Abstract—We propose a method for adaptive nonlinear sequential modeling of time series data. Data is modeled as a nonlinear function of past values corrupted by noise, and the underlying non-linear function is assumed to be approximately expandable in a spline basis. We cast the modeling of data as finding a good fit representation in the linear span of multi-dimensional spline basis, and use a variant of $l_1$-penalty regularization in order to reduce the dimensionality of representation. Using adaptive filtering techniques, we design our online algorithm to automatically tune the underlying parameters based on the minimization of the regularized sequential prediction error. We demonstrate the generality and flexibility of the proposed approach on both synthetic and real-world datasets. Moreover, we analytically investigate the performance of our algorithm by obtaining both bounds on prediction errors and consistency in variable selection.

Index Terms—Adaptive Filtering, Data prediction, Group LASSO, Nonlinearity, Sequential Modeling, SLANTS, Spline, Time Series.

I. INTRODUCTION

SEQUENTIALLY observed multi-dimensional time series are emerging in various applications. In most of these applications, modeling nonlinear functional inter-dependency between present and past data is crucial for both representation and prediction. This is a challenging problem, especially when fast online implementation, adaptivity to new data generating processes, and ability to handle high dimensions need to be simultaneously taken into account in nonlinear modeling. For example, environmental science combines high dimensional weather signals for real time prediction [1]. In epidemics, huge amount of online search data is used to form fast prediction of influenza epidemics [2]. In finance, algorithmic traders demand adaptive models to accommodate a fast changing stock market. In robot autonomy, there is the challenge of learning the high dimensional movement systems [3]. These tasks usually take high dimensional input signals which may contain a large number of irrelevant signals. In all these applications, methods to remove redundant signals and learn the nonlinear model with low computational complexity are well sought after. This motivates our work in this paper, where we propose an approach to sequential nonlinear adaptive modeling of potentially high dimensional time series.

Inference of nonlinear models has been a notoriously difficult problem, especially for large dimensional data [3]– [5]. In low dimensional settings, there have been remarkable parametric and nonparametric nonlinear time series models that have been applied successfully to data from various domains. Examples include threshold models [6], generalized autoregressive conditional hetero-scedasticity models [7], multivariate adaptive regression splines (MARS) [4], generalized additive models [8], functional coefficient regression models [9], etc. However, some of these methods may suffer from prohibitive computational complexity. Variable selection using some of these approaches is yet another challenge as they may not guarantee the selection of significant predictors (variables that contribute to the true data generating process) given limited data size. In contrast, there exist high dimensional nonlinear time series models that are mostly inspired by high dimensional statistical methods. There are typically two kinds of approaches. In one approach, a small subset of significant variables is first selected and then nonlinear time series models are applied to selected variables. For example, independence screening techniques such as [10]–[12] or the MARS may be used to do variable selection. In another approach, dimension reduction method such as least absolute shrinkage and selection operator (LASSO) [13] are directly applied to nonlinear modeling. Sparse additive models have been developed in recent works of Ravikumar et al. [14] and Huang et al. [5]. In the work of Bazerque et al. [15], splines additive models together with group-sparsity penalty was proposed and applied to spectrum cartography. These offline approaches seem promising and may benefit from additional reductions in computational complexity.

In this work, inspired by the second approach, we develop a new method referred to as Sequential Learning Algorithm for Nonlinear Time Series (SLANTS). A challenging problem in sequential inference is that the data generating process varies with time, which is common in many practical applications [1]–[3]. We propose a method that can help address sequential inference of potentially time-varying models. Moreover, the proposed method provides computational benefits as we avoid repeating batch estimation upon sequential arrival of data. Specifically, we use the spline basis to dynamically approximate the nonlinear functions. The algorithm can efficiently give unequal weights to data points by design, as typical in adaptive filtering. We also develop an online version of group LASSO for dimensionality reduction (i.e. simultaneous
estimation and variable selection). To this end, the group LASSO regularization is re-formulated into a recursive estimation problem that produces an estimator close to the maximum likelihood estimator from batch data. We theoretically analyze the performance of SLANTS. Under reasonable assumptions, we also provide an estimation error bound, and a backward stepwise procedure that guarantees consistency in variable selection.

