Degree Papers for the Master's Program in Computational and Applied Mathematics

Statistical Analysis of High-dimensional Time Series

Qin Wen

Advisors: Rebecca Willett, Daren Wang

Approved Daren Wang

Date May 14, 2021

May-14th, 2021

Abstract

This paper consists of two projects: localizing change points in high dimensional regression setting and functional autoregressive (FAR) processes in reproducing kernel Hilbert spaces and the 2 dimensional extension. The changepoint project is the joint work with Alessandro Rinaldo, Daren Wang, Rebecca Willett and Yi Yu. (see Rinaldo et al. (2021)). It addresses the problem of localizing change points in high-dimensional linear regression models with piecewise constant regression coefficients. We developed dynamic programming and a computationally efficient refinement procedure that consistently estimates the unknown number of change points K and the time points' locations by performing group lasso in parallel on different segments. To tune parameters for numerical experiments on both simulated data and a real public dataset, we developed a data-driven tuning method by combining cross-validation with a grid search method. This part of the paper has appeared at AISTATS 2021. In the FAR project, we study the inference (estimation and prediction) of a functional autoregressive (FAR) process, a statistical tool for modeling functional time series data. We proposed a Frobenius norm regularization method to estimate the transition operator of the FAR process directly from discrete measurements of the functional time series. We extend the one dimensional framework into two dimensional scenario. We also developed the two dimensional prediction method based on functional principal component analysis (FPCA) which serves as a baseline method in this paper.

Contents

1 Acknowledgement

I would like to express my deep gratitude to Professor Rebecca Willett and Professor Daren Wang, my research supervisors, for their patient guidance, enthusiastic encouragement and useful critiques of this research work. I would also like to thank Professor Yi Yu, Professor Alessandro Rinaldo, for their advice and assistance in the change point detection project.

My grateful thanks are also extended to Dr. Eric Baer for his valuable guidance throughout my master's studies. I would also like to extend my thanks to the Committee on Computational and Applied Mathematics for all the resources provided.

Finally, I wish to thank my parents and my friends for their support and encouragement throughout my study.

2 Localizing Changes in High-Dimensional Regression Models

This is the joint work with Alessandro Rinaldo, Daren Wang, Rebecca Willett and Yi Yu. (see Rinaldo et al. (2021)). It has appeared at AISTATS 2021.

2.1 Introduction

High-dimensional linear regression modeling has been extensively applied and studied over the last two decades due to the technological advancements in collecting and storing data from a wide range of application areas, including biology, neuroscience, climatology, finance, cybersecurity, to name but a few. There exist now a host of methodologies available to practitioners to fit high-dimensional sparse linear models, and their properties have been thoroughly investigated and are now well understood. See Bühlmann and van de Geer (2011) for recent reviews.

In this paper, we are concerned with a non-stationary variant of the high-dim linear regression model in which the data are observed as a time series and the regression coefficients are piece-wise stationary, with changes occurring at unknown times. We formally introduce our model settings next.

Model 1. *Let the data $\{(x_t, y_t)\}_{t=1}^n \subset \mathbb{R}^p \times \mathbb{R}$ satisfy the model*

$$y_t = x_t^\top \beta_t^* + \varepsilon_t, \quad t = 1, \ldots, n$$

where $\{\beta_t^*\}_{t=1}^n \subset \mathbb{R}^p$ is the unknown coefficient vector, $\{x_t\}_{t=1}^n$ are independent and identically distributed mean-zero sub-Gaussian random vectors with $\mathbb{E}(x_t x_t^\top) = \Sigma$, and $\{\varepsilon_t\}_{t=1}^n$ are independent mean-zero sub-Gaussian random variables with sub-Gaussian parameter bounded by σ_ε^2 and independent of $\{x_t\}_{t=1}^n$. In addition, there exists a sequence of change points $1 = \eta_0 < \eta_1 < \ldots < \eta_{K+1} = n$ such that $\beta_t^* \neq \beta_{t-1}^*$, if and only if $t \in \{\eta_k\}_{k=1}^K$.

We consider a high-dimensional framework where the features of the above change-point model are allowed to change with the sample size n; see Assumption 1 below for details.

Given data sampled from Model 1, our main task is to develop computationally-efficient algorithms that can consistently estimate both the unknown number K of change points and the time points $\{\eta_k\}_{k=1}^K$, at which the regression coefficients change. That is, we seek *consistent* estimators $\{\hat{\eta}_k\}_{k=1}^{\widehat{K}}$, such that, as the sample size n grows unbounded, it holds with probability tending to 1 that

$$\widehat{K} = K \quad \text{and} \quad \epsilon = \max_{k=1,\dots,K} |\widehat{\eta}_k - \eta_k| = o(\Delta),$$

where $\Delta = \min_{k=1,\dots,K_1} \eta_k - \eta_{k-1}$ is the minimal spacing between consecutive change-points. We refer to the quantity ϵ as the *localization error rate*.

The model detailed above has already been considered in the recent literature. Lee et al. (2016), Kaul et al. (2018), Lee et al. (2018), among others, focused on the cases where there exists at most one true change point. Leonardi and Bühlmann (2016) and Zhang et al. (2015) considered multiple change points and devised consistent change point estimators, albeit with localization error rates worse than the one we establish in Theorem 1. Wang et al. (2019) also allowed for multiple change points in a regression setting and proposed a variant of the wild binary segmentation (WBS) method (Fryzlewicz, 2014), the performances thereof match the one of the procedure we study next. More detailed comparisons are further commentary can be found in Section 2.3.3.

In this paper, we make several theoretical and methodological contributions, summarized next, that improve the existing literature.

- We provide consistent change point estimators for Model 1. We allow for model parameters to change with the sample size n, including the dimensionality of the data, the entry-wise sparsity of the coefficient vectors, the number of change points, the smallest distance between two consecutive change points, and the smallest difference between two consecutive different regression coefficients. To the best of our knowledge, the theoretical results we provide in this paper are the sharpest in the existing literature. Furthermore, the proposed algorithms, based on the general framework described in (1), can be implemented using dynamic programming approaches and are computationally efficient.

- We devise a additional second step (Algorithm 1), called local refinement, that is guaranteed to deliver an even better localization error rate, even though directly optimizing (1) already provides the sharpest rates among the ones existing in the literature.

- We present information-theoretic lower bounds on both detection and localization, establishing the fundamental limits of localizing change points in Model 1. To the best of our knowledge, this is the first time such results are developed for Model 1. The lower bounds on the localisation and detection nearly match the upper bounds we obtained under mild conditions.

- We present extensive experimental results including simulated data and real data analysis, supporting our theoretical findings, and confirming the practicality of our procedures.

- To tune parameters for numerical experiments on both simulated data and a real public dataset, I developed a data-driven tuning method by combining cross-validation with a grid search method. The results of my tuning method showed that the proposed local refinement algorithm significantly improves the initial estimator, especially when the estimated number of initial points $\hat{K}$ is correct($=K$).

Throughout this paper, we adopt the following notation. For any set S, $|S|$ denotes its cardinality. For any vector v, let $\|v\|_2$, $\|v\|_1$, $\|v\|_0$ and $\|v\|_\infty$ be its ℓ_2-, ℓ_1-, ℓ_0- and entry-wise maximum norms, respectively; and let $v(j)$ be the jth coordinate of v. For any square matrix $A \in \mathbb{R}^{n\times n}$, let $\Lambda_{\min}(A)$ and $\Lambda_{\max}(A)$ be the smallest and largest eigenvalues of matrix A, respectively. For any pair of integers $s, e \in \{0, 1, \ldots, n\}$ with $s < e$, we let $(s, e] = \{s + 1, \ldots, e\}$ and $[s, e] = \{s, \ldots, e\}$ be the corresponding integer intervals.

2.2 Methods

2.2.1 A Dynamic Programming Approach

To achieve the goal of obtaining consistent change point estimators, we adopt a dynamic programming approach, whiuch we summarize next. Let $\mathcal{P}$ be an integer interval partition of $\{1, \ldots, n\}$ into $K_{\mathcal{P}}$ intervals, i.e.

$$\mathcal{P} = \big\{\{1, \ldots, i_1 - 1\}, \{i_1, \ldots, i_2 - 1\}, \ldots, \{i_{K_{\mathcal{P}}-1}, \\ \ldots, i_{K_{\mathcal{P}}} - 1\}\big\},$$

for some integers $1 < i_1 < \cdots < i_{K_{\mathcal{P}}} = n+1$, where $K_{\mathcal{P}} \geq 1$. For a positive tuning parameter $\gamma > 0$, let

$$\widehat{\mathcal{P}} \in \operatorname*{arg\,min}_{\mathcal{P}} \left\{ \sum_{I \in \mathcal{P}} \mathcal{L}(I) + \gamma |\mathcal{P}| \right\}, \tag{1}$$

where $\mathcal{L}(\cdot)$ is an appropriate loss function to be specified below, $|\mathcal{P}|$ is the cardinality of $\mathcal{P}$ and the minimization is taken over all possible interval partitions of $\{1, \ldots, n\}$.

The change point estimator resulting from the solution to (1) is simply obtained by taking all the left endpoints of the intervals $I \in \widehat{\mathcal{P}}$, except 1. The optimization problem (1) is known as the *minimal partition problem* and can be solved using dynamic programming with an overall computational cost of order $O(n^2 \mathcal{T}(n))$, where $\mathcal{T}(n)$ denotes the computational cost of solving $\mathcal{L}(I)$ with $|I| = n$ (see e.g. Algorithm 1 in Friedrich et al., 2008).

To specialize the dynamic programming algorithm to the high-dimensional linear regression model of interest by setting the loss function to be (1) with

$$\mathcal{L}(I) = \sum_{t \in I} (y_t - x_t^\top \widehat{\beta}_I^\lambda)^2, \tag{2}$$

where

$$\widehat{\beta}_I^\lambda = \operatorname*{arg\,min}_{v \in \mathbb{R}^p} \Bigg\{ \sum_{t \in I} (y_t - x_t^\top v)^2 \\ + \lambda \sqrt{\max\{|I|, \log(n \vee p)\}} \|v\|_1 \Bigg\}, \tag{3}$$

and $\lambda \geq 0$ is a tuning parameter. The penalty is multiplied by the quantity $\max\{|I|, \log(n \vee p)\}$ in order to fulfill certain types of large deviation inequalities that are needed to ensure consistency.

Algorithms based on dynamic programming are widely used in the change point detection literature. Friedrich et al. (2008), Killick et al. (2012), Rigaill (2010), Maidstone et al. (2017), Wang et al. (2018b), among others, studied dynamic programming approaches for change point analysis involving a univariate time series with piecewise-constant means. Leonardi and Bühlmann (2016) examined high-dimensional linear regression change point detection problems by using a version of dynamic programming approach.

2.2.2 Local Refinement

We will show later in Theorem 1 that the localization error afforded by the dynamic programming approach in (1), (2), and (3) is linear in K, the number of change points. Although the corresponding localization rate is already sharper than any other rates previously established in the literature (see Section 2.3.3), it is possible to improve it by removing the dependence on K through an additional step, which we refer to as local refinement, detailed in Algorithm 1.

Algorithm 1 Local Refinement.

```
INPUT: Data $\{(x_t, y_t)\}_{t=1}^{n}$, $\{\widetilde{\eta}_k\}_{k=1}^{\widetilde{K}}$ , $\zeta > 0$.
  $(\widetilde{\eta}_0, \widetilde{\eta}_{\widetilde{K}+1}) \leftarrow (0, n)$
  for $k = 1, \ldots, \widetilde{K}$ do
      $(s_k, e_k) \leftarrow (2\widetilde{\eta}_{k-1}/3 + \widetilde{\eta}_k/3, \widetilde{\eta}_k/3 + 2\widetilde{\eta}_{k+1}/3)$
      $\left(\widehat{\beta}_1, \widehat{\beta}_2, \widehat{\eta}_k\right) \leftarrow \mathop{\arg\min}_{\substack{\eta \in \{s_k+1, \ldots, e_k-1\} \\ \beta_1, \beta_2 \in \mathbb{R}^p,\, \beta_1 \neq \beta_2}} \Bigg\{ \sum_{t=s_k+1}^{\eta} \left\|y_t - \beta_1^{\top} x_t\right\|_2^2 + \sum_{t=\eta+1}^{e_k} \left\|y_t - \beta_2 x_t\right\|_2^2 + \zeta \sum_{i=1}^{p} \sqrt{(\eta - s_k)(\beta_1)_i^2 + (e_k - \eta)(\beta_2)_i^2} \Bigg\}$   (4)
  end for
OUTPUT: $\{\widehat{\eta}_k\}_{k=1}^{\widetilde{K}}$.
```

Algorithm 1 takes a sequence of preliminary change point estimators $\{\widetilde{\eta}_k\}_{k=1}^{\widetilde{K}}$, and refines each of the estimator $\widetilde{\eta}_k$ within the interval (s_k, e_k), which is a shrunken version of $(\widetilde{\eta}_{k-1}, \widetilde{\eta}_{k+1})$ (the constants $2/3$ and $1/3$ specifying the shrinking factor in the definition of $(\widetilde{\eta}_{k-1}, \widetilde{\eta}_{k+1})$ are not special and can be replaced by other values without affecting the rates). The shrinkage is applied to eliminate false positives, which are more likely to occur in the immediate proximity of a preliminary estimate of a change point. Since the refinement is done locally within each disjoint interval, the procedure is parallelizable. A group Lasso penalty is deployed in (4), and this is key to the success of the refinement. Intuitively, the group Lasso penalty integrates the information that the regression coefficients are piecewise-constant within each coordinate. Previously, Wang et al. (2019) also proposed a similar local screening algorithm based on the group Lasso estimators to refine the estimates of the regression change points. While Wang et al. (2019) assumed all the covariates to be uniformly bounded, we show that Algorithm 1 can achieve optimal localization error rates in a more general setting.

2.3 Main Results

In this section, we derive high-probability bounds on the localization errors of our main procedure based on the dynamic programming algorithm as detailed in equations 1, 2, and 3, and of the local refinement procedure of Algorithm 1.

2.3.1 Assumptions

We begin by stating the assumptions we require in order to derive localization error bounds.

Assumption 1. *Consider the model defined in Model 1. We assume that, for some fixed positive constants C_β, c_x, C_x, ξ, and C_{SNR} the following holds:*

a. *(Sparsity). Let $d_0 = |S|$. There exists a subset $S \subset \{1, \ldots, p\}$ such that*

$$\beta_t^*(j) = 0, \quad t = 1, \ldots, n, \quad j \in S^c = \{1, \ldots, p\} \setminus S.$$

b. *(Boundedness). For some absolute constant $C_\beta > 0$, $\max_{t=1,\ldots,n} \|\beta_t^*\|_\infty \leq C_\beta$.*

c. *(Minimal eigenvalue). We have that $\Lambda_{\min}(\Sigma) = c_x^2 > 0$ and $\max_{j=1,\ldots,p}(\Sigma)_{jj} = C_x^2 > 0$.*

d. *(Signal-to-noise ratio). Let* $\kappa = \min_{k=1,\ldots,K+1} \|\beta^*_{\eta_k} - \beta^*_{\eta_{k-1}}\|_2$ *and* $\Delta = \min_{k=1,\ldots,K+1}(\eta_k - \eta_{k-1})$ *be the minimal jump size and minimal spacing defined as follows, respectively. Then,*

$$\Delta\kappa^2 \geq C_{\mathrm{SNR}} d_0^2 K \sigma_\varepsilon^2 \log^{1+\xi}(n \vee p). \tag{5}$$

Assumption 1(**a**) and (**c**) are standard conditions required for consistency of Lasso-based estimators. Assumption 1(**d**) specifies a minimal signal-to-noise ratio condition that allows to detect the presence of a change point. Interestingly, if $K = d_0 = 1$, (5) reduces to $\Delta\kappa^2\sigma_\epsilon^{-2} \gtrsim \log^{1+\xi}(n \vee p)$, matching the information theoretic lower bound (up to constants and logarithmic terms) for the univariate mean change point detection problem (see e.g. Chan and Walther, 2013; Frick et al., 2014; Wang et al., 2018b).

