order to cope with situations where parametric assumptions are not appropriate, many nonparametric methods have recently been proposed. In these methods, the common approach is to focus on two subsets of data that arrive in intervals before and after time t, which we call as the *reference set* and the *test set* respectively, and to evaluate the dissimilarity between the two sequences. The effectiveness and the performance of change-point detection techniques depend on how the two sequences are modeled, and how the two models are compared. For example, in [9], two one-class support vector machines are trained independently on the reference set and the test set, and the two resulting hyperplanes are compared in the feature space to evaluate the dissimilarity between the two sets. There are also methods that focus on the subspace spanned by the trajectory matrix of the sequence [10], [11], or methods that focus on density ratio estimation [12].

Many of these methods have demonstrated great performance in different settings. However, there is a downside that these methods have in common which is that all of these methods assume that there is only *one* vector associated with each time step.

We could easily think of situations where these assumptions do not hold. One scenario would be where each observation at each time step is a collection of multi-dimensional vector, the size of which may vary over time, and the interest lies in the behavior of the group as a whole, and not each individual vector in the collection. An example of this situation would be to conduct a questionnaire survey periodically, and monitor for any changes in the overall characteristic of the group.

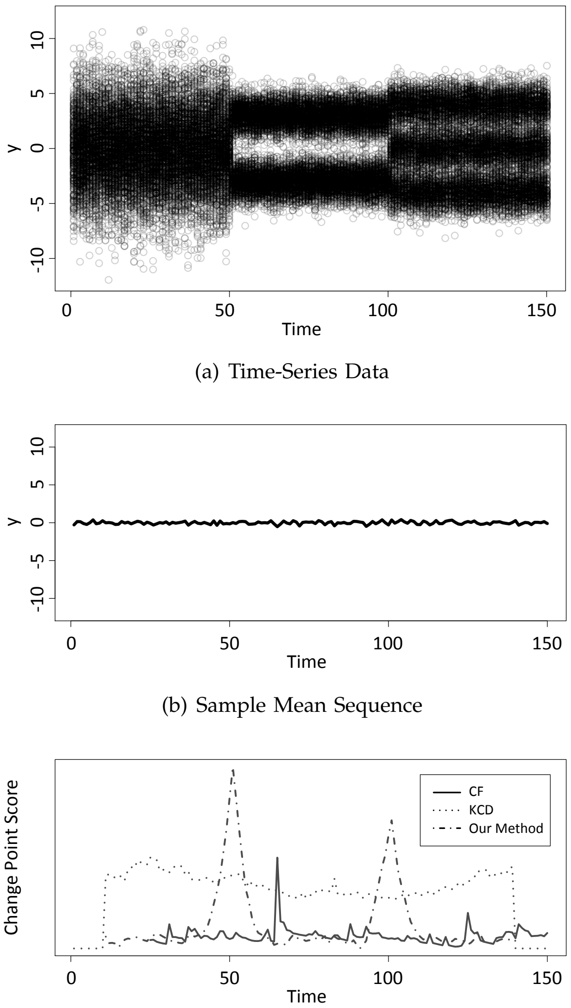
- *K. Koshijima and N. Murata are with the School of Science and Engineering, Waseda University, 3-4-1 Ohkubo, Shinjuku, Tokyo, Japan, and with the Institute of Systems, Information Technologies and Nanotechnologies (ISIT), Fukuoka SRP Center Building 7F 2-1-22, Momochihama, Sawara-ku, Fukuoka, Japan.*
- *H. Hino is with the Department of Computer Science, University of Tsukuba, 1-1-1 Tennodai, Ibaraki, Tsukuba, Japan.*

Another scenario is when the frequency of observation is not constant. Suppose that the goal is to detect an outbreak of a disease in a town, and that we are monitoring for any changes in the characteristics of patients that come to a hospital. The analysis is conducted every day with the data collected from patients that come to the hospital. Here again, we face the same problem that we have discussed above. Multiple patients come each day, and the number of patients that come each day varies.

In these situations, one could attempt to apply existing methods by computing descriptive statistics from the group at each time step, such as the sample mean, and apply existing methods on the resulting sequence of the descriptive statistics. However, the loss of information associated with the summarization is not desirable.

To illustrate this, we present a simple example in Fig. 1. Here, we have a sequence of data with change-points at $t = 50$ and $t = 100$. Data observed at each time step are generated from a single Gaussian distribution from $t = 1$ to $t = 50$, a mixture of two Gaussian distributions from $t = 51$ to $t = 100$, and a mixture of three Gaussian distributions from $t = 101$ to $t = 150$. There are about 300 instances at each step. Figure 1(b) shows a sequence obtained by taking the sample mean of data observed at each time step. Figure 1(c) shows the results of three change-point detection methods applied to this data. Our method, which we will discuss in detail in later sections, is applied directly to data in Fig.1(a) and it accurately detects the changes that occurred at $t = 50$ and $t = 100$. On the other hand, two existing methods proposed in [8], [9] were applied to data in Fig.1(b). The two methods exhibit scores that are totally unrelated to the change-points, which is fairly obvious because the sample mean sequence is losing too much information and it clearly does not capture the change-points occurring at $t = 50$ and $t = 100$. In later sections, we show other numerical examples that are common in the real world. In summary, when the analysis requires some sort of aggregation (over groups of individuals in the first scenario, and over time in the second scenario) resulting in datasets of different sizes, existing methods may not be appropriate.

In this paper, we propose a method to detect change points in a stream of aggregated data with varying sizes. In our setting, the observation at time t is not a single random variable, but a collection of random variables, which we call a *bag* of data. We estimate the underlying distribution that generates each bag, embed the distributions in a metric space, and then evaluate how those distributions are fluctuating in the metric space. Furthermore, we also incorporate a procedure that automatically determines the appropriate threshold for the change-point scores, which is a task that is often overlooked in past works. We also demonstrate that our method works for detecting changes in bipartite graphs that have different numbers of nodes, using simple statistics obtained from each node or edge.



(a) Time-Series Data

(b) Sample Mean Sequence

(c) Change Point Scores

Fig. 1. An example of a situation where the proposed method is advantageous over existing methods. (a) At each time step, about 300 one-dimensional vectors are observed. At $t = 50$ and $t = 100$, the underlying distribution changes from a single Gaussian distribution to a mixture of two Gaussian distributions, and then to a mixture of three Gaussian distributions. (b) The sample mean of each set of observations are taken so that existing methods can be applied. (c) Existing methods can not detect the changes that occurred at $t = 50$ and $t = 100$.

The remainder of this paper is organized as follows. In Section 2, we formulate our problem setting and discuss the differences of our setting compared to the setting considered in past works. In Section 3, we explain how the change-point scores are derived. In Section 4, we discuss how the confidence intervals of the change-point scores are estimated, and how they are used to adaptively determine where significant changes actually took place. Section 5 shows numerical examples using both simulated data and real data which include two of the most common examples of real world applications: the problem of detecting changes in collections of random variables and the problem of detecting changes in a sender-receiver bipartite network. Section 6 gives concluding remarks. A short and preliminary version of this paper appeared in [13].

2 Problem formulation

In the typical problem setting of change-point detection, we observes an $\boldsymbol{x}_t$ at each time step t where $\boldsymbol{x}_t$ is a multi-dimensional vector, i.e. $\boldsymbol{x}_t \in \mathbb{R}^d$, generated from some stochastic process. Typically, we consider two sets of data around time step t each having τ, τ' data points; the reference (past) data set $\mathcal{X}_{\text{ref}}^{(\tau)}$ and the test (future) data set $\mathcal{X}_{\text{test}}^{(\tau')}$, each of which is written as below:

$$\mathcal{X}_{\text{ref}}^{(\tau)} := \{\boldsymbol{x}_{t-\tau}, \boldsymbol{x}_{t-\tau+1}, \ldots, \boldsymbol{x}_{t-1}\}, \quad (1)$$

$$\mathcal{X}_{\text{test}}^{(\tau')} := \{\boldsymbol{x}_{t}, \boldsymbol{x}_{t+1}, \ldots, \boldsymbol{x}_{t+\tau'-1}\}. \quad (2)$$

For the sake of notational simplicity, we omit τ and τ' hereafter, unless needed. Letting $P_{\mathcal{X}_{\text{ref}}}$ and $P_{\mathcal{X}_{\text{test}}}$ represent the underlying probability distributions behind $\mathcal{X}_{\text{ref}}$ and $\mathcal{X}_{\text{test}}$ respectively, the objective here is to evaluate the difference between $P_{\mathcal{X}_{\text{ref}}}$ and $P_{\mathcal{X}_{\text{test}}}$ and raise an alarm if there is enough difference in the two underlying models.

As is shown in Fig. 2(a), this approach assumes that each data point is a single multi-dimensional vector. There are many situations where this condition does not hold. Often, data is obtained in groups, sometimes of different sizes, in which case we would have to deal with a collection of multi-dimensional vectors at each time step. Another situation would be when data points arrive randomly and analysis is done on a regular basis, such as daily or weekly. Here, the number of data points associated with each time step is the number of data points that arrive in the same time window.