The outline of this paper is given next. In Section II, we formulate the problem mathematically and present our inference algorithm. In Section III, we present our theoretical results regarding prediction error and model consistency. In Section IV, we provide numerical results using both synthetic and real data examples. The results demonstrate excellent performance of our method. We make our conclusions in Section V.

II. SEQUENTIAL MODELING OF NONLINEAR TIME SERIES

In this section, we first present our mathematical model and cast our problem as $l_1$-regularized linear regression. We then propose an expectation-maximization (EM) type algorithm to sequentially estimate the underlying parameters. Finally, we disclose methods for tuning the underlying parameters, and in Section V, we provide numerical results using both synthetic and real data examples. The results demonstrate excellent performance of our method.

A. Formulation of SLANTS

Consider a multi-dimensional time series given by

$$X_t = [X_{1,t}, \ldots, X_{D,t}]^T \in \mathbb{R}^D, \ t = 1, 2, \ldots$$

Our main objective in this paper is to predict the value of $X_T$ at time $T$ given the past observations $X_{T-1}, \ldots, X_1$. For simplicity, we present our results for the prediction of scalar random variable $X_{1:T+1}$. We start with the general formulation

$$X_{1:T} = f(X_{T-1}, \ldots, X_{T-L}) + \varepsilon_T,$$

where $f(\cdot, \ldots, \cdot)$ is smooth (or at least piece-wise smooth), $\varepsilon_T$ are independent and identically distributed (i.i.d.) zero mean random variables and the lag order $L$ is a finite but unknown nonnegative integer.

We rewrite the model in (1) as

$$X_{1:T} = f(X_{1:T-1}, \ldots, X_{T-L-1}, \ldots, X_{D,T-1}, \ldots, X_{D,T-L}) + \varepsilon_T.$$ 

With a slight abuse of notation, we rewrite the above model (1) as

$$Y_T = f(X_{1:T}, \ldots, X_{D,T}) + \varepsilon_T,$$ 

with observations $Y_T = X_{1:T}$ and $[X_{1:T}, \ldots, X_{D,T}] = [X_{1:T-1}, \ldots, X_{T-L-1}, \ldots, X_{D,T-1}, \ldots, X_{D,T-L}]$, where $D = DL$. To estimate $f(\cdot, \ldots, \cdot)$, we consider the following least squares formulation

$$\min_{\beta} \sum_{t=1}^{T} w_{T,t} (Y_t - f(X_{1:t}, \ldots, X_{D,t}))^2$$

where $\{w_{T,t} \in [0, 1]\}$ are weights used to emphasize varying influences of the past data. The weights may also be used to accommodate different variance levels across dimensions. The appropriate choice of $\{w_{T,t} \in [0, 1]\}$ will be later discussed in Section III-C.

In order to estimate the nonlinear function $f(\cdot, \ldots, \cdot)$, we further assume a nonlinear additive model, i.e.

$$f(X_{1:t}, \ldots, X_{D,t}) = \mu + \sum_{i=1}^{D} f_i(X_i), \quad E\{f_i(X_i)\} = 0,$$ 

where $f_i$ are scalar functions, and expectation is with respect to the stationary distribution of $X_i$. The second condition is required for identifiability. To estimate $f_i$, we use B-splines (extensions of polynomial regression techniques [16]). In our presentation, we consider the additive model mainly for brevity. Our methods can be extended to models where there exist interactions among $X_1, \ldots, X_D$ using multidimensional splines in a straightforward manner.

