Latent Space Tracking from Heterogeneous Data with an Application for Anomaly Detection

Jiaji Huang[1(✉)] and Xia Ning[2]

[1] Department of Electrical Engineering, Duke University, Durham, NC 27708, USA
jiaji.huang@duke.edu
[2] Department of Computer and Information Science,
IUPUI, Indianapolis, IN 46202, USA
xning@cs.iupui.edu

Abstract. Streaming heterogeneous information is ubiquitous in the era of Big Data, which provides versatile perspectives for more comprehensive understanding of behaviors of an underlying system/process. Human analysis of these volumes is infeasible, leading to unprecedented demands for mathematical tools which effectively parse and distill such data. However, the complicated nature of streaming heterogeneous data prevents the conventional multivariate data analysis methods being applied immediately. In this paper, we propose a novel framework together with an online algorithm, denoted as LSTH, for latent space tracking from heterogeneous data. Our method leverages the advantages of dimension reduction, correlation analysis and sparse learning to better reveal the latent relations among heterogeneous information and adapt to slow variations in streaming data. We applied our method on both synthetic and real data, and it achieves results competitive with or superior to the state-of-the-art in detecting several different types of anomalies.

1 Introduction

In the era of Big Data, heterogeneity of various information generated from a same yet complex underlying system/process has become ubiquitous. Examples of such heterogeneous data include video and audio from a sensor network, acoustic and articulatory signals during a speech, etc. Such heterogeneous data provides complimentary or augmented depiction of the system from different perspectives, allowing more comprehensive understanding of the system than that from homogeneous data. Albeit the high dimensionality and heterogeneity, these data often exhibits low dimensional nature and can be characterized by a (low dimensional) latent space. Correctly identifying the latent space benefits classical machine learning tasks (e.g., classification [6]), as well as more novel applications (e.g., the anomaly detection). However, learning from heterogeneous data is highly nontrivial. The requirement of operating in real time imposes further challenges and prevents straightforward extensions of existing methods.

T. Cao et al. (Eds.): PAKDD 2015, Part I, LNAI 9077, pp. 429–441, 2015.
DOI: 10.1007/978-3-319-18038-0_34

Principal Component Analysis (PCA) [7] is arguably the most well-known method for extracting the low dimensional latent space. A common assumption in applying PCA is that most data is near the low dimensional space. The anomalies are assumed to be significantly deviated from the space such that using some simple statistics is sufficient to identify them. Inspired by this assumption, online PCA [12] techniques are developed to conduct anomaly detection on data streams. Representative online PCA algorithms include [4] as well as its extension [17] under union-of-subspace assumption. However, PCA based methods do not model the relations between the heterogeneous data sources. Therefore, PCA cannot identify anomalies corresponding to violation of the relations. In contrast, Canonical Correspondence Analysis (CCA) [5] is a classical method for analyzing the relation between multiple data sources. And online CCA through stochastic gradient on generalized Stiefel manifold has been applied to anomaly detection on time series [19]. However, it still does not fully consider the heterogeneous nature of the data.

Recently, learning from heterogeneous data has attracted much attention in machine learning community, particularly in transfer learning, multi-task learning and multi-view learning. Transfer learning utilizes an auxiliary source domain data to learn a better model in a target domain, where the two domains are often heterogeneous [13]. Multi-task learning leverages the relation between multiple tasks, each of which may work on a different/heterogeneous data domain [6]. Multi-view learning leverages multiple views of same instances for better models [18]. Many of these works assume a common low dimensional latent space, and learn a mapping from each data source/view to the latent space in a supervised fashion. However, adapting these methods to an online and unsupervised setting (e.g., anomaly detection task) is not straightforward.

In this paper, we tackle the problem of online learning of heterogeneous data via latent space tracking. In specific, we propose a framework to track the low-dimensional latent structures of heterogeneous data and learn their inherent relations. Our formulation incorporates the key insights underlying PCA, CCA, and sparse learning to enable dimension reduction together with feature selection for anomaly detection from heterogeneous data. We develop an efficient online algorithm that effectively conducts **L**atent **S**pace **T**racking from **H**eterogeneous data, denoted as LSTH. Based on the learned latent space, we further design an anomaly detection method that reports anomalies significantly outlying the latent space. We test LSTH on both synthetic and real datasets. Experimental results demonstrate that LSTH is effective in revealing relations among heterogeneous data for anomaly detection.

The paper is organized as follows. Section 2 formulates the latent space tracking problem. Section 3 presents the tracking algorithm. Section 4 further designs an anomaly detection method as an application of the learned latent space. Experimental results and conclusions are in Section 5 and 6 respectively.