In addition, we have

$$\begin{aligned}\Delta &\geq \frac{C_{\mathrm{SNR}} d_0^2 K \sigma_\epsilon^2 \log^{1+\xi}(n \vee p)}{\kappa^2} \\ &\geq \frac{C_{\mathrm{SNR}} d_0^2 K \sigma_\epsilon^2 \log^{1+\xi}(n \vee p)}{4C_\beta^2 d_0} \\ &\geq \frac{C_{\mathrm{SNR}}}{4C_\beta^2} d_0 K \sigma_\epsilon^2 \log^{1+\xi}(n \vee p),\end{aligned} \tag{6}$$

where the second inequality follows from the bound

$$\kappa^2 = \min_{k=1,\ldots,K+1} \|\beta^*_{\eta_k} - \beta^*_{\eta_{k-1}}\|_2^2 \leq d_0(2C_\beta)^2 = 4C_\beta^2 d_0.$$

If $\Delta = \Theta(n)$ and $K = O(1)$, then (6) becomes $n \gtrsim d_0 \log^{1+\xi}(n \vee p)$, which resembles the effective sample size condition needed in the Lasso estimation literature.

Another way to interpret the signal-to-noise ratio Assumption 1(**d**) is to introduce a normalized jump size $\kappa_0 = \kappa/\sqrt{d_0}$, which leads to the equivalent condition

$$\Delta\kappa_0^2 \geq C_{\mathrm{SNR}} d_0 K \sigma_\epsilon^2 \log^{1+\xi}(n \vee p).$$

Analogous constraints on the model parameters are required in other change point detection problems, including high-dimensional mean change point detection (Wang and Samworth, 2018), high-dimensional covariance change point detection (Wang et al., 2017), sparse dynamic network change point detection (Wang et al., 2018a), high-dimensional regression change point detection (Wang et al., 2019), to name but a few. Note that in these aforementioned papers, when variants of wild binary segmentation (Fryzlewicz, 2014) were deployed, additional knowledge is needed to get rid of K in the lower bound of the signal-to-noise ratio. We refer the reader to Wang et al. (2018a) for more discussions regarding this point.

The constant ξ is needed to guarantee consistency when Δ is of the same irder as n but can be set to zero if $\Delta = o(n)$. We may instead replace it with a weaker condition of the form

$$\Delta\kappa^2 \gtrsim C_{\mathrm{SNR}} d_0^2 K\{\log(n \vee p) + a_n\},$$

where $a_n \to \infty$ arbitrarily slow as $n \to \infty$. We stick with the signal-to-noise ratio condition (5) for simplicity.

2.3.2 Localization Rates

We are now ready to state one of the main results of the paper.

Theorem 1. *Assume Model 1 and the conditions in Assumption 1. Then, the change point estimators* $\{\widetilde{\eta}_k\}_{k=1}^{\widehat{K}}$ *obtained as a solution to the dynamic programming optimization problem given in* (1), (2), *and* (3) *with tuning parameters*

$$\lambda = C_\lambda \sigma_\varepsilon \sqrt{d_0 \log(n \vee p)}$$

and

$$\gamma = C_\gamma \sigma_\epsilon^2 (K+1) d_0^2 \log(n \vee p),$$

are such that

$$\begin{aligned}\mathbb{P}\left\{\widehat{K} = K, \max_{k=1,\ldots,K} |\widetilde{\eta}_k - \eta_k| \le \frac{K C_\epsilon d_0^2 \sigma_\epsilon^2 \log(n \vee p)}{\kappa^2}\right\} \\ \ge 1 - C(n \vee p)^{-c},\end{aligned} \tag{7}$$

where $C_\lambda, C_\gamma, C_\epsilon, C, c > 0$ *are absolute constants depending only on* C_β, C_x*, and* c_x*.*

The above result implies that, with probability tending to 1 as n grows,

$$\begin{aligned}\max_{k=1,\ldots,K} \frac{|\hat{\eta}_k - \eta_k|}{\Delta} &\le \frac{K C_\epsilon d_0^2 \sigma_\epsilon^2 \log(n \vee p)}{\kappa^2 \Delta} \\ &\le \frac{C_\epsilon}{C_{\mathrm{SNR}} \log^\xi (n \vee p)} \to 0,\end{aligned}$$

where in the second inequality we have used Assumption 1(**c**). Thus, the localization error converges to zero in probability.

It is worth emphasizing that the bound in (7) along with Model 1 provide a *family* of rates, depending on how the model parameters (p, d_0, κ, Δ, K and σ_ϵ) scale with n.

The tuning parameter λ affects the performance of the Lasso estimator. The second tuning parameter γ prevents overfitting while searching the optimal partition as a solution to the problem (1). In particular, γ is determined by the squared ℓ_2-loss of the Lasso estimator and is of order $\lambda^2 d_0$. We will elaborate more on this point in **??**.

We now turn to the analysis of the local refinement algorithm, which takes as input a preliminary collection of change point estimators $\{\widetilde{\eta}_k\}_{k=1}^K$ such that $\max_{k=1,\ldots,K} |\widetilde{\eta}_k - \eta_k|$, such as the ones returned by our estimator based on the dynamic programming approach. The only assumption required for local refinement is that the localization error of the preliminary estimators be a small enough fraction of the minimal spacing Δ (see (8) below). Then local refinement returns an improved collection of change point estimators $\{\widehat{\eta}_k\}_{k=1}^K$ with a vanishing localization error rate of order $O\left(\frac{d_0 \log(n \vee p)}{n\kappa^2}\right)$. Interestingly, the initial estimators need not be consistent in order for local refinement to work: all that is required is essentially that the each of the working intervals in Algorithm 1 contains one and only one true change point. This fact allows us to refine the search within each working intervals separately, yielding better rates.

In particular, if we use the outputs of (1), (2), and (3) as the inputs of Algorithm 1, then it follows from (7) and (5) that, for any $k \in \{1, \ldots, K\}$,

$$\begin{aligned} s_k - \eta_{k-1} &> \frac{2}{3}\widetilde{\eta}_{k-1} + \frac{1}{3}\widetilde{\eta}_k - \widetilde{\eta}_{k-1} - \epsilon \\ &= \frac{1}{3}(\widetilde{\eta}_k - \widetilde{\eta}_{k-1}) - \epsilon > \Delta/3 - 5\epsilon/3 > 0 \end{aligned}$$

and

$$\begin{aligned} s_k - \eta_k &< \frac{2}{3}\widetilde{\eta}_{k-1} + \frac{1}{3}\widetilde{\eta}_k - \widetilde{\eta}_k + \epsilon \\ &= -\frac{2}{3}(\widetilde{\eta}_{k-1} - \widetilde{\eta}_k) + \epsilon < -2\Delta/3 + 5\epsilon/3 < 0. \end{aligned}$$

Corollary 1. *Assume the same conditions of Theorem 1. Let $\{\widetilde{\eta}_k\}_{k=1}^K$ be a set of time points* *satisfying*

$$\max_{k=1,\ldots,K} |\widetilde{\eta}_k - \eta_k| \leq \Delta/7. \tag{8}$$

Let $\{\widehat{\eta}_k\}_{k=1}^{\widehat{K}}$ *be the change point estimators generated from Algorithm 1 with* $\{\widetilde{\eta}_k\}_{k=1}^{K}$ *and*

$$\zeta = C_\zeta \sqrt{\log(n \vee p)}$$

as inputs. Then,

$$\begin{aligned} \mathbb{P}\left\{\widehat{K} = K, \max_{k=1,\ldots,K} |\hat{\eta}_k - \eta_k| \leq \frac{C_\epsilon d_0 \log(n \vee p)}{\kappa^2}\right\} \\ \geq 1 - Cn^{-c}, \end{aligned}$$

where $C_\zeta, C_\epsilon, C, c > 0$ *are absolute constants depending only on* $C_\beta, \mathcal{M}$ *and* c_x.

Compared to the localization error given in Theorem 1, the improved localization error guaranteed by the local refinement algorithm does not have a direct dependence on K, the number of change points. The intuition for this is as follows. First, due to the nature of the change point detection problem, there is a natural group structure. This justifies the use of the group Lasso-type penalty, which reduces the localization error by bringing down d_0^2 to d_0. Second, using condition (8), there is one and only one true change point in every working interval used by the local refinement algorithm. The true change points can then be estimated separately using K independent searches, in such a way that the final localization rate that does not depend on the number of searches, namely K.

2.3.3 Comparisons

We now discuss how our contributions compared with the results of Wang et al. (2019) and of Leonardi and Bühlmann (2016), which investigate the same high-dimensional change-point linear regression model.

Wang et al. (2019) proposed different algorithms, all of which are variants of wild binary segmentation, with or without additional Lasso estimation procedures. Those methods inherit both the advantages and the disadvantages of WBS. Compared with dynamic programming, WBS-based methods require additional tuning parameters such as randomly selected intervals as inputs. With these additional tuning parameters, Theorem 1 in Wang et al. (2019) achieved the same statistical accuracy in terms of the localization error rate as Theorem 1 above. In terms of computational cost, the methods in Wang et al. (2019) are of order $O(K^2 n \cdot \mathrm{Lasso}(n))$, where K, n and $\mathrm{Lasso}(n)$ denote the number of change points, the sample size and the computational cost of Lasso algorithm with sample size n, respectively, while the dynamic programming approach of this paper is of order $O(n^2 \cdot \mathrm{Lasso}(n))$. Thus, when $K \lesssim \sqrt{n}$, the algorithm in Wang et al. (2019) is computationally more efficient, but when $K \gtrsim \sqrt{n}$, the method in this paper has smaller complexity.

Leonardi and Bühlmann (2016) analysed two algorithms, one based on a dynamic programming approach, and the other on binary segmentation, and claimed that they both yield the same localization, which is, in our notation,

$$\sum_{k=1}^{K} |\hat{\eta}_k - \eta_k| \lesssim \frac{d_0^2 \sqrt{n \log(np)}}{\kappa^2}. \tag{9}$$

Note that, the error bound in Leonardi and Bühlmann (2016) is originally of the form $\sum_{k=1}^{K} |\hat{\eta}_k - \eta_k| \lesssim \frac{d_0 \sqrt{n \log(np)}}{\kappa^2}$ under a slightly stronger assumption than ours. In the more general settings of Assumption 1, the localization error bound of Leonardi and Bühlmann (2016) is of the form (9), based on personal communication with the authors.

It is not immediate to directly compare the sum of all localization errors, used by Leonardi and Bühlmann (2016), with the maximum localization error, which is the target in this paper. Using a worst-case upper bound, Theorem 1 yields that

$$\sum_{k=1}^{K} |\hat{\eta}_k - \eta_k| \lesssim \frac{K^2 d_0^2 \sigma_\varepsilon^2 \log(n \vee p)}{\kappa^2}.$$

In light of Corollary 1, this error bound can be sharpened, using the local refinement Algorithm 1 to

$$\sum_{k=1}^{K} |\hat{\eta}_k - \eta_k| \lesssim \frac{K d_0 \sigma_\varepsilon^2 \log(n \vee p)}{\kappa^2}.$$

As long as $K^2 \lesssim \sqrt{\frac{n}{\log(np)}}$, or, using the local refinement algorithm, $K \lesssim \sqrt{\frac{n}{\log(np)}}$, our localization rates are better than the one implied by (9). It is not immediate to compare directly the assumptions used in Leonardi and Bühlmann (2016) with the ones we formulate here due to the different ways we use to present them. For instance, the conditions in Theorem 3.1 of Leonardi and Bühlmann (2016) imply, in our notation, that condition

$$\Delta \gtrsim \sqrt{n\log(p)}$$

is needed for consistency, even if the sparsity parameter $d_0 = \Theta(1)$. However in our case, in view of (5), if we assume $d_0 = \kappa = \Theta(1)$, then we only require $\Delta \gtrsim \log^{1+\xi}(n \vee p)$ for consistency.

2.3.4 Lower Bounds

In Section 2.3.2, we show that as long as

$$\kappa^2\Delta \gtrsim d_0^2 K \sigma_\varepsilon^2 \log^{1+\xi}(n \vee p),$$

we demonstrate provide change point estimators with localization errors upper bounded by

$$d_0\sigma_\varepsilon^2\kappa^{-2}\log(n \vee p).$$

In this section, we show that no algorithm is guaranteed to be consistent in the regime

$$\kappa^2\Delta \lesssim d_0\sigma_\varepsilon^2,$$

and otherwise, a minimax lower bound on the localization errors is

$$d_0\sigma_\varepsilon^2\kappa^{-2}.$$

These findings are formally stated next, in Lemmas 1 and 2, respectively.

Lemma 1. *Let $\{(x_t, y_t)\}_{t=1}^T \subset \mathbb{R}^p \times \mathbb{R}$ satisfy Model 1 and Assumption 1, with $K = 1$. In addition, assume that $\{x_t\}_{t=1}^n \overset{iid}{\sim} \mathcal{N}(0, I_p)$ and $\{\varepsilon_t\}_{t=1}^n \overset{iid}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2)$. Let $P^T_{\kappa,\Delta,\sigma_\varepsilon,d}$ be the corresponding joint distribution. For any $0 < c < \frac{2}{8e+1}$, consider the class of distributions*

$$\mathcal{P} = \Big\{P^T_{\kappa,\Delta,\sigma_\varepsilon,d} : \Delta = \min\left\{\lfloor c d_0\sigma_\varepsilon^2\kappa^{-2}\rfloor, \lfloor T/4\rfloor\right\}, \\ 2cd_0\max\{d_0, 2\} \le \Delta\Big\}.$$

There exists a $T(c)$, which depends on c, such that for all $T \ge T(c)$,

$$\inf_{\widehat{\eta}}\sup_{P\in\mathcal{P}} \mathbb{E}_P(|\widehat{\eta} - \eta(P)|) \ge \Delta,$$

where $\eta(P)$ is the location of the change point of distribution P and the infimum is over all estimators of the change point.

Lemma 1 shows that if $\kappa^2\Delta \lesssim d_0\sigma_\varepsilon^2$, then

$$\frac{\inf_{\widehat{\eta}}\sup_{P\in\mathcal{P}}\mathbb{E}_P(|\widehat{\eta}-\eta(P)|)}{\Delta} \geq 1,$$

which implies that the localization error is not a vanishing fraction of Δ as the sample size grows unbounded.

Lemma 2. *Let $\{(x_t, y_t)\}_{t=1}^T \subset \mathbb{R}^p \times \mathbb{R}$ satisfy Model 1 and Assumption 1, with $K = 1$. In addition, assume $\{x_t\}_{t=1}^n \overset{iid}{\sim} \mathcal{N}(0, I_p)$ and $\{\varepsilon_t\}_{t=1}^n \overset{iid}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2)$. Let $P^T_{\kappa,\Delta,\sigma_\varepsilon,d}$ be the corresponding joint distribution. For any diverging sequence ζ_T, consider the class of distributions*

$$\mathcal{P} = \left\{P^T_{\kappa,\Delta,\sigma_\varepsilon,d} : \Delta = \min\left\{\lfloor\zeta_T d_0\sigma_\varepsilon^2\kappa^{-2}\rfloor, \lfloor T/4\rfloor\right\}\right\}.$$

Then

$$\inf_{\widehat{\eta}}\sup_{P\in\mathcal{P}}\mathbb{E}_P(|\widehat{\eta}-\eta(P)|) \geq \frac{cd_0\sigma_\varepsilon^2}{\kappa^2},$$

where $\eta(P)$ is the location of the change point of distribution P, the infimum is over all estimators of the change point and $c > 0$ is an absolute constant.