We assume that there are $v$ spline basis of degree $\ell$ for each $f_i$. Incorporating the B-spline basis into regression, we write

$$f_i(x) = \sum_{j=1}^{v} c_{i,j} b_{i,j}(x),$$

$$b_{i,j}(x) = B(x | s_i, \ldots, s_{i,v-\ell+1})$$

where $s_1, \ldots, s_{i,v-\ell+1}$ are the knots and $c_{i,j}$ are the coefficients associated with the B-spline basis. Replacing these into (5), the problem of interest is now the minimization of

$$\hat{\varepsilon}_T = \sum_{t=1}^{T} w_{T,t} \left\{ Y_t - \mu - \sum_{i=1}^{D} \sum_{j=1}^{v} c_{i,j} b_{i,j}(X_{i,t}) \right\}^2$$

over $c_{i,j}, i = 1, \ldots, D, j = 1, \ldots, v$, under the constraint

$$\sum_{t=1}^{T} \sum_{j=1}^{v} c_{i,j} b_{i,j}(x_i) = 0, \quad i = 1, \ldots, L.$$ 

which is the sample analog of the constraint in (4). Equivalently, we obtain an unconstrained optimization problem by centering the basis functions. Let $b_{i,j}(x_{i,t})$ be replaced by $b_{i,j}(x_{i,t}) - T^{-1} \sum_{t=1}^{T} b_{i,j}(x_{i,t}).$ By proper rearrangement, (6) can be rewritten into a linear regression form

$$\hat{\varepsilon}_T = \sum_{t=1}^{T} w_{T,t} (Y_t - z_i^T \beta_T)^2$$

where $\beta_T$ is a $(1 + \hat{D}v) \times 1$ column vector to be estimated and $z_i$ is a $(1 + \hat{D}v) \times 1$ column vector $z_i = [1, b_{1,1}(x_{i,1}), \ldots, b_{1,v}(x_{i,1}), \ldots, b_{D,1}(x_{i,1}), \ldots, b_{D,v}(x_{i,1})].$ Let $Z_T$ be the design matrix of stacking the row vectors $z_i^T, t = 1, \ldots, T.$ Note that we have used $\beta_T$ instead of a fixed $\beta$ to emphasize that $\beta_T$ may vary with time. We have used bold style for vectors to distinguish them from matrices. Let $W_T$ be the diagonal matrix whose elements are $w_{T,t}, t = 1, \ldots, T.$ Then the optimal $\beta_T$ in (8) can be
recognized as the MLE of the following linear Gaussian model
\[
Y_T = Z_T \beta_T + \varepsilon
\]  
where \( \varepsilon \in \mathcal{N}(0, W_0^{-1}) \). Here, we have used \( \mathcal{N}(\mu, V) \) to denote Gaussian distribution with mean \( \mu \) and covariance matrix \( V \).

To obtain a sharp model from large \( L \), we further assume that the expansion of \( f(\cdot, \ldots, \cdot) \) is sparse, i.e., only a few additive components \( f_i \) are active. Selecting a sparse model is critical as models of over large dimensions lead to inflated variance, thus compromising the predictive power. To this end, we give independent Laplace priors for each sub-vector of \( \beta_T \) corresponding to each \( f_i \). Our objective now reduces to obtaining the maximum a posteriori estimator (MAP)
\[
\log p(Y_T | \beta_T, Z_T) - \lambda_T \sum_{i=1}^{\tilde{D}} \| \beta_{T,i} \|_2
\]
\[
= -\frac{1}{2} \sum_{i=1}^{T} w_{T,i} (Y_i - z_{T}^T \beta) + \lambda_T \sum_{i=1}^{\tilde{D}} \| \beta_{T,i} \|_2 + c
\]
where \( c \) is a constant that depends only on \( W_T \). The above prior corresponds to the so called group LASSO \[17\]. The bold \( \beta_{T,i} \) is to emphasize that it is not a scalar element of \( \beta_T \) but a sub-vector of it. It will be interesting to consider adaptive group LASSO \[18\], i.e., to use \( \lambda_{T,i} \) instead of a unified \( \lambda_T \) and this is currently being investigated. We refer to \[5\] for a study of adaptive group LASSO for batch estimation.

**B. Implementation of SLANTS**

In order to solve the optimization problem given by (10), we build on an EM-based solution originally proposed for wavelet image restoration \[19\]. This was further applied to online adaptive filtering for sparse linear models \[20\] and nonlinear models approximated by Volterra series \[21\]. The basic idea is to decompose the optimization (10) into two parts that are easier to solve and iterate between them. One part involves linear updates, and the other involves group LASSO in the form of orthogonal covariance which leads to closed-form solution.