2 Problem Formulation

Throughout this paper, vectors are represented by lower-case letters (e.g., x), and matrices are represented by upper-case letters (e.g., U). By default, all the vectors are column vectors, while row vectors are represented by having a transpose superscript$^\top$ (e.g., $x^\top$). We use subscript i and j to index an element in a matrix (e.g., $V_{i,j}$) and subscript t to index a data point at timestamp t (e.g., x_t) in a data steam. The estimate of a variable is represented by having a hat over the variable (e.g., $\widehat{U}$ represents the estimate of U).

We assume $x_t \in \mathbb{R}^{D_x}$ and $y_t \in \mathbb{R}^{D_y}$ are the high-dimensional heterogeneous data samples from a same system at timestamp t, where D_x and D_y are the number of features in x_t and y_t, respectively. The heterogeneity of particular interest in this paper is that x_t's features are correlated, whereas only very few features in y_t describe the states of the system. Heterogeneous data in many real-life applications exhibits such kind of property. For example, during a speech, y_t can be data recorded by articulatory sensors, which are highly correlated [3] due to connected muscles. In contrast, x_t can be Mel-frequency cepstrum coefficients (MFCC). Obtained by appending higher order derivatives of acoustic signal, it contains much redundancy and often need a feature selection [11] step before further processing. In a stock market, x_t could be the prices of multiple correlated stocks, and y_t is massive news about the market [14].

In order to learn the underlying structures and relations among x_t and y_t, we monitor the joint probability density $p(x_t, y_t)$ at each timestamp t:

$$p(x_t, y_t) = p(y_t|x_t)p(x_t). \tag{1}$$

However, since both x_t and y_t are of high dimensionality, online density estimation for $p(y_t|x_t)$ or $p(x_t)$ is prohibitively difficult. Therefore, we assume there is a d dimensional latent space ($d \ll D_x, D_y$) underlying the data, into which x_t and y_t can be transformed via two linear projectors $U \in \mathbb{R}^{D_x \times d}$ and $V \in \mathbb{R}^{D_y \times d}$. Their projections are denoted as $U^\top x_t$ and $V^\top y_t$, respectively, which can be considered as realizations of a common latent variable that determines the states of the underlying system. U and V will exhibit different structures. Specifically, while U may span a low-rank subspace as in `PCA`, V may model a latent space impacting only a subset of the features in y_t.

3 Proposed Approach

We constrain U to be orthonormal (i.e., $U^\top U = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix) to preserve the magnitude of x_t. Thus, the reconstruction error of x_t is $\|x_t - UU^\top x_t\|^2$. In this case, we measure the probability distribution of x_t by the reconstruction error [17]:

$$p(x_t) \propto \exp\left(-\|x_t - UU^\top x_t\|^2/\sigma_x^2\right), \tag{2}$$

where σ_x^2 is the variance of reconstruction error in each dimension. Since the projections of x_t and y_t are considered as realizations of a common latent variable, they are expected to be close. Hence, we measure $p(y_t|x_t)$ by the distance of the projections in the latent space:

$$p(y_t|x_t) \propto \exp\left(-\|V^\top y_t - U^\top x_t\|^2/\sigma_y^2\right), \tag{3}$$

where σ_y^2 is the variance of the difference between x_t and y_t in the latent space.

By substituting Equation (2) and (3) into (1) and taking the logarithm, the log-likelihood can be represented as

$$\log p(x_t, y_t) \propto -\left[\frac{\|V^\top y_t - U^\top x_t\|^2}{\sigma_y^2} + \frac{\|(\mathbf{I} - UU^\top)x_t\|^2}{\sigma_x^2}\right].$$

In addition, we constrain V to exhibit "group sparse" structure so that applying V performs feature selection from y_t to identify the most informative features. We use the mixed norm $\|V\|_{1,2} \triangleq \sum_{i=1}^{D_y} \|v_i^\top\|_2$ to introduce sparsity into V, where $v_i^\top$ is the i-th row of V.

To enable tracking in a slowly evolving environment, we apply an exponentially decaying window to downweigh the historical samples. In addition, we define $\sigma = \sigma_y^2/\sigma_x^2$, and denote the estimates of U and V at timestamp t as $\widehat{U}_t$ and $\widehat{V}_t$, respectively. Then we formulate the following optimization problem to find the projectors U and V at timestamp t:

$$\begin{aligned}(\widehat{U}_t, \widehat{V}_t) &= \underset{U^\top U = \mathbf{I}, V}{\arg\min}\ F(U, V; t, \alpha, \sigma, \lambda) \\ &= \underset{U^\top U = \mathbf{I}, V}{\arg\min} \sum_{k=0}^{t-1} \frac{\alpha^k}{2}\left(\|U^\top x_{t-k} - V^\top y_{t-k}\|^2 + \sigma\|(\mathbf{I} - UU^\top)x_{t-k}\|^2\right) + \lambda\|V\|_{1,2},\end{aligned} \tag{4}$$

where $\alpha \in (0, 1]$ is a forgetting factor over historical samples to implement the decaying window, σ balances between projection residual and discrepancy in the latent space, and λ is the regularization parameter for sparsity. Note that the data stream starts from $t = 1$.