Recalling all the results we have obtained, the change point localization task is either impossible when $\kappa^2\Delta \lesssim d_0\sigma_\varepsilon^2$ (in the sense that no algorithm is guaranteed to be consistent) or, when $\kappa^2\Delta \gtrsim d_0^2K\sigma_\varepsilon^2\log^{1+\xi}(n\vee p)$, it can be solved by our algorithms at nearly a minimax optimal rate.

In the intermediate case

$$d_0\sigma_\varepsilon^2 \lesssim \kappa^2\Delta \lesssim d_0^2K\sigma_\varepsilon^2\log^{1+\xi}(n\vee p)$$

we are unable to provide a result one way or another. However, we remark that, if in addition, in Theorem 1, we assume $\kappa \leq C$, for an absolute constant $C > 0$, then we are able to weaken the condition from $\kappa^2\Delta \gtrsim d_0^2K\sigma_\varepsilon^2\log^{1+\xi}(n\vee p)$ to $\kappa^2\Delta \gtrsim d_0K\sigma_\varepsilon^2\log^{1+\xi}(n\vee p)$, by almost identical arguments. This shows that, under the additional conditions

$$\max\{\kappa,\, K\} \leq C,$$

for an absolute constant $C > 0$, the condition is nearly optimal, off by a logarithmic factor.

2.4 Numerical Experiments

In this section, we investigate the numerical performances of our proposed methods, with efficient binary segmentation algorithm (EBSA) of Leonardi and Bühlmann (2016) as the competitor. We compare four methods: dynamic programming (DP, see 1, 2, and 3), EBSA, local refinement (Algorithm 1) initialized by DP (DP.LR), and local refinement (Algorithm 1) initialized by EBSA (EBSA.LR).

The evaluation metric considered is the scaled Hausdorff distance between the estimators $\{\hat{\eta}_k\}_{k=1}^{\hat{K}}$ and the truth $\{\eta_k\}_{k=1}^{K}$. To be specific, we report $d(\hat{\mathcal{C}}, \mathcal{C}) = n^{-1}\mathcal{D}(\hat{\mathcal{C}}, \mathcal{C})$, where

$$\mathcal{D}(\hat{\mathcal{C}}, \mathcal{C}) = \max\{\max_{\hat{\eta}\in\hat{\mathcal{C}}} \min_{\eta\in\mathcal{C}} |\hat{\eta} - \eta|, \max_{\eta\in\mathcal{C}} \min_{\hat{\eta}\in\hat{\mathcal{C}}} |\hat{\eta} - \eta|\},$$

$\mathcal{C} = \{\eta_k\}_{k=1}^{K}$ and $\hat{\mathcal{C}} = \{\hat{\eta}_k\}_{k=1}^{\hat{K}}$.

We consider both simulated data and a real-life public dataset on air quality indicators in Taiwan.

2.4.1 Tuning Parameter Selection

We adopt a cross-validation approach to choosing tuning parameters. Let samples with odd indices be the training set and even ones be the validation set. Recall that for the DP, we have two tuning parameters λ and γ, which we tune using a brute-force grid search. For each pair of tuning parameters, we conduct DP on the training set and obtain estimated change points. Within each estimated segment of the training set, we obtain $\hat{\beta}_t$ by (3). On the validation set, let $\hat{y}_t = x_t^\top \hat{\beta}_t$ and calculate the validation loss $(n/2)^{-1} \sum_{t \mod 2 \equiv 0} (\hat{y}_t - y_t)^2$. The pair (λ, γ) is chosen to be the one corresponding to the lowest validation loss.

As for the simulated data, we use some prior knowledge of the truth to save some computational cost. To be specific, we let the odd index set be partitioned by the true change points and estimate β_t on these intervals. We then plot the mean squared errors of $\hat{\beta}_t$ across a range of values of λ and obtain an "optimal" λ. We choose the grid range of λ around the "optimal" λ. This step is to approximately locate the range of λ's value but this step will not be used in the real data experiment. The same procedure is conducted for the tuning parameter selection in EBSA.

For the local refinement algorithm, we let the estimated change points of DP or EBSA be the initializers of the local refinement algorithm. We then regard the initialization algorithm and local refinement as a self-contained method and tune all three parameters λ, γ, and ζ jointly. The tuning procedure is almost the same as we described above, except that we use

$$\widehat{\beta}_I^\lambda = \arg\min_{v \in \mathbb{R}^p} \left\{ \sum_{t \in I} (y_t - x_t^\top v)^2 + \zeta \sqrt{|I|} \|v\|_1 \right\}$$

to estimate β_t.

2.4.2 Simulations

Throughout this section, we let $n = 600$, $p = 200$, $K = 4$, $\Sigma = I$ and $\sigma_\epsilon = 1$. The true change points are at 121, 221, 351 and 451. Let $\beta_0 = (\beta_{0i}, i = 1, \ldots, p)^\top$, with $\beta_{0i} = 2^{-1} d_0^{-1/2} \kappa$, $i \in \{1, \ldots, d_0\}$, and zero otherwise. Let

$$\beta_t = \begin{cases} \beta_0, & t \in \{1, \ldots, 120\}, \\ -\beta_0, & t \in \{121, \ldots, 220\}, \\ \beta_0, & t \in \{221, \ldots, 350\}, \\ -\beta_0, & t \in \{351, \ldots, 450\}, \\ \beta_0, & t \in \{451, \ldots, 600\}. \end{cases}$$

We let $\kappa \in \{4, 5, 6\}$ and $d_0 \in \{10, 15, 20\}$. For each pair of κ and d_0, the experiment is repeated 100 times. The results are reported in Table 1 and Figure ??

Generally speaking, DP outperforms EBSA, and LR significantly improves upon EBSA and DP when DP doesn't give accurate results. LR is comparable with DP when the initial points estimated by DP are already good enough. Note that since for EBSA.LR we tune EBSA to optimize EBSA.LR's performance, it may well happen that the estimated number of change points K from EBSA.LR is much different than from EBSA.

Table 1: Scaled Hausdorff Distance. The numbers in the brackets indicate the corresponding standard errors of the scaled Hausdorff distance.

Setting	Cases	DP	DP.LR	EBSA	EBSA.LR
$\kappa = 4, d_0 = 10$	All	0.023(0.015)	0.008(0.004)	0.104(0.031)	0.034(0.045)
$\kappa = 4, d_0 = 15$		0.031(0.020)	0.017(0.047)	0.104(0.029)	0.038(0.050)
$\kappa = 4, d_0 = 20$		0.038(0.032)	0.019(0.042)	0.104(0.027)	0.036(0.051)
$\kappa = 4, d_0 = 10$	$\hat{K} = K$	0.022(0.015)	0.008(0.004)	0.061(0.047)	0.008(0.008)
$\kappa = 4, d_0 = 15$		0.025(0.018)	0.008(0.007)	0.071(0.045)	0.010(0.016)
$\kappa = 4, d_0 = 20$		0.028(0.020)	0.014(0.012)	0.076(0.048)	0.010(0.011)
$\kappa = 5, d_0 = 10$	All	0.022(0.022)	0.007(0.004)	0.102(0.033)	0.033(0.046)
$\kappa = 5, d_0 = 15$		0.025(0.023)	0.015(0.025)	0.102(0.030)	0.027(0.042)
$\kappa = 5, d_0 = 20$		0.030(0.027)	0.016(0.030)	0.102(0.030)	0.041(0.048)
$\kappa = 5, d_0 = 10$	$\hat{K} = K$	0.020(0.015)	0.007(0.004)	0.068(0.073)	0.007(0.008)
$\kappa = 5, d_0 = 15$		0.021(0.012)	0.010(0.006)	0.075(0.049)	0.007(0.008)
$\kappa = 5, d_0 = 20$		0.025(0.018)	0.010(0.007)	0.076(0.065)	0.010(0.012)
$\kappa = 6, d_0 = 10$	All	0.009(0.010)	0.007(0.004)	0.100(0.028)	0.034(0.049)
$\kappa = 6, d_0 = 15$		0.022(0.017)	0.009(0.005)	0.101(0.029)	0.037(0.049)
$\kappa = 6, d_0 = 20$		0.023(0.017)	0.010(0.006)	0.102(0.031)	0.028(0.043)
$\kappa = 6, d_0 = 10$	$\hat{K} = K$	0.009(0.010)	0.007(0.004)	0.061(0.064)	0.006(0.010)
$\kappa = 6, d_0 = 15$		0.022(0.017)	0.009(0.005)	0.064(0.050)	0.007(0.007)
$\kappa = 6, d_0 = 20$		0.023(0.017)	0.010(0.006)	0.076(0.041)	0.009(0.013)

2.4.3 Air Quality Data

In this subsection, we consider the air quality data from https://www.kaggle.com/nelsonchu/air-quality-in-northern-taiwan. It collects environment information and air quality data from Northern Taiwan in 2015. We choose the PM10 in Banqiao as the response variables, with covariates being the temperature, the CO level, the NO level, the NO_2 level, the NO_x level, the rainfall quantity, the humidity quantity, the SO_2 level, the ultraviolet index, the wind speed, the wind direction and the PM10 levels in Guanyin, Longtan, Taoyuan, Xindian, Tamsui, Wanli and Keelung District. We transfer the hourly data into daily by averaging across 24 hours. After removing all dates containing missing values, we obtain a data set with $n = 343$ days and $p = 18$ covariates. Our goal is to detect potential change points of this data set and determine if they are consistent with the historical information.

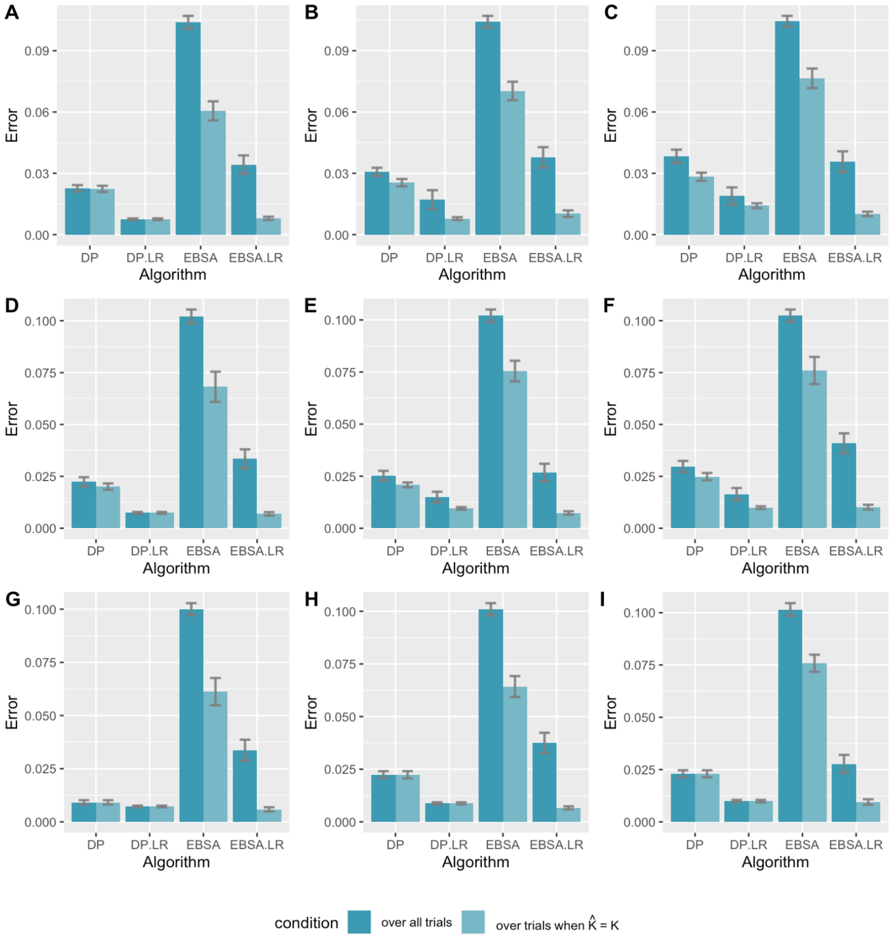

Figure 1: Bar plots for results in Table 1. Plots A-C are settings with $\kappa = 4$ and $d_0 \in \{10, 15, 20\}$; Plots D-F are settings with $\kappa = 5$ and $d_0 \in \{10, 15, 20\}$; and Plots G-I are settings with $\kappa = 6$ and $d_0 \in \{10, 15, 20\}$.

We standardize the data so that the variance of $y_t = 1$, for all t. We then conduct DP, DP.LR, EBSA and EBSA.LR. DP estimates 2 change points which are March 16th and November 1st, 2015. No change points are detected by EBSA. DP.LR and EBSA.LR both detect May 15th and October 25th, 2015 as the change points.

The first change point detected by DP.LR and EBSA.LR seems to correspond with the first strong-enough typhoon near Northern Taiwan in 2015, which happened during May 6th-20th (e.g. Wikipedia, 2020a). The second change points from EBSA.LR, DP.LR and DP are relatively close and they all could be explained by the severe air pollution at the beginning of November in Taiwan, which reached the hazardous purple alert on November 8th (e.g. Wikipedia, 2020b). The visualization is shown in Figure 2.

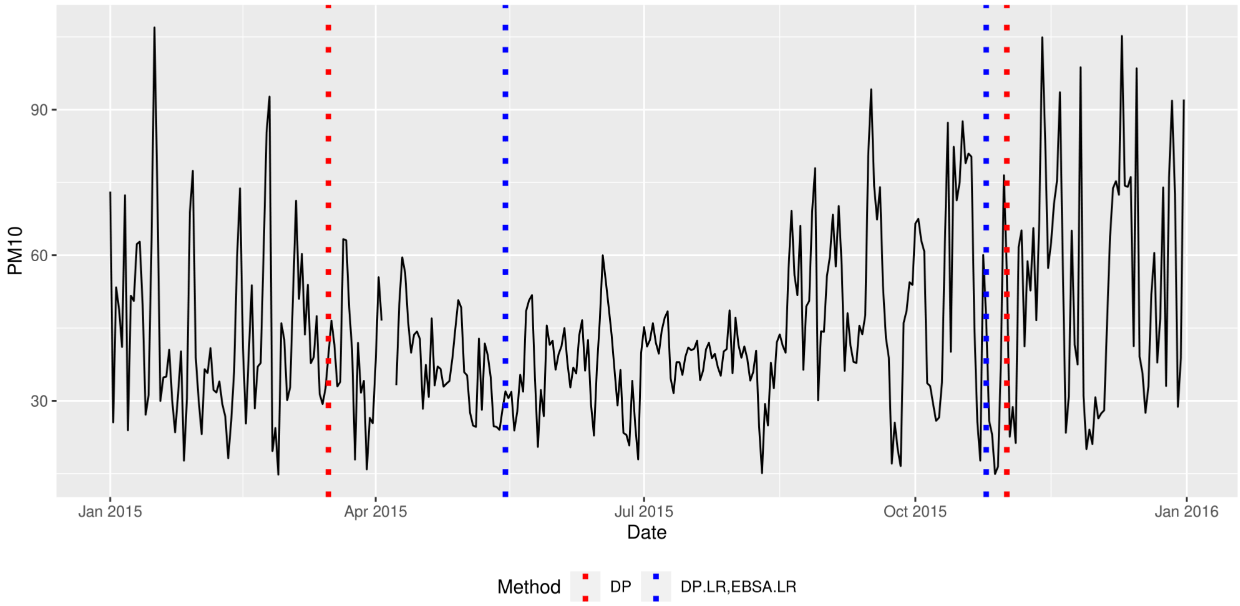

Figure 2: Real air quality example. When we tune EBSA for its standalone performance, we obtain zero change points, but when we tune EBSA to optimize the EBSA.LR's performance, we obtain two change points.