In order to deal with the above issues, we formulate the problem in a different way. In our setting, observation at time t is not a single random variable, but a collection of random variables, which we call a *bag* of data (Fig. 2(b)),

$$B_t = \left\{\boldsymbol{x}_i^{(t)}\right\}_{i=1}^{n_t}, \quad (3)$$

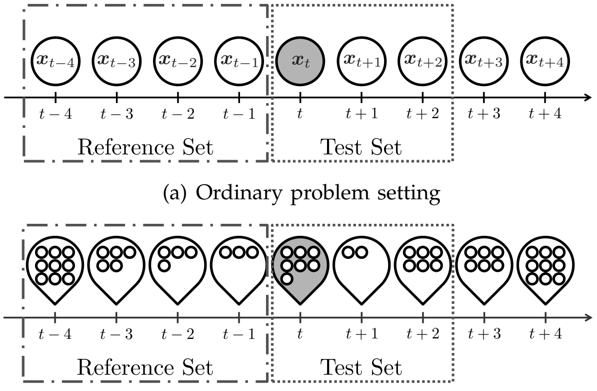
where $\boldsymbol{x}_i^{(t)} \in \mathbb{R}^d$, and n_t is the number of observations at time t, which can be different over time. The goal is to detect changes in the sequence of bags, and therefore, our interest lies in the behavior of the bags themselves and not the individual vectors in the bags.

3 Derivation of change-point scores

Let P_{B_t} represent the underlying distribution that generates the elements of B_t, i.e. $\boldsymbol{x}_i^{(t)} \overset{iid}{\sim} P_{B_t}, \forall i$. Similar to the ordinary approach that we discussed in Section 2, we consider a reference set and a test set as

$$\mathcal{B}_{\text{ref}} := \{B_{t-\tau}, B_{t-\tau+1}, \ldots, B_{t-1}\}, \quad (4)$$

$$\mathcal{B}_{\text{test}} := \{B_{t}, B_{t+1}, \ldots, B_{t+\tau'-1}\}. \quad (5)$$



(a) Ordinary problem setting

(b) Our problem setting

Fig. 2. The concept of our problem setting. In both of the figures, each circle represents a multi-dimensional vector. We want to know if there is a change-point at time step t. (a) One observes a single multi-dimensional vector at each time step and then attempts to evaluate the disparity between the underlying distribution behind the reference set and the test set. (b) We are considering a setting in which one has to deal with multiple vectors at each time step, where the number of vectors could differ over time. This is a situation that often arises in the real world.

Here, the objective is to evaluate the difference between the underlying distributions behind each of the bags in the reference set and each of the bags in the test set.

In a nut-shell our method works as follows. First, we embed distributions P_{B_t} in an appropriate metric space $\mathcal{M}$ with a distance measure between distributions (see Fig. 3). Then, we examine the fluctuation of the sequence of P_{B_t} in that metric space by using a distance-based information estimator. Intuitively, we consider evaluating the disparity between two *sets* of distributions, as opposed to the ordinary setting where only two distributions are compared. We will go into the details in the following sections.

3.1 Modeling distributions of each bag

There are two approaches in modeling P_{B_t}: parametric or non-parametric. If we could model P_{B_t} parametrically, we can reduce the problem to the ordinary change-point detection problem of the parameters of each P_{B_t}. Parametric approaches are known to perform better in situations where data come from a specific family of distributions, and are also known to work well with a small amount of data. However, applicability of parametric models are limited in real-world situations since real-world data often do not follow any standard parametric models or distributions. Therefore, we propose a non-parametric approach to this problem. It would allow us to estimate distributions in more general settings.

More specifically, we propose to represent the densities P_{B_t} for each time step t using *signatures*, which we

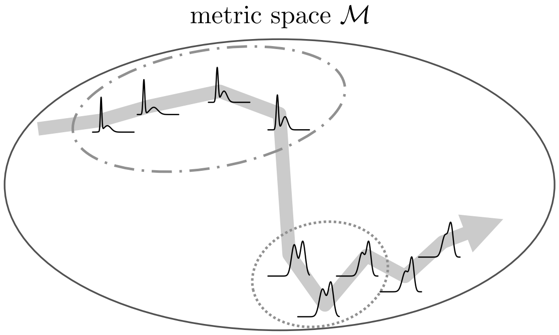


Fig. 3. The overall illustration of our approach. The underlying distributions for all of the bags in the reference set and the test set are estimated, and then are embedded in a metric space. Then, the fluctuation of these distributions are evaluated; the change-point scores are computed based on how much the reference set (distributions surrounded by the dashed lines) and the test set (distributions surrounded by the dotted lines) are different, in a statistical sense, from each other.

denote as S_t. Signature S_t takes the form

$$S_t = \left\{ \left(\boldsymbol{u}_k^{(t)}, w_k^{(t)} \right) \right\}_{k=1}^{K} \tag{6}$$

and it is a result of quantizing vectors in each bag using methods such as k-means, k-medoids, or learning vector quantization [14]. In other words, $\boldsymbol{u}_k^{(t)} \in \mathbb{R}^d$ are vectors that represent the cluster centers of the vectors in B_t, and $w_k^{(t)}$ are the number of observations that belong to the cluster of $\boldsymbol{u}_k^{(t)}$. This is a common approach especially taken in the area of computer vision [15], and it also follows the motivation described in [16], where they use histograms, which is a special case of signatures, as a means to represent the underlying distribution of a bag of data.

Another very simple way to make signatures is to make them as histograms. In other words, the signatures could be obtained simply by partitioning $\mathbb{R}^d$ into distinct bins of fixed width and then count the number of observations that fall in each bin. This would be a common approach especially when the vectors $\boldsymbol{x}$ are 1-dimensional vectors.

Similar to descriptive statistics like centroids, signatures are a form of summarization and, to some extent, they involve loss of information. However, signatures allow us to capture the shape of P_{B_t}, which is something that can not be accomplished by using descriptive statistics like centroids.

3.2 Embedding signatures in a metric space

An appropriate distance measure of signatures is necessary in order to embed the signatures in a metric space. Here, we employ the Earth Mover's distance (EMD) [17]. The EMD is a useful, efficiently calculated metric between two distributions which is known to behave naturally in terms of human perception. It is also known to be mathematically equivalent to the Wasserstein/Mallows distance [18].

Given two signatures written as S_t and $S_{t'}$, where $S_{t'} = \left\{ (\boldsymbol{v}_l^{(t')}, w_l^{(t')}) \right\}_{l=1}^{L}$, and the *ground distance* d_{kl}, which is an arbitrarily given measure that gives the dissimilarity between $\boldsymbol{u}_k^{(t)}$ and $\boldsymbol{v}_l^{(t')}$, the EMD is obtained by solving for the optimal *flow* f_{kl} through the following transportation problem:

$$f_{kl}^* = \arg\min_{f_{kl}} \sum_{k=1}^{K} \sum_{l=1}^{L} f_{kl} d_{kl} \tag{7}$$

subject to the following constraints.

$$f_{kl} \geq 0, \qquad 1 \leq k \leq K,\ 1 \leq l \leq L, \tag{8}$$

$$\sum_{l=1}^{L} f_{kl} \leq w_k^{(t)}, \qquad 1 \leq k \leq K, \tag{9}$$

$$\sum_{k=1}^{K} f_{kl} \leq w_l^{(t')}, \qquad 1 \leq l \leq L, \tag{10}$$

$$\sum_{k=1}^{K} \sum_{l=1}^{L} f_{kl} = \min \left(\sum_{k=1}^{K} w_k^{(t)}, \sum_{l=1}^{L} w_l^{(t')} \right). \tag{11}$$

Once the solution to Eq. (7) is obtained, the EMD between S_t and $S_{t'}$ is defined as

$$\mathrm{EMD}(S_t, S_{t'}) = \frac{\sum_{k=1}^{K} \sum_{l=1}^{L} f_{kl}^* d_{kl}}{\sum_{k=1}^{K} \sum_{l=1}^{L} f_{kl}^*}. \tag{12}$$

Intuitively speaking, as shown in Fig. 4, the EMD measures the minimum amount of work needed to make one signature out of the other signature.

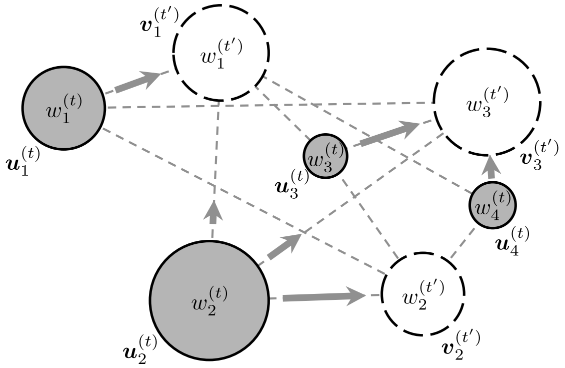


Fig. 4. Earth Mover's distance. Here S_t is drawn with the gray circles, and $S_{t'}$ is drawn with the white circles with dashed lines. The size of the circle represent the number of observations that belong to each cluster center. Given d_{kl} (dashed gray lines), the goal is to find the optimal flow f_{kl}^* which reconstructs one signature out of the other one. Once the solution is obtained (gray arrows), the EMD is calculated according to Eq. (12).