In the above $F(U, V; t, \alpha, \sigma, \lambda)$, the first term measures the discrepancy of two data sources in the latent space. It has the flavor of `CCA` that maximizes the correlation of two projections. Same as `PCA`, the second term imposes low-dimensional structure in x_t. It is important to highlight the $\|V\|_{1,2}$ term here. $\|v_i^\top\|_2$ indicates the significance of the i-th feature in y_t. In addition, $\|V\|_{1,2}$ is invariant if multiplying an unitary matrix to the right of V. Therefore, the cost of (4) depends on the subspace spanned by $\widehat{U}_t$ and $\widehat{V}_t$ rather than the particular basis chosen.

3.1 A Batch Algorithm

We first present a batch algorithm, denoted as `bLSTH`, to solve U and V for simplicity. The `bLSTH` algorithm will be further modified into an online version in Section 3.2.

Algorithm 1. The Batch Algorithm `bLSTH`

Input: samples $X \in \mathbb{R}^{D_x \times L}$, $Y \in \mathbb{R}^{D_y \times L}$
Parameters: λ, σ, latent dimension d
Output: $\widehat{U}$ and $\widehat{V}$

$i \leftarrow 0$, $U[0] \leftarrow$ the first d principal components of X
repeat
$i \leftarrow i + 1$
$Z \leftarrow U[i-1]^\top X$
$V[i] \leftarrow \arg\min_V \frac{1}{2}\|V^\top Y - Z\|_F^2 + \lambda\|V\|_{1,2}$ (6)
$W \leftarrow V[i]^\top Y$
$U[i] \leftarrow \arg\min_{U^\top U = \mathbf{I}} \frac{1}{2}\left(\|U^\top X - W\|_F^2 + \sigma\|(\mathbf{I} - UU^\top)X\|_F^2\right)$ (7)
until $U[i], V[i]$ converge or i is large enough
$\widehat{U} \leftarrow U[i]$, $\widehat{V} \leftarrow V[i]$

In `bLSTH`, L buffered samples $X = [x_{-L+1}, \cdots, x_0]$, $Y = [y_{-L+1}, \cdots, y_0]$ are used to solve the following optimization problem:

$$(\widehat{U}, \widehat{V}) = \arg\min_{U^\top U = \mathbf{I}, V} \frac{1}{2}\Big(\|U^\top X - V^\top Y\|_F^2 + \sigma\|(\mathbf{I} - UU^\top)X\|_F^2\Big) + \lambda\|V\|_{1,2}.$$

We use an alternating method to solve for $\widehat{U}$ and $\widehat{V}$, as presented in Algorithm 1. The optimization problem in Equation (6) of Algorithm 1 is a well-studied convex optimization problem. Now we focus on the optimization problem in Equation (7). The objective can be reformulated as:

$$\begin{aligned} f(U;\sigma) &\triangleq \frac{1}{2}\left(\|U^\top X - W\|_F^2 + \sigma\|(\mathbf{I} - UU^\top)X\|_F^2\right) \\ &= \frac{1}{2}(1-\sigma)\,\mathtt{tr}\left\{U^\top X X^\top U\right\} - \mathtt{tr}\left\{\left(XW^\top\right)U^\top\right\}, \end{aligned} \tag{5}$$

where $U^\top U = \mathbf{I}$ and $W = V^\top Y$. This orthonormality constrained problem is non-convex. However, we are able to find a local minimum within a few iterations and our experiments show that even local minimum is able to give good results. Following the idea in [8], we use a majorization minimization scheme. The basic idea is to construct a non-decreasing sequence $f(U[1]), \ldots, f(U[k]), \ldots$ that converges to a local minimum of $f(U)$. Specifically, suppose we are at $U[k]$, we construct a surrogate function $g_k(U)$ that satisfies

$$f(U) \le g_k(U) \text{ and } f(U[k]) = g_k(U[k]). \tag{8}$$

That is, $g_k(U)$ is an upper bound of $f(U)$ and the equality holds when $U = U[k]$. Assign the global minimizer of $g_k(U)$ to $U[k+1]$, thus the sequence

$f(U[1]), \ldots, f(U[k]), \ldots$ is guaranteed to be non-increasing due to the properties of $g_k(U)$ as in Equation (8) and the notion of global minimizer. In practice, a surrogate function should be constructed such that its global minimizer is easily obtained. The following two lemmas suggest one form of such $g_k(U)$ and its global minimizer.