2.5 Discussion

In this paper, we in fact provide a general framework for analyzing general regression-type change point localization problems that include the linear regression model above as a special case. The analysis in this paper may be utilized as a blueprint for more complex change point localization problems. In our analysis, we develop a new and refined toolbox for the change point detection community to study more complex data generating mechanisms above and beyond linear regression models.

3 Functional Autoregressive Processes in Reproducing Kernel Hilbert Spaces and the 2 dimensional Extension

3.1 Introduction

Functional data analysis (FDA) deal with data in the form of curves, surfaces or anything else varying over a continuum. As described in Ramsay and Silverman (2005), Ferraty and Vieu (2006) and Horváth and Kokoszka (2012), FDA framework considers each sample element to be a function, offering a natural solution to a variety of problems which are not suitable for the classical statistical frameworks.

When the functional data is observed in a sequential manner, it is called functional time series (Hörmann and Kokoszka, 2010; Aue et al., 2015). Formally speaking, a functional time series takes the form $\{X_t(s), s \in [a,b]\}_{t\in\mathbb{Z}}$, where each observation $X_t(s)$ is a (random) function defined for s taking values in some compact interval $[a,b]$. Throughout the paper, we assume that $[a,b] = [0,1]$. It may well happen that the classical vector autoregressive models would fail to track the dynamics of functional time series in high-dimensional spaces. (Bosq (2000), Hyndman and Ullah (2007) and Shang (2013)). Therefore, an important goal for functional time series analysis is the estimation and prediction of a reliable statistical model, which will offer a better understanding of the behavior of the data and providing accurate prediction for future realization.

In this paper, we are going to talk about a widely-used functional time series model: functional autoregressive(FAR) process. Formally speaking, a functional time series $\{X_t\}$ follows an FAR process (in $\mathcal{L}^2$) of order D if

$$X_t(\cdot) = \mu(\cdot) + \sum_{d=1}^{D} \Psi_d(X_{t-d})(\cdot) + \epsilon_t(\cdot), \tag{10}$$

where $\mu(\cdot) \in \mathcal{L}^2$ is a deterministic function, $\epsilon_t(\cdot) \in \mathcal{L}^2$ are *i.i.d.* zero-mean noise functions, and $\{\Psi_d\}_{d=1}^{D}$ are bounded linear operators mapping $\mathcal{L}^2 \to \mathcal{L}^2$. An important goal of this time series analysis is providing reliable predictions for future realizations.

Dimension reduction is mainly used to address the inference of FAR in the existing literature due to the infinite-dimensional nature of the functional space $\mathcal{L}^2$. Most literature reduces the dimension via functional principal component analysis (FPCA), where the finite basis is chosen as the leading p functional principal components (FPC) of an estimated covariance operator of the underlying FAR process. More works on FPCA-based approaches can be found at Besse and Cardot (1996), Bosq (2000), Besse et al. (2000), Hyndman and Shang (2009), Didericksen et al. (2012) and Aue et al. (2015).

Existing estimation methods and theoretical results require fully observed functional time series $\{X_t(s), s \in [0,1]\}_{t=1}^T$. However, in reality, this is an unrealistic assumption as the FAR process is measured discretely and observations instead take the form $\{X_t(s_i), 1 \le i \le n\}_{t=1}^T$, where $\{s_i\}_{i=1}^n$ denotes n discrete grid points in $[0,1]$. In practice, the FPCA-based methods typically rely on an extra smoothing step to convert discrete measurements $\{X_t(s_i), 1 \le i \le n\}_{t=1}^T$ into (estimated) fully functional data $\{\widetilde{X}_t(s), s \in [0,1]\}_{t=1}^T$, and the statistical analysis is performed on $\{\widetilde{X}_t(s), s \in [0,1]\}_{t=1}^T$. This will bring the smoothing error when establishing theoretical guarantees.

Wang et al. (2020a) proposed new estimation and prediction procedures for FAR without dimension reduction through the lens of Reproducing Kernel Hilbert Spaces (RKHS, Wahba (1990)). To be more specific, the paper considered a refined FAR process in RKHS,

$$X_t(\cdot) = \mu(\cdot) + \sum_{d=1}^{D} \int_0^1 A_d(\cdot, s) X_{t-d}(s) ds + \epsilon_t(\cdot), \tag{11}$$

where the bounded linear operators $\{\Psi_d\}_{d=1}^D$ take the explicit form of integral operators with bivariate kernels $\{A_d(r,s) : [0,1] \times [0,1] \to \mathbb{R}\}_{d=1}^D$. The paper construct theoretical guarantee for estimation and prediction of FAR processes based on discrete observations of functional time series.

A limitation of this paper is that they used the accelerated gradient method(AGM) to solve the optimization problem. Notice that this algorithm does singular value decomposition in every iteration, which will be time-consuming, especially when we would like to apply this framework into multi-dimension. Therefore, in our paper, we explored the complexity and regularization of the regression problem to develop a computational efficient algorithm, and extended the 1 dimensional framework into multi-dimension. We also extended the fPCA method into 2 dimensional scenario, which serves as a baseline method and applied them to our problem.

The rest of the paper is organized as follows. Section 3.2 proposes the penalized Frobenius norm estimator for FAR and studies its theoretical properties. The performance of the 1 dimensional proposed method over existing procedures is demonstrated via extensive numerical experiments in Section 3.3. Section 3.4 extend both methods based on regression and based on fPCA into 2-dimension. The performance of 2 dimensional methods can be found in section 3.5.

Some notations used throughout the paper are defined as follows. Denote $\|f\|_{\mathcal{L}^2}^2 = \langle f, f\rangle_{\mathcal{L}^2}$ and $\|f\|_\infty := \sup_{s\in[0,1]} |f(s)|$. For a matrix W, denote $\|W\|_F$ as its Frobenius norm and $\|W\|_*$ as its trace norm. We omit $[0, 1]$ in the integral whenever the domain of functions is clear.

3.2 Estimation Methodology and Main Results

Section 3.2.1 proposes the RKHS-based penalized estimation procedure for the transition operators $\{A_d^*\}_{d=1}^D$ with discrete realizations of $\{X_t\}_{t=1}^T$. Section 3.2.2 formulates the penalized estimation as a Frobenius norm minimization problem rather than nuclear norm as in Wang et al. (2020a) and discusses its numerical implementation.

Let us define the FAR(D) process in RKHS.

Definition 1. *A functional time series $\{X_t\}_{t=1}^T \subset \mathcal{H}$ is said to follow a functional autoregressive process of order D in $\mathcal{H}$(FAR(D)) for an RKHS $\mathcal{H}$, if*

$$X_t(r) = \sum_{d=1}^{D} \int A_d^*(r, s)X_{t-d}(s)ds + \epsilon_t(r), \text{ for } r \in [0, 1], \tag{12}$$

where $\{\epsilon_t\}_{t=1}^T \subset \mathcal{H}$ is a collection of i.i.d. functional noise and the transition operators $\{A_d^*\}_{d=1}^D \subset \mathcal{C}$ are compact linear operators on $\mathcal{H}$.

3.2.1 Penalized estimation and Representer theorem

Existing literature requires fully observed functional time series, but this is impractical because the FAR process is measured discretely in reality. Following the standard RKHS literature, we assume $\{s_i\}_{i=1}^n$ to be a collection of random variables uniformly sampled from $[0,1]$. Given the discrete observations $\{X_t(s_i)\}_{1\le t\le T, 1\le i\le n}$, our goal is to estimate the D unknown transition operators $\{A_d^*\}_{d=1}^D$, which will facilitate the prediction task.

We construct the estimator $\{\widehat{A}_d\}_{d=1}^D$ through a penalized Frobenius norm optimization such that

$$\{\widehat{A}_d\}_{d=1}^D = \underset{\{A_d\}_{d=1}^D \in \mathcal{C}_{\boldsymbol{\tau}}}{\arg\min} \frac{1}{Tn} \sum_{t=D+1}^{T} \sum_{i=1}^{n} \left(X_t(s_i) - \sum_{d=1}^{D} \frac{1}{n} \sum_{j=1}^{n} A_d(s_i, s_j) X_{t-d}(s_j) \right)^2 \tag{13}$$

where $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_D)$ is the tuning parameter and $\mathcal{C}_{\boldsymbol{\tau}} := \{(A_1, \cdots, A_D) : A_d \in \mathcal{C}$ and $\|A_d\|_{\mathcal{H},\mathcal{F}} \le \tau_d,\ d = 1, \cdots, D\}$ is the constraint space. We name $\{\widehat{A}_d\}_{d=1}^D$ in (13) the constrained Frobenius norm estimator for transition operators of FAR.

The motivation for the formulation of (13), is that consider the integral scenario (ideal yet infeasible) in which $\{X_t\}_{t=1}^T$ are fully observed in the entire domain $[0,1]$, thus we can solve

$$\{\widetilde{A}_d\}_{d=1}^D = \underset{\{A_d\}_{d=1}^D \in \mathcal{C}_{\boldsymbol{\tau}}}{\arg\min} \frac{1}{T} \sum_{t=D+1}^{T} \int \left(X_t(r) - \sum_{d=1}^{D} \int A_d(r,s) X_{t-d}(s) ds \right)^2 dr.$$

After using the integral approximation $\int A_d(s_i, r) X_{t-d}(r) dr \approx \frac{1}{n} \sum_{j=1}^n A_d(s_i, s_j) X_{t-d}(s_j)$, we instead solve (13) since only discrete data $\{X_t(s_i)\}_{1\le t\le T, 1\le i\le n}$ are observed.

Proposition 1 (Representer theorem)**.** *There exists a minimizer $\{\widehat{A}_d\}_{d=1}^D$ of the constrained Frobenius/nuclear norm optimization* (13) *such that for any* $(r,s) \in [0,1] \times [0,1]$,

$$\widehat{A}_d(r,s) = \sum_{1\le i,j\le n} \widehat{a}_{d,ij} \mathbb{K}(r, s_i) \mathbb{K}(s, s_j), \ \textit{for } d = 1, 2, \cdots, D. \tag{14}$$

Wang et al. (2020a) propose a representer theorem of the nuclear norm optimization, whose proof is also suitable to establish the Representer theorem for Frobenius norm optimization.

After obtaining the estimated transition operators $\{\widehat{A}_d\}_{d=1}^D$, the one-step ahead prediction of X_{T+1} can be computed as

$$\widehat{X}_{T+1}(r)=\sum_{d=1}^{D}\frac{1}{n}\sum_{j=1}^{n}\widehat{A}_d(r,s_j)X_{T+1-d}(s_j) \text{ for } r\in[0,1]. \tag{15}$$

3.2.2 Optimization for 1 Dimension Scenario

In this section, we discuss the numerical implementation of the proposed RKHS-based penalized estimator by reformulating the constrained optimization in (13) into a Frobenius norm minimization problem.

We first introduce some notations. Denote the estimator $A_d(r,s)=\sum_{1\le i,j\le n}a_{d,ij}\mathbb{K}(r,s_i)\mathbb{K}(s,s_j)$, where $a_{d,ij}$s are the coefficients to be estimated. Define the coefficient matrix $R_d\in\mathbb{R}^{n\times n}$ with $R_{d,ij}=a_{d,ij}$. Define the kernel vector $k_i=(\mathbb{K}(s_1,s_i),\mathbb{K}(s_2,s_i),\cdots,\mathbb{K}(s_n,s_i))^\top$ and the kernel matrix $K=[k_1,k_2,\cdots,k_n]$. Note that the kernel matrix K is symmetric such that $K=K^\top$. Denote the observation of the functional time series at time t as $X_t=(X_t(s_1),X_t(s_2),\cdots,X_t(s_n))^\top$. Define the observation matrix $X=[X_T,X_{T-1},\cdots,X_{D+1}]$ and the lagged observation matrix $X^{(d)}=[X_{T-d},X_{T-d-1},\cdots,X_{D+1-d}]$ for $d=1,\cdots,D$.

Using the well-known equivalence between constrained and penalized optimization (see Hastie et al. (2009)), we can reformulate (13) into a penalized Frobenius norm optimization such that

$$\{\widehat{A}_d\}_{d=1}^{D}=\arg\min\sum_{t=D+1}^{T}\sum_{i=1}^{n}\left(X_t(s_i)-\sum_{d=1}^{D}\frac{1}{n}\sum_{j=1}^{n}A_d(s_i,s_j)X_{t-d}(s_j)\right)^2+\sum_{d=1}^{D}\lambda_d\|A_d\|^2_{\mathcal{H},\mathcal{F}},$$

where $(\lambda_1,\cdots,\lambda_D)$ is the tuning parameter. With simple linear algebra, we can rewrite the penalized optimization as

$$\begin{aligned}&\min_{R_1,\cdots,R_D}\sum_{t=D+1}^{T}\left(X_t-\frac{1}{n}\sum_{d=1}^{D}K^\top R_dKX_{t-d}\right)^\top\left(X_t-\frac{1}{n}\sum_{d=1}^{D}K^\top R_dKX_{t-d}\right)+\sum_{d=1}^{D}\lambda_d\|A_d\|^2_{\mathcal{H},\mathcal{F}}\\ =&\min_{R_1,\cdots,R_D}\left\|X-\frac{1}{n}\sum_{d=1}^{D}KR_dKX^{(d)}\right\|_F^2+\sum_{d=1}^{D}\lambda_d\|A_d\|^2_{\mathcal{H},\mathcal{F}}.\end{aligned} \tag{16}$$

We now write the Frobenius norm $\|A_d\|_{\mathcal{H},\mathcal{F}}$ as a function of R_d. By the Representer theorem, $A_d(r,s)=\sum_{i,j}a_{d,ij}\mathbb{K}(r,s_i)\mathbb{K}(s,s_j)$, thus the adjoint operator $A_d^\top(r,s)=A_d(s,r)$. Define $k(s)=(\mathbb{K}(s,s_1),\mathbb{K}(s,s_2),\cdots,\mathbb{K}(s,s_n))^\top$, we have $A_d(r,s)=k(r)^\top R_dk(s)$ and $\langle k(s),k(s)^\top\rangle_{\mathcal{H}}=$

K. Define $u(s) = k(s)^\top b$, where $b = (b_1, b_2, \cdots, b_n)^\top$. To calculate $\|A_d\|_{\mathcal{H},\mathcal{F}}$, note that

$$\begin{aligned} A_d^\top A_d[u](s) &= \langle A_d^\top(s,r), A_d[u](r)\rangle_{\mathcal{H}} = \langle A_d(r,s), \langle A_d(r,s), u(s)\rangle_{\mathcal{H}}\rangle_{\mathcal{H}} \\ &= \langle k(r)^\top R_d k(s), \langle k(r)^\top R_d k(s), k(s)^\top b\rangle_{\mathcal{H}}\rangle_{\mathcal{H}} = k(s)^\top R_d^\top K R_d K b. \end{aligned}$$

In other words, the eigenvalues of the operator $A_d^\top A$ correspond to the eigenvalues of the matrix $R_d^\top K R_d K$. Thus, (16) can be further written as

$$\begin{aligned} &\min_{R_1,\cdots,R_D} \left\| X - \frac{1}{n}\sum_{d=1}^{D} K R_d K X^{(d)} \right\|_F^2 + \sum_{d=1}^{D} \lambda_d \cdot \mathrm{trace}((R_d^\top K R_d K)) \\ =& \min_{R_1,\cdots,R_D} \left\| X - \frac{1}{n}\sum_{d=1}^{D} K R_d K X^{(d)} \right\|_F^2 + \sum_{d=1}^{D} \lambda_d \|K^{\frac{1}{2}} R_d K^{\frac{1}{2}}\|_{\mathcal{F}}^2 \\ =& \min_{W_1,\cdots,W_D} \left\| X - \sum_{d=1}^{D} \mathcal{K} W_d Z_d \right\|_F^2 + \sum_{d=1}^{D} \|W_d\|_{\mathcal{F}}^2 \end{aligned} \tag{17}$$

where $W_d = K^{\frac{1}{2}} R_d K^{\frac{1}{2}}$, $\mathcal{K} = K^{\frac{1}{2}}$, $Z_d = \frac{1}{n} K^{\frac{1}{2}} X^{(d)}$ and the first equality comes from the fact that $R_d^\top K R_d K$ and $K^{1/2} R_d^\top K R_d K^{1/2}$ share the same eigenvalues for $d = 1, \cdots, D$.