3.3 Assessing fluctuations in the metric space

Once the distributions corresponding to each bag are embedded in a metric space, we would want to consider how the distributions are fluctuating in the metric space. We would expect to see a large difference between the reference set and the test set in the metric space if there is a change, and vice versa. To measure the statistical difference between the two sets, we adopt the distance-based information estimators proposed in [19].

In [19], three types of computationally efficient information estimators for weighted data that are calculated using distance measures between data points are proposed. Using the signatures that we have obtained in Eq. (6), we define the weighted dataset that we consider as $\mathcal{S} := \{(S_i, \psi_i); i = 1, \ldots, n\}$, which is a set of signatures each associated with weight coefficients ψ_i that satisfy $\sum_i \psi_i = 1$ and $\psi_i \geq 0$. Details about ψ_i will be given later. Then, using the Earth Mover's distance, which we denote as $\mathrm{EMD}(\cdot, \cdot)$, we obtain the following quantities:

- Information content of a signature S with respect to $\mathcal{S}'(\mathcal{S}' := \{(S'_j, \psi'_j); j = 1, \ldots, m\})$:
$$I(S; \mathcal{S}') = c + d \sum_{j=1}^{m} \psi'_j \log \mathrm{EMD}(S'_j, S).$$
- Auto-entropy of $\mathcal{S}$:
$$H(\mathcal{S}) = c + d \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{m} \frac{\psi_i \psi_j}{1 - \psi_i} \log \mathrm{EMD}(S_i, S_j).$$
- Cross-entropy between $\mathcal{S}$ and $\mathcal{S}'$:
$$H(\mathcal{S}; \mathcal{S}') = c + d \sum_{i=1}^{n} \sum_{j=1}^{m} \psi_i \psi'_j \log \mathrm{EMD}(S_i, S'_j).$$

Here, c is a constant and d represents the dimension of S in the metric space which, theoretically, would be infinite. However, we make an assumption that the distributions have in common a finite effective dimension. This would be a reasonable assumption to make because N points in a metric space can be isometrically embedded to a Euclidean space of at most $N-1$ dimensions. Furthermore, the change-point scores that we compute takes the difference of the values written above and not the values themselves, and thus d is not essential.

Following Eq. (4) and Eq. (5), we will denote the reference set and the test set of the signatures as

$$\begin{aligned} \mathcal{S}_{\text{ref}} &:= \{(S_{t-\tau}, \psi_{t-\tau}), \ldots, (S_{t-1}, \psi_{t-1})\}, \quad (13) \\ \mathcal{S}_{\text{test}} &:= \{(S_t, \psi_t), \ldots, (S_{t+\tau'-1}, \psi_{t+\tau'-1})\}. \quad (14) \end{aligned}$$

We could make use of the weights ψ_i to reflect discounting to give more importance on data that is closer to the inspection point, e.g.

$$\psi_i \propto \frac{1}{|t-i|} \text{ (for } \mathcal{S}_{\text{ref}}), \quad \propto \frac{1}{|t-i+1|} \text{ (for } \mathcal{S}_{\text{ref}}). \quad (15)$$

This is a common scheme in time series analysis or time signal processing. Otherwise, we could simply make $\psi_i = 1/\tau$ or $1/\tau'$.

Based on these estimators, we propose two change-point scores as follows:

- Change-point score based on log likelihood ratio:
$$\begin{aligned} \mathrm{score}_{LR}(S_t) &= \log \left(\frac{p(S_t; \mathcal{S}_{\text{test}} \setminus S_t)}{p(S_t; \mathcal{S}_{\text{ref}})} \right) \\ &= I(S_t; \mathcal{S}_{\text{ref}}) - I(S_t; \mathcal{S}_{\text{test}} \setminus S_t). \quad (16) \end{aligned}$$
- Change-point score based on symmetrized KL divergence:
$$\begin{aligned} \mathrm{score}_{KL}(S_t) &= \frac{D_{KL}(\mathcal{S}_{\text{ref}} || \mathcal{S}_{\text{test}}) + D_{KL}(\mathcal{S}_{\text{test}} || \mathcal{S}_{\text{ref}})}{2} \\ &= \frac{1}{2} \Big(H(\mathcal{S}_{\text{ref}}; \mathcal{S}_{\text{test}}) - H(\mathcal{S}_{\text{ref}}) \\ &\quad + H(\mathcal{S}_{\text{ref}}; \mathcal{S}_{\text{test}}) - H(\mathcal{S}_{\text{test}}) \Big). \quad (17) \end{aligned}$$

The change-point score based on symmetrized KL divergence tends to be more conservative and robust, but at the same time insensitive to minor changes. On the other hand, the change-point score based on log likelihood ratio tends to behave in the opposite way.

4 Adaptive Thresholding of Change-Point Scores

The change-point score described in the previous section, as with change-point scores proposed in existing works, indicates the degree of change occurring at each inspection point, but do not offer a clear signal of where a significant, anomalous change has actually taken place. In practice, this is a crucial problem because one would need to be certain, to some extent, that an anomalous change has happened before taking action against the anomaly. Despite its importance, this is a problem that is overlooked in most existing works. In most cases, a threshold σ is assumed to be given beforehand, and one is instructed to simply compare the obtained change-point score with σ and raise an alarm whenever the change-point score goes above the predetermined threshold value. However, this approach is far from being reliable. In situations where the observed time-series is highly noisy or where the underlying distribution is constantly changing, high values in the change-point scores may not be true indications of a significant change. Therefore, it is important that the threshold σ is determined adaptively in a *data-centric* way.

4.1 Testing for significant changes

In order to determine if the change-point score obtained by Eq. (16) or Eq. (17) is indicating a true significant change or not, we propose to perform a statistical test at each time step. More specifically, letting γ_t denote the test statistic for time step t, we determine that the change-point score at time step t indicates a significant change if

$$\gamma_t > 0. \tag{18}$$

We compute γ_t by using confidence intervals of the change-point scores. The confidence intervals of the change-point scores are calculated for each time step using the Bayesian bootstrap method [20], which we will discuss further in the next section. Given a prespecified significance level η, the $100(1-\eta)\%$ confidence interval of the change-point score at time t, which we denote as $\left(\xi_{lo}^{(t)}, \xi_{up}^{(t)}\right)$, is obtained such that

$$\Pr\left(\xi_{lo}^{(t)} < \text{score}(t) < \xi_{up}^{(t)}\right) = 1 - \eta. \tag{19}$$

Once $\left(\xi_{lo}^{(t)}, \xi_{up}^{(t)}\right)$ is obtained, γ_t is computed as follows.

$$\gamma_t = \xi_{lo}^{(t)} - \xi_{up}^{(t-\tau')}. \tag{20}$$

Intuitively, γ_t considers the overlap between the confidence interval at time t and the confidence interval at time $t-\tau'$, as in Fig. 5. Note that τ and τ' were defined as the number of bags in the reference set and the test set, respectively. The reason for comparing the $\xi_{lo}^{(t)}$ with $\xi_{up}^{(t-\tau')}$ is that we want to make sure that the test set for the two confidence intervals do not share the same bags.

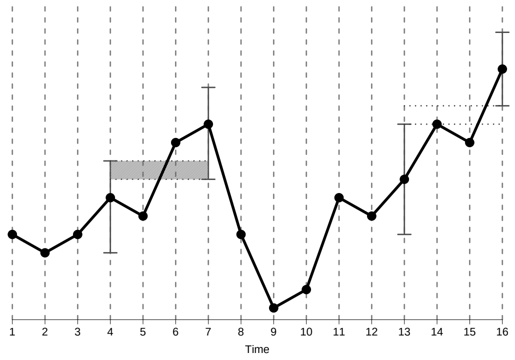


Fig. 5. An example of a sequence of change-point scores calculated with $\tau' = 3$. At both $t = 7$ and $t = 16$, we observe high change-point scores. We conclude that there is a significant change at $t = 16$ since there is no overlap between the confidence intervals, i.e. $\gamma_{16} > 0$. On the other hand, we do not conclude that there is a change-point at $t = 7$ because here $\gamma_7 < 0$.

Taking advantage of the fact that our proposed change-point scores take account of the weights associated with each signature, we propose to compute the confidence interval of the change-point scores using the Bayesian bootstrap method. In the next section, we will give a brief explanation of the Bayesian bootstrap and the method to compute the confidence interval, and then discuss about the advantages of the Bayesian bootstrap over standard bootstrap methods.