Lemma 1. *For any given orthonormal matrix $U[k] \in \mathbb{R}^{D_x \times d}$, the following $g_k(U; a)$ defined on the set of orthonormal matrices $U \in \mathbb{R}^{D_x \times d}$*

$$g_k(U; a) = \mathtt{tr}\left\{ \left[(1-\sigma)(XX^\top - a\,\mathbf{I})U[k] - (XW^\top) \right]^\top U \right\} + c$$

is a surrogate function for the $f(U; \sigma)$ in Equation (5), where c is some constant independent of U. And the scalar a chosen as

$$a = \begin{cases} \lambda^* & \sigma < 1 \\ 0 & \sigma \geq 1 \end{cases},$$

where λ^* is the maximum eigenvalue of $XX^\top$.

Proof. The proof leverages Rayleigh quotient inequality and is omitted for conciseness.

Lemma 2. *[10] The global minimizer of*

$$\min_{U^\top U = \mathbf{I}} -\mathtt{tr}\{A^\top U\}$$

*is $PQ^\top$, where $P\Sigma Q^\top = A$ is the Singular Value Decomposition (*SVD*) of A.*

Using Lemma 2, the global minimizer of the surrogate function $g_k(U; a)$ has a closed form $\arg\min_{U^\top U = \mathbf{I}} g_k(U; a) = PQ^\top$,where $P\Sigma Q^\top$ is the SVD of $XW^\top - (1-\sigma)(XX^\top - a\,\mathbf{I})U[k]$.Thus, by applying Lemma 1 and 2, the problem in Equation (7) can be solved via the iterative majorization minimization process as presented in Algorithm 2, where $G \triangleq XW^\top = XY^\top V$ and $C_x \triangleq XX^\top$. A special case is when $\sigma = 1$, in which the minimizer of $f(U; \sigma)$ is given by the closed-form solution directly by Lemma 2.

3.2 An Online Algorithm

Here we derive the online algorithm LSTH from bLSTH. We use the solution ($\widehat{U}$, $\widehat{V}$) by bLSTH on the samples $X = [x_{-L+1}, \cdots, x_0]$, $Y = [y_{-L+1}, \cdots, y_0]$ as the initialization ($\widehat{U}_0$, $\widehat{V}_0$) for the online updates, assuming the online process starts from timestamp $t = 1$. We also use an alternating method to track (U_t, V_t) with the following definition of projections of x_t and y_t into the latent space:

$$z_t \triangleq U_t^\top x_t, \qquad w_t \triangleq V_t^\top y_t.$$

The online algorithm LSTH consists of an initialization via bLSTH and iterative online updates of U and V, as presented in Algorithm 3.

Algorithm 2. Updating U for `bLSTH`

Input: orthonormal U, scalar σ, cross-covariance matrix G
auto-covariance matrix C_x
Output: U_{updated}
$k \leftarrow 0,\ U[k] \leftarrow U$
repeat
$k \leftarrow k+1$
compute `SVD`: $P\Sigma Q^\top = G - (1-\sigma)(C_x - a\,\mathbf{I})U[k-1]$ (9)
$U[k] \leftarrow PQ^\top$
until $U[k]$ converged or k is large enough
$U_{\text{updated}} \leftarrow U[k]$

Online Tracking of U_t. Upon arrival of new data (x_t, y_t) at t, we use $\widehat{V}_{t-1}$ to estimate the projection of y_t at t as follows:

$$\widehat{w}_t = \widehat{V}_{t-1}^\top y_t. \tag{10}$$

Substituting the $\widehat{w}_t$ into Equation (4), we will see that the objective function of U is of the same form as (5), except that the historical x_t are downweighed. Therefore it can be minimized via Algorithm 2 with the only modification that G in Equation (9) is replaced by $\sum_{k=0}^{t-1} \alpha^k x_{t-k}\widehat{w}_{t-k}^\top$, and C_x is replaced by $\sum_{k=0}^{t-1} \alpha^k x_{t-k}x_{t-k}^\top$. Both of these two summations can be incrementally updated.

Online Tracking of V_t. Given $\widehat{U}_t$ solved as in Section 3.2, we use $\widehat{U}_t$ to estimate z_t at current timestamp t as follows

$$\widehat{z}_t = \widehat{U}_t^\top x_t. \tag{11}$$

Substituting $\widehat{z}_t$ into Equation (4), we can get the following objective function w.r.t V,

$$F_V(V;t) = \sum_{k=0}^{t-1}\left[\frac{\alpha^k}{2}\left\|V^\top y_{t-k} - \widehat{z}_{t-k}\right\|_2^2\right] + \lambda\|V\|_{1,2}. \tag{12}$$

For the above problem, we derive a Stochastic Coordinate Descent (SCD) method with a similar spirit as [9]. The SCD admits a row-wise updating of $\widehat{V}_t$, details can be found in Equation (13) in Algorithm 3.