In the following experiment, we set d = 1. Hence, the objective function becomes

$$\begin{aligned} &\min_{R} \left\| X - \frac{1}{n} K R K X^{(1)} \right\|_F^2 + \lambda \cdot \|K^{\frac{1}{2}} R K^{\frac{1}{2}}\|_{\mathcal{F}}^2 \\ =& \min_{W} \left\| X - \frac{1}{n} K^{\frac{1}{2}} W K^{\frac{1}{2}} X^{(1)} \right\|_F^2 + \lambda \|W\|_{\mathcal{F}}^2 \\ =& \min_{W} \|X - \mathcal{K} W Z\|_F^2 + \lambda \|W\|_{\mathcal{F}}^2 \end{aligned} \tag{18}$$

Note that (18) is a convex function of W with a unique global minimizer.

Here we provide two solutions for this objective function.

Solution 1: Iterative optimization

We can rewrite (18) as

$$G(W) = \min_{W} \mathrm{trace}((X - \mathcal{K}WZ)^T (X - \mathcal{K}WZ)) + \lambda \mathrm{trace}(W^T W)$$

Taking the derivative with respect to W,

$$\nabla G(W) = \lambda W + KWZZ^T - Z\mathcal{K} = \lambda W + KWZZ^T - ZK^{\frac{1}{2}}$$

We find the optimality by using gradient descent.

Solution 2: Find the explicit solution of (18) by vectorizing the matrices

Vectorizing the matrices inside (18), the objective function is equivalent to

$$\begin{aligned} g(w) &= \min_{w=\mathrm{vec}(W)} \left\| x - \frac{1}{n}\mathrm{vec}(K^{\frac{1}{2}} W K^{\frac{1}{2}} X^{(1)}) \right\|_2^2 + \lambda \left\| w \right\|_2^2 \\ &= \min_{w=\mathrm{vec}(W)} \left\| x - \frac{1}{n}(X^{(1)^T} K^{\frac{1}{2}}) \otimes K^{\frac{1}{2}} w \right\|_2^2 + \lambda \left\| w \right\|_2^2, \end{aligned} \tag{19}$$

where $x = \mathrm{vec}(X), w = \mathrm{vec}(W)$.

Denote $S = \frac{1}{n}(X^{(1)^T} K^{\frac{1}{2}}) \otimes K^{\frac{1}{2}}$, the objective function becomes

$$g(w) = \min_{w} \left\| x - Sw \right\|_2^2 + \lambda \left\| w \right\|_2^2 \tag{20}$$

We can obtain the explicit solution by solving the linear equation

$$(S^T S + \lambda I)w = S^T x \tag{21}$$

Notice that S is a $n^2 \times n^2$ matrix, solving (21) is very likely to be problematic. Here we further exploit the structure of $S^T S + \lambda I$. Denote $S_0 = K^{\frac{1}{2}} X^{(1)}/n$, we have $S^T S = S_0 S_0^T \otimes K$.

Do SVD to $S_0 S_0^T$ and K, we have $S_0 S_0^T = U_1 D_1 U_1^T$, $K = U_2 D_2 U_2^T$, $U_1 = (U_{11}, ..., U_{1n})$, $U_2 = (U_{21}, ..., U_{2n})$.

Hence,

$$\begin{aligned} &S^T S + \lambda I = S_0 S_0^T \otimes K + \lambda I = (U_1 \otimes U_2)(D_1 \otimes D_2 + \lambda I)(U_1^T \otimes U_2^T), \\ &(S^T S + \lambda I)^{-1} = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{\lambda + D_{1i} D_{2j}} (U_{1i} U_{1i}^T) \otimes (U_{2j} U_{2j}^T), \\ &\hat{w} = (S^T S + \lambda I)^{-1} S^T x = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{\lambda + D_{1i} D_{2j}} [(U_{1i} U_{1i}^T) \otimes (U_{2j} U_{2j}^T)][S_0 \otimes K^{\frac{1}{2}}]x \end{aligned}$$

Therefore, the explicit expression of $\widehat{W}$ is

$$\widehat{W} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{\lambda + D_{1i} D_{2j}} U_{2j} U_{2j}^T (K^{\frac{1}{2}} X X^{(1)^T} K^{\frac{1}{2}}) U_{1i} U_{1i}^T \tag{22}$$

Similar techniques can be found in Wang et al. (2020b).

Given $\widehat{W}$, the estimated transition operators $\{\widehat{A}\}$ can be recovered by

$$\widehat{A}(r, s) = k(r)^\top \widehat{R} k(s) = \frac{1}{\lambda} k(r)^\top K^{-\frac{1}{2}} \widehat{W} K^{-\frac{1}{2}} k(s) \tag{23}$$

For this solution, we use generalized cross validation to select the tuning parameter λ. The

GCV score is

$$V(\lambda) = \frac{\frac{1}{nT}\left\|X - \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{\lambda+D_{1i}D_{2j}}(K^{\frac{1}{2}}U_{2j}U_{2j}^{T}K^{\frac{1}{2}})X(S_0^TU_{1i}U_{1i}^TS_0)\right\|_F^2}{(1-\frac{1}{nT}\mathrm{tr}(M(\lambda))^2}, \tag{24}$$

where

$$M(\lambda) = \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{\lambda+D_{1i}D_{2j}}\mathrm{tr}(U_{1i}U_{1i}^TS_0S_0^T)\mathrm{tr}(U_{2j}U_{2j}^TK)$$

Plugging into (15), the one-step ahead prediction of X_{T+1} is then

$$\widehat{X}_{T+1}(r) = \frac{1}{n}\frac{1}{\lambda}k(r)^\top K^{-\frac{1}{2}}\widehat{W}K^{\frac{1}{2}}X_T, \text{ for } r \in [0,1]. \tag{25}$$

3.3 1 dim Simulation Studies

In this section, we conduct simulation studies to study the estimation and prediction performance of the proposed penalized Frobenius and nuclear norm estimator and compare it with the state-of-the-art functional time series prediction method in Aue et al. (2015).

3.3.1 Basic simulation setting

Data generating process: We borrowed from the simulation setting in Aue et al. (2015). For $d = 1, 2, \cdots, D$, we assume the dth transition operator $A_d(r,s)$ is of rank q_d and is generated by q_d basis functions $\{u_i(s)\}_{i=1}^{q_d}$ such that

$$A_d(r,s) = \textstyle\sum_{i,j=1}^{q_d}\lambda_{d,ij}u_i(r)u_j(s),$$

where $\{u_i(s)\}_{i=1}^{q_d}$ consists of orthonormal basis of $\mathcal{L}^2[0,1]$ that will be specified later. Define matrix Λ_d such that $\Lambda_{d,ij} = \lambda_{d,ij}$ and define $\mathbf{u}_{q_d}(s) = (u_1(s), u_2(s), \cdots, u_{q_d}(s))^\top$. We have $A_d(r,s) = \mathbf{u}_{q_d}(r)^\top\Lambda_d\mathbf{u}_{q_d}(s)$. We further set the noise function ϵ_t to be of finite rank q_ϵ such that $\epsilon_t(s) = \sum_{i=1}^{q_\epsilon} z_{ti}u_i(s)$, where $z_{ti} \overset{i.i.d.}{\sim} U(-a_i, a_i)$ or $z_{ti} \overset{i.i.d.}{\sim} N(0, \sigma_i^2)$. Without loss of generality, we set $q_1 = q_2 = \cdots = q_D = q_\epsilon = q$ for simplicity. Hence, the FAR(D) process $\{X_t(s)\}_{t=1}^T$ nests in a finite dimensional subspace spanned by the orthonormal basis $\{u_i(s)\}_{i=1}^q$. Denote $X_t(r) = \sum_{i=1}^q x_{ti}u_i(r)$ where $x_{ti} = \int X_t(r)u_i(r)dr$, and denote $x_t =$

$(x_{t1}, \cdots, x_{tq})^\top$ and $z_t = (z_{t1}, \cdots, z_{tq})^\top$. We have

$$\begin{aligned} X_t(r) =& \sum_{d=1}^{D} \int A_d(r,s) X_{t-d}(s) ds + \epsilon_t(r) = \sum_{d=1}^{D} \int \mathbf{u}_q(r)^\top \Lambda_d \mathbf{u}_q(s) X_{t-d}(s) ds + z_t^\top \mathbf{u}_q(r) \\ =& \sum_{d=1}^{D} \int \mathbf{u}_q(r)^\top \Lambda_d \mathbf{u}_q(s) \mathbf{u}_q(s)^\top x_{t-d} ds + z_t^\top \mathbf{u}_q(r) = \mathbf{u}_q(r)^\top \left(\sum_{d=1}^{D} \Lambda_d x_{t-d} + z_t \right). \end{aligned} \tag{26}$$

For FAR(1) process, the expression simplifies to

$$X_t(r) = \mathbf{u}_q(r)^\top \left(\Lambda x_{t-1} + z_t \right) \tag{27}$$

(49) leads to $x_t = \sum_{d=1}^{D} \Lambda_d x_{t-d} + z_t$. Thus, the FAR($D$) process can be exactly simulated via a VAR(D) process. Following the simulation setting in Yuan and Cai (2010) and Sun et al. (2018), we set $u_i(s) = 1$ if $i = 1$ and $u_i(s) = \sqrt{2}\cos((i-1)\pi s)$ for $i = 2, \cdots, q$.

Given the transition operators $A_1, \cdots, A_D$ (i.e. $\Lambda_1, \cdots, \Lambda_D$) and the distribution of noise z_t, the true FAR(D) process $\{X_t(s), s \in [0,1]\}_{t=1}^T$ can be simulated and discrete measurements of the functional time series are taken at the sampling points $\{s_i\}_{i=1}^n$. For simplicity, we set $\{s_i\}_{i=1}^n$ to be the n equal-spaced points in $[0,1]$, which resembles the typical sampling scheme of functional time series in real data applications.

Evaluation criteria: We evaluate the performance of a method via (a). estimation error of $\widehat{A}_1, \widehat{A}_2, \cdots, \widehat{A}_D$ and (b). prediction error of the estimated FAR(D) model.

Specifically, given sample size (n, T), we simulate the observed functional time series $\{X_t(s_i), i = 1, \cdots, n\}_{t=1}^{T+0.2T}$, which we then partition into training data $\{X_t(s_i), i = 1, \cdots, n\}_{t=1}^{T}$ for estimation of $A_1, \cdots, A_D$ and test data $\{X_t(s_i), i = 1, \cdots, n\}_{t=T+1}^{T+0.2T}$ for evaluation of prediction performance. Denote $\{\widehat{X}_t(s_i), i = 1, \cdots, n\}_{t=T+1}^{T+0.2T}$ as the one-step ahead prediction given by the estimated FAR(D) model. We define

$$\text{MISE}(\widehat{A}_d, A_d) = \int_{[0,1]} \int_{[0,1]} (A_d(r,s) - \widehat{A}_d(r,s))^2 dr ds \Big/ \int_{[0,1]} \int_{[0,1]} A_d(r,s)^2 dr ds, \tag{28}$$

$$\text{PE} = \frac{1}{0.2nT} \sum_{t=T+1}^{T+0.2T} \sum_{i=1}^{n} (X_t(s_i) - \widehat{X}_t(s_i))^2, \tag{29}$$

where MISE (mean integrated squared error) measures the estimation error and PE measures the prediction error.

3.3.2 Estimation methods and implementation details

We implement the functional PCA (FPCA) based estimation approach for FAR: the vector autoregressive based approach in Aue et al. (2015) for comparison. The estimator makes use of the FPCA conducted on the sample covariance operator $\widetilde{C}(s,r) = \frac{1}{T}\sum_{t=1}^{T} X_t(s)X_t(r)$ such that $\widetilde{C}(s,r) = \sum_{i=1}^{\infty} \hat{\lambda}_i \hat{f}_i(s)\hat{f}_i(r)$, where $(\hat{\lambda}_i, \hat{f}_i)$ is the eigenvalue-eigenfunction pair.

Functional PCA-VAR estimator in Aue et al. (2015) [ANH]: The basic idea of Aue et al. (2015) is the combination of FPCA-based dimension reduction and the classical vector autoregressive (VAR) model, designed for prediction of FAR processes. Specifically, the infinite dimensional functional time series $\{X_t\}_{t=1}^T$ is first projected to the p eigenfunctions $\hat{f}(s) = (\hat{f}_1(s), \cdots, \hat{f}_p(s))^\top$ of the sample covariance operator. After projection, X_t is represented by a p-dimensional functional principal score $x_t = (x_{t1}, \cdots, x_{tp})^\top$ with $x_{ti} = \int X_t(s)\hat{f}_i(s)ds$. A VAR($D$) model is then fitted on the p-dimensional time series $\{x_t\}_{t=1}^T$ such that $x_t = B_1 x_{t-1} + \cdots + B_D x_{t-D} + \epsilon_t$. Denote the estimated coefficient matrices as $\hat{B}_1, \cdots, \hat{B}_D \in \mathbb{R}^{p\times p}$, the one-step ahead prediction of $X_t(s)$ is then $\widehat{X}_t(s) = \hat{f}(s)^\top \hat{x}_t = \hat{f}(s)^\top \sum_{d=1}^{D} \hat{B}_d x_{t-d}$.

Note that this implies $\hat{X}_t(s) = \hat{f}(s)^\top \hat{x}_t = \hat{f}(s)^\top \sum_{d=1}^{D} \hat{B}_d x_{t-d} = \hat{f}(s)^\top \sum_{d=1}^{D} \hat{B}_d \int \hat{f}(r) X_{t-d}(r)dr = \sum_{d=1}^{D} \int \hat{f}(s)^\top \hat{B}_d \hat{f}(r) X_{t-d}(r)dr$. Thus, the FPCA-based prediction algorithm in Aue et al. (2015) induces an estimator for the transition operators $\{A_d\}_{d=1}^D$ such that

$$\widehat{A}_d(s,r) = \hat{f}(s)^\top \hat{B}_d \hat{f}(r), \text{ for } d = 1, \cdots, D.$$

We refer to this estimator by ANH.

Implementation of FPCA-based estimators (ANH): For the implementation of ANH, the functional time series is required to be fully observed over the entire interval $[0,1]$. However, under the current simulation setting, only discrete measurements $\{X_t(s_i), i = 1, \cdots, n\}_{t=1}^T$ are available. Following Aue et al. (2015), for each t, the function $X_t(s), s \in [0,1]$ is estimated using 10 cubic B-spline basis functions based on the discrete measurements $(X_t(s_1), \cdots, X_t(s_n))$. We also use 20 cubic B-spline basis functions for more flexibility (see more details later).

Implementation of penalized nuclear norm estimator (FAR-Tr): For the implementation of the proposed RKHS-based penalized nuclear norm estimator, we use the rescaled Bernoulli polynomial as the reproducing kernel $\mathbb{K}$, such that

$$\mathbb{K}(x,y) = 1 + k_1(x)k_1(y) + k_2(x)k_2(y) - k_4(x-y),$$

where $k_1(x) = x - 0.5$, $k_2(x) = \frac{1}{2}(k_1^2(x) - \frac{1}{12})$ and $k_4(x) = \frac{1}{24}(k_1^4(x) - \frac{k_1^2(x)}{2} + \frac{7}{240})$ for $x \in [0,1]$, and $k_4(x-y) = k_4(|x-y|)$ for $x, y \in [0,1]$. Such $\mathbb{K}$ is the reproducing kernel for $W^{2,2}$. See Chapter 2.3.3 of Gu (2013) for more detail. The accelerated gradient algorithm in Ji and Ye (2009) is used to solve the trace norm minimization

Implementation of penalized Frobenius norm estimator (FAR-F2): We implemented the two solutions provided in 3.2.2, and they give the same result. However, when tuning parameter λ is relatively small, say 10^{-5}, gradient descent becomes very slow to converge. Hence, in the following simulations, we use (22) to solve the optimization problem. Generalized cross validation is applied to select the tuning parameter λ. More details are in 3.2.2. A standard 5-fold cross validation is used to select the tuning parameter λ.