4.2 Computing confidence intervals using the Bayesian bootstrap

The bootstrap is a computer-based method used for evaluating the properties of a statistic $\hat{\phi}$ that is calculated based on a set of samples obtained from an unknown probability function F. One application in using the bootstrap is to compute the confidence interval of $\hat{\phi}$. In our setting, we wish to evaluate the confidence interval for the change-point score calculated at each time step.

The Bayesian bootstrap is the Bayesian analogue of the bootstrap [20]. As opposed to the standard bootstrap, which simulates the estimated sampling distribution of a statistic, the Bayesian bootstrap simulates the posterior distribution of the parameter.

One advantage of using the Bayesian bootstrap instead of the standard bootstrap method is that the Bayesian bootstrap is able to generate a smooth distribution of the statistic ϕ even when the number of samples is small. When applying our change-point detection method, it may often occur that τ and τ', which are the number of bags in the reference set and the test set respectively, are set to small numbers, depending on the situation considered. The Bayesian bootstrap would therefore be a better choice.

The procedure is as follows. For T times, the weight coefficients are resampled from Dirichlet distributions with different parameters.

$$\begin{aligned} \{\psi_{t-\tau}, \ldots, \psi_{t-1}\} &\sim \text{Dir}(\tau\psi_{t-\tau}, \ldots, \tau\psi_{t-1}), \quad (21) \\ \{\psi_t, \ldots, \psi_{t+\tau'-1}\} &\sim \text{Dir}(\tau'\psi_t, \ldots, \tau'\psi_{t+\tau'-1}). \quad (22) \end{aligned}$$

With each sample of $\{\psi_{t-\tau}, \ldots, \psi_{t-1}\}$ and $\{\psi_t, \ldots, \psi_{t+\tau'-1}\}$, the change-point scores in Eq. (16) or Eq. (17) are calculated, resulting in T values of change-point scores for time step t. Then, $\xi_{lo}^{(t)}$ and $\xi_{up}^{(t)}$ are determined as the $\eta/2$ quantile point and the $1-\eta/2$ quantile point, respectively, of the T change-point scores. More on the derivation and the theoretic aspect of this procedure is written in the Appendices A and B. Appendix A and B explain about using the Bayesian bootstrap for unweighted and weighted data as in (15), respectively.

5 Numerical Examples

We conduct several experiments to demonstrate the effectiveness of our approach. First, we demonstrate the validity of the confidence interval generated by the Bayesian bootstrap method by studying how the confidence interval behaves in different settings. Next, we apply our method to the PAMAP human activity dataset [21]. Finally, we demonstrate that our method could also be applied to detect changes in time-series sequence of bipartite graphs using simple statistics. For all of the results presented in this section, the signatures are weighted equally.

5.1 Behavior of confidence intervals

Suppose that at each time step t we observe a bag of two dimensional vectors $B_t = \{\boldsymbol{x}_i^{(t)}\}_{i=1}^{n_t}$ ($\boldsymbol{x}_i \in \mathbb{R}^2$), where the number of vectors in each bag n_t follows a Poisson distribution with $\lambda = 50$. We consider five different situations; all $\boldsymbol{x}_i$ are sampled from normal distributions, i.e. $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, with different parameters used for different datasets. Here, each sequence is consisted of 20 bags. For all the datasets, we make both the reference set and the test set to have five bags each, i.e. $\tau = \tau' = 5$.

- Dataset 1: All $\boldsymbol{x}_i$ are generated from a normal distribution with a large variance. For all time steps, $\boldsymbol{x}_i$ are generated from the following parameters, and there are no change points.

$$\forall t, i, \quad \boldsymbol{\mu} = \mathbf{0}, \Sigma = 15 I_2.$$

- Dataset 2: Approximately 80% of $\boldsymbol{x}_i$ are generated from a standard normal distribution. The remaining 20% are generated to simulate noise. There are no change points.

$$\forall t, \quad \begin{cases} \boldsymbol{\mu} = \mathbf{0}, \Sigma = I_2 & (1 \le i < \frac{4}{5} n_t), \\ \boldsymbol{\mu} \sim \mathcal{N}(\mathbf{0}, 20 I_2), \Sigma = 5 I_2 & (otherwise). \end{cases}$$

- Dataset 3: Here, $\boldsymbol{\mu}$ moves in a circular path and $\Sigma = I_2$. This is a simulation of situations where the distribution is constantly going through a gradual change. There are no significant change points.

$$\forall t, i, \quad \boldsymbol{\mu} = \sqrt{3}\left(\cos(\frac{(t-0.5)\pi}{5}) \quad \sin(\frac{(t-0.5)\pi}{5})\right)^{\top}.$$

- Dataset 4: There is a significant change at $t = 11$ where $\boldsymbol{\mu}$ moves from (3,0) to (-3,0).

$$\forall i, \quad \begin{cases} \boldsymbol{\mu} = (3 \quad 0)^{\top}, \Sigma = I_2 & (1 \le t \le 10), \\ \boldsymbol{\mu} = (-3 \quad 0)^{\top}, \Sigma = I_2 & (11 \le t \le 20). \end{cases}$$

- Dataset 5: The rate of change in $\boldsymbol{\mu}$ changes. Starting from $t = 11$, $\boldsymbol{\mu}$ starts to move faster.

$$\boldsymbol{\mu} = \epsilon\left(\cos(\frac{(t-0.5)\pi}{5}) \quad \sin(\frac{(t-0.5)\pi}{5})\right)^{\top}, \Sigma = I_2$$

$$\text{where} \begin{cases} \epsilon = \sqrt{3} & (1 \le t \le 10), \\ \epsilon = 3 & (11 \le t \le 20). \end{cases}$$

The results are shown in Fig. 6. The values of EMD are shown on the left. To offer an intuitive understanding of the datasets, the bags are mapped to a two dimensional space using multi-dimensional scaling. On the right, the change-point scores along with the confidence intervals are plotted. From the results, we could first see that our method did not raise any alarms for datasets that have no significant change points. We could also take note of the fact that the width of the confidence interval is larger for datasets 2, 3, and 5, which supports the fact that our method can reduce the risk of raising false alarms in highly noisy or unstationary situations. As for datasets that have a significant change point, our method was able to raise alerts successfully for dataset 4, but not for Dataset 5.

5.2 PAMAP Dataset

In order to evaluate the performance of our method, we experimented our method on the PAMAP2 physical activity monitoring dataset [21], which can be found at the UCI Machine Learning Repository. This is a dataset that contains data of 18 different physical activities, performed by nine subjects wearing three inertial measurement units and a heart rate monitor. Following a protocol, the nine subjects performed the activities listed in Table 1. Our objective is to detect when the subject changes his or her activity based on the data collected from the four sensors.

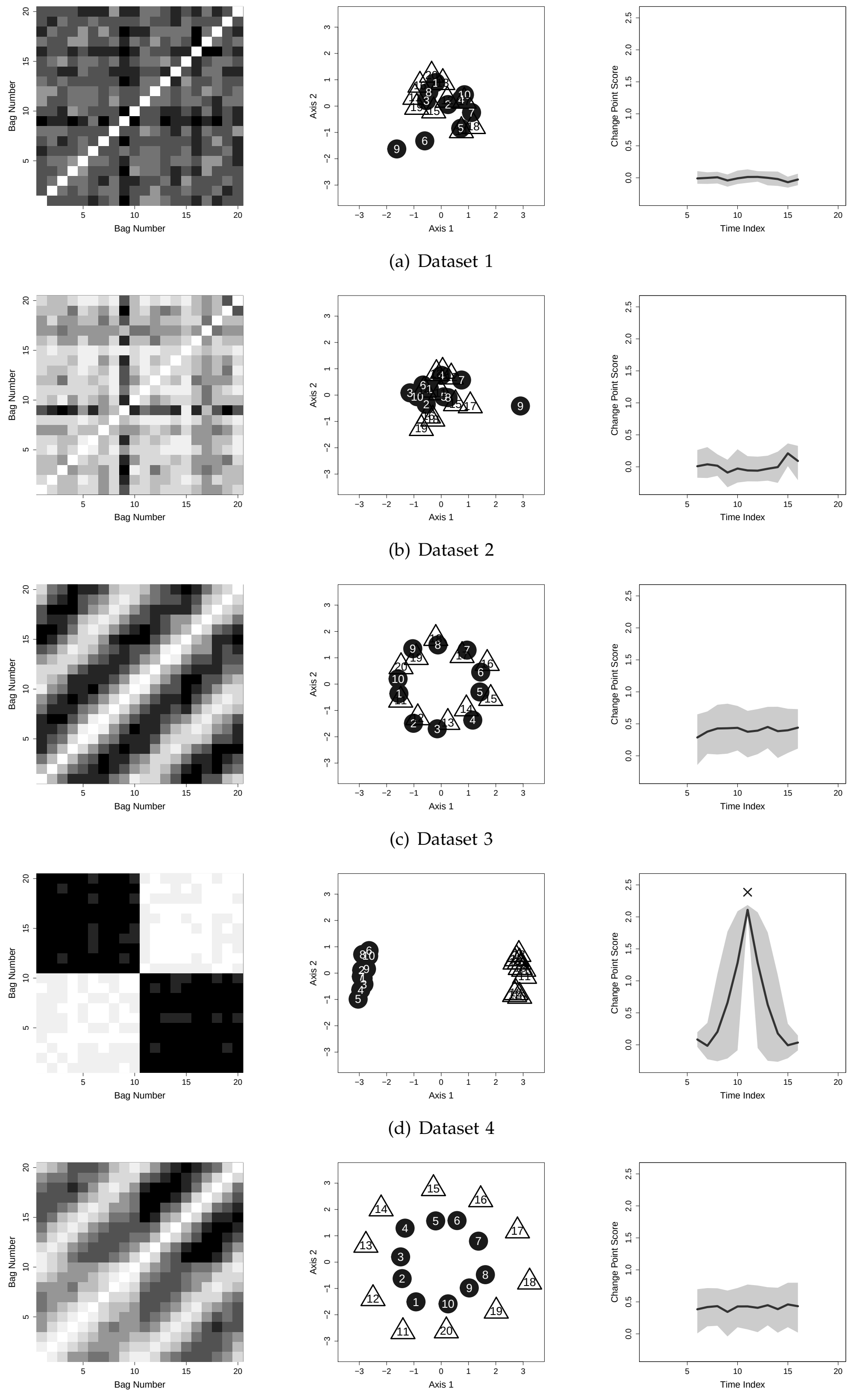
TABLE 1
Activities and their IDs.