3.3 Complexity Analysis

The complexity of `LSTH` is $O(c \cdot D_x^2 d + D_y^2 d)$, where $c \cdot D_x^2 d$ is due to the `SVD` step in Equation (9) and c is the number of iterations in majorization minimization for U ($c = 1$ suffices in practice). Efficient algorithms for computing the `SVD` of a sequentially updated matrix [2] can be applied to reduce the complexity. $D_y^2 \cdot d$ is

due to the coordinate descent algorithm on V, for which further acceleration can be achieved via active set tricks. Our experiments show that LSTH is sufficiently fast for real applications, for example, 20 ms for the XRMB dataset (sampling interval: 25 ms/sample). The experimental details will be presented in Section 5. To reduce the complexity of LSTH is very important and it is left for future exploration for now.

4 Application: Anomaly Detection

The basic idea of our anomaly detection method is to monitor $\|U^\top x_t - V^\top y_t\|^2 + \sigma\|(\mathbf{I} - UU^\top)x_t\|^2$. We define the *a priori* error:

$$\xi_t \triangleq \|\widehat{U}_{t-1}^\top x_t - \widehat{V}_{t-1}^\top y_t\|^2 + \sigma\|x_t - \widehat{U}_{t-1}\widehat{U}_{t-1}^\top x_t\|^2, \tag{14}$$

and use ξ_t as the detection statistic. An anomaly is claimed only when $p(x_t, y_t)$ appears to be significantly small, corresponding to ξ_t being significantly large. We maintain a sliding window over ξ_t with the mean μ_t and standard deviation ν_t within the window. When the new (x_{t+1}, y_{t+1}) arrives, we compare its ξ_{t+1}

Algorithm 3. The Online Algorithm LSTH

Parameters: d, α, λ, σ
Input: data stream: $\cdots, (x_0, y_0), \cdots, (x_t, y_t), \cdots$
Obtain $\widehat{U}_0$ and $\widehat{V}_0$ by Algorithm 1
for $t = 1, 2, \ldots$ **do**
  //update $\widehat{U}_t$
  $\widehat{w}_t \leftarrow \widehat{V}_{t-1}^\top y_t$
  $G_t \leftarrow \alpha G_{t-1} + x_t \widehat{w}_t^\top$ /*$G_0 = \sum_{\tau=-L+1}^{0} x_\tau w_\tau^\top$*/
  $C_{x,t} \leftarrow \alpha C_{x,t-1} + x_t x_t^\top$ /*$C_{x,0} = \sum_{\tau=-L+1}^{0} x_\tau x_\tau^\top$*/
  get $\widehat{U}_t$ via Algorithm 2 with ($\widehat{U}_{t-1}$, σ, G_t, $C_{x,t}$) as input
  //update $\widehat{V}_t$
  $\widehat{z}_t \leftarrow \widehat{U}_t^\top x_t$
  $H_t \leftarrow \alpha H_{t-1} + y_t \widehat{z}_t^\top$ /*$H_0 = \sum_{\tau=-L+1}^{0} y_\tau z_\tau^\top$*/
  $C_{y,t} \leftarrow \alpha C_{y,t-1} + y_t y_t^\top$ /*$C_{y,0} = \sum_{\tau=-L+1}^{0} y_\tau y_\tau^\top$*/
  for $i = 1, 2, \ldots, D_y$ **do**
    Calculate the i-th row of $\widehat{V}_t$:

$$\widehat{v}_{t,i}^\top = \frac{S(\|h_{t,i}^\top - \sum_{j\neq i} v_{t-1,j}^\top C_{y,t,i,j}\|, \lambda)}{C_{y,t,i,i}} \times \frac{h_{t,i}^\top - \sum_{j\neq i} v_{t-1,j}^\top C_{y,t,i,j}}{\|h_{t,i}^\top - \sum_{j\neq i} v_{t-1,j}^\top C_{y,t,i,j}\|}, \tag{13}$$

    where $S(\cdot, \lambda)$ is the soft thresholding function with parameter λ.
  end for
end for

with a threshold $b_t = \mu_t + \gamma\nu_t$, where $\gamma > 0$ indicates the effect of variance. Once ξ_{t+1} exceeds the threshold, an anomaly is claimed.

Additional care need to be taken for the claimed anomalous data points. In specific, if the anomaly behaves as a sudden outlier after which the data stream goes back to normal state, then the anomalous data point should be excluded for model updating. The other case is that the anomaly is in fact the start of a different stage in the data stream, then the anomalous data point should be included in model updating. These two cases will be addressed in synthetic and real data experiments respectively.

5 Experiments

In this section, we conduct comparative experiments to demonstrate the performance of `LSTH` in tracking the latent space for anomaly detection. All types of tracking methods as well as their corresponding anomaly detection statistics are summarized in Table 1.

