Computing Correlation Anomaly Scores using Stochastic Nearest Neighbors

Tsuyoshi Idé
IBM Research, Tokyo Research Laboratory
Yamato, Kanagawa, Japan
goodidea@jp.ibm.com

Spiros Papadimitriou Michail Vlachos
IBM T.J. Watson Research Center
Hawthorne, NY, USA
{spapadim, vlachos}@us.ibm.com

Abstract

This paper addresses the task of change analysis of correlated multi-sensor systems. The goal of change analysis is to compute the anomaly score of each sensor when we know that the system has some potential difference from a reference state. Examples include validating the proper performance of various car sensors in the automobile industry. We solve this problem based on a neighborhood preservation principle — If the system is working normally, the neighborhood graph of each sensor is almost invariant against the fluctuations of experimental conditions. Here a neighborhood graph is defined based on the correlation between sensor signals. With the notion of stochastic neighborhood, our method is capable of robustly computing the anomaly score of each sensor under conditions that are hard to be detected by other naive methods.

1 Introduction

Knowledge discovery from data streams is one of the major research topics in data mining. In recent years, growing attention has been paid to mining techniques from multivariate time series data, which are naturally represented as a stream of weighted graphs [7, 11, 13, 12]. In such a graph, each node corresponds to each time series, and each edge is weighted by the (dis)similarity between a pair of time series.

One of the typical tasks of stream mining is change (or anomaly) detection. Change detection is an unsupervised learning task, which aims at deciding on whether the data generating mechanism has been changed or not. When considering multivariate systems, however, more advanced tasks involving anomaly or change detection are also of importance. After detecting a change, we are generally interested in which variables (or, more generally, degrees of freedom) are responsible for the change. We call this step *change analysis*.

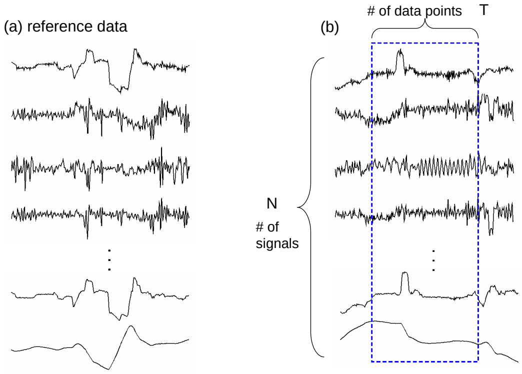

Figure 1. Problem setting. We wish to compute the anomaly score of each sensor in a target run (b) using a reference data set (a).

In this paper, we address the problem of change analysis of multivariate time-series data. One motivating application is the task of sensor validation, where sensor signals are inspected for detecting proper operation. Our assumption about the input data is as follows: (1) The signals are highly dynamic, so a different experimental run can have a completely different trend (see Fig. 1). (2) The signals are heterogeneous, i.e. a mixture of seemingly different types of signals. (3) There can be strong correlations between signals, so that individual analysis of each sensor can overlook interesting anomalies. (4) Supervised information about the behavior of each sensor is generally not given. Thus, we need to treat the problem in an unsupervised learning fashion.

Note that the highly dynamic nature makes it impossible to use existing alignment techniques [1, 8, 9] over different runs to compute the *anomaly score*, which represents how much a variable is responsible for the difference from a reference state. Thus, instead of making direct alignment with corresponding variables, we focus on the (dis)similarity between signals in the same run. In other words, we wish to perform change analysis based on complete graphs whose weights are defined by the (dis)similarities between signals.

This paper solves this task using a notion of *neighborhood preservation*: Under normal system operation, the neighborhood graph of each node is almost invariant against the fluctuations of experimental conditions. In particular, we compute the anomaly scores by stochastically evaluating how much this assumption is broken. Focusing only on local structures of the graph, this principle works surprisingly well even for data of the heterogeneous nature, where global approaches such as principal component analysis [10] are clearly less useful. To the best of our knowledge, this is the first work which successfully solves the change analysis task for highly dynamic, correlated, and heterogeneous sensor data.

The layout of this paper is as follows. Section 2 provides the problem setting rather formally. Section 3 defines an anomaly metric. Section 4 presents some experimental results, including a real-world task of car sensor validation. Finally, Section 5 summarizes the paper.

2 Problem setting and overview

In this section, we formalize the problem, and introduce the key concept of neighborhood preservation.