Activity	ID	Activity	ID
lying	1	descending stairs	7
sitting	2	walking	8
standing	3	Nordic walking	9
ironing	4	cycling	10
vacuum cleaning	5	running	11
ascending stairs	6	rope jumping	12

Due to the slight difference in the sampling frequencies of the inertial measurement units, and other hardware faults such as connection loss or system crash, the number of observations recorded for each second is different throughout the whole dataset. Therefore, this is a situation where it is suitable to use bags to analyze the time-series data.

The time-series sequence of data recorded at each sensor is splitted every 10 seconds, which means that all the data that was observed in the same 10 second interval belong to the same bag. As a result, the data for each subject is separated in 251.8 bags in average with a standard deviation of 32.5. We set both the reference set and the test set to have five bags, i.e. $\tau = \tau' = 5$. The average number of records in each bag is 947.8 with a standard deviation of 162.3.

We present our results in Fig. 7. We present the results for three of the nine subjects whose changes were easier to see. For all subjects, the change points were detected with a plausible accuracy. Although alerts were not raised for all of the change points, we could still see a raise in the change-point scores when there is a change. Also, we could take note of the fact that no alerts were raised when the change-point score is oscillating rapidly. This is one of the advantageous effects of using confidence intervals to test for significant changes. Another advantage is that we could accurately raise alerts even when the change-point score is lower compared to those of other change points. This would not be achievable if we were to fix the threshold value σ to a single value.



(a) Dataset 1

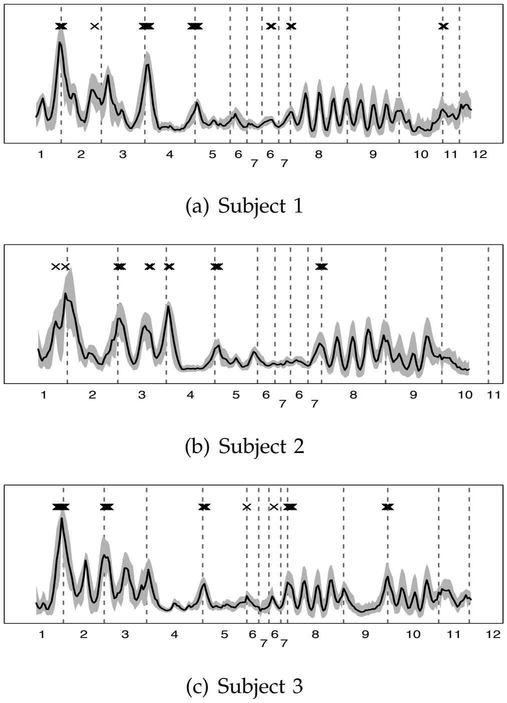
(b) Dataset 2

(c) Dataset 3

(d) Dataset 4

(e) Dataset 5

Fig. 6. The result of our method applied to five synthetic datasets. Left: a symmetric matrix whose ij elements is the EMD between bag i and bag j where darker color indicates further distance. Center: using the distance values on the left, the bags are mapped to a two dimensional space using multi-dimensional scaling. The number of the bag is indicated. The first ten bags are drawn with circles, and the next ten bags are drawn with triangles. Right: the change point scores are plotted. The black solid line is the change-point score, the gray shades are the 95% confidence intervals, and the cross mark shows where an alert was raised.



(a) Subject 1

(b) Subject 2

(c) Subject 3

Fig. 7. Results for the PAMAP Dataset. The activity IDs are indicated on the horizontal axis. Gray vertical dashed lines indicate where the change points are, and the cross marks indicate where our method raised alarms.

5.3 Sequence of bipartite graphs: synthetic data

There have been numerous works on detecting changes in a stream of bipartite graphs. In [22], Sun et al. take an information theoretic approach to discover communities and detect changes in dynamic networks in an online manner. In [23], [24], dissimilarity measures of bipartite graphs are proposed by comparing the so-called "behavior" (or "activity") vectors, which is the principal eigenvector of the correlation matrix (or dependency matrix in the latter case). All of these works assume that the graphs that they deal with have the same nodes throughout the whole sequence. In other words, they focus on detecting anomalous changes in how the nodes communicate with each other in a community whose *members do not change*. This becomes a bottleneck when the size of the network is overly massive since it becomes practically impossible to analyze the whole network.

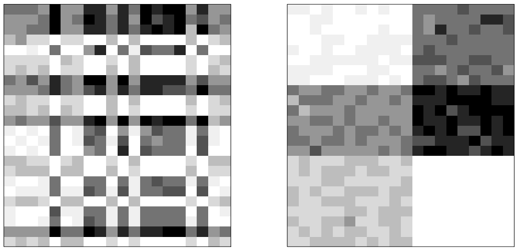
In these situations, an alternative approach would be to observe the network for a fixed amount of time, and then analyze the network that was observed in that time window. For example, if the task were to monitor a large e-mail network, like the one in the Enron dataset [25], we might consider analyzing the time-series on a daily basis, in which case we work with a bipartite graph whose source and destination nodes are comprised of people who sent and received e-mails on that day, respectively. Obviously, this would mean that the bipartite graphs for different days have different nodes of different numbers.

Here, we will demonstrate that our method could be applied to detect changes in a time-series sequence of bipartite graphs that have different numbers of nodes. More specifically, we extract features from each node or edge and then consider the feature vectors obtained for each bipartite graph as a *bag*, and then apply our method. Our rationale for doing this is that changes in the underlying communication patterns of bipartite graphs often cause changes in features obtained at each node. Our method does not have the capability of discovering communities as in [22] or specifying the nodes that largely contribute to the detected change as in [23], but if the task is solely to detect changes, our method would often suffice. Also, the method in [22] requires that all of the nodes in the network are known beforehand and that the network is always comprised of the same members. This assumption is unrealistic to make for many situations in the real world.

First, we evaluate our method with synthetic data simulating several situations. As mentioned above, we consider a situation where we are continuously observing communication among senders and receivers. We split the sequence with a fixed time window, and for each subset of data in each window, we form a bipartite graph. Therefore, each bipartite graph that we work with is a representation of communication that took place in the respective time windows.

When monitoring dynamic bipartite graphs, one's interest lies in discovering changes in two things: a change in the amount of traffic in the network and a change in how the source nodes and the destination nodes are forming groups, or clusters. Therefore, we design our synthetic data to simulate both situations. An example is shown in Fig. 8(a). It is a representation of a bipartite graph that has 24 source nodes and 20 destination nodes. Its ij element is the weight of the edge between the ith source node and jth destination node, which represents the number of queries that went from source node i to destination node j. Darker colors indicate heavier weights. We think of a setting where we can assume that both the source nodes and the destination nodes are comprised of clusters. We will use the word *community* to refer to groups of source nodes and destination nodes that are communicating in a similar way. For example, if we rearrange the nodes in Fig. 8(a), we obtain Fig. 8(b). Here, we can clearly see that the source nodes and the destination nodes are comprised of 3 and 2 clusters, respectively, and that there are 6 communities.

