High-dimensional Time Series Clustering via Cross-Predictability

Dezhi Hong
University of Virginia

Quanquan Gu
University of Virginia

Kamin Whitehouse
University of Virginia

Abstract

The key to time series clustering is how to characterize the similarity between any two time series. In this paper, we explore a new similarity metric called "cross-predictability": the degree to which a future value in each time series is predicted by past values of the others. However, it is challenging to estimate such cross-predictability among time series in the high-dimensional regime, where the number of time series is much larger than the length of each time series. We address this challenge with a sparsity assumption: only time series in the same cluster have significant cross-predictability with each other. We demonstrate that this approach is computationally attractive, and provide a theoretical proof that the proposed algorithm will identify the correct clustering structure with high probability under certain conditions. To the best of our knowledge, this is the first practical high-dimensional time series clustering algorithm with a provable guarantee. We evaluate with experiments on both synthetic data and real-world data, and results indicate that our method can achieve more than 80% clustering accuracy on real-world data, which is 20% higher than the state-of-art baselines.

1 INTRODUCTION

The proliferation of cheap, ubiquitous sensing infrastructure has enabled continuous monitoring of the world, and many expect the Internet of Things to have over 25 billion devices by 2020 [12]. In this paradigm, time series data will often be high-dimensional: the number of time series d (i.e., number of sensors) will be much larger than the length of each time series T. Time series clustering often serves as an important first step for many applications and poses long-standing challenges. In this paper, we explore the challenge of time series clustering in the high-dimensional regime.

The key to time series clustering is how to characterize the similarity between any two time series. In the past several decades, various metrics for measuring the similarity/distance between time series have been investigated [8, 13, 6, 11, 18, 26, 10, 20, 34, 3], and so on. Hidden Markov Models [29, 24] have also been utilized to derive the distances between time series for clustering. Recently, a few new metrics [27, 19] to measure the similarity between time series have been proposed and applied to cluster brain-computer interface data and motion capture data. However, all the aforementioned work either did not provide theoretical guarantees for their methods, or only considered scenarios where the number of observations per time series T far exceeds the number of time series d.

In this paper, we explore a new similarity metric called "cross-predictability": the degree to which a future value in each time series is predicted by past values of the others. This metric captures causal relationships between time series, such as seasonal or diurnal effects on multiple environment sensors, market effects on multiple stock prices, and so on. However, it is challenging to estimate such cross-predictability among time series in the high-dimensional setting where $d > T$: a conventional regression task, for example, would have d variables and T equations, which is under-constrained. Intuitively, only time series in the same cluster would have significant cross-predictability for each other, thus yielding sparse relationships that are indicative of the cluster structure. Consequently, we propose to estimate cross-predictability by imposing a sparsity assumption on the cross-predictability matrix, i.e., that only time series in the same cluster have significant cross-predictability with each other. To do this, we propose a new regularized Dantzig selector, which is a variant of standard Dantzig selector [7], to estimate the similarity among the time series. We demonstrate that this approach is computationally attractive because it

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54. Copyright 2017 by the author(s).

involves solving d regularized Dantzig selectors that can be optimized by alternating direction method of multipliers (ADMM) [4] in parallel.

Additionally, we provide a theoretical proof that the proposed algorithm will identify the correct clustering structure with high probability, if two conditions hold: 1) the individual time series themselves can be modeled with an autoregressive model [14], and 2) the transition matrix for the vector autoregressive model is block diagonal, i.e., that it is actually possible to create clusters such that time series in the same cluster are cross-predictive while those in different clusters are not.

To the best of our knowledge, this is the first practical high-dimensional time series clustering algorithm with a provable guarantee. It is worth noting that the proposed algorithm can be generally applied to cluster any high dimensional times series, regardless of the underlying data distribution. We make the autoregressive model assumption solely for the purpose of providing the theoretical guarantees for our method.

To demonstrate the effectiveness of our method, we conduct experiments on a real-world data set of sensor time series as well as simulations with synthetic data. Our method can achieve more than 80% clustering accuracy on the real-world data set, which is 20% higher than the state-of-art baselines.

Notations We compile here some standard notations used throughout the paper. In this paper, we use lowercase letters $x, y, \dots$ to denote scalars, bold lowercase letters $\mathbf{x}, \mathbf{y}, \dots$ for vectors, and bold uppercase letters $\mathbf{X}, \mathbf{Y}, \dots$ for matrices. We denote random vectors by $\boldsymbol{X}, \boldsymbol{Y}$. We denote the (i, j) entry of a matrix as M_{ij}, and use $\mathbf{M}_{i*}$ to index the i-th row of a matrix (likewise, $\mathbf{M}_{*j}$ for the j-th column). We also use $\mathbf{M}_{S,T}$ to represent a submatrix of $\mathbf{M}$ with its rows indexed by the indices in set S and columns indexed by T. In addition, we write S^c to denote the complement of a set S. For any matrix $\mathbf{M}$, $\mathcal{P}(\mathbf{M})$ represents the symmetric convex hull of its columns, i.e., $\mathcal{P}(\mathbf{M}) = \text{conv}(\pm\mathbf{X})$. For any matrices $\mathbf{M}_1, \mathbf{M}_2, \dots \mathbf{M}_k$, we denote a block diagonal matrix by $\text{diag}(\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_k)$ such that the k-th diagonal block is $\mathbf{M}_k$. Throughout the paper, we will use vector norm ℓ_q for $0 < q < \infty$ and ℓ_∞ of $\mathbf{v}$ defined as $\|\mathbf{v}\|_q = \Big(\sum_i |v_i|^q\Big)^{1/q}$, $\|\mathbf{v}\|_{\infty,\infty} = \max_i |v_i|$, and matrix norm ℓ_q, element-wise ℓ_∞ and ℓ_F of $\mathbf{M}$ as $\|\mathbf{M}\|_q = \max_{\|\mathbf{v}\|_q=1} \|\mathbf{M}\mathbf{v}\|_q$, $\|\mathbf{M}\|_{\infty,\infty} = \max_{ij} |M_{ij}|$, $\|\mathbf{M}\|_F = \Big(\sum_{i,j} |M_{ij}|^2\Big)^{1/2}$.