For now, we assume that the knot sequence \( t_{i,1}, \ldots, t_{i,v} \) for each \( i \) and \( v \) is fixed. Suppose that all the tuning parameters are well-defined. We introduce an auxiliary variable \( \tau_T \) that we refer to as the innovation parameter. This helps us to decompose the problem so that underlying coefficients can be iteratively updated. It also allows the sufficient statistics to be rapidly updated in a sequential manner. The model in (10) now can be rewritten as
\[
Y_T = Z_T \theta_T + W_T^{\frac{1}{2}} \varepsilon_1, \quad \theta_T = \beta_T + \tau_T \varepsilon_2,
\]
where
\[
\varepsilon_1 \in \mathcal{N}(0, (I - \tau_T^2 W_T^{\frac{1}{2}} Z_T Z_T^T W_T^{\frac{1}{2}})) \quad \varepsilon_2 \in \mathcal{N}(0, I)
\]
(11)

We treat \( \theta_T \) as the missing data, so that an EM algorithm can be derived. By basic calculations similar to that of \[19\], we obtain the \( k^{th} \) step of EM algorithm

**E step:**
\[
Q(\beta | \hat{\beta}^{(k)}_T) = -\frac{1}{2T \tau_T} \| \beta - r^{(k)} \|_2^2 - \lambda_T \sum_{i=1}^{\tilde{D}} \| \beta_i \|_2
\]
where
\[
r^{(k)} = (I - \tau_T^2 \Lambda_T) \hat{\beta}^{(k)} + \tau_T^2 B_T,
\]
\[
A_T = Z_T^T W_T Z_T, \quad B_T = Z_T^T W_T Y_T.
\]
(13)

The derivation of Equation (12) is included in the appendix. Theorem 1: \( \hat{\beta}^{(k+1)}_T \) is the maximum of \( Q(\beta | \beta^{(k)}_T) \) given by
\[
\hat{\beta}^{(k+1)}_T = \left[ 1 - \frac{\lambda_T \tau_T^2}{\| r^{(k)} \|_2} \right] r^{(k)} + i = 1, \ldots, \tilde{D}.
\]
Suppose that we have obtained the estimator \( \hat{\beta}_T \) at time step \( T \). Consider the arrival of the \((T+1)\)th point \((y_{T+1}, z_{T+1})\), respectively corresponding to the response and covariates of time step \( T+1 \). We first compute \( r^{(0)}_{T+1} \), the initial value of \( r \) to be input the EM at time step \( T + 1 \):
\[
r^{(0)}_{T+1} = (I - \tau_T^2 \Lambda_{T+1}) \hat{\beta}_T + \tau_T^2 B_{T+1},
\]
where
\[
A_{T+1} = (1 - \gamma_{T+1}) A_T + \gamma_{T+1} z_{T+1} z_{T+1}^T, \\
B_{T+1} = (1 - \gamma_{T+1}) B_T + \gamma_{T+1} y_{T+1} z_{T+1}.
\]
(17)
Then we run the above EM for \( K > 0 \) iterations to obtain an updated \( \hat{\beta}_{T+1} \).

**Remark 1:** In the above equation, \( \{\gamma_t\} \) is a nonnegative sequence which we refer to as the step sizes. We shall elaborate on its relation with \( \{W_t\} \) in Subsection II-C.

SLANTS can be efficiently implemented. The recursive computation of \( A_T \) (resp. \( B_T \)) reduces the complexity from \( O(D^3) \) to \( O(D^2) \) (resp. from \( O(D^2) \) to \( O(D) \)). Moreover, straightforward computations indicate that the complexity of SLANTS at each time \( t \) is \( O(D^2) \), which does not depend on \( T \). Coordinate descent \[22\] is perhaps the most widely used algorithm for batch LASSO. Adapting coordinate descent to sequential setting has the same complexity for updating sufficient statistics. But straightforward use of batch LASSO has complexity \( O(D^3) \).