2.1 Correlation anomaly analysis

Consider a dynamic system having N physical sensors such as pressure, acceleration, and luminance sensors. Each sensor produces real-valued time-series data with T time points (see Fig.1). We call such a data unit an experimental run. We assume that the measurements are done synchronously with a fixed frequency. In a single run, let $x_i^{(t)}$ be the observation of the i-th sensor ($i = 1, 2, ..., N$) at a time index t ($t = 1, 2, ..., T$). Let D and $\bar{\mathsf{D}} \in \mathbb{R}^{N \times N}$ be the dissimilarity matrices of target and reference runs, respectively. By thinking of D and $\bar{\mathsf{D}}$ as weight matrices of graphs, our problem is stated as follows.

Definition 1 (Correlation anomaly analysis) *Given a target graph with* D *and a reference graph with* $\bar{\mathsf{D}}$*, provide the score of each node which accounts for the difference between the graphs.*

Hereafter we use the bar ($\bar{\ }$) to represent the corresponding quantity of a reference run. We denote the (i, j) element of D ($\bar{\mathsf{D}}$) by $d_{i,j}$ ($\bar{d}_{i,j}$), representing the dissimilarity between the i-th and j-th signals.

This problem can be easily extended to an online change analysis by thinking of $\bar{\mathsf{D}}$ and D as dissimilarity matrices at two different times. Even in this scenario, our implicit assumption is that the same sensors are used in any run.

For the definition of the dissimilarities, we make use of the correlation coefficients, and take an "inverse". Specifically, given the correlation coefficients $\{a_{i,j}\}$, we define $d_{i,j}$ so as to satisfy the following conditions. (1) $d_{i,i} = 0$ for $\forall i$, (2) $d_{i,i} \approx 0$ for highly-correlated pairs, and (3) $d_{i,i} \to \infty$ for almost uncorrelated pairs. Here it is important to understand the difference between the second and the third conditions. A large correlation coefficient should be thought of as a representation of the internal structure of the system, so it should be treated carefully. On the other hand, a value of small correlation coefficients is considered to carry no useful information of the system. Hereafter, we take a simple definition of $d_{ij} = -\log |a_{i,j}|$. As usual, $a_{i,j}$ is defined by $c_{i,j}/\sqrt{c_{i,i}c_{j,j}}$, where

$$c_{i,j} \equiv \frac{1}{T}\sum_{t=1}^{T}[x_i^{(t)} - \langle x_i \rangle][x_j^{(t)} - \langle x_j \rangle],$$

and $\langle x_i \rangle \equiv \frac{1}{T}\sum_{t=1}^{T} x_i^{(t)}$. For constant signals, we define $a_{i,j} = \delta_{i,j}$, where $\delta_{i,j}$ is the Kronecker's delta function.

2.2 The principle of neighborhood preservation

We define the l-th nearest neighbor (NN) to a node i as the one that has the l-th smallest dissimilarity to i (except for i itself). We also define the k-NN set w.r.t. i as the collection of the 1st, 2nd, ..., k-th NNs to i, and denote by $\mathcal{N}_i$. Now let us define the k-neighborhood graph of the i-th node as follows:

Definition 2 (k-neighborhood graph) *The k-neighborhood graph of the i-th node is a graph that contains $\mathcal{N}_i$ and the i-th node itself, connected with edges between the i-th node and its neighbors.*

In this definition, we call the i-th node the central node of the k-neighborhood graph.

As described in the introduction, one of the basic assumptions of ours is that multi-sensor systems, such as automobiles and artificial satellites, include pairs of highly correlated sensors. Otherwise, the change analysis problem would be trivial since it can be solved by analyzing each sensor separately. From Fig. 1, one might think that the fluctuations of sensor signals is so strong that the change analysis is extremely challenging. However, our observation shows that most of the unimportant fluctuations in a highly dynamic system are due to weakly correlated pairs of sensors. Conversely, highly correlated pairs of sensors tend to be hardly affected by the change in experimental conditions, under normal system operation. These observations lead to the following principle.

Definition 3 (Neighborhood preservation principle) *If the system is working normally, the neighborhood graph of each node is almost invariant against the fluctuations of experimental conditions.*

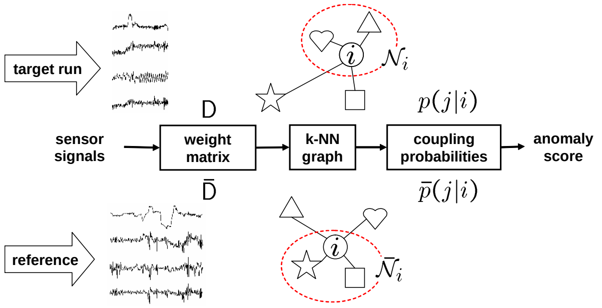

Figure 2. Flowchart of our method for computing anomaly scores.