Based on $\{\widehat{A}\}$, the one-step ahead prediction of $X_t(s_i)$ for $t = T+1, \cdots, T+0.2T$ in the test data can be calculated via $\widehat{X}_t(s_i) = \frac{1}{n}\sum_{j=1}^{n} \widehat{A}(s_i, s_j)X_t(s_j)$ for $i = 1, \cdots, n$ as in (15).

3.3.3 Simulation result for FAR(1)

For FAR(1), there is only one transition operator $A(r,s) = A_1(r,s)$. The simulation setting involves the transition matrix $\Lambda = \Lambda_1 \in \mathbb{R}^{q\times q}$ (signal) and the noise range $a_{1:q} = (a_1, a_2, \cdots, a_q)$ or the noise variance $\sigma_{1:q}^2 = (\sigma_1^2, \sigma_2^2, \cdots, \sigma_q^2)$ for $\{z_{ti}\}_{i=1}^q$ (driving noise). Denote $\sigma(\Lambda)$ as the leading singular value for a matrix Λ. We consider two different signal-noise settings:

- Scenario A (Diag Λ): $\Lambda = \text{diag}(\kappa, \cdots, \kappa)$ and $z_{ti} \overset{i.i.d.}{\sim} U(-a, a)$ with $a = 0.1$ for $i = 1, \cdots, q$.
- Scenario B (Random Λ): A random matrix Λ^* is first generated via $\Lambda_{ij}^* \overset{i.i.d.}{\sim} N(0,1)$ and we set $\Lambda = \kappa \cdot \Lambda^*/\sigma(\Lambda^*)$, and $z_{ti} \overset{i.i.d.}{\sim} U(-a, a)$ with $a = 0.1$ for $i = 1, \cdots, q$.

For ANH, the function $X_t(s)$ is first estimated using 10 cubic B-spline basis functions. We use 20 cubic B-splines when $q = 21$ for more flexibility. With the FAR order fixed at

$D = 1$, the fFPE criterion is used to select the number of FPCs p for ANH. For RKHS, we implemented the two solutions provided in 3.2.2, and they gives the same result. However, when tuning parameter λ is relatively small, say 10^{-5}, gradient descent becomes very slow to converge. Hence, in the following simulations, we use (22) to solve the optimization problem. Generalized cross validation is applied to select the tuning parameter λ. More details are in 3.2.2.

Numerical result for FAR(1): For Scenarios A and B, we consider three sample sizes: $(1)q = 6, n = 20, T = 100, (2)q = 12, n = 20, T = 400, (3)q = 21, n = 40, T = 400$. As for the signal level, we vary the spectral norm of Λ by $\kappa = 0.2, 0.5, 0.8$. For each simulation setting, i.e. different combination of Scenario A,B and (q, n, T, κ), we conduct 20 experiments. Note that the transition matrix Λ is randomly generated for each experiment under Scenario B.

We summarize the numerical performance of ANH, FAR-F2 and FAR-Tr in Table 3, where we report the mean MISE (MISE$_{avg}$) and mean PE (PE$_{avg}$) across the 20 experiments. For each experiment, we also calculate the percentage improvement of prediction by FAR-F2 or FAR-Tr over ANH via Ratio= (PE(ANH) / PE(FAR-x)-1) × 100%. A positive ratio indicates improvement by FAR-F2 or FAR-Tr. We report the mean ratio (denoted by R$_{avg}$) across the 20 experiments.

Overall, FAR-F2, FAR-Tr and ANH are suitable for different situations. Within each scenario, the improvement of RKHS based methods over comparison methods increase with a higher dimension q and a stronger signal κ, while for the same (q, κ), the 2 RKHS based method yield the most improvement in Scenario A when spectral norm κ and effective dimension q are relatively large.

		Scenario A: $q = 6, n = 20, T = 100$			Scenario B: $q = 6, n = 20, T = 100$		
	Method	MISE$_{avg}$	PE$_{avg}\times$100	R$_{avg}$(%)	MISE$_{avg}$	PE$_{avg}\times$100	R$_{avg}$(%)
$\kappa = 0.2$	FAR-F2	1.373	2.198	-2.61	2.066	2.228	-4.85
	ANH	**0.898**	**2.141**		**1.958**	**2.120**	
$\kappa = 0.5$	FAR-F2	**0.274**	**2.231**	1.82	**0.587**	**2.226**	0.00
	ANH	0.316	2.271		0.742	2.226	
$\kappa = 0.8$	FAR-F2	0.106	2.350	-1.50	**0.307**	**2.206**	1.38
	ANH	**0.066**	**2.315**		0.349	2.237	
		Scenario A: $q = 12, n = 20, T = 400$			Scenario B: $q = 12, n = 20, T = 400$		
	Method	MISE$_{avg}$	PE$_{avg}\times$100	R$_{avg}$(%)	MISE$_{avg}$	PE$_{avg}\times$100	R$_{avg}$(%)
$\kappa = 0.2$	FAR-F2	0.809	4.299	-0.53	1.368	4.224	-0.28
	ANH	**0.715**	**4.276**		**1.112**	**4.212**	
$\kappa = 0.5$	FAR-F2	**0.150**	**4.365**	5.63	**0.416**	**4.327**	1.62
	ANH	0.417	4.611		1.088	4.400	
$\kappa = 0.8$	FAR-F2	**0.078**	**4.365**	38.94	**0.154**	**4.306**	8.42
	ANH	0.343	6.065		0.923	4.668	
		Scenario A: $q = 21, n = 40, T = 400$			Scenario B: $q = 21, n = 40, T = 400$		
	Method	MISE$_{avg}$	PE$_{avg}\times$100	R$_{avg}$(%)	MISE$_{avg}$	PE$_{avg}\times$100	R$_{avg}$(%)
$\kappa = 0.2$	FAR-F2	0.976	7.534	-1.59	**1.009**	7.386	-1.23
	ANH	**0.833**	**7.414**		1.065	**7.259**	
$\kappa = 0.5$	FAR-F2	**0.231**	**7.769**	2.83	**0.653**	**7.512**	1.18
	ANH	0.340	7.988		3.342	7.601	
$\kappa = 0.8$	FAR-F2	**0.080**	**7.794**	19.52	**0.268**	**7.591**	4.39
	ANH	0.308	9.315		3.698	7.925	

Table 2: Numerical performance of various methods for FAR(1) processes. Methods considered are RKHS (this paper), ANH (Aue et al., 2015). Bold font indicates the best performance, where the proposed RKHS method is generally the best performer in Scenarios A and B with large spectral norm κ.

		Scenario A: $q = 6, n = 20, T = 100$			Scenario B: $q = 6, n = 20, T = 100$		
	Method	MISE$_{avg}$	PE$_{avg}$ $\times$ 100	R$_{avg}$(%)	MISE$_{avg}$	PE$_{avg}$ $\times$ 100	R$_{avg}$(%)
$\kappa = 0.2$	FAR-F2	1.386	2.191	-1.69	2.077	2.166	-1.06
	FAR-Tr	0.970	**2.144**	0.47	**1.529**	**2.138**	0.23
	ANH	**0.882**	2.154		1.532	2.143	
$\kappa = 0.5$	FAR-F2	0.278	2.138	3.32	**0.741**	2.163	1.80
	FAR-Tr	**0.232**	**2.137**	3.32	**0.741**	**2.160**	1.94
	ANH	0.312	2.209		0.845	2.202	
$\kappa = 0.8$	FAR-F2	0.113	2.399	0.79	0.222	**2.265**	0.08
	FAR-Tr	**0.083**	**2.392**	1.09	**0.195**	2.266	0.04
	ANH	0.086	2.418		0.296	2.267	
		Scenario A: $q = 12, n = 20, T = 400$			Scenario B: $q = 12, n = 20, T = 400$		
	Method	MISE$_{avg}$	PE$_{avg}$ $\times$ 100	R$_{avg}$(%)	MISE$_{avg}$	PE$_{avg}$ $\times$ 100	R$_{avg}$(%)
$\kappa = 0.2$	FAR-F2	0.871	**4.249**	0.87	1.388	4.173	0.02
	FAR-Tr	**0.635**	4.260	0.61	**1.025**	**4.169**	0.12
	ANH	0.715	4.286		1.062	4.174	
$\kappa = 0.5$	FAR-F2	0.152	4.495	6.85	0.388	4.412	2.79
	FAR-Tr	**0.137**	**4.458**	7.74	**0.329**	**4.380**	3.54
	ANH	0.454	4.803		1.341	4.535	
$\kappa = 0.8$	FAR-F2	0.081	4.287	33.68	**0.154**	**4.122**	15.14
	FAR-Tr	**0.079**	**4.221**	35.77	0.166	4.129	14.94
	ANH	0.338	5.731		0.737	4.746	

Table 3: Numerical performance of various methods for FAR(1) processes. Bold font indicates the best performance.

Notice that although FAR-Tr is better than FAR-F2 in some cases, the nuclear norm penalty is much slower than FAR-F2, which might cause problem when it comes to multi-dimensional setting.

3.4 2 dim Scenario Extension

3.4.1 2 Dimensional RKHS-based Penalized Estimator

We construct its estimator $\{\widehat{A}_d\}_{d=1}^D$ via a constrained Frobenius norm optimization such that

$$\{\widehat{A}_d\}_{d=1}^D = \mathop{\arg\min}_{\{A_d\}_{d=1}^D \in \mathcal{C}_\tau} \frac{1}{Tn} \sum_{t=D+1}^{T} \sum_{i=1}^{n} \left(X_t(s_i, r_i) - \sum_{d=1}^{D} \frac{1}{n} \sum_{j=1}^{n} A_d((s_i, r_i), (s_j, r_j)) X_{t-d}(s_j, r_j) \right)^2 \tag{30}$$

where $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_D)$ is the tuning parameter and $\mathcal{C}_{\boldsymbol{\tau}} := \{(A_1, \cdots, A_D) : A_d \in \mathcal{C}$ and $\|A_d\|_{\mathcal{H},\mathcal{F}} \leq \tau_d,\ d = 1, \cdots, D\}$ is the constraint space. We name $\{\widehat{A}_d\}_{d=1}^D$ in (30) the penalized/constrained Frobenius norm estimator for transition operators of FAR. Similarly, we extend the representer theorem into 2 dimensional case.

Proposition 2 (Representer theorem). *There exists a minimizer $\{\widehat{A}_d\}_{d=1}^D$ of the constrained nuclear/Frobenius norm optimization* (30) *such that for any $(r, s) = ((r^1, r^2), (s^1, s^2)) \in [0, 1] \times [0, 1] \times [0, 1] \times [0, 1]$,*

$$\widehat{A}_d(r, s) = \widehat{A}_d((r^1, r^2), (s^1, s^2)) = \sum_{1 \leq i,j \leq n} \widehat{a}_{d,ij} \mathbb{K}((r^1, r^2), (s_i^1, s_i^2)) \mathbb{K}((s^1, s^2), (s_j^1, s_j^2)), \tag{31}$$

for $d = 1, 2, \cdots, D$.

Plugging the representer theorem into (30), the optimization problem is almost the same as in 1 dimensional scenario, except that the input is spatial correlated. Hence, the extension to 2 dimensional Frobenius norm penalty estimator is trivial(See Algorithm 2). In the following section, we will introduce the 2 dimensional penalized functional PCA in detail, which will serve as a baseline method.

Algorithm 2 FAR-F2

```
1: INPUT:$X, X^{(1)}$, n sample points $(s_1^1, s_1^2), ..., (s_n^1, s_n^2)$, kernel matrix $K$.
2: Do SVD to $\frac{1}{n^2} K^{\frac{1}{2}} X^{(1)} X^{(1)^T} K^{\frac{1}{2}} = U_1 D_1 U_1^T, K = U_2 D_2 U_2^T$.
3: Plug into 22 ▷ $U_{ij}$ is the $j_{\text{th}}$ column of matrix $U_i$, $D_{ij}$ is the $j_{\text{th}}$ element of the diagonal of matrix $D_i$.
4: OUTPUT:$\hat{W}, \hat{A}, \hat{X}$ by 23 and 25
```

3.4.2 Penalized Functional PCA

The framework for 1 dimensional functional PCA is described in Aue et al. (2015). The basic idea of Aue et al. (2015) is a canny combination of FPCA-based dimension reduction and the classical vector autoregressive (VAR) model, designed for prediction of FAR processes. Specifically, the infinite dimensional functional time series $\{X_t\}_{t=1}^T$ is first projected to the p eigenfunctions $\hat{f}(s) = (\hat{f}_1(s), \cdots, \hat{f}_p(s))^\top$ of the sample covariance operator. After projection, X_t is represented by a p-dimensional functional principal score

$x_t = (x_{t1}, \cdots, x_{tp})^\top$ with $x_{ti} = \int X_t(s)\hat{f}_i(s)ds$. A VAR($D$) model is then fitted on the p-dimensional time series $\{x_t\}_{t=1}^T$ such that $x_t = B_1 x_{t-1} + \cdots + B_D x_{t-D} + \epsilon_t$. Denote the estimated coefficient matrices as $\hat{B}_1, \cdots, \hat{B}_D \in \mathbb{R}^{p\times p}$, the one-step ahead prediction of $X_t(s)$ is then $\widehat{X}_t(s) = \hat{f}(s)^\top \hat{x}_t = \hat{f}(s)^\top \sum_{d=1}^D \hat{B}_d x_{t-d}$.

Based on the 1 dimension framework, the only difference for 2 dimension algorithm is to obtain the eigenfunctions with spatial data input. To be more specific, the only thing we need to work on is to extend functional PCA into 2 dimensional case. The one dimensional fPCA method has been stated in Ramsay and Silverman (2005). In this section, we will extend functional PCA to 2 dimension scenario together with smoothing spline method by using the tensor products of B-splines.

Given sample points (s_1, r_1), ..., (s_n, r_n), the response functional data $X_t(s, r)$ where $(s, r) \in \mathbb{R}^2$ and the spline basis $\Phi_1(s) = (\phi_{11}(s), ..., \phi_{1k_1}(s))^T$, $\Phi_2(r) = (\phi_{21}(r), ..., \phi_{2k_2}(r))^T$ for 2 dimensions respectively, we define the 2 dimensional basis function as

$$\begin{aligned}
\Phi(s,r) &= (\phi_{11}(s)\phi_{21}(r), ..., \phi_{1k_1}(s)\phi_{21}(r), \phi_{11}(s)\phi_{22}(r), ...)^T \\
&= \Phi_2(r) \otimes \Phi_1(s) \\
&= (\Phi_2(r) \otimes \mathbb{1}_{k_1}(s)) \odot (\mathbb{1}_{k_2}(r) \otimes \Phi_1(s)) \qquad (32)
\end{aligned}$$

where $\odot$ is the Hadamard product. (32) was proposed by Lee and Durbán (2011) in multi-dimensional smoothing.