**Theorem 1:** At each iteration, the mapping from \( \hat{\beta}^{(k)}_T \) to \( \hat{\beta}^{(k+1)}_T \) is a contraction mapping for any \( \tau_T \), whenever the absolute values of all eigenvalues of \( I - \tau_T^2 \Lambda_{T+1} \) are less than one. In addition, there exists a unique global maximum point of (10) denoted by \( \beta_T \), and the error \( \| \beta_T - \beta_T \|_2 \) decays exponentially in \( k \).

**Remark 2:** The theorem states that EM can converge exponentially fast to the MAP of (10). From its assumption, it can be directly calculated that (10) as a function of \( \beta_T \) is strictly concave. We note that the assumption is not mild, so the application of Theorem 1 is limited. But the proposed algorithm does converge exponentially fast in our various synthetic and real data experiments. The proof of Theorem 1 is given in the appendix.
C. The choice of tuning parameters: from a prequential perspective

To evaluate the predictive power of an inferential model estimated from all the currently available data, ideally we would apply it to independent and identically generated datasets. However, it is not realistic to apply this cross-validation idea to real-world time series data, since real data is not permutable and has a “once in a lifetime” nature. As an alternative, we adopt a prequential perspective [23] that the goodness of a sequential predictive model shall be assessed by its forecasting ability.

Specifically, we evaluate the model in terms of the one-step prediction errors upon each newly arrived data point and subsequently tune the necessary control parameters, including regularization parameter $\lambda$ and innovation parameter $\tau$ (see details below). Automatic tuning of the control parameters are almost a necessity in many real-world applications in which any theoretical guidance (e.g., our Theorem 2) may be insufficient or unrealistic. Throughout our algorithmic design, we have adhered to the prequential principle and implemented the following strategies.

The choice of $w_{T,t}$: In view of equation (17), $w_{T,t}$ is determined by $w_{1,1} = \gamma_1$, and

$$w_{t,t} = \gamma_t, \quad w_{t,j} = w_{t-1,j}(1 - \gamma_t), \quad j = 1, \ldots, t - 1,$$

for $t > 1$.

It includes two special cases that have been commonly used in the literature. The first case is $\gamma_1 = 1/t$. It is easy to verify that $w_{T,t} = 1/T, t = 1, \ldots, T$ for any $T$. This leads to the usual least squares. The second case is $\gamma_1 = c$ where $c$ is a positive constant. It gives $w_{T,t} = c(1 - c)^T - t, t = 1, \ldots, T$. From (3), the estimator of $r$ remains unchanged by rescaling $w_{T,t}$ by $1/c$, i.e. $w_{T,t} = (1 - c)^{T-t}$ which is a series of powers of $1 - c$. The value $1 - c$ has been called the “forgetting factor” in the signal processing literature and used to achieve adaptive filtering [20].

The choice of $\tau_T$: Because the optimization problem

$$\log p(Y_T | \beta_T) - \lambda_T \sum_{i=1}^{L} ||\beta_{T,i}||_2$$

is convex, as long as $\tau_T$ is proper, the EM algorithm converges to the optimum regardless of what $\tau_T$ is. But $\tau_T$ affects the speed of convergence of EM as $\lambda_T \gamma_T^2$ determines how fast $\beta_T$ shrinks. Intuitively the larger $\tau_T$ is, the faster is the convergence. Therefore we prefer $\tau_T$ to be large and proper. A necessary condition for $\tau_T$ to be proper is to ensure that the covariance matrix of $\varepsilon_1$ in

$$\varepsilon_1 \in N'(0, I - \tau_T^2 W^{1/2} Z_T Z_T^T W^{1/2}), \quad \varepsilon_2 \in N'(0, I)$$

is positive definite. Therefore, there is an upper bound $\tau_T$ for $\tau_T$, and $\tau_T$ converges to a constant $\bar{\tau}$ under some mild assumptions (e.g. the stochastic process $X_i$ is stationary). Extensive experiments have shown that $\tau_T/2