Figure 2 summarizes our implementation of this principle. The key of our approach is to use stochastic neighborhood graphs. In particular, we use a probability distribution $p(j|i)$, which represents a coupling probability between the i- and j-th sensors.

3 Stochastic k-neighborhood graph

The neighborhood preservation principle suggests that the change in the k-neighborhood graph around the i-th node should be related to the anomaly score of the i-th sensor. In this section, we describe how to evaluate the change.

3.1 Stochastic nearest neighbors

Let $\mathcal{N}_i$ and $\bar{\mathcal{N}}_i$ be the k-NN sets w.r.t. the i-th node in target and reference runs, respectively. To quantitatively evaluate the change between the k-neighborhood graphs, we introduce the coupling probability of the j-th node with the i-th node, and denote by $p(j|i)$. For each i, the normalization condition is given by

$$p(i|i) + \sum_{j \in \mathcal{N}_i} p(j|i) = 1, \tag{1}$$

where we assume $p(j|i) = 0$ if $j \notin \mathcal{N}_i$ and $j \neq i$. Note that we have included the self-coupling probability $p(i|i)$ here. This term corresponds to the probability that the central node has no coupling with its neighbors.

To determine $p(j|i)$, consider the following problem: Under the condition that the central node i takes a constant number of couplings with neighbors on average, construct the neighborhood graph as compact as possible. Here it is helpful to consider the case where all the neighbors are equal. In this case, the distribution may be $p(j|i) = \frac{1}{k}(1 - \delta_{i,j})$. For this distribution, the conditional entropy

$$H_i \equiv - \sum_{j \in i \cup \mathcal{N}_i} p(j|i) \ln p(j|i)$$

is readily calculated as $\ln k$. Thus the *perplexity*, which is defined by e^{H_i}, is just k. This example shows that the perplexity is a probabilistic counterpart of the number of nearest neighbors [1].

Since the compactness of the neighborhood graph around the i-th node is naturally represented by

$$\langle d_i \rangle \equiv \sum_{j \in \mathcal{N}_i} d_{i,j} p(j|i),$$

which is the expectation of dissimilarity around i, we can get the distribution $p(j|i)$ by solving the following optimization problem

$$\min \langle d_i \rangle \ \text{ s.t. } \ \mathrm{e}^{H_i} = \text{const.} \quad \text{and Eq. (1).} \tag{2}$$

By introducing Lagrange's multipliers for H_i and Eq. (1), and by differentiating w.r.t. $p(j|i)$, we finally get

$$p(j|i) = \frac{1}{Z_i} \mathrm{e}^{-\frac{d_{ij}}{\sigma_i}}, \tag{3}$$

where the partition function Z_i is defnined by

$$Z_i \equiv 1 + \sum_{l \in \mathcal{N}_i} \mathrm{e}^{-\frac{d_{il}}{\sigma_i}}. \tag{4}$$

The multiplier σ_i is to be determined by the condition about the perplexity. Otherwise, one may include σ_i in the definition of $d_{i,j}$ so that $\sigma_i = 1$.

3.2 Anomaly score

Now we have distributions $p(j|i)$ and $\bar{p}(j|i)$ for test and reference runs, respectively. According to the neighborhood preservation principle, the difference between the following quantities should be small if the system is working normally.

$$e_i(\mathcal{N}_i) \equiv \sum_{j \in \mathcal{N}_i} p(j|i) \tag{5}$$

$$\bar{e}_i(\mathcal{N}_i) \equiv \sum_{j \in \mathcal{N}_i} \bar{p}(j|i). \tag{6}$$

Clearly, e_i measures the tightness of the coupling between the central node and its neighbors in a target run in terms of probability. Also, $\bar{e}_i(\mathcal{N}_i)$ measures the tightness of the coupling around the i-th node in a reference run using the k-NN set defined in the target set (i.e. not necessarily the same as the k-NN set of the reference data). For defining $\bar{e}_i(\mathcal{N}_i)$ (as well as $e_i(\bar{\mathcal{N}}_i)$), we assume one-to-one correspondence of sensor identities between target and reference runs.

[1] In the context of stochastic neighborhood graph, this fact was first pointed out by Hinton and Roweis [6].