In our experiment, we consider bipartite graphs that have two source node clusters and two destination node clusters. Let n_s and n_d denote the number of source nodes and destination nodes, respectively. At each time step, both n_s and n_d are generated from a Poisson distribution of $\lambda = 200$, meaning that the number of source nodes and destination nodes vary over time. For each community, we assume that the weight of the edges follows a Poisson distribution, and we will denote the parameter of the Poisson distribution for the community comprised by source node cluster k and destination node



(a) Observed (b) Rearranged

Fig. 8. An illustrative representation of the bipartite graph. (a) When observed, the clusters of nodes that are formed in a bipartite graph may not be obvious. (b) When rearranged, we could see that the source nodes and the destination nodes are formed in groups.

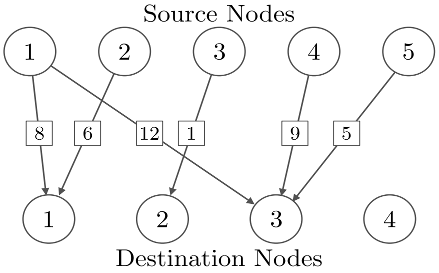


Fig. 9. An example of a bipartite graph that shows five source nodes sending queries to four destination nodes. The numbers of the nodes are labeled in the circle. The numbers in the boxes are the weights of the edges.

cluster l as $\lambda_{k,l}$. We will also denote the number of source nodes in each partition as αn_s and $(1-\alpha)n_s$ $(0 \leq \alpha \leq 1)$, and the number of destination nodes in each partition as βn_d and $(1-\beta)n_d$ $(0 \leq \beta \leq 1)$. The initial state is set to $\lambda_{1,1} = 10, \lambda_{1,2} = 3, \lambda_{2,1} = 1, \lambda_{2,2} = 5, \alpha = 0.5, \beta = 0.5$. We observe 200 points in total, and for every 20 points we alter the parameters to simulate a change point. The magnitude of the changes gradually becomes larger in later time steps, making the changes more obvious and easier to detect at later time steps. We will change parameters $\lambda_{1,1}, \lambda_{1,2}, \lambda_{2,1}, \lambda_{2,2}, \alpha, \beta$ in several different ways as described below to simulate different situations.

- Dataset 1: The partitioning of the source nodes and the destination nodes do not change, but the total amount of traffic changes.

  $$\forall k, l, \quad \lambda_{k,l} = \begin{cases} a+1 & (t = t_{a,1}, \ldots, t_{a,2}), \\ 1 & (otherwise), \end{cases}$$

  where $t_{a,1} = 20(a+1)+1$ and $t_{a,2} = 20(a+1)+20$, and $a = 1, \ldots, 5$. Here, α and β do not change.

- Dataset 2: The partitioning of the source nodes and the destination nodes changes. The total amount of traffic also changes since $\lambda_{k,l}$ do not change.

  $$\alpha = \beta = \begin{cases} 0.5 + 0.1a(-1)^{\nu} & (t = t_{a,1}, \ldots, t_{a,2}), \\ 0.5 & (otherwise), \end{cases}$$

  where $t_{a,1} = 20(a+1)+1, t_{a,2} = 20(a+1)+20, a = 1, \ldots, 5$, and $\nu \in \{0, 1\}$ is a random variable which takes one of the two values with the same probability. Here, $\lambda_{k,l}$ do not change.

- Dataset 3: This is a variant of dataset 2. The partitions are changed in the same way, but the total amount of traffic stays the same. Here, instead of sampling from Poisson distributions with parameters $\lambda_{1,1}, \lambda_{1,2}, \lambda_{2,1}, \lambda_{2,2}$, we fix the total weight of the edges to 100,000 and assign them to each community according to the ratio of the parameters $\lambda_{1,1}, \lambda_{1,2}, \lambda_{2,1}, \lambda_{2,2}$. In each community, the weights of the edges are distributed randomly.

- Dataset 4: Here, we do not change α and β, so the partitioning does not change. Instead, we interchange the values of $\lambda_{1,1}, \lambda_{1,2}, \lambda_{2,1}, \lambda_{2,2}$ in different ways.

For each bipartite graph that we observe, we obtain the seven features listed below. We will try to detect the changes using these features.

1) Degrees of source nodes:
   For each source node, the number of destination nodes that are connected to that node is counted. For example, in Fig. 9, source node 1 is connected to 2 destination nodes, so its degree is 2.
2) Degrees of destination nodes:
   For each destination node, the number of source nodes that are connected to that node is counted. In Fig. 9, destination node 1 is connected to 2 source nodes, so its degree is 2.
3) Second degrees of source nodes:
   For each source node, we count the number of source nodes connected to that node via a destination node. In Fig. 9, source node 1 is connected to destination nodes 1 and 3, which are connected to source nodes 2, and source nodes 4 and 5, respectively. Therefore, its second degree is 3.
4) Second degrees of destination nodes:
   For each destination node, we count the number of destination nodes connected to that node via a source node. In Fig. 9, destination node 1 is connected to source node 1, which is connected to destination node 3. Therefore, its second degree is 1. Note that source node 2 connects to destination node 1, but does not connect to any other destination nodes.
5) Total weight of the edges coming out from a source node:
   For each source node, we take the total weight of the edges coming out from that node. In Fig. 9, it would be 20 for source node 1, and 9 for source node 4.
6) Total weight of the edges going in to a destination node:
   For each destination node, we take the total weight

of the edges coming into that node. In Fig. 9, it would be 14 for destination node 1, and 26 for destination node 3.

7) Weight of each edge:
Here, we simply take the weight of each edge.

Since each of the statistics are computed for each node or edge and the number of nodes and edges differ for each graph, we analyze each graph using bags. The change-point scores calculated using Eq. (17) for datasets 1, 2, 3, and 4 are shown in Fig. 10. The change-points are indicated with dashed lines and the alerts are indicated with cross marks. In all situations, the changes are accurately detected when using statistics 5 and 6 as the features. This is even true for change-points in earlier stages where the magnitude of changes are small. Statistics 3 and 4 do not seem to be working in this case since the synthetic data that we produced do not consider the correspondence between the source nodes and the destination nodes.

5.4 Case study: ENRON corpus

Next, we examine the effectiveness of our method using the ENRON Corpus [25]. This is a dataset that contains all the email communications in Enron Inc. from January 1999 to July 2002. This was right before the collapse of Enron Inc. and we could expect to see some dramatic change points in how people communicated each other using emails. Following [25], we cleaned the dataset for use in this experiment by removing duplicate emails and computer-generated messages. We also focused our experiment on records from July 1, 2000 to May 31, 2002 since the amount of emails before and after this period was very scarce. The resulting corpus included 278,274 messages. We constructed the bipartite graphs on a weekly basis. The duration of the reference set and the test set are five weeks and three weeks, respectively.

For each bipartite graph, we computed the same seven statistics that we used in the previous section. The change-point scores calculated with Eq. (17) are shown in Fig. 11 along with the dates of critical events that involved Enron Inc. labeled with dashed lines. From the results, we could see that the change-point scores coincide with many of the important events. Also, note that there are two columns with X's in the table next to the graph. The X's in the left column indicate that the corresponding events were detected by our method with at least one of the seven features, and the X's in the right column indicate that the events were detected by [22]. We were able to detect most of the events that were detected in [22] along with some extras that were not detected in [22]. It is most likely valid in saying that we have detected the changes in communication patterns that happened along with these events.

6 Concluding Remarks

In this paper, we proposed a scheme for nonparametric change-point detection in the setting where each observation is a bag of data, i.e. a collection of random variables. By modeling each bag in the form of signatures, and employing EMD as a distance measure, the change-point scores are efficiently calculated using the distance-based information estimators. To test for the significance of the changes, we used the Bayesian bootstrap to construct confidence intervals of each change-point score. Through experimental studies, we have shown that our method could be applied to a wide range of datasets, even for detecting changes in a stream of bipartite graphs. Although we did not discuss about it in this paper, we have used this method to detect cyber attacks in a darknet, and it has performed very well.

An important future challenge would be to implement an online feature selection algorithm. It might often occur that only a couple of dimensions of $\boldsymbol{x}$ are relevant to changes, while the other features are completely irrelevant. There could also be an underlying structure in a lower dimension $d' < d$ that separates normal and abnormal behaviors more correctly. Using data that have the class labels ("change" or "no change") for each time step, which could be obtained in an online manner, we could think of learning a mapping and apply it on all $\boldsymbol{x}$ before constructing signatures in order to improve the accuracy of this change-point detection scheme.

Also, another future work might be to consider a situation where the elements in each bag are correlated. In time-series analysis, signals are often preprocessed by removing the predictable component. The resulting innovation time series is an i.i.d. sequence [26], and this is the assumption we have made in this paper. However, considering correlation in the data could be an interesting topic for additional research.