Table 1. Latent space tracking methods and corresponding detection statistics

method	detection statistics	semantics
`LSTH`	$\xi_t = \|\widehat{U}_{t-1}^\top x_t - \widehat{V}_{t-1}^\top y_t\|^2 + \sigma\|x_t - \widehat{U}_{t-1}\widehat{U}_{t-1}^\top x_t\|^2$	latent discrepancy and projection residual
(online) `CCA`	$\delta_t = \widehat{C}_{x,t-1} r_{x,t}/D_x + r_{y,t}^\top \widehat{C}_{y,t-1} r_{y,t}/D_y$ where $r_{x,t} = (\widehat{C}_{x,t-1}^{-1} - \widehat{U}_{t-1}\widehat{U}_{t-1}^\top)x_t$ and $r_{y,t} = (\widehat{C}_{y,t-1}^{-1} - \widehat{V}_{t-1}\widehat{V}_{t-1}^\top)y_t$	projection residual onto Generalized Stiefel manifold [19]
(online) `PCAx`	$\epsilon_{x,t} = \|(\mathbf{I} - \widehat{U}_{t-1}\widehat{U}_{t-1}^\top)x_t\|^2$	projection residual onto individual or joint signal subspace [4]
`PCAy`	$\epsilon_{y,t} = \|(\mathbf{I} - \widehat{U}_{t-1}\widehat{U}_{t-1}^\top)y_t\|^2$	
`PCAxy`	$\epsilon_{xy,t} = \|(\mathbf{I} - \widehat{U}_{t-1}\widehat{U}_{t-1}^\top)[x_t; y_t]\|^2$	

5.1 Experiments on Synthetic Data

We generated a synthetic dataset with continuous data $x_t \in \mathbb{R}^{500}$ and sparse, discrete and non-negative data $y_t \in \mathbb{R}^{1000}$. The x_t's are generated via a linear model $x_t = A\theta_t + n_t,\ t = 1, \ldots, 10500.$ where $A \in \mathbb{R}^{500\times 10}$, $\theta_t \in \mathbb{R}^{10}$ and n_t is white Gaussian noise. The y_t's are generated as of dimension 1000. The first 50 features of y_t's are relevant to the underlying system, generated via $B\theta_t + m_t,\ t = 1, \ldots, 10500$, where $B \in \mathbb{R}^{50\times 10}$ and m_t is white Gaussian noise. The rest 950 dimensions are padded as noise. We introduced sparsity into y_t by randomly setting half of its values to zero. In the end we round the y_t to non-negative integers. In this way, y_t is analogous to the real-world documents in bag-of-words representation. In this generated dataset, we introduced three types of anomalies, all of them are sudden outliers.

Table 2. Synthetic dataset: AUC and parameters

method	Type-1		Type-2		Type-3	
	AUC	parameters	AUC	parameters	AUC	parameters
LSTH	**0.863/0.860**	10,10,1,10	**0.995/0.993**	10,10,1,10	**0.984/0.979**	10,20,1,0
CCA	0.848/0.859	10	0.020/0.019	10	0.971/0.950	500
PCAx	0.500/0.525	10, 1	0.015/0.018	10, 1	0.013/0.016	10, 1
PCAxy	0.644/0.662	20, 1	0.744/0.730	20, 1	0.977/0.971	20, 1
PCAy	0.298/0.365	10, 1	0.015/0.015	10, 1	0.977/0.960	20, 1

The parameters for LSTH are d (dimension of the latent space), λ, α and σ, respectively, The parameter for CCA is d. The parameters for PCAx, PCAxy and PCAy are d and the forgetting factor, respectively. AUC of the precision-recall plot is used for evaluation; the larger the AUC value is, the better the performance is. The values under AUC column (i.e., x/y) are the performance on training and testing set, respectively. **Bold** numbers correspond to the best performance for each anomaly type among all the methods.

Type-1 anomaly: at $t = 500, 600, \ldots, 10400$, x_t is distorted to $\tilde{x}_t = \tilde{A}\theta_t + n_t$, where $\tilde{A}$ is identical to A except that one row of $\tilde{A}$ is randomly re-drawn from $\mathcal{N}(0, 1)$. At the same timestamps when A is distorted, B in generating y_t is also distorted to $\tilde{B}$ by randomly re-drawing 5 of its rows from $\mathcal{N}(1, 0.3^2)$. This corresponds to the scenario when both x_t and y_t behave anomalously at same time.

Type-2 anomaly: at $t = 500, 600, \ldots, 10400$, only x_t is distorted to $\tilde{x}_t = A\tilde{\theta}_t + n_t$ with $\tilde{\theta}_t \sim \mathcal{N}(3.5, 1)$, that is, the latent variable θ_t is distorted. In this way, a discrepancy is introduced between the latencies of $\tilde{x}_t$ and y_t. This corresponds to the scenario when x_t has anomalies but y_t behaves normally.