Similarly, by replacing $\mathcal{N}_i$ with $\bar{\mathcal{N}_i}$, $e_i(\bar{\mathcal{N}_i})$ and $\bar{e}_i(\bar{\mathcal{N}_i})$ can be defined. Here it is easy to see

$$0 \leq e_i \leq \frac{k}{k+1}. \tag{7}$$

The same holds for $\bar{e}_i$. The lower bound is obtained when the i-th node is totally uncorrelated to others, and the upper bound is obtained when the nodes are perfectly correlated.

Using the tightnesses, we define the anomaly score of the i-th node as

$$E \equiv \max\left\{ \left| e_i(\mathcal{N}_i) - \bar{e}_i(\mathcal{N}_i) \right|, \left| e_i(\bar{\mathcal{N}_i}) - \bar{e}_i(\bar{\mathcal{N}_i}) \right| \right\}. \tag{8}$$

We call this the *E-score* hereafter. Note that the E-score is given by the difference between probabilities, having the same bound as Eq. (7). This feature makes interpretation of the E-score very clear.

3.3 Properties of E-score

Role of $p(i|i)$. Our formulation of the neighborhood graph is capable of naturally discounting such nodes that are almost uncorrelated with any of the others. Since $d_{i,j} \ll 1$ for $j \neq i$ in such a case, the self-coupling probability $p(i|i)$ dominates the others. As a result, the E-score will be always negligible. This feature provides robustness over the highly dynamic nature of data, and contrasts with previous stochastic formulations of neighborhood [6, 5], where self-coupling terms are not included.

Choosing parameters. The only input parameter of our approach is k (the number of NNs). In theory, k should be chosen as the minimum size of tight clusters. While optimally determining k from the data is a challenging research issue [14], a value of two or three was turned to work well in our applications.

Complexity. For computing the dissimilarity matrix and the k-neighborhood graphs, the complexities are TN^2 and kN^2 times, respectively. If N is approximately more than O(10^3), a fast method for computing the dissimilarity matrix and k-neighborhood graphs will be needed. However, such an issue is out of scope of this paper, and naive direct computation suffices in applications explained in the next section.

4 Experiments

In this section, we demonstrate the utility of our approach using real-world data. Note that there is no standard method of change analysis which can handle such data that have the four features explained in the introduction.

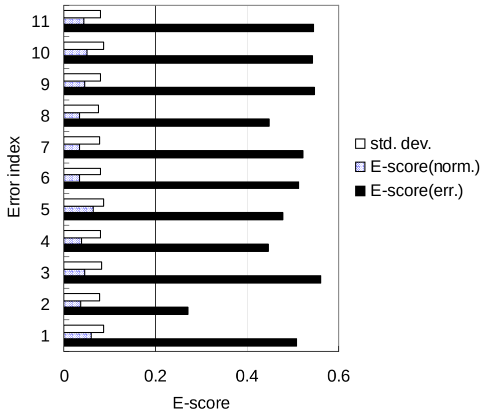

Figure 3. Averaged E-scores for $k = 3$.

4.1 Car sensor validation

In this experiment, we focus on picking up sensor misinstallation errors. For example, some acceleration sensors has a small cubic shape, so that, say, the z-axis might be installed as the y-axis by human error. Since misplaced signals themselves can be still active, and their mean and variance may remain almost the same, errors of this type are extremely hard to detect by the human eye.

Data Set. Sensor signals were collected from embedded sensors of an automobile running on the street, in collaboration with an automaker. Several instances of the sensor signals were shown in Fig. 1. As shown, the trend of each signal in one run can be completely different from another run. We excluded inactive signals to use $N = 61$ signals for change analysis. We generated test (i.e. faulty) and reference runs by splitting long data stored in a database, so that each run has about one minute length. Before analysis, the data is resampled so that all of the time series have the same time interval of 0.2 seconds (so $T \approx 300$). We performed change analysis over 11 types of misplacement errors. For each error, we had 25 reference-target pairs.

Analysis. Figure 3 compares the average scores between faulty and normal sensors for the 11 error patterns. Each of the bars is the mean of the 25 reference-target pairs. We fixed $\sigma_i = 1$ to avoid numerical instabilities, and set $k = 2$. We also computed the averaged standard deviation (computed as the square root of the averaged variance) of correctly working sensors. As shown, we have a clear contrast between faulty and normal ones even when considering the standard deviations. Considering the fact that the upper bound of the E-score is $\frac{2}{3}$ for $k = 2$, we see that some of the error patterns almost attain the upper bound. This means that the sensor misplacement error completely changes the neighborhood graphs.