Therefore, we have $X_t(s, r) = c_t^T \Phi(s, r)$. Let y be the vector of coefficients of any potential principle component curve $\xi(s, r)$, so that $\xi(s, r) = y^T \Phi(s, r)$. As stated in Ramsay and Silverman (2005), we need to minimize the penalized sample variance, which is defined as

$$\text{PCAPSV}(\xi) = \frac{\mathbf{var} \iint \xi X_t}{||\xi||^2 + \lambda_1 \times \text{PEN}_2(\xi)} \qquad (33)$$

In our paper, we define

$$\begin{aligned}\mathrm{PEN}_2(\xi) &\coloneqq ||D^2\xi||_F^2 = \iint (\frac{\partial^2\xi}{\partial s^2})^2 + 2(\frac{\partial^2\xi}{\partial s\partial r})^2 + (\frac{\partial^2\xi}{\partial r^2})^2 \\ &= y^T\left[\iint\left(\frac{\partial^2\Phi}{\partial s^2}\left(\frac{\partial^2\Phi}{\partial s^2}\right)^T + 2\frac{\partial^2\Phi}{\partial s\partial r}\left(\frac{\partial^2\Phi}{\partial s\partial r}\right)^T + \frac{\partial^2\Phi}{\partial r^2}\left(\frac{\partial^2\Phi}{\partial r^2}\right)^T\right)\right]y \\ &\coloneqq y^T(K_1 + 2K_2 + K_3)y \\ &\coloneqq y^T K y. \end{aligned} \tag{34}$$

From (32) we have,

$$\begin{aligned}\frac{\partial^2\Phi}{\partial s^2} &= (\Phi_2(r)\otimes\mathbb{1}_{k_1}(s))\odot\big(\mathbb{1}_{k_2}(r)\otimes D^2\Phi_1(s)\big) \\ &= (\Phi_2(r)\odot\mathbb{1}_{k_2}(r))\otimes\big(\mathbb{1}_{k_1}(s)\odot D^2\Phi_1(s)\big) \\ &= \Phi_2(r)\otimes D^2\Phi_1(s)\end{aligned} \tag{35}$$

Similarly,

$$\frac{\partial^2\Phi}{\partial s\partial r} = (D\Phi_2(r)\otimes\mathbb{1}_{k_1}(s))\odot(\mathbb{1}_{k_2}(r)\otimes D\Phi_1(s)) = D\Phi_2(r)\otimes D\Phi_1(s) \tag{36}$$

$$\frac{\partial^2\Phi}{\partial r^2} = \big(D^2\Phi_2(r)\otimes\mathbb{1}_{k_1}(s)\big)\odot(\mathbb{1}_{k_2}(r)\otimes\Phi_1(s)) = D^2\Phi_2(r)\otimes\Phi_1(s) \tag{37}$$

Therefore,

$$\begin{aligned}K_1 &= \iint\frac{\partial^2\Phi}{\partial s^2}\left(\frac{\partial^2\Phi}{\partial s^2}\right)^T \\ &= \iint\big(\Phi_2(r)\otimes D^2\Phi_1(s)\big)\cdot\big(\Phi_2^T(r)\otimes D^2\Phi_1^T(s)\big) \\ &= \iint\big(\Phi_2(r)\Phi_2^T(r)\big)\otimes\big(D^2\Phi_1(s)D^2\Phi_1^T(s)\big) \\ &= \int\Phi_2(r)\Phi_2^T(r)\otimes\int D^2\Phi_1(s)D^2\Phi_1^T(s)\end{aligned} \tag{38}$$

Likewise,

$$\begin{aligned}K_2 &= \int D\Phi_2(r)D\Phi_2^T(r)\otimes\int D\Phi_1(s)D\Phi_1^T(s) \\ K_3 &= \int D^2\Phi_2(r)D^2\Phi_2^T(r)\otimes\int\Phi_1(s)\Phi_1^T(s)\end{aligned} \tag{39}$$

We then calculate other terms in (33)

$$\textbf{var} \iint \xi X_t = \textbf{var}\ y^T \iint \Phi(s,r)\Phi(s,r)^T c_t := y^T J V J^T y \tag{40}$$

$$||\xi||^2 = \iint \xi^2(s,r)dsdr = y^T J y \tag{41}$$

where $V = \textbf{var}\ c_t$, $J = \iint \Phi(s,r)\Phi(s,r)^T$. Let $C = (c_1, ..., c_T)$, $V = CC^T/T$.

To further exploit the structure of J, we have

$$J = \iint \Phi(s,r)\Phi(s,r)^T = \int \Phi_2(r)\Phi_2^T(r) \otimes \int \Phi_2(s)\Phi_2^T(s) := J_2 \otimes J_1 \tag{42}$$

Hence, $K = J_2 \otimes D^2 J_1 + 2DJ_2 \otimes DJ_1 + D^2 J_2 \otimes J_1$.

Plug (34), (40), (38) and (39) into (33), the minimization problem becomes the eigenequation

$$JVJy = \rho(J + \lambda_1 K)y \tag{43}$$

J and K can be expressed as the tensor product of the corresponding matrices in 1 dimensional case. In the following context, we will find the expression of coefficient matrix C.

Denote the observation of the functional time series at time t as $X_t = (X_t(s_1, r_1), X_t(s_2, r_2), \cdots, X_t(s_n, r_n))^\top$. Data X_ts are assumed to be the result of the model

$$X_t = \widetilde{\Phi} c_t + \epsilon_t \tag{44}$$

$\widetilde{\Phi}$ is the matrix of B-splines,

$$\widetilde{\Phi} = (\Phi(s_1, r_1), .., \Phi(s_n, r_n))^T = (\Phi_2(r_1) \otimes \Phi_1(s_1), ..., \Phi_2(r_n) \otimes \Phi_1(s_n))^T \tag{45}$$

Define $X = (X_1, ..., X_T)$, we have

$$X = \widetilde{\Phi} C + \epsilon \tag{46}$$

where $C = (c_1, ..., c_T)$.

To estimate θ, the penalized B-spline is here considered, minimizing the penalized optimization function

$$f = (X - \widetilde{\Phi} C)^T (X - \widetilde{\Phi} C) + \lambda_2 C^T K C \tag{47}$$

The solution of minimization problem (47) is

$$C = (\widetilde{\Phi}^T\widetilde{\Phi} + \lambda_2 K)^{-1}\widetilde{\Phi}^T X \tag{48}$$

Plug (48) into (43), where $V = \textbf{var } c_t = CC^T/T$. Solve (43) and we will have coefficients of the eigenfunction.

The explicit algorithm is summarized in Algorithm 3. We coded it based on 1 dimensional functional PCA method, and took advantage of the R package 'fda'. Hence, we could get basis $\Phi(s, r), \widetilde{\Phi}$ by combining 1 dimensional b-spline basis with 32 and 45. $J_1 = \int \Phi_1(s)\Phi_1(s)^T, J_2 = \int \Phi_2(s)\Phi_2(s)^T, DJ_1, DJ_2, D^2J_1, D^2J_2$ defined in 35 - 42 can be obtained using functions in **fda** package.

Algorithm 3 2D FPCA

```
INPUT: $X \in \mathbb{R}^{n\times T}$, n sample points $(s_1^1, s_1^2), ..., (s_n^1, s_n^2)$, $\lambda_1$, $\lambda_2$.
Obtain $J_1, J_2, \widetilde{\Phi}, \Phi$ using package fda.
$K = J_2 \otimes D^2J_1 + 2DJ_2 \otimes DJ_1 + D^2J_2 \otimes J_1$ and plug into 48 to get $\hat{C}$.
Solve 43, where $V = \hat{C}\hat{C}^T/T$
OUTPUT: y is the coefficient of the eigenfunction.
```

Algorithm 3 outputs the 2 dimensional eigenfunction. Combining with Algorithm 1 in Aue et al. (2015), the method could be used in 2 dimensional functional prediction problem.

3.5 2 dim Simulation Studies

In this section, we conduct simulation studies to investigate the estimation and prediction performance of the 2 dimensional penalized Frobenius norm estimator and compare it with the 2 dimensional functional PCA method in section 3.

3.5.1 Estimation methods and implementation details

Data generating process: Similar to 1 dimensional data generation process, we first define an FAR(D) process, borrowed from the simulation setting in Aue et al. (2015), that is used in the simulation study. For $d = 1, 2, \cdots, D$, we assume the dth transition operator $A_d(r, s)$ is of rank q_d and is generated by q_d basis functions $\{u_i(s)\}_{i=1}^{q_d}$ such that

$$A_d((s,r),(l,w)) = \sum_{i,j=1}^{q_d} \lambda_{d,ij} u_i(s,r) u_j(l,w),$$

where $\{u_i(s)\}_{i=1}^{q_d}$ consists of orthonormal basis of $\mathcal{L}^2[0,1]$ that will be specified later. Define matrix Λ_d such that $\Lambda_{d,ij} = \lambda_{d,ij}$ and define $\mathbf{u}_{q_d}(s) = (u_1(s,r), u_2(s,r), \cdots, u_{q_d}(s,r))^\top$. We have $A_d(r,s) = \mathbf{u}_{q_d}(r)^\top \Lambda_d \mathbf{u}_{q_d}(s)$. We further set the noise function ϵ_t to be of finite rank q_ϵ such that $\epsilon_t(s) = \sum_{i=1}^{q_\epsilon} z_{ti} u_i(s,r)$, where $z_{ti} \overset{i.i.d.}{\sim} U(-a_i, a_i)$ or $z_{ti} \overset{i.i.d.}{\sim} N(0, \sigma_i^2)$.

Without loss of generality, we set $q_1 = q_2 = \cdots = q_D = q_\epsilon = q$ for simplicity. Thus, the FAR(D) process $\{X_t(s)\}_{t=1}^T$ resides in a finite dimensional subspace spanned by the orthonormal basis $\{u_i(s)\}_{i=1}^q$. Denote $X_t(r) = \sum_{i=1}^q x_{ti} u_i(r)$ where $x_{ti} = \int X_t(r) u_i(r) dr$, and denote $x_t = (x_{t1}, \cdots, x_{tq})^\top$ and $z_t = (z_{t1}, \cdots, z_{tq})^\top$. We have

$$\begin{aligned}
X_t(s,r) =& \sum_{d=1}^D \iint A_d((s,r),(l,w)) X_{t-d}(l,w) dl dw + \epsilon_t(s,r) \\
=& \sum_{d=1}^D \iint \mathbf{u}_q(s,r)^\top \Lambda_d \mathbf{u}_q(l,w) X_{t-d}(l,w) dl dw + z_t^\top \mathbf{u}_q(s,r) \\
=& \sum_{d=1}^D \iint \mathbf{u}_q(s,r)^\top \Lambda_d \mathbf{u}_q(l,w) \mathbf{u}_q(l,w)^\top x_{t-d} dl dw + z_t^\top \mathbf{u}_q(s,r) \\
=& \mathbf{u}_q(s,r)^\top \left(\sum_{d=1}^D \Lambda_d x_{t-d} + z_t \right). \qquad (49)
\end{aligned}$$

For FAR(1) process, the expression simplifies to

$$X_t(s,r) = \mathbf{u}_q(s,r)^\top \left(\sum_{d=1}^D \Lambda_d x_{t-d} + z_t \right) \qquad (50)$$

(49) leads to $x_t = \sum_{d=1}^D \Lambda_d x_{t-d} + z_t$. Thus, the FAR($D$) process can be exactly simulated via a VAR(D) process. We set $u_i(s,r) = 1$ if $i = 1$ and $u_i(s,r) = \sqrt{2}\cos(2(i-1)\pi(s+r))$ for $i = 2, \cdots, q$.

Given the transition operators $A_1, \cdots, A_D$ (i.e. $\Lambda_1, \cdots, \Lambda_D$) and the distribution of noise z_t, the true FAR(D) process $\{X_t(s,r), (s,r) \in [0,1] \times [0,1]\}_{t=1}^T$ can be simulated and discrete measurements of the functional time series are taken at the sampling points $\{(s_i, r_i)\}_{i=1}^n$. For simplicity, we set $\{(s_i, r_i)\}_{i=1}^n$ to be the n equal-spaced points on $[0,1] \times [0,1]$, which resembles the typical sampling scheme of functional time series in real data applications.

Implementation of penalized Functional PCA-VAR estimator(ANH): There are 2 tuning parameters in the 2 dimensional Functional PCA-VAR method: λ_1 from the penalized functional PCA, λ_2 is from smoothing spline. In the following experiments, we simplify the case into $\lambda_1 = \lambda_2 = \lambda$. We select the number of functional principal component p by looking at the proportion of variance explained. Further work on dimension selection could be exploit in the future.

Implementation of penalized Frobenius norm estimator(FAR-F2): For the implementation of the proposed RKHS-based penalized Frobenius norm estimator, we use the Gaussian function as the reproducing kernel $\mathbb{K}$, such that

$$\mathbb{K}(\mathbf{x}, \mathbf{y}) = \frac{e^{||\mathbf{x}-\mathbf{y}||^2}}{2\sigma^2}$$

Generalized cross validation is applied to select the tuning parameter λ. More details are in 3.2.2.

Implementation of penalized nuclear norm estimator(FAR-Tr): We use the accelerated gradient algorithm in Ji and Ye (2009) to solve the nuclear norm optimization problem. The standard 5 fold cross validation is used to tune the parameter λ. The reproducing kernel is the same as in method FAR-F2.

Based on $\{\widehat{A}\}$, the one-step ahead prediction of $X_t(s_i, r_i)$ for $t = T+1, \cdots, T+0.2T$ in the test data can be calculated via $\widehat{X}_t(s_i) = \frac{1}{n}\sum_{j=1}^{n} \widehat{A}((s_i, r_i), (s_j, r_j))X_t(s_j, r_j)$ for $i = 1, \cdots, n$ as in (15).

Evaluation criteria: We evaluate the performance of a method via (a). estimation error of $\widehat{A}_1, \widehat{A}_2, \cdots, \widehat{A}_D$ and (b). prediction error of the estimated FAR(D) model.

Specifically, given sample size (n, T), we simulate the observed functional time series $\{X_t(s_i, r_i), i = 1, \cdots, n\}_{t=1}^{T+0.2T}$, which we then partition into training data $\{X_t(s_i, r_i), i = 1, \cdots, n\}_{t=1}^{T}$ for estimation of $A_1, \cdots, A_D$ and test data $\{X_t(s_i, r_i), i = 1, \cdots, n\}_{t=T+1}^{T+0.2T}$ for evaluation of prediction performance. Denote $\{\widehat{X}_t(s_i), i = 1, \cdots, n\}_{t=T+1}^{T+0.2T}$ as the one-step ahead prediction given by the estimated FAR(D) model. We define

$$\mathrm{MISE}(\widehat{A}_d, A_d) = \frac{\iint_{[0,1]\times[0,1]}\iint_{[0,1]\times[0,1]}(A_d((s,r),(l,w)) - \widehat{A}_d((s,r),(l,w)))^2 dsdrdldw}{\iint_{[0,1]\times[0,1]}\iint_{[0,1]\times[0,1]} A_d((s,r),(l,w))^2 dsdrdldw} \tag{51}$$

$$\mathrm{PE} = \frac{1}{0.2nT}\sum_{t=T+1}^{T+0.2T}\sum_{i=1}^{n}(X_t(s_i,r_i) - \widehat{X}_t(s_i,r_i))^2, \tag{52}$$

where MISE (mean integrated squared error) measures the estimation error and PE measures the prediction error.