2 RELATED WORK

There has been a substantial body of work on time series clustering, and in this section we briefly overview two related categories: clustering based on similarity and subspace clustering.

Similarity/Distance-based Time Series Clustering A wide range of classical similarity/distance metrics have been developed and studied [8], including Pearson's correlation coefficient [13], cosine similarity [6], autocorrelation [11], dynamic time warping [18, 26], Euclidean Distance [10], edit distance [20], distance metric learning [34, 3], and so on. Studies have also shown that time series can be modeled as generated from Hidden Markov Models [29, 24], and the estimated weight for each mixture can be used to cluster the time series. Recently, Ryabko et al. [27] considered brain-computer interface data for which independence assumptions do not hold, and for clustering they proposed a new distance metric to measure the similarity between two time series distributions. Khaleghi et al. [19] formulated a novel metric to quantify the distance between time series and proved the consistency of k-means for clustering processes according only to their distributions. However, in the aforementioned studies, they either did not provide theoretical analysis of the performance or only handled settings where the number of time series d is smaller than the number of observations T. Different from the above similarity or distance metrics, we define the similarity between time series from a new perspective - time series are clustered based on how much they can be predicted by each other.

Subspace Clustering (SC) Another relevant line of research on high-dimensional data analysis is subspace clustering [9], where the assumption is that data lie on the union of multiple lower-dimensional linear spaces and data points can be clustered into the subspace they belong to. SC has been widely applied to face images clustering [1], social graphs [17] and so on. Recently, extensions to handling noisy data [21, 31, 33] and data with irrelevant features [25] have been studied as well. SC achieves the state-of-art performance while enjoying rigorous theoretical guarantees. The key difference between SC and our method is two-fold: first, SC assumes data lie on different subspaces and even data in the same subspace are independent and identically distributed (i.i.d.), while we assume the time series follow a VAR model and are dependent for the ones in the same cluster; second, SC mathematically solves for each data point a linear regression problem with all other data points being the candidate, while in contrast, our study solves the regression problem to estimate the prediction weights between observations from different time stamps using all the time series.

3 METHODOLOGY

3.1 The VAR Model

Our algorithm is motivated by the autoregressive model, and the later-on theoretical guarantee for our algorithm also relies on the autoregressive model assumption, so we briefly review the stationary first-order vector autoregressive model with Gaussian noise here. Let random vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_T$ be from a stationary process $(\boldsymbol{X}_t)_{t=-\infty}^{\infty}$, and we further define $\mathbf{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_t, \ldots \boldsymbol{X}_T]^\top \in \mathbb{R}^{T \times d}$, where $\boldsymbol{X}_t = (x_1, \ldots, x_d)^\top \in \mathbb{R}^d$ is a d-dimensional vector and each column of $\mathbf{X}$ is a one-dimensional time series with T samples. In particular, we assume each $\boldsymbol{X}_t$ can be modeled by a first-order vector autoregressive model:

$$\boldsymbol{X}_{t+1} = \mathbf{A}\boldsymbol{X}_t + \boldsymbol{Z}_t, \text{ for } t = 1, 2, \ldots, T-1. \tag{3.1}$$

To secure the above process to be stationary, the transition matrix $\mathbf{A}$ must have bounded spectral norm, i.e., $\|\mathbf{A}\|_2 < 1$. We also assume $\boldsymbol{Z}_t \sim N(0, \boldsymbol{\Psi})$ is i.i.d. additive noise independent of $\boldsymbol{X}_t$, and $\boldsymbol{X}_t$ has zero mean and a covariance matrix $\boldsymbol{\Sigma}$, i.e., $\boldsymbol{X}_t \sim N(0, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = \mathbb{E}[\boldsymbol{X}_t \boldsymbol{X}_t^\top]$ is the autocovariance matrix. In addition, we have the lag-1 autocovariance matrix as $\boldsymbol{\Sigma}_1 = \mathbb{E}[\boldsymbol{X}_t \boldsymbol{X}_{t+1}^\top]$. Since $(\mathbf{X}_t)_{t=-\infty}^{\infty}$ is stationary, it is easy to observe that the covariance matrix $\boldsymbol{\Sigma}$ depends on $\mathbf{A}$ and $\boldsymbol{\Psi}$, i.e., $\boldsymbol{\Sigma} = \mathbf{A}^\top \boldsymbol{\Sigma} \mathbf{A} + \boldsymbol{\Psi}$, and we further have:

$$\boldsymbol{\Sigma}\mathbf{A}^\top = \boldsymbol{\Sigma}_1. \tag{3.2}$$

Essentially, the zero and nonzero entries in the transition matrix $\mathbf{A}$ directly reflect the Granger non-causalities and causalities with regard to the stochastic time series. In other words, a nonzero entry A_{ij} implies that the j-th time series is predictive for the i-th time series, with the magnitude $|A_{ij}|$ indicating how much the predictive power is. The new similarity metric in our clustering algorithm is built upon such cross-predictive relationship between time series. Now we set to introduce the clustering algorithm.

3.2 The Proposed Clustering Algorithm

Our algorithm first estimates the cross-predictability among the time series, and then identifies the clustering structure based on the estimated relationship. To introduce our proposed algorithm, we need the following notations: $\mathbf{X}_{\mathcal{S}} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{T-1}]^\top \in \mathbb{R}^{(T-1) \times d}$, $\mathbf{X}_{\mathcal{T}} = [\boldsymbol{X}_2, \ldots, \boldsymbol{X}_T]^\top \in \mathbb{R}^{(T-1) \times d}$, $\hat{\boldsymbol{\Sigma}} = \mathbf{X}_{\mathcal{S}}^\top \mathbf{X}_{\mathcal{S}}/(T-1)$, and $\hat{\boldsymbol{\Sigma}}_1 = \mathbf{X}_{\mathcal{S}}^\top \mathbf{X}_{\mathcal{T}}/(T-1)$. Inspired by the relationship in Eq. (3.2), our main idea is to estimate $\mathbf{A}$ based on the relationship between $\mathbf{A}$ and the autocovariance and lag-1 autocovariance matrices. This motivates the following Dantzig selector type estimator [15],