Type-3 anomaly: At $t = 500, 600, \ldots, 10400$, three relevant features and three among the rest 950 features of y_t are exchanged. This corresponds to the scenario when some relevant features in y_t are changed while x_t remains normal.

Experimental Results on Synthetic Data. We compare all methods in Table 1 for anomaly detection task. For all the methods, the first 100 samples are used for initialization. The γ in computing detection threshold is varied to produce a full precision-recall plot. The parameters are selected as the ones that maximize the Area Under Curve (AUC) of the precision-recall plot on a training set generated separately from the same data generation protocol. Results are presented in Table 2. For the three types of anomalies, LSTH consistently achieves the

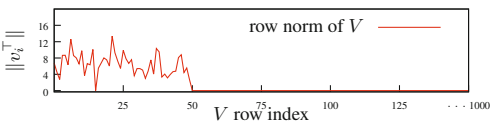

Fig. 1. Feature selection effects of LSTH

Table 3. XRMB results

method	AUC	parameters
LSTH	**0.342**	20,300,0.95,1000
CCA	0.045	30
PCAx	0.035	20, 1e-5
PCAxy	0.035	30, 1e-5
PCAy	0.033	30, 1e-5

The parameters for each method are same as those in Table 2 in paper.

best detection performance. CCA is competitive for Type-1 and Type-3 anomalies but completely fails for Type-2, due to the fact that its detection statistic cannot capture the changes in the signal/latent space. The failure of PCAx on Type-2 has a same reason as that of CCA on Type-2. On average, PCA based methods perform worst among all the methods except for Type-3. However, by joining two data sources properly, PCAxy is able to detect the change of the "joint" subspace so as to achieve better performance than PCAx and PCAy.

Figure 1 shows the norm of each row of the learned V, after all the updates of LSTH at $t = 10500$. For the relevant $(i = 1, \ldots, 50)$ features in y_t, $\|v_i^\top\|$ are non-zero. For the irrelevant features $(i > 50)$, $\|v_i^\top\|$ are zero or very small. This demonstrates that LSTH can successfully identify the relevant features via the mixed norm on V.

5.2 Experimental Results on Real Data: XRMB

XRMB [16] contains synchronous 273-dim MFCC and 112-dim articulatory information of length 51K. Each timestamp has a label indicating which word it corresponds to. Details on the data are available in [1]. Speech segmentation has attracted lots of attention for treating related diseases [15]. The task in our experiment is to detect the boundary of words from acoustic and articulatory features. During each segment, a tracking algorithm, e.g., LSTH, gradually learns the underlying latent subspace. Upon arrival of a new segment, the underlying latent space has a sudden change. This event may induce a drastic change of the detection statistics provided by the tracking algorithms, and therefore is considered as an anomaly. In this case, the claimed anomalous data point should be incorporated in learning the new latent space in the new segment.

When applying LSTH, we assign to x_t the articulatory features with highly correlated dimensions [3]. And y_t is designated as the MFCC, which is redundant and sparse filtering has been shown necessary for feature selection [11]. We randomly select 1000 frames for parameter tuning for all the methods, and use the tuned parameters for testing on the rest of the frames. Figure 2 shows the detection statistics of all methods on the parameter tuning dataset. Out of 25 words within the 1000 frames, LSTH is able to identify 15 words with clear and strong spikes in the detection statistics. After each alarm of anomaly (start of a new segment), it quickly adapts to the new latent space in the new segment. PCA based methods only show weak spikes. CCA fails in this case, as the conclusion in [1]. Based on their results, kernel CCA should be a better approach on this dataset than CCA. However, there is not a meaningful detection statistic for kernel CCA, so we leave this approach for later research.

We then applied all the methods on the rest of the data with their optimal parameters tuned on the training set. The parameters and the performance of different methods are presented in Table 3. LSTH has an AUC value 0.342 (note that a random guess would give an AUC of $412/51000 = 0.008$) and it is the only method that can detect the boundaries of the words from XRMB dataset. All the other methods fail with AUC values smaller than 0.05.