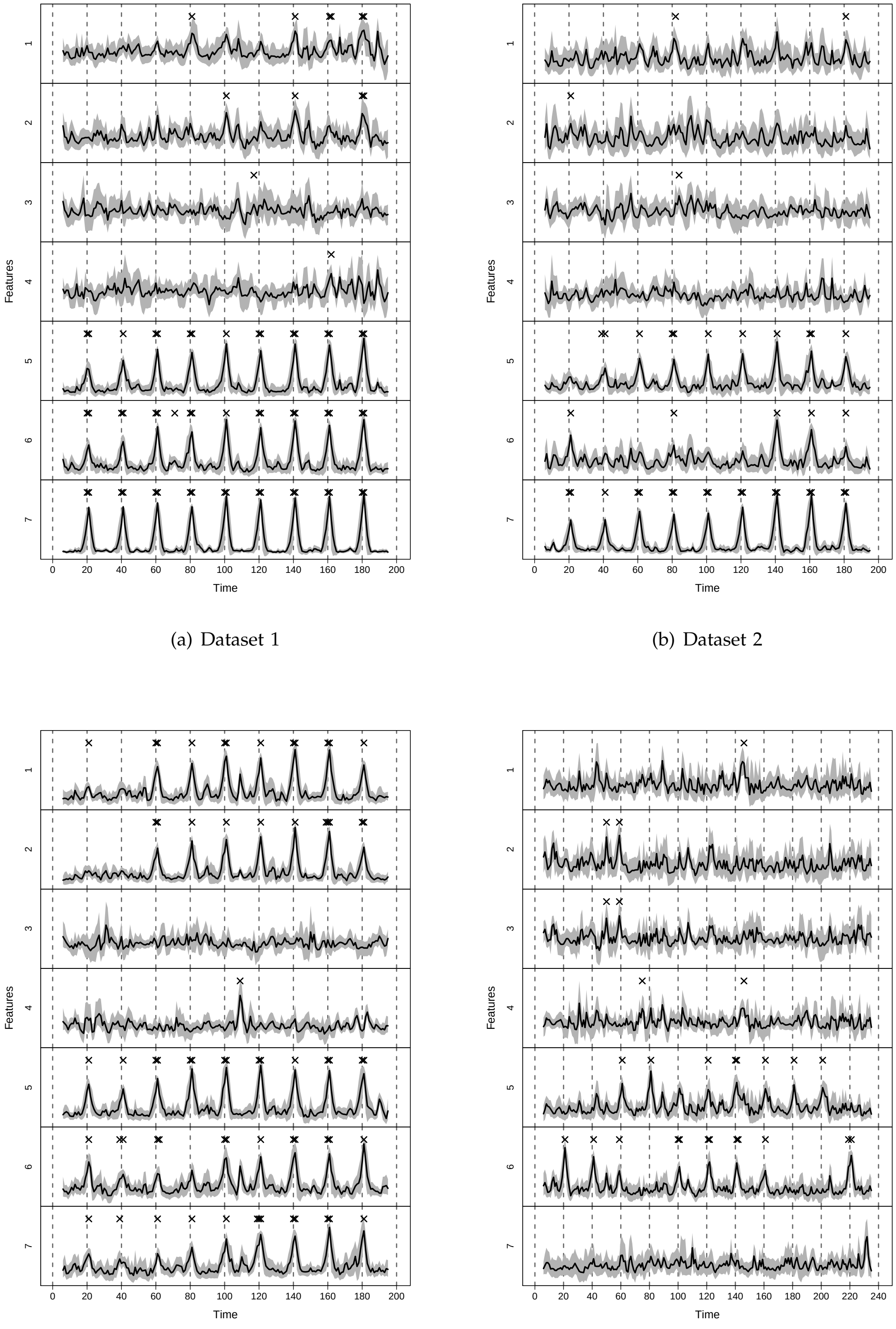
Appendix A
Bayesian Bootstrap

Suppose we have a sample $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, which is viewed as n i.i.d. realizations of a random variable X. The Bayesian bootstrap can be used when the computation of the statistic of our interest involves the "probability" (or the "weight") associated with each value in $\mathbf{x}$. An example of this is the sample mean $\hat{\mu}$. In the standard bootstrap setting, if we let f_i be the proportion of times x_i is drawn in a bootstrap replication, the sample mean for that bootstrap replication can be computed as $\sum_{j=1}^{n} f_j x_j$. Here, each f_i can be viewed as the probability associated with each x_i which is obtained *as a result of* each resampling process. In the Bayesian bootstrap, on the other hand, we evaluate the same statistic by *directly* assigning a probability to each x_i.

Let $\boldsymbol{v} = (v_1, \ldots, v_K)$ be the vector of all possible distinct values of X, and let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ be the corresponding vector of probabilities, i.e.,

$$P(X = v_k | \boldsymbol{\theta}) = \theta_k, \qquad \sum_k \theta_k = 1.$$

Let $\mathbf{x}$ be an i.i.d. sample and let n_k be the number of x_i



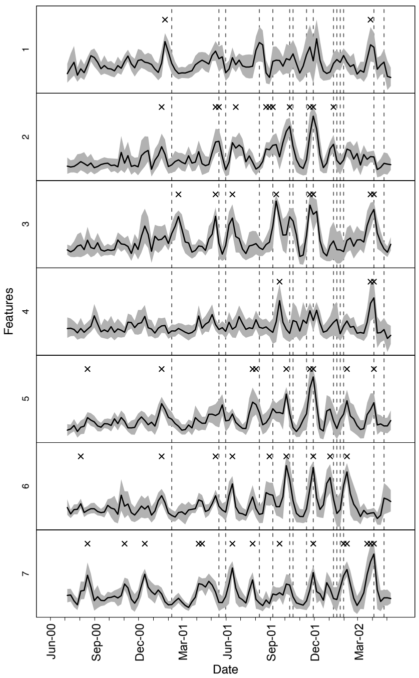
(a) Dataset 1

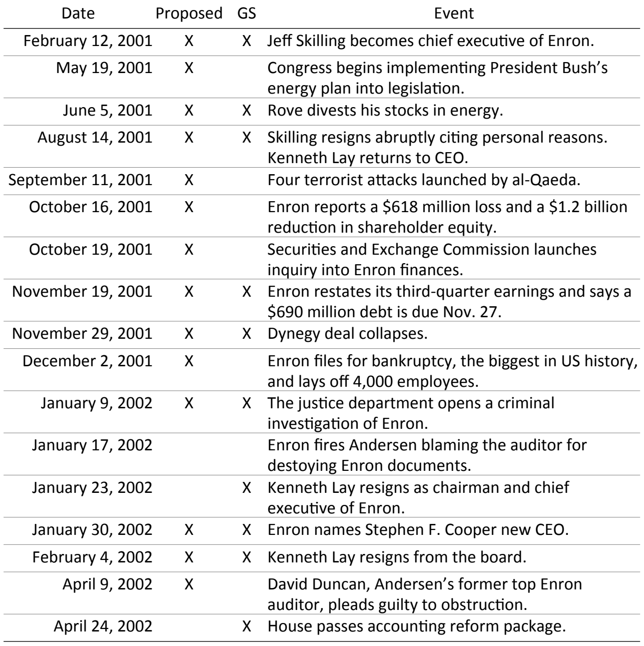
(b) Dataset 2

(c) Dataset 3

(d) Dataset 4

Fig. 10. Results for synthetic datasets of bipartite graphs. The artificial change-points are indicated with vertical dashed lines, and the alerts are indicated with cross marks. All the change-points are accurately detected with at least one of the features, regardless of the magnitude of the changes. In some cases, the change-point scores take high values when there is no change, but the confidence intervals are properly functioning to avoid raising false alerts.





Date	Proposed	GS	Event
February 12, 2001	X	X	Jeff Skilling becomes chief executive of Enron.
May 19, 2001	X		Congress begins implementing President Bush's energy plan into legislation.
June 5, 2001	X	X	Rove divests his stocks in energy.
August 14, 2001	X	X	Skilling resigns abruptly citing personal reasons. Kenneth Lay returns to CEO.
September 11, 2001	X		Four terrorist attacks launched by al-Qaeda.
October 16, 2001	X		Enron reports a $618 million loss and a $1.2 billion reduction in shareholder equity.
October 19, 2001	X		Securities and Exchange Commission launches inquiry into Enron finances.
November 19, 2001	X	X	Enron restates its third-quarter earnings and says a $690 million debt is due Nov. 27.
November 29, 2001	X	X	Dynegy deal collapses.
December 2, 2001	X		Enron files for bankruptcy, the biggest in US history, and lays off 4,000 employees.
January 9, 2002	X	X	The justice department opens a criminal investigation of Enron.
January 17, 2002			Enron fires Andersen blaming the auditor for destoying Enron documents.
January 23, 2002		X	Kenneth Lay resigns as chairman and chief executive of Enron.
January 30, 2002	X	X	Enron names Stephen F. Cooper new CEO.
February 4, 2002	X	X	Kenneth Lay resigns from the board.
April 9, 2002	X		David Duncan, Andersen's former top Enron auditor, pleads guilty to obstruction.
April 24, 2002		X	House passes accounting reform package.