$$\hat{\mathbf{A}} = \arg\min_{\mathbf{A}} \|\mathbf{A}\|_1 \quad \text{subject to} \quad \|\hat{\boldsymbol{\Sigma}}\mathbf{A}^\top - \hat{\boldsymbol{\Sigma}}_1\|_{\infty,\infty} \le \mu, \tag{3.3}$$

where $\mu > 0$ is a tuning parameter. Since each row of $\mathbf{A}$ is independent, the above optimization problem can be decomposed into d independent sub-problems and solved individually as follows:

$$\hat{\boldsymbol{\beta}}_i = \arg\min_{\boldsymbol{\beta}_i} \|\boldsymbol{\beta}_i\|_1 \quad \text{subject to} \quad \|\hat{\boldsymbol{\Sigma}}\boldsymbol{\beta}_i - \hat{\boldsymbol{\gamma}}_i\|_{\infty,\infty} \le \mu, \tag{3.4}$$

where $\hat{\boldsymbol{\gamma}}_i = (\hat{\boldsymbol{\Sigma}}_1)_{*i} = \mathbf{X}_{\mathcal{S}}^\top (\mathbf{X}_{\mathcal{T}})_{*i}/(T-1)$, i.e., $\hat{\boldsymbol{\gamma}}_i$ is the i-th column of $\hat{\boldsymbol{\Sigma}}_1$, and $\hat{\mathbf{A}} = \left[\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_d\right]^\top \in \mathbb{R}^{d \times d}$ with each $\hat{\boldsymbol{\beta}}_i \in \mathbb{R}^d$. Therefore, the $\hat{\boldsymbol{\beta}}_i$ in (3.4) is an estimation of the i-th row of the transition matrix $\mathbf{A}$. Furthermore, for each $\mu > 0$, there always exists a $\lambda > 0$ such that (3.4) is equivalent to the following regularized Dantzig selector type estimator:

$$\hat{\boldsymbol{\beta}}_i = \arg\min_{\boldsymbol{\beta}_i} \ \lambda \|\hat{\boldsymbol{\Sigma}}\boldsymbol{\beta}_i - \hat{\boldsymbol{\gamma}}_i\|_{\infty,\infty} + \|\boldsymbol{\beta}_i\|_1, \tag{3.5}$$

where λ is a regularization parameter to determine the sparsity of the estimation. (3.4) can be solved by alternating direction method of multipliers (ADMM) [4]. All the d optimization problems can be solved in parallel, thus computationally efficient.

After solving the problem in (3.5), we construct an affinity matrix $\mathbf{W}$ based on $\hat{\mathbf{A}} = \left[\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_d\right]^\top$ by symmetrization, and compute the corresponding Laplacian to perform standard spectral clustering [23, 28] to recover the clusters in the input time series. The procedure is summarized in Algorithm 1.

3.3 Discussion

At first glance, the regularized Dantzig selector in (3.5) and Lasso appear similar. However, different from Lasso, the "input" of the regression problem in Eq (3.5) is the lag-0 covariance matrix, and the "response" is the lag-1 covariance matrix. Here the lag-one covariance matrix encodes and includes into consideration the first-order temporal information, which is missing in conventional similarity metrics such as correlation. Additionally, different from the Lasso-based estimation procedure [2], which penalizes the square loss, the regularized Dantzig selector estimator penalizes the $\ell_{\infty,\infty}$ loss.

It is also worth noting that Algorithm 1 shares a similar high level idea as the subspace clustering (SC) algorithm [30, 33, 31], but the key difference is that SC considers the relationship between each data point and all the other points, while in contrast, our estimator solves the regression problem to estimate the predictive relationship between the observations from time

Algorithm 1: Time Series Clustering Algorithm

```
Input: Time series $\mathbf{X} = [\boldsymbol{X}_1, \dots \boldsymbol{X}_T]^\top \in \mathbb{R}^{T \times d}$,
$\mathbf{X}_\mathcal{S} = [\boldsymbol{X}_1, \dots, \boldsymbol{X}_{T-1}]^\top \in \mathbb{R}^{(T-1) \times d}$,
$\mathbf{X}_\mathcal{T} = [\boldsymbol{X}_2, \dots, \boldsymbol{X}_T]^\top \in \mathbb{R}^{(T-1) \times d}$,
$\hat{\mathbf{\Sigma}} = \mathbf{X}_\mathcal{S}^\top \mathbf{X}_\mathcal{S} / (T-1)$, and $\hat{\boldsymbol{\gamma}}_i = \mathbf{X}_\mathcal{S}^\top (\mathbf{X}_\mathcal{T})_{*i} / (T-1)$
Output: Cluster membership of each time series $\boldsymbol{Y}$
1. Solve for each $i = 1, \dots, d$:
       $\hat{\boldsymbol{\beta}}_i = \arg\min_{\boldsymbol{\beta}_i} \ \lambda \|\hat{\mathbf{\Sigma}} \boldsymbol{\beta}_i - \hat{\boldsymbol{\gamma}}_i\|_{\infty,\infty} + \|\boldsymbol{\beta}_i\|_1$;
2. Set $\hat{\mathbf{A}} = \left[\hat{\boldsymbol{\beta}}_1, \dots, \hat{\boldsymbol{\beta}}_d\right]^\top$;
3. Construct the affinity graph $G$ with nodes being
   the $d$ time series in $\mathbf{X}$, and edge weights given by the
   matrix $\mathbf{W} = |\hat{\mathbf{A}}| + |\hat{\mathbf{A}}|^\top$;
4. Compute the unnormalized Laplacian $\mathbf{L} = \mathbf{M} - \mathbf{W}$
   of graph $G$, with $\mathbf{M} = \mathrm{diag}(m_1, m_2, \dots, m_d)$ and
   $m_i = \sum_{j=1}^d W_{ij}$;
5. Compute the first $k$ eigenvectors $\sigma_1, \dots, \sigma_k$ of $\mathbf{L}$
   and let $\mathbf{V} \in \mathbb{R}^{d \times k}$ be the matrix containing as
   columns the first $k$ eigenvectors;
6. Cluster time series $\mathbf{x}'_i \in \mathbb{R}^k$, as the $i$-th row of $\mathbf{V}$,
   with the $k$-means algorithm into clusters
   $\mathcal{C}_1, \dots, \mathcal{C}_l, \dots, \mathcal{C}_k$.
```