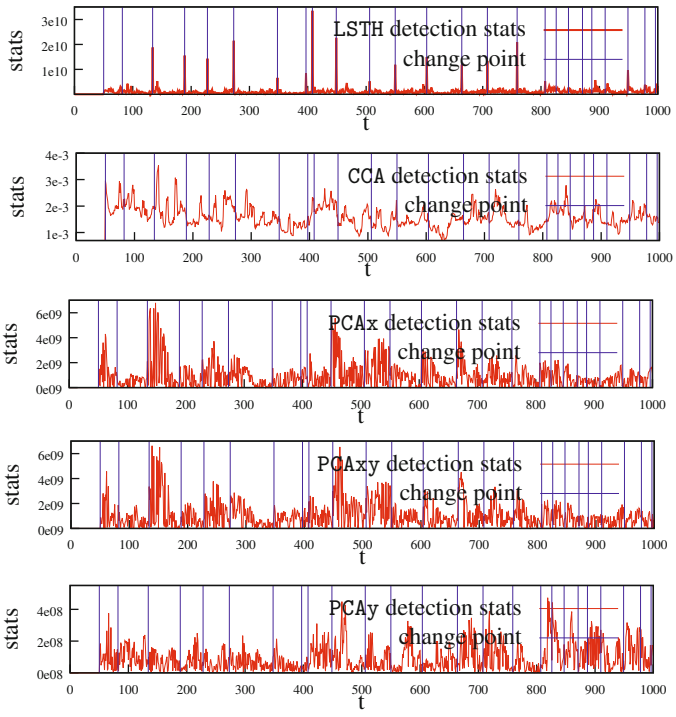

Fig. 2. Detection statistics on XRMB training data

6 Conclusions and Discussions

We developed LSTH, a latent space tracking method for heterogeneous streaming data. Under the assumption that anomalies significantly deviate from the latent space, we further designed an anomaly detection method based on LSTH. Experimental results demonstrate that LSTH's detection statistics outperform the other state-of-the-art in identifying anomalies. Therefore LSTH better characterizes the latent structure of heterogeneous data than does the other methods. Future work on LSTH includes non-linear mapping into the latent space via kernelization, online supervised learning in the latent space, and extending to cases with more than two views of a system.

References

1. Andrew, G., Arora, R., Bilmes, J., Livescu, K.: Deep canonical correlation analysis. In: Proceedings of the 30th International Conference on Machine Learning (ICML 2013), vol. 28, pp. 1247–1255 (2013)
2. Brand, M.: Fast low-rank modifications of the thin singular value decomposition. Linear algebra and its applications **415**(1), 20–30 (2006)

3. Cai, J., Laprie, Y., Busset, J., Hirsch, F.: Articulatory modeling based on semi-polar coordinates and guided pca technique. In: 10th Annual Conference of the International Speech Communication Association-INTERSPEECH (2009)
4. Chi, Y., Eldar, Y., Calderbank, R.: Petrels: Parallel subspace estimation and tracking by recursive least squares from partial observations. IEEE Transactions on Signal Processing **61**(23), 5947–5959 (2013)
5. Hotelling, H.: Canonical correlation analysis. Journal of Educational Psychology (1935)
6. Jin, X., Zhuang, F., Xiong, H., Du, C., Luo, P., He, Q.: Multi-task multi-view learning for heterogeneous tasks. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 441–450 (2014)
7. Jolliffe, I.T.: Principal Component Analysis. John Wiley & Sons, Ltd. (2005)
8. Kiers, H.A.L.: Majorization as a tool for optimizing a class of matrix functions. Psychometrika **55**(3), 417–428 (1990)
9. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: The 26th Annual International Conference on Machine Learning, pp. 689–696. ACM (2009)
10. Mardia, K.V., Kent, J.T., Bibby, J.M.: Multivariate Analysis. Academic Press, New York (1979)
11. Ngiam, J., Koh, P.W., Chen, Z., Bhaskar, S., Ng, A.: Sparse filtering. In: Advances in Neural Information Processing Systems, pp. 1125–1133 (2011)
12. dos Santos Teixeira, P.H., Milidiú, R.L.: Data stream anomaly detection through principal subspace tracking. In: Proceedings of the 2010 ACM Symposium on Applied Computing (2010)
13. Shi, X., Liu, Q., Fan, W., Yu, P.S., Zhu, R.: Transfer learning on heterogenous feature spaces via spectral transformation. In: 2010 IEEE 10th International Conference on Data Mining (ICDM), pp. 1049–1054 (2010)
14. Tang, X., Yang, C., Zhou, J.: Stock price forecasting by combining news mining and time series analysis. In: IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, vol. 1, pp. 279–282 (2009)
15. Todder, D., Avissar, S., Schreiber, G.: Non-linear dynamic analysis of inter-word time intervals in psychotic speech. IEEE Journal of Translational Engineering in Health and Medicine **1**, 1–7 (2013)
16. Westbury, J.R.: X-ray microbeam speech production database user's handbook. Tech. rep., University of Wisconsin, Madison (1994)
17. Xie, Y., Huang, J., Willett, R.: Change-point detection for high-dimensional time series with missing data. IEEE Journal of Selected Topics in Signal Processing **7**(1), 12–27 (2013)
18. Xu, C., Tao, D., Xu, C.: A survey on multi-view learning. ArXiv e-prints (April 2013)
19. Yger, F., Berar, M., Gasso, G., Rakotomamonjy, A.: Adaptive canonical correlation analysis based on matrix manifolds. In: Proceedings of the 29th International Conference on Machine Learning, pp. 1071–1078 (2012)