3.5.2 Simulation result for FAR(1)

For FAR(1), there is only one transition operator $A((s,r),(l,w)) = A_1((s,r),(l,w))$. The simulation setting involves the transition matrix $\Lambda = \Lambda_1 \in \mathbb{R}^{q\times q}$ (signal) and the noise range $a_{1:q} = (a_1, a_2, \cdots, a_q)$ or the noise variance $\sigma^2_{1:q} = (\sigma^2_1, \sigma^2_2, \cdots, \sigma^2_q)$ for $\{z_{ti}\}_{i=1}^q$ (driving noise). Denote $\sigma(\Lambda)$ as the leading singular value for a matrix Λ. We consider two different signal-noise settings:

- Scenario A (Diag Λ): $\Lambda = \mathrm{diag}(\kappa, \cdots, \kappa)$ and $z_{ti} \overset{i.i.d.}{\sim} U(-a, a)$ with $a = 0.1$ for $i = 1, \cdots, q.$, σ of kernel function $= 0.1$. The function $X_t(s)$ is first estimated using 9*10 cubic penalized B-spline basis functions. We let sample size $n = 100$, effective dimension $q = 6, 6^2$.
- Scenario C (Low-rank setting): Partition the spatial domain $[0,1]\times[0,1]$ into 4 regions I, II, III, IV.

$$A((s,r),(s',r')) = \begin{cases} \kappa & (s,r),(s',r') \text{ are in the same region} \\ 0 & \text{Otherwise} \end{cases}$$

In scenario C, Let FAR(1) process $\{X_t(s)\}_{t=1}^T$ resides in a finite dimensional subspace spanned by the orthonormal basis $\{u_i(s)\}_{i=1}^q$. Denote $X_t(s,r) = \sum_{i=1}^q x_{ti}u_i(s,r)$ where $x_{ti} = \iint X_t(s,r)u_i(s,r)dsdr$, and denote $x_t = (x_{t1}, \cdots, x_{tq})^\top$ and $z_t = (z_{t1}, \cdots, z_{tq})^\top$, we will have $X_t(s,r) = x_t^T u(s,r), x_t = \iint X_t(s,r)u_(s,r)dsdr$. Assume $(s,r) \in$ region R, we have

$$
\begin{aligned}
X_t(s,r) =& \iint A((s,r),(s',r'))X_{t-1}(s',r')ds'dr' + \epsilon_t(s,r) \\
=& \iint_R X_{t-1}(s',r')ds'dr' + \epsilon_t(s,r) \\
=& x_{t-1}^T \iint_R u(s',r')ds'dr' + \epsilon_t(s,r) \\
=& x_{t-1}^T \widetilde{u}(s,r) + \epsilon_t(s,r) \qquad (53)
\end{aligned}
$$

where piecewise function $\widetilde{u}(s,r) = \iint_R u(s',r')ds'dr'$, if $(s,r) \in$ region R, for $R = I, II, III, IV$. Denote $\widetilde{u}(R) = \widetilde{u}(s,r)$, if $(s,r) \in$ region R,

$$
\begin{aligned}
x_t =& \iint X_t(s,r)u_(s,r)dsdr \\
=& \iint (x_{t-1}^T \widetilde{u}(s,r))u(s,r)dsdr + \iint \epsilon_t(s,r)u_(s,r)dsdr \\
=& \sum_{R=I,II,III,IV} (\iint_R u(s,r)\widetilde{u}(s,r)^T dsdr)x_{t-1} + z_t \\
=& \sum_{R=I,II,III,IV} (\iint_R u(s,r)dsdr)\widetilde{u}(R)^T x_{t-1} + z_t \\
=& \sum_{R=I,II,III,IV} \widetilde{u}(R)\widetilde{u}(R)^T x_{t-1} + z_t \\
:=& \Lambda x_{t-1} + z_t \qquad (54)
\end{aligned}
$$

where $\Lambda = \sum_{R=I,II,III,IV} \widetilde{u}(R)\widetilde{u}(R)^T$, $\widetilde{u}(R) = \iint_R u(s',r')ds'dr'$.

In this setting, we let region $I = [0, \frac{1}{2}] \times [\frac{1}{2}, 1]$, $II = [\frac{1}{2}, 1] \times [\frac{1}{2}, 1]$, $III = [0, \frac{1}{2}] \times [0, \frac{1}{2}]$, $IV = [\frac{1}{2}, 1] \times [0, \frac{1}{2}]$. Let $u_i(s,r) = 1$ if $i = 1$ and $u_i(s,r) = \sqrt{2}\cos(2(i-1)\pi(s+r))$ for $i = 2, \cdots, q$,

$$
\widetilde{u}_i(R) = \begin{cases} \frac{1}{4} & i = 1 \\ \sqrt{2}\frac{S_i(R)}{\lambda_i^2} & i = 2, ..., q \end{cases} \qquad (55)
$$

where denote the range for x axis of region R is from w_1 to w_2, while the range for y axis of region R is from z_1 to z_2, $\lambda_i = 2(i-1)\pi$, $S_i(R) = (sin(kw_2) - sin(kw_1))(sin(kz_2) - sin(kz_1)) - (cos(kw_2) - cos(kw_1))(cos(kz_2) - cos(kz_1))$.

Numerical result for FAR(1): For Scenarios A and C, we consider four sample sizes: $(1)q = 6, n = 36, T = 100, (2)q = 6, n = 100, T = 400, (3)q = 6^2, n = 36, T = 400, (3)q = 6^2, n = 100, T = 400$. As for the signal level, we vary the spectral norm of Λ by $\kappa = 0.5, 0.8, 1$. For each simulation setting, i.e. different combination of Scenario A, C and (q, n, T, κ), we conduct 20 experiments.

We summarize the numerical performance of ANH, FAR-F2 and FAR-Tr in Table 4 and in Table 5, where we report the mean MISE (MISE$_{avg}$) and mean PE (PE$_{avg}$) across the 20 experiments. For each experiment, we also calculate the percentage improvement of prediction by RKHS based methods over ANH via Ratio= (PE(ANH) / PE(FAR-x)−1) × 100%. A positive ratio indicates improvement by FAR-F2 or FAR-Tr. We also report the mean ratio (denoted by R$_{avg}$).

		Scenario A: $q = 6, n = 36, T = 100$			Scenario A: $q = 6, n = 100, T = 400$		
	Method	MISE$_{avg}$	PE$_{avg}$	R$_{avg}$(%)	MISE$_{avg}$	PE$_{avg}$	R$_{avg}$(%)
$\kappa = 0.5$	FAR-F2	**0.835**	**2.445**$\times 10^{-2}$	11.74	**0.812**	**2.299** $\times 10^{-2}$	33.23
	FAR-Tr	0.835	2.455$\times 10^{-2}$	11.28	0.812	2.298$\times 10^{-2}$	33.29
	ANH	1.573	2.732$\times 10^{-2}$		1.003	3.063$\times 10^{-2}$	
$\kappa = 0.8$	FAR-F2	**0.833**	**2.403**$\times 10^{-2}$	74.78	**0.803**	**2.381**$\times 10^{-2}$	179.88
	FAR-Tr	0.833	2.405$\times 10^{-2}$	74.64	0.803	2.381$\times 10^{-2}$	179.88
	ANH	1.216	4.200$\times 10^{-2}$		0.923	6.664$\times 10^{-2}$	
		Scenario A: $q = 6^2, n = 36, T = 400$			Scenario A: $q = 6^2, n = 100, T = 400$		
	Method	MISE$_{avg}$	PE$_{avg}$	R$_{avg}$(%)	MISE$_{avg}$	PE$_{avg}$	R$_{avg}$(%)
$\kappa = 0.5$	FAR-F2	**0.964**	**0.159**	19.50	**0.963**	**0.147**	28.57
	FAR-Tr	0.964	0.162	17.28	0.964	0.147	28.57
	ANH	0.967	0.190		0.999	0.189	
$\kappa = 0.8$	FAR-F2	0.966	0.158	71.52	**0.962**	**0.161**	188.20
	FAR-Tr	0.965	**0.157**	72.61	0.963	0.161	188.20
	ANH	**0.961**	0.271		0.993	0.464	

Table 4: Bold font indicates the best performance.

		Scenario C: $q = 6, n = 36, T = 100$		Scenario C: $q = 6, n = 100, T = 400$	
	Method	MISE$_{avg}$	PE$_{avg}$	MISE$_{avg}$	PE$_{avg}$
$\kappa = 0.5$	FAR-F2	0.892	2.381×10^{-2}	0.576	2.361×10^{-2}
	FAR-Tr	**0.597**	$\mathbf{2.341}\times10^{-2}$	**0.449**	$\mathbf{2.341}\times10^{-2}$
	ANH	3.705	2.350×10^{-2}	2.824	2.342×10^{-2}
$\kappa = 0.8$	FAR-F2	0.436	2.297×10^{-2}	0.247	2.394×10^{-2}
	FAR-Tr	**0.329**	$\mathbf{2.252}\times10^{-2}$	**0.208**	$\mathbf{2.382}\times10^{-2}$
	ANH	1.414	2.268×10^{-2}	2.512	2.382×10^{-2}
		Scenario C: $q = 6^2, n = 36, T = 400$		Scenario C: $q = 6^2, n = 100, T = 400$	
	Method	MISE$_{avg}$	PE$_{avg}$	MISE$_{avg}$	PE$_{avg}$
$\kappa = 0.8$	FAR-F2	0.715	0.157	0.822	0.140
	FAR-Tr	**0.711**	**0.156**	**0.819**	**0.140**
	ANH	1.002	0.156	3.163	0.140
$\kappa = 1$	FAR-F2	0.777	**0.160**	0.839	0.148
	FAR-Tr	**0.775**	**0.160**	**0.837**	0.148
	ANH	1.002	**0.160**	2.062	**0.146**

Table 5: Numerical performance in scenario C for FAR(1) processes. Bold font indicates the best performance.

In scenario A, the two RKHS based method do not have much difference in terms of error on operator A, and the Frobenius penalty estimator is slightly better than nuclear penalty estimator in terms of prediction error. They outperform the fPCA method in both criteria.

In the low rank setting, although the improvement on the prediction error is not quite clear, the nuclear penalty estimator has the lowest estimation error among these 3 methods.

3.6 Conclusions and Future Work

This paper studies the estimation and prediction of a functional autoregressive process and extends the inference framework from one dimensional to multidimensional via the tools of Reproducing Kernel Hilbert Spaces (RKHS) of functional autoregressive time series. We work on the complexity and regularization to develop a computationally efficient algorithm. The paper also extend functional PCA method to multidimensional scenario, which serve as a baseline method for simulations. Future work could be applying this method to ARGO float data, which collects the temperature, salinity and other measurements pertaining to the biology/chemistry of the ocean.

References

Aue, A., Norinho, D. D., and Hörmann, S. (2015). On the prediction of stationary functional time series. *Journal of the American Statistical Association*, 110(509):378–392.

Besse, P. C. and Cardot, H. (1996). Approximation spline de la prevision d'un processus fonctionnel autorégressif d'ordre 1. *Canadian Journal of Statistics*, 24(4):467–487.

Besse, P. C., Cardot, H., and Stephenson, D. B. (2000). Autoregressive forecasting of some functional climatic variations. *Scandinavian Journal of Statistics*, 27(4):673–687.

Bosq, D. (2000). *Linear processes in function spaces: theory and applications.* Springer-Verlag New York.

Bühlmann, P. and van de Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications.* Springer Science & Business Media.

Chan, H. P. and Walther, G. (2013). Detection with the scan and the average likelihood ratio. *Statistica Sinica*, 1(23):409–428.

Didericksen, D., Kokoszka, P., and Zhang, X. (2012). Empirical properties of forecasts with the functionalautoregressive model. *Computational Statistics*, (27):285–298.

Ferraty, F. and Vieu, P. (2006). *Nonparametric Functional Data Analysis.* Springer-Verlag New York.

Frick, K., Munk, A., and Sieling, H. (2014). Multiscale change point inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76:495–580.

Friedrich, F., Kempe, A., Liebscher, V., and Winkler, G. (2008). Complexity penalized m-estimation: Fast computation. *Journal of Computational and Graphical Statistics*, 17:201–204.

Fryzlewicz, P. (2014). Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281.

Gu, C. (2013). *Smoothing Spline ANOVA Models.* Springer-Verlag New York, 2 edition.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning.* Springer-Verlag New York, 2 edition.

Horváth, L. and Kokoszka, P. (2012). *Inference for Functional Data with Applications.* Springer.

Hyndman, R. J. and Shang, H. L. (2009). Forecasting functional time series. *Journal of the Korean Statistical Society*, 38(3):199–211.

Hyndman, R. J. and Ullah, M. S. (2007). Robust forecasting of mortality and fertility rates: A functional data approach. *Computational Statistics & Data Analysis*, 51(10):4942–4956.

Hörmann, S. and Kokoszka, P. (2010). Weakly dependent functional data. *Annals of Statistics*, 38(3):1845–1884.

Ji, S. and Ye, J. (2009). An accelerated gradient method for trace norm minimization. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, page 457–464.

Kaul, A., Jandhyala, V. K., and Fotopoulos, S. B. (2018). Parameter estimation for high dimensional change point regression models without grid search. *arXiv preprint arXiv:1805.03719*.

Killick, R., Fearnhead, P., and Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598.

Lee, D.-J. and Durbán, M. (2011). P-spline anova-type interaction models for spatio-temporal smoothing. *Statistical Modelling*, 11(1):49–69.

Lee, S., Liao, Y., Seo, M. H., and Shin, Y. (2018). Oracle estimation of a change point in high-dimensional quantile regression. *Journal of the American Statistical Association*, 113(523):1184–1194.

Lee, S., Seo, M. H., and Shin, Y. (2016). The lasso for high dimensional regression with a possible change point. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(1):193–210.

Leonardi, F. and Bühlmann, P. (2016). Computationally efficient change point detection for high-dimensional regression. *arXiv preprint arXiv:1601.03704*.

Maidstone, R., Hocking, T., Rigaill, G., and Fearnhead, P. (2017). On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, 27:519–533.

Ramsay, J. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer-Verlag New York.

Rigaill, G. (2010). Pruned dynamic programming for optimal multiple change-point detection. *arXiv preprint arXiv:1004.0887*.

Rinaldo, A., Wang, D., Wen, Q., Willett, R., and Yu, Y. (2021). Localizing changes in high-dimensional regression models. *International Conference on Artificial Intelligence and Statistics*, 130:2089–2097.

Shang, H. L. (2013). Functional time series approach for forecasting very short-term electricity demand. *Journal of Applied Statistics*, 40(1):152–168.

Sun, X., Du, P., Wang, X., and Ma, P. (2018). Optimal penalized function-on-function regression under a reproducing kernel hilbert space framework. *Journal of the American Statistical Association*, 113(524):1601–1611.

Wahba, G. (1990). *Spline Models for Observational Data*. SIAM, Philadelphia.

Wang, D., Lin, K., and Willett, R. (2019). Statistically and computationally efficient change point localization in regression settings. *arXiv preprint arXiv:1906.11364*.

Wang, D., Yu, Y., and Rinaldo, A. (2017). Optimal covariance change point localization in high dimension. *arXiv preprint arXiv:1712.09912*.

Wang, D., Yu, Y., and Rinaldo, A. (2018a). Optimal change point detection and localization in sparse dynamic networks. *arXiv preprint arXiv:1809.09602*.

Wang, D., Yu, Y., and Rinaldo, A. (2018b). Univariate mean change point detection: Penalization, cusum and optimality. *arXiv preprint arXiv:1810.09498*.

Wang, D., Zhao, Z., Willett, R., and Yau, C. Y. (2020a). Functional autoregressive processes in reproducing kernel hilbert spaces. *arXiv preprint arXiv:2011.13993*.

Wang, D., Zhao, Z., Yu, Y., and Willet, R. (2020b). Functional linear regression with mixed predictors. *arXiv preprint arXiv:2012.00460*.

Wang, T. and Samworth, R. J. (2018). High dimensional change point estimation via sparse projection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(1):57–83.

Wikipedia (2020a). 2015 pacific typhoon season. https://en.wikipedia.org/wiki/2015_Pacific_typhoon_season.

Wikipedia (2020b). Air pollution in taiwan. https://en.wikipedia.org/wiki/Air_pollution_in_Taiwan.

Yuan, M. and Cai, T. T. (2010). A reproducing kernel Hilbert space approach to functional linear regression. *Annals of Statistics*, 38(6):3412–3444.

Zhang, B., Geng, J., and Lai, L. (2015). Change-point estimation in high dimensional linear regression models via sparse group lasso. In *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*, pages 815–821. IEEE.