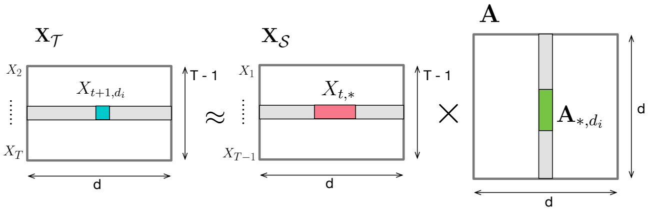


Figure 1: Illustration of the Proposed Regularized Dantzig Selector: it solves the regression problem to estimate the predictive relationship between the observations from time $t+1$ and time t considering all the time series.

$t+1$ and the observations from time t considering all the time series, as illustrated in Figure 1. Another fundamental difference here is, SC assumes data are i.i.d. and lie on different subspaces, while here the time series data are obviously dependent. This poses a big challenge to the theoretical analysis of our algorithm.

4 MAIN RESULTS

In this section, we state our main theory - a provable guarantee for successfully recovering the underlying clustering structure of the input time series. We first introduce some necessary definitions for understanding our main theorem.

4.1 Preliminaries

To define the clusters among time series $\mathbf{X}$ under the context of VAR model, we assume $\mathbf{A} = \mathrm{diag}(\mathbf{A}_1, \dots, \mathbf{A}_l, \dots, \mathbf{A}_k)$ to be block diagonal, where $\mathbf{A}_l \in \mathbb{R}^{d_l \times d_l}$ and the number of time series d satisfies $d = \sum_{l=1}^k d_l$. Consequently, we can rewrite $\mathbf{X}$ as $\mathbf{X} = \left[\mathbf{X}^1, \dots, \mathbf{X}^l, \dots, \mathbf{X}^k\right]$ with each $\mathbf{X}^l \in \mathbb{R}^{T \times d_l}$ obeying:

$$\boldsymbol{X}_{t+1}^l = \mathbf{A}_l \boldsymbol{X}_t^l + \boldsymbol{Z}_t^l, \text{ for } t = 1, 2, \dots, T-1,$$

which essentially defines the clustering structure in the time series, such that the data $\boldsymbol{X}_{t+1}^l \in \mathbb{R}^{d_l}$ at time point $t+1$ depends only on the data $\boldsymbol{X}_t^l$ from the previous time point t in the same block indexed by $\mathbf{A}_l$. In other words, as an effect of $\mathbf{A}_l$, data are more predictive for each other in the same block, rather than for those in the other blocks. The block diagonal transition matrix $\mathbf{A}$ gives rise to the fact that the time series in $\mathbf{X} \in \mathbb{R}^{T \times d}$ formulate k clusters $\mathcal{C}_1, \dots, \mathcal{C}_l, \dots, \mathcal{C}_k$ of $\mathbb{R}^{T \times d_l}$, and each $\mathcal{C}_l$ contains d_l one-dimensional time series of $\mathbb{R}^T$ denoted as $\mathbf{X}^l$. Without loss of generality, let $\mathbf{X} = \left[\mathbf{X}^1, \dots, \mathbf{X}^l, \dots, \mathbf{X}^k\right]$ be ordered. We further write $\mathcal{S}_l$ to denote the set of indices corresponding to the columns of $\mathbf{X}$ that belong to cluster $\mathcal{C}_l$.

Definition 4.1 (Cluster Recovery Property)**.** *The clusters $\{\mathcal{C}_l\}_{l=1}^k$ and the time series $\mathbf{X}$ from these clusters obey the cluster recovery property (CRP) with a parameter λ, if and only if it holds that for all i, the optimal solution $\hat{\boldsymbol{\beta}}_i$ to* (3.5) *satisfies: (1) $\hat{\boldsymbol{\beta}}_i$ is nonzero; (2) the indices of nonzero entries in $\hat{\boldsymbol{\beta}}_i$ correspond to only the columns of $\mathbf{X}$ that are in the same cluster as $\mathbf{X}_{*i}$.*

This property ensures that the output coefficient matrix $\hat{\mathbf{A}}$ and affinity matrix $\mathbf{W}$ will be exactly block diagonal, with each cluster represented in a disjoint block. Particularly, recall that we assume the transition matrix $\mathbf{A}$ in the VAR model to be block diagonal, and therefore the CRP is guaranteed to hold for data generated from such a model. For convenience, we will refer to the second requirement as the "*Self-Reconstruction Property (SRP)*" from now on.

Definition 4.2 (Inradius [30])**.** *The inradius of a convex body $\mathcal{P}$, denoted by $r(\mathcal{P})$, is defined as the radius of the largest Euclidean ball inscribed in $\mathcal{P}$.*

By the definition, the radius of a $\mathcal{P}(\mathbf{X})$ measures the dispersion of the time series in $\mathbf{X}$. Naturally, well-dispersed data will yield a large inradius while data with skewed distribution will have a small inradius.

4.2 Theoretical Guarantees

One of our major contributions in this paper is to provide theoretical guarantees for successfully recovering the clustering structure in the data.