Fig. 11. Results for ENRON Corpus. Some important events are indicated with vertical dashed lines, and the specific dates are listed on the right. We could see that the change-point scores coincide with many of the events.

equal to v_k. Then we have,

$$P(\mathbf{x}|\boldsymbol{\theta}) \propto \prod_{k=1}^{K} \theta_k^{n_k}. \tag{23}$$

Letting the prior distribution of $\boldsymbol{\theta}$ be

$$P(\boldsymbol{\theta}) \propto \prod_{k=1}^{K} \theta_k^{l_k}, \tag{24}$$

the posterior distribution of $\boldsymbol{x}$ becomes

$$P(\boldsymbol{\theta}|\mathbf{x}) \propto \prod_{k=1}^{K} \theta_k^{(n_k+l_k)}. \tag{25}$$

By letting all $n_k = 1$ and $l_k = -1$, which means that the prior distribution in Eq. (24) would be an improper prior, $P(\boldsymbol{\theta}|\mathbf{x})$ becomes $\mathrm{Dir}(1, 1, \ldots, 1)$. As a result, the Bayesian bootstrap produces the posterior probability g_i for each x_i, i.e. $\{g_1, \ldots, g_n\} \sim \mathrm{Dir}(1, 1, \ldots, 1)$, and then computes the statistic using this probability, e.g. $\sum_{j=1}^{n} g_j x_j$ in the sample mean case. The rationale for making all $n_k = 1$ is that probabilities corresponding to the same value would be added up. Also, by making all $l_k = -1$, we can guarantee that the mean, variance, and correlation of f_i and g_i would be very similar.

$$\begin{aligned} E[f_i] &= E[g_i] = \frac{1}{n}, \\ \mathrm{var}[f_i] &= \mathrm{var}[g_i] \cdot \frac{n+1}{n} = \frac{n-1}{n^3}, \\ \mathrm{cor}[f_i, f_j] &= \mathrm{cor}[g_i, g_j] = -\frac{1}{n-1}. \end{aligned}$$

Appendix B
Bayesian bootstrap for weighted data

Let w_i be the weights assigned to each x_i in $\mathbf{x}$. In the standard bootstrap setting, instead of sampling from $\mathbf{x}$ *randomly*, we would want to draw samples from $\mathbf{x}$ according to the weights w_i. In other words, the probability of drawing x_i from $\mathbf{x}$ would be w_i/w_0, where $w_0 = \sum_{i=1}^{n} w_i$. Since the number of times x_i is drawn in each bootstrap sample, say m_i, follows a multinomial distribution, i.e.

$$P(m_1, m_2, \ldots, m_n) = \frac{1}{Z} \prod_{i=1}^{n} \rho_i^{m_i}$$

where $\rho_i = w_i/w_0$, the mean and the variance of the *proportion* of times x_i is drawn in a bootstrap sample (which we will again denote as f_i) would be as follows.

$$E[f_i] = \rho_i, \quad \mathrm{var}[f_i] = \frac{\rho_i(1-\rho_i)}{n}.$$

On the other hand, the mean and the variance of a Dirichlet distribution with parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$, i.e. $\{g_1, \ldots, g_n\} \sim \mathrm{Dir}(\alpha_1, \ldots, \alpha_n)$, are given as follows.

$$\begin{aligned} E[g_i] &= \frac{\alpha_i}{\alpha_0}, \\ \mathrm{var}[g_i] &= \frac{\alpha_i(\alpha_0 - \alpha_i)}{(\alpha_0)^2(\alpha_0+1)} = \frac{E[g_i](1-E[g_i])}{\alpha_0+1}, \end{aligned}$$

where $\alpha_0 = \sum_{i=1}^{n} \alpha_i$. Therefore, if we let $\alpha_i = n\rho_i$, we would have the following relation.

$$\begin{aligned} E[f_i] &= E[g_i] = \rho_i, \\ \mathrm{var}[f_i] &= \mathrm{var}[g_i] \cdot \frac{n+1}{n} = \frac{\rho_i(1-\rho_i)}{n}, \\ \mathrm{cor}[f_i, f_j] &= \mathrm{cor}[g_i, g_j] = -\frac{\sqrt{\rho_i \rho_j}}{\sqrt{(1-\rho_i)(1-\rho_j)}}. \end{aligned}$$

Acknowledgments

The authors would like to express our special thanks to the editor and reviewers whose comments led to valuable improvements of the manuscripts. We are also grateful to Ken Takano and Takahiro Nukushina for helping programming. Part of this work was supported by JSPS KAKENHI No. 25120009, 25870811, and 26120504, and by Proactive Response Against Cyber-attacks Through International Collaborative Exchange (PRACTICE), Ministry of Internal Affairs and Communications, Japan.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[2] M. E. Basseville and I. V. Nikiforov, "Detection of abrupt changes: theory and application," 1993.

[3] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.

[4] R. Fujimaki, T. Yairi, and K. Machida, "An approach to spacecraft anomaly detection problem using kernel feature space," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 401–410.

[5] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011.

[6] M. Davy and S. Godsill, "Detection of abrupt spectral changes using support vector machines an application to audio signal segmentation," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 2. IEEE, 2002, pp. II–1313.

[7] F. Gustafsson, "The marginalized likelihood ratio test for detecting abrupt changes," *Automatic Control, IEEE Transactions on*, vol. 41, no. 1, pp. 66–78, 1996.

[8] J. Takeuchi and K. Yamanishi, "A unifying framework for detecting outliers and change points from time series," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 4, pp. 482–492, 2006.

[9] F. Desobry, M. Davy, and C. Doncarli, "An online kernel change detection algorithm," *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 2961–2974, 2005.

[10] V. Moskvina and A. Zhigljavsky, "An algorithm based on singular spectrum analysis for change-point detection," *Communications in Statistics-Simulation and Computation*, vol. 32, no. 2, pp. 319–352, 2003.

[11] T. Idé and K. Tsuda, "Change-point detection using krylov subspace learning." in *SDM*. SIAM, 2007, pp. 515–520.

[12] S. Liu, M. Yamada, N. Collier, and M. Sugiyama, "Change-point detection in time-series data by relative density-ratio estimation," *Neural Networks*, 2013.

[13] N. Murata, K. Koshijima, and H. Hino, "Distance-based change-point detection with entropy estimation," in *Proceedings of the Sixth Workshop on Information Theoretic Methods in Science and Engineering*, 2013, pp. 22–25.

[14] T. Kohonen, *Learning vector quantization*. Springer, 2001.

[15] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, 2004, p. 22.

[16] J. Arroyo and C. Maté, "Forecasting histogram time series with k-nearest neighbours methods," *International Journal of Forecasting*, vol. 25, no. 1, pp. 192–207, 2009.

[17] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth Mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[18] E. Levina and P. Bickel, "The Earth Mover's distance is the Mallows distance: some insights from statistics," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 251–256.

[19] H. Hino and N. Murata, "Information estimators for weighted observations," *Neural Networks*, vol. 46, pp. 260–275, 2013.

[20] D. B. Rubin, "The Bayesian bootstrap," *The annals of statistics*, vol. 9, no. 1, pp. 130–134, 1981.

[21] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Wearable Computers (ISWC), 2012 16th International Symposium on*. IEEE, 2012, pp. 108–109.

[22] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, "GraphScope: parameter-free mining of large time-evolving graphs," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 687–696.

[23] L. Akoglu and C. Faloutsos, "Event detection in time series of mobile communication graphs," in *Army Science Conference*, 2010, pp. 77–79.

[24] 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*. ACM, 2004, pp. 440–449.

[25] B. Klimt and Y. Yang, "The ENRON corpus: A new dataset for email classification research," in *Machine learning: ECML 2004*. Springer, 2004, pp. 217–226.

[26] P. J. Brockwell and R. A. Davis, *Time series: theory and methods*. Springer, 2009.

Kensuke Koshijima received his Bachelor's degree in engineering in 2012, and Master's degree in engineering in 2014 from Waseda University, Japan. His research interest includes data mining and machine learning.

Hideitsu Hino received his B. Eng. in 2003, and M. Informatics in Applied Mathematics and Physics in 2005 from Kyoto University, Japan. He was with Hitachi's Systems Development Laboratory from April 2005 to August 2007. He earned D. Eng. in 2010 from Waseda University. From April 2013, he is an Assistant Professor at University of Tsukuba. His research interests includes the analysis of learning algorithms. He is a member of IEEE.

Noboru Murata received the B. Eng., M. Eng. and Dr. Eng in Mathematical Engineering and Information Physics from the University of Tokyo in 1987, 1989 and 1992, respectively. After working at the University of Tokyo, GMD FIRST in Germany, and RIKEN in Japan, in April of 2000, he joined Waseda University in Japan where he is presently a professor. His research interest includes the theoretical aspects of learning machines such as neural networks, focusing on the dynamics and statistical properties of learning.