To look at the result in more detail, we picked one of the

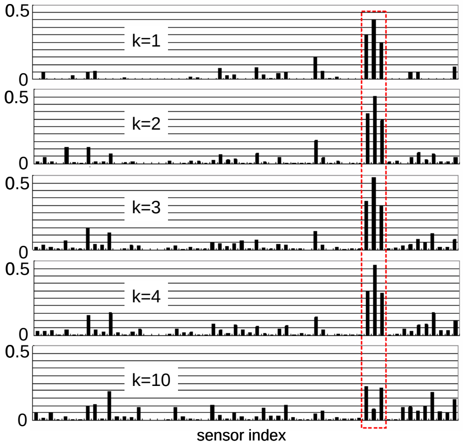

Figure 4. The k-dependence of the E-scores.

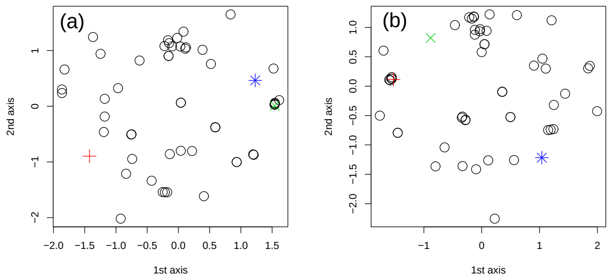

Figure 5. Sammon maps for a reference-target pair of the second error pattern. (a) The reference and (b) target data.

25 reference-target pairs in the second error pattern, which marks the worst contrast in Fig. 3, and showed the E-scores in Fig. 4 for different ks. From the names of sensors, the misplaced signals seem to come from a single three-axis acceleration sensor. Despite the fact that this is picked from the worst cases, the misplaced sensors (highlighted with the rectangle of dashed line) are clearly pinpointed by exceptionally high E-scores when $k \leq 4$.

We visualized this reference-target pair using Sammon map [2] in Fig. 5, where the misplaced sensors are marked with '+', '$\times$', and '$*$'. We see that the '+' sensor is almost isolated from others in the reference data, but it gets into a tightly connected cluster in (b). Close inspection shows the size of this cluster is five, which explain why the E-score graph is robust within $k \leq 4$ in Fig. 4. Comparing between the two maps, we also see that the result of embedding is not stable at all. In other words, the global structure of the data manifold is quite vulnerable to the fluctuation. This result clearly supports our neighborhood preservation strategy.

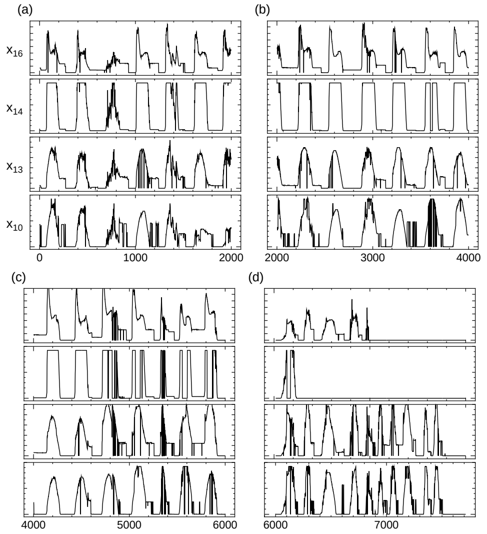

Figure 6. A part of raw time-series data of the LightSensor data (x_{10}, x_{13}, x_{14}, and x_{16}).

4.2 LightSensor data

So far we have based on batch-style scenarios. This subsection presents an application of our method to online sensor data analysis.

Data set. The LightSensor data [4] is a subset of the garden data set [3], containing $N = 48$ sensor signals recorded at the UC Botanical Garden in Berkeley. The sensors are installed in a single redwood tree, and placed at 4 different altitudes in the tree, where they collect luminance values once every 5 minutes. We split this data set into four non-overlapping subsets with $T \approx 2,000$, as shown in Fig. 6, where one tic corresponds to 5 minutes. In the figure, only x_{10}, x_{13}, x_{14}, and x_{16} signals are shown out of the 48 signals. We took the first part (from zero through 2,000 tics) as the reference, and studied the time dependence of the E-scores.