Theorem 4.3. *Under the assumption of VAR model with a block diagonal transition matrix, we compactly denote* $\mathcal{P}_0^l = \mathcal{P}(\mathbf{\Sigma}_{\mathcal{S}_l,\mathcal{S}_l})$, $\mathcal{P}_1^l = \mathcal{P}((\mathbf{\Sigma}_1)_{\mathcal{S}_l,\mathcal{S}_l})$, $r_0^l = r(\mathcal{P}_0^l)$, $r_1^l = r(\mathcal{P}_1^l)$, *and* $r_0 r_1 = \min_l r_0^l r_1^l$ *for* $l = 1, 2, ..., k$, *and let*

$$\rho = \frac{16\|\mathbf{\Sigma}\|_2 \max_j \Sigma_{jj}}{\min_j \Sigma_{jj}(1 - \|\mathbf{A}\|_2)} \sqrt{\frac{6\log d + 4}{T}}. \tag{4.1}$$

Furthermore, if

$$r_0 r_1 > \frac{\|\mathbf{\Sigma}_{\mathcal{S}_l^c,\mathcal{S}_l}\|_{\infty,\infty} + 2\rho}{\|\boldsymbol{\gamma}_{\mathcal{S}_l}\|_{\infty,\infty} - 2\rho}, \tag{4.2}$$

where $\boldsymbol{\gamma}_{\mathcal{S}_l} \in \mathbb{R}^{d_l}$ *is a column of* $\mathbf{\Sigma}_1$*, then with probability at least* $1 - 6d^{-1}$ *the cluster recovery property holds for all the values of the regularization parameter* λ *in the range:*

$$\frac{1}{r_0 r_1(\|\boldsymbol{\gamma}_{\mathcal{S}_l}\|_{\infty,\infty} - 2\rho) - \rho} < \lambda < \frac{1}{\rho + \|\mathbf{\Sigma}_{\mathcal{S}_l^c,\mathcal{S}_l}\|_{\infty,\infty}}, \tag{4.3}$$

which is guaranteed to be non-empty.

We defer the full proof of the theorem in the supplementary material. The theorem provides an upper bound and a lower bound for the regularization parameter λ, to successfully recover the underlying clustering structure in the time series: on the one hand, λ cannot be too large, otherwise $\mathbf{A}$ will be too dense to perform clustering on. On the other hand, as λ approximates 0 the connectivity among time series decreases because the optimal solution to (3.5) becomes more sparse. To guarantee the obtained solution is nontrivial (i.e., $\hat{\boldsymbol{\beta}}_i$ is nonzero), λ must be larger than a certain value. In addition, a lower bound on $r_0 r_1$ is established, which imposes a requirement on the dispersion of the covariance between time series within the same cluster. We further make the following remarks:

Remark 4.4 (Tolerance of Noise across Clusters)**.** *From* (4.2) *we see that, for the CRP to hold, the dispersion of the columns of* $\mathbf{\Sigma}$ *(each column is taken as a data point* $\in \mathbb{R}^d$*) needs to be sufficiently large.* r_0 *is the dispersion of covariance between time series in a cluster* $\mathcal{S}_l$*, and* r_1 *is the dispersion of lag-1 covariance between time series in a cluster* $\mathcal{S}_l$*. The RHS of* (4.2) *depends on the scale of* $\mathbf{\Sigma}_{\mathcal{S}_l^c,\mathcal{S}_l}$ *and reflects the maximum correlation between the time series in one cluster* $\mathcal{S}_l$ *and any time series from all the other clusters.*

Remark 4.5 (Sample Complexity)**.** *We can observe that the factor before the square root in* (4.1) *is bounded by the largest and smallest eigenvalue of* $\mathbf{\Sigma}$*, and therefore we can rewrite* $\rho = \kappa\sqrt{(6\log d + 4)/T}$ *where* κ *is a constant dependent on* $\mathbf{\Sigma}$*. We can further derive from* (4.2) *that* $T > 4\kappa^2(6\log d + 4)(r_0 r_1 + 1)^2/(r_0 r_1\|\boldsymbol{\gamma}_{\mathcal{S}_l}\|_{\infty,\infty} - \|\mathbf{\Sigma}_{\mathcal{S}_l^c,\mathcal{S}_l}\|_{\infty,\infty})^2$*, which essentially indicates that the sample complexity for CRP to hold is* $O(\log d)$*. In other words, for the algorithm to succeed, the number of time series* d *is allowed to grow exponentially with the length of time series* T*; as long as* $\log d$ *is smaller than the length of time series* T*, our theory holds. Indeed, it is desired to see such a property for high-dimensional data, as* d *can often possibly far exceeds the number of samples* T*.*

Remark 4.6 (A Uniform Parameter λ)**.** *Another direct observation from the main theorem is that we can find a uniform value for* λ*, within the range as specified in* (4.3)*, which can work for the regression task in* (3.5) *for all* $i = 1, ..., d$*. In other words, problem in* (3.5) *is solvable with a single* λ *and can be solved in parallel for each* $i = 1, ..., d$*.*

5 EXPERIMENTS

In this section, we demonstrate the correctness of our theoretical findings and the effectiveness of our proposed clustering algorithm on both synthetic and real-world data. We first experiment with different sets of parameters, including the number of time series d, the length of the time series T, and the number of clusters k in the data. The experimental results confirm that, when the required condition in Theorem 4.3 is satisfied, the clusters in the data can be recovered perfectly. We further apply the algorithm to a real-world data set where the task is to group sensor time series by their type of measurement (e.g., a temperature sensor vs. a humidity sensor). Our algorithm is able to outperform the state-of-art baselines by more than 20% measured by adjusted rand index.

5.1 Baselines

In our proposed clustering algorithm, we estimate the similarity matrix with the regularized Dantzig selector (referred to as CP). As baselines, instead of using our estimator, we consider the following methods to obtain the similarity matrix, and the rest of the clustering procedure remains the same as ours:

Correlation Coefficient (CC): In this baseline, we compute the Pearson correlation coefficient between all pairs of time series, and use these coefficients to construct the similarity matrix.

Cosine Similarity (Cosine): In the second baseline, we compute the pairwise cosine similarity for all the time series, and preserve only the similarity scores for the top-k nearest neighbors for each time series and put them as the row of the similarity matrix. Our experiment shows that the results are not sensitive to k and we set $k = 5$.

Autocorrelation (ACF): This baseline first computes the autocorrelation vectors (with a lag up to 50) for each time series, and then further calculates the Euclidean distance between each pair of time series based on the autocorrelation vectors. We use the implementation in [22] to obtain the distance matrix first, and then convert the distance into similarity score with Gaussian kernel function. (Smaller distances should map to larger similarity scores.)

Dynamic Time Warping (DTW): DTW is a popular method to compute the similarity between time series. Here we compute the pairwise DTW similarity score for all the time series, and then normalize similarity scores to between 0 and 1.

We also implement a baseline that does not rely on the similarity between time series:

Principal Component Analysis (PCA): In this method, PCA is first applied to reduce the dimensionality of each original time series by preserving $d(=4)$ principle components, and then k-means is applied to these PCA scores for clustering.

5.2 Synthetic Data

In this section, we show the effectiveness of our proposed clustering algorithm via numerical simulations. Particularly, the data $\mathbf{X}_t$ at a time point t are generated from a VAR model as defined in (3.1), and we generate the input time series $\mathbf{X}$ as follows: (1) We first generate the block diagonal transition matrix $\mathbf{A}$ with k clusters, and the values within each block are generated with a Bernoulli distribution; (2) Since we assume $(\mathbf{X})_{t=-\infty}^{\infty}$to be stationary, we then rescale $\mathbf{A}$ such that its spectral norm $\|\mathbf{A}\|_2 = \alpha < 1$; (3) Given $\mathbf{A}$, $\mathbf{\Sigma}$ is generated such that the elements on the diagonal equal to 1 and the off-diagonal elements are set to a same small value, e.g., 0.1. Then we rescale $\mathbf{\Sigma}$ to have its spectral norm satisfy $\|\mathbf{\Sigma}\|_2 = 2\|\mathbf{A}\|_2$; (4) Next, according to the stationary property, the covariance matrix of the additive noise $\mathbf{Z}_t$ follows $\mathbf{\Psi} = \mathbf{\Sigma} - \mathbf{A}^\top \mathbf{\Sigma} \mathbf{A}$, where $\mathbf{\Psi}$ must be a positive definite matrix; (5) We can then generate $\mathbf{X}_1$ from the multivariate normal distribution with the parameters generated in previous steps, and obtain the following $\mathbf{X}_t$ with the VAR model. We fix the number of time series d at 100, and choose the length of the time series T from a grid of $\{1, 3, 5, 7, 9\} \times \log(d)$ (rounded to the closet integer), i.e, the ratio of $T/\log(d)$ varies from 1 to 9. For each value of T, we repeat the data generation process for 100 times and report the average of the experimental results.

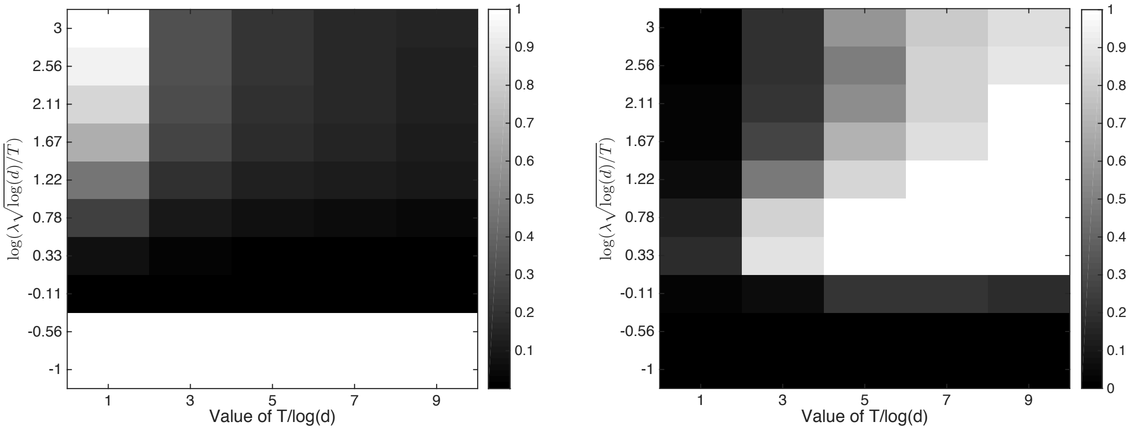
We first experiment with different values for the regularization parameter λ and examine if the two requirements as stated in Definition 4.1 are satisfied. We scan through an exponential space of λ from $1/(\log(d)/T) \times 10^{-1}$ to $1/(\log(d)/T) \times 10^{3}$ and define the metric *Self-Reconstruction Property Violation Rate* (VioRate) of the estimated transition matrix $\mathbf{A}$ as follows:

$$VioRate = \frac{\sum_{i,j \notin \mathcal{C}_l} |A_{ij}|}{\sum_{i,j \in \mathcal{C}_l} |A_{ij}|},$$

where $(i, j) \in \mathcal{C}_l$ denotes that the i-th time series $\mathbf{X}_{*i}$ and the j-th time series $\mathbf{X}_{*j}$ are in the same cluster $\mathcal{C}_l$ for some l (likewise for $(i, j) \notin \mathcal{C}_l$). By definition, $VioRate$ measures relatively how significant the predictive weights are for pairs of time series across different clusters, compared to the weights for pairs in the same cluster. For a trivial solution, i.e., $\mathbf{A} = \mathbf{0}$, the $VioRate$ is defined to be 1 while for a solution satisfying the self-reconstruction property, the $VioRate$ should be exactly 0. The violation rates for different $T/\log(d)$ and λ values when $k = 25$ are illustrated in Figure 2a; the results confirm our theoretical findings. We observe that when λ is small, the solution violates the nonzero requirement, thus the $VioRate$ being 1 (refer to the two rows at the bottom). When λ is sufficiently large within a range, the violation rates are zero, indicating all the entries in the off-diagonal blocks of the estimated $\mathbf{A}$ are zero, which satisfies the SRP. In Figure 2b, we show the quality of time series clustering (measured by adjusted rand index and higher is better) with the corresponding $\mathbf{A}$ obtained in Figure 2a. We can notice that cases perfectly satisfying the nonzero and SRP requirements can produce perfect clustering results. Furthermore, it is also clear that exact self-reconstruction condition is not necessary for perfect clustering.