As shown in Fig. 6, the signals have approximately 24-hour periodicity. This is because the luminance is almost zero at night. However, even during the daytime, climate changes and other random factors also affect the values of luminance. Since climate changes are at random, deciding on the sensor state is not easy even when a sensor signal outputs some very low value; it would be almost impossible only by looking at individual sensors separately. Thus focusing on correlation anomaly will be a reasonable approach.

Analysis. While detailed information about what was happening when the measurement was done is not available, some of the sensors seem to be dying towards the end of the data, according to Fig. 6. The overall trend would be that

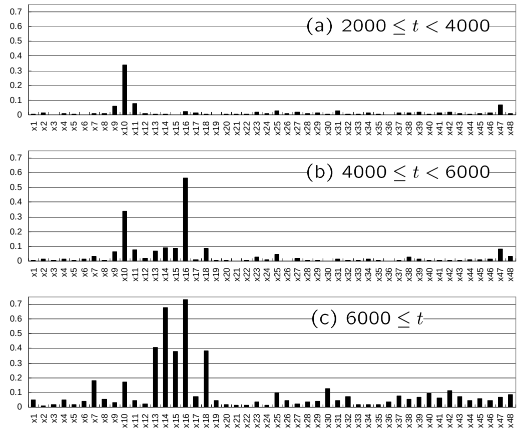

Figure 7. Time dependence of the E-score for the LightSensor data.

the signals are relatively regular in earlier periods while out of shape in later periods. Our goal is to pick up this trend in terms of the E-score. Since the signals are quite noisy, early detection is challenging.

Figure 7 shows E-score graphs at different periods. We used $k = 3$ and $\sigma_i = 1$. We see that the E-scores exhibit interesting time dependence. As expected, several sensors takes very high scores (almost attain the upper bound of 3/4) in the last period. Interestingly, the 16th sensor takes a very high score also in Fig. 7 (c). This anomaly is hard to detect just by looking at individual sensors, showing the utility of our method.

5 Summary

We have formulated a task of change analysis of correlated sensor signals. This task can be viewed as one that account for the difference between two weighted graphs, whose weights are computed based on the correlation between the signals. We showed that the neighborhood preservation principle makes the algorithm surprisingly robust over the variability of time series data. In addition, the anomaly score can be related to a probability value by introducing coupling probabilities between nodes in a k-neighborhood graph. Finally, we demonstrated the utility our approach using real-world data.

References

[1] D. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *AAAI-94 Workshop on Knowledge Discovery in Databases*. AAAI, 1994.

[2] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling, 2nd ed.* Chapman and Hall, 2001.

[3] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *Proc. Intl. Conf. Very Large Data Bases*, pages 588–599, 2004.

[4] C. Faloutsos, J. Sun, E. Hoke, and S. Papadimitriou. Spirit: Streaming pattern discovery [`http://www.cs.cmu.edu/afs/cs/project/spirit-1/www/`].

[5] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood component analysis. In *Advances in Neural Information Processing Systems, 17*, pages 513–520, 2005.

[6] G. Hinton and S. T. Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems, 15*, pages 833–840, 2003.

[7] T. Idé and H. Kashima. Eigenspace-based anomaly detection in computer systems. In *Proc. ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, pages 440–449, 2004.

[8] E. Keogh and M. Pazzani. Scaling up dynamic time warping for data mining applications. In *Proc. ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, pages 285–289, 2000.

[9] J. Listgarten, R. M. Neal, S. T. Roweis, and A. Emili. Multiple alignment of continuous time series. In *Advances in Neural Information Processing Systems 17*, pages 817–824, 2005.

[10] S. Papadimitriou, J. Sun, and C. Faloutsos. Streaming pattern discovery in multiple time-series. In *Proc. Intl. Conf. Very Large Data Bases*, pages 697–708, 2005.

[11] S. Papadimitriou, J. Sun, and P. Yu. Local correlation tracking in time series. In *Proc. IEEE Intl. Conf. Data Mining*, pages 456–465. IEEE, 2006.

[12] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *Proc. IEEE Intl. Conf. Data Mining*, pages 418–425, 2005.

[13] J. Sun, Y. Xie, H. Zhang, and C. Faloutsos. Less is more: Compact matrix decomposition for large sparse graphs. In *Proc. SIAM Intl. Conf. Data Mining*, 2007.

[14] X. Yang, S. Michea, and H. Zha. Conical dimension as an intrinsic dimension estimator and its applications. In *Proc. SIAM Intl. Conf. Data Mining*, 2007.