We next investigate how the number of clusters k affects the clustering performance, where we vary the value of k from 5 to 25. For the regularization parameter λ, we scan through the same exponential space as the above experiment with 5-fold cross-validation, and choose the one with the minimal cross-validation error. We fix $\|\mathbf{A}\|_2$ at 0.4 and report the average results of the 100 runs for each set of parameters as illustrated in Figure 3. We clearly see that a larger k leads to better clustering results, which makes sense since the more the number of clusters is, the sparser $\mathbf{A}$ is, and therefore the more accurate the estimation of $\mathbf{A}$ is.

We also examine the effect of the transition matrix's spectral norm $\|\mathbf{A}\|_2$ on the clustering quality. To this end, we set $\|\mathbf{A}\|_2 = \alpha$ and vary α from 0.1 to 0.9, and the covariance matrix $\mathbf{\Sigma}$ and $\mathbf{\Psi}$ are generated in the same way as described earlier. For the parameter λ, we take the same cross-validation procedure as above. We fix the number of clusters k at 25 and report the average results of the 100 runs for each set of parameters, as shown in Figure 4. We observe that, for a certain value of $T/\log(d)$, the clustering quality increases as the spectral norm of the transition matrix decreases.



(a) Self-Reconstruction Property (SRP) violation rate for different $T/\log(d)$ against different λ with $k = 25$: too small a λ will produce trivial solutions ($\mathbf{A} = 0$, thus $VioRate = 1$) while a sufficiently large λ gives a solution satisfying both the nonzero and SRP requirements ($VioRate = 0$).

(b) Clustering quality for different $T/\log(d)$ against different λ with $k = 25$: cases satisfying the nonzero and SRP conditions yield perfect clustering results. It is also clear that the exact self-reconstruction condition ($VioRate = 0$) is not necessary for perfect clustering.

Figure 2: Self-Reconstruction Property Violation Rate and the Corresponding Clustering Quality (measured by Adjusted Rand Index) with Different $T/\log(d)$ Against Different λ.

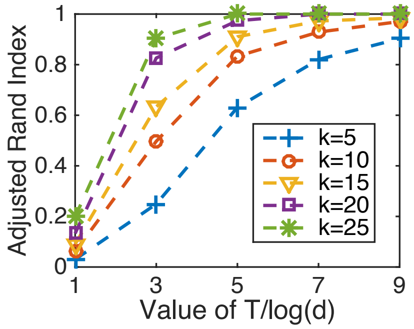


Figure 3: Clustering Quality for Different $T/\log(d)$ Against Different Number of Clusters k: larger k is better.

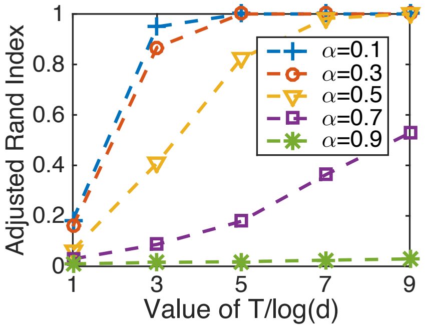


Figure 4: Clustering Quality for Different $T/\log(d)$ Against Different Values α for Spectral Norm $\|\mathbf{A}\|_2$: smaller α is better.

This indicates that the spectral norm of the transition matrix is a critical factor and verifies the theoretical findings in (4.1).

To compare our method with the baselines described in §5.1, we further conduct two sets of experiments on synthetic data with different parameters. To generate the synthetic data in the first experiment (referred to as Synthetic Data-1 in Table 1), we set the number of time series $d = 50$, the length of time series $T = 50$, the number of clusters $k = 5$, and the transition matrix's spectral norm $\|\mathbf{A}\|_2 = 0.5$. For the second experiment (Synthetic Data-2 in Table 1), we change T to 100, and the rest of parameters remain the same. We see that, when d is comparable to T in the first experiment, our method (CP) performs significantly better than the baselines. When the number of samples T is increased to 100, all the baselines see performance boost, while our method produces perfect clustering results.

One shall note the better performance of PCA, our understanding is that PCA extracts better explanatory components out of the sample covariance matrix, which still captures the underlying causal relationship between variables, though it does not consider the first-order temporal information as our proposed method does. For the other baselines, they simply compute similarity directly between variables, which is not sufficiently effective in characterizing the relationship between time series in the high-dimensional setting.

Table 1: Experimental Comparisons with Baselines: results on synthetic and real data demonstrate the advantage of our proposed algorithm (CP), and each cell includes the average clustering performance (adjusted rand index) of 10 runs with standard deviation.

	CC	Cosine	ACF	DTW	PCA	CP
Synthetic Data-1 ($d = 50, T = 50$)	0.383 ± 0.171	0.521 ± 0.123	0.240 ± 0.147	0.282 ± 0.184	0.786 ± 0.139	0.943 ± 0.165
Synthetic Data-2 ($d = 50, T = 100$)	0.603 ± 0.181	0.551 ± 0.143	0.253 ± 0.109	0.410 ± 0.105	0.912 ± 0.141	1.000 ± 0.000
Real Data	0.617 ± 0.031	0.542 ± 0.014	0.362 ± 0.113	0.523 ± 0.119	0.456 ± 0.144	0.824 ± 0.025

5.3 Real-world Data

To further examine how effective our proposed algorithm is in practice, we also apply it to a real-world data set, where the assumption of VAR model with block diagonal transition matrix might not be perfectly satisfied. The data set [16] contains data collected from 204 sensor time series from 51 rooms on 4 different floors of a large office building on a university campus. Each room is instrumented with 4 different types of sensors: a CO_2 sensor, a temperature sensor, a humidity sensor and a light sensor. The data from each sensor is recorded every 15 minutes and the data set contains one-week worth of data. There are missing values in the one-week period, so the total number of observations T is smaller than the number of sensor time series d. Our goal is to assign each sensor time series into the correct type cluster, e.g., a temperature cluster or a CO_2 cluster. Recognizing the type of sensors is often an important step for many useful applications. For instance, when applying analytics stacks comprised of a bundle of analytics jobs to a building for energy savings, every particular analytics job requires as input some specific types of sensors.

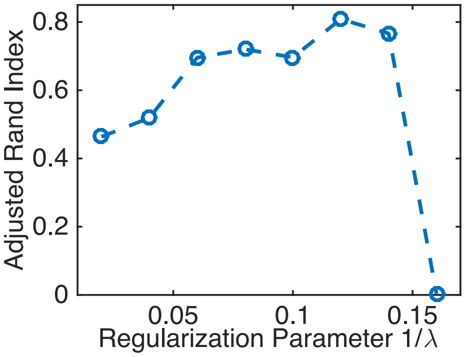


Figure 5: Clustering Quality of Our Regularized Dantzig Selector-based Spectral Clustering Algorithm: the algorithm works with a wide range of λ.

In this case, we do not know the values of the parameters in the sufficient condition in Theorem 4.3, so we cannot fine-tune λ. We roughly scan through the entire range of [0,1] for $1/\lambda$ and the results are shown in Figure 5 (the data points beyond 0.15 all drop to zero, thus omitted in the figure). It again confirms our theoretical findings in the sense that the proposed clustering algorithm can work when λ is sufficiently large, even not perfectly. We also examine how well the baselines (detailed in §5.1) perform on the real data set, and the results are summarized in Table 1. Our method can achieve more than 80% accuracy and outperforms the best baseline by more than 20%, indicating that our method can still be effective when the assumption of VAR with a block diagonal transition matrix might not be satisfied.

6 CONCLUSIONS

In this paper, we study the time series clustering problem with a new similarity metric in the high-dimensional regime, where the number of time series is much larger than the length of time series. Different from existing metrics, our similarity metric measures the "cross-predictability" between time series, i.e., the degree to which a future value in each time series is predicted by past values of the others. We impose a sparsity assumption and propose a regularized Dantzig selector estimator to learn the cross-predictability among time series for clustering. We further provide a theoretical proof that the proposed algorithm will successfully recover the clustering structure in the data with high probability under certain conditions. Experiments on both synthetic and real-world data verify the correctness of our findings, and demonstrate the effectiveness of the algorithm. For the real-world task of sensor type clustering, our method is able to outperform the state-of-art baselines by more than 20% with regard to clustering quality.

7 Acknowledgments

We would like to thank the anonymous reviewers for their invaluable comments. This work was funded by NSF awards CNS-1305362, IIS-1618948, and IIS-165253. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

[1] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(2):218–233, 2003.

[2] S. Basu, G. Michailidis, et al. Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, 43(4):1535–1567, 2015.

[3] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the 21st international conference on Machine learning*, page 11. ACM, 2004.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[5] R. Brandenberg, A. Dattasharma, P. Gritzmann, and D. Larman. Isoradial bodies. *Discrete & Computational Geometry*, 32(4):447–457, 2004.

[6] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *Knowledge and Data Engineering, IEEE Transactions on*, 17(12):1624–1637, 2005.

[7] E. Candes and T. Tao. The dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, pages 2313–2351, 2007.

[8] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

[9] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2790–2797. IEEE, 2009.

[10] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. *Fast subsequence matching in time-series databases*, volume 23. ACM, 1994.

[11] P. Galeano and D. Peña. Multivariate analysis in vector time series. 2001.

[12] Gartner Inc. `http://www.gartner.com/newsroom/id/2636073`.

[13] X. Golay, S. Kollias, G. Stoll, D. Meier, A. Valavanis, and P. Boesiger. A new correlation-based fuzzy logic clustering algorithm for fmri. *Magnetic Resonance in Medicine*, 40(2):249–260, 1998.

[14] J. D. Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, 1994.

[15] F. Han, H. Lu, and H. Liu. A direct estimation of high dimensional stationary vector autoregressions. *Journal of Machine Learning Research*, 16:3115–3150, 2015.

[16] D. Hong, J. Ortiz, K. Whitehouse, and D. Culler. Towards automatic spatial verification of sensor placement in buildings. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pages 1–8. ACM, 2013.

[17] A. Jalali, Y. Chen, S. Sanghavi, and H. Xu. Clustering partially observed graphs via convex optimization. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1001–1008, 2011.

[18] E. Keogh. Exact indexing of dynamic time warping. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 406–417. VLDB Endowment, 2002.

[19] A. Khaleghi, D. Ryabko, J. Mary, and P. Preux. Consistent algorithms for clustering time series. *Journal of Machine Learning Research*, 17(3):1–32, 2016.

[20] J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604. ACM, 2007.

[21] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th international conference on machine learning*, pages 663–670, 2010.

[22] P. Montero and J. A. Vilar. Tsclust: An r package for time series clustering.

[23] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, volume 14, pages 849–856, 2001.

[24] A. Panuccio, M. Bicego, and V. Murino. A hidden markov model-based approach to sequential data clustering. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 734–743. Springer, 2002.

[25] C. Qu and H. Xu. Subspace clustering with irrelevant features via robust dantzig selector. In *Advances in Neural Information Processing Systems*, pages 757–765, 2015.

[26] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 262–270. ACM, 2012.

[27] D. Ryabko and J. Mary. Reducing statistical time-series problems to binary classification. In *Advances in Neural Information Processing Systems*, pages 2060–2068, 2012.

[28] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[29] P. Smyth. Clustering sequences with hidden markov models. *Advances in neural information processing systems*, pages 648–654, 1997.

[30] M. Soltanolkotabi and E. J. Candes. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, pages 2195–2238, 2012.

[31] M. Soltanolkotabi, E. Elhamifar, E. J. Candes, et al. Robust subspace clustering. *The Annals of Statistics*, 42(2):669–699, 2014.

[32] M. J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using-constrained quadratic programming (lasso). *Information Theory, IEEE Transactions on*, 55(5):2183–2202, 2009.

[33] Y.-X. Wang and H. Xu. Noisy sparse subspace clustering. *Journal of Machine Learning Research*, 17(12):1–41, 2016.

[34] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems*, 15:505–512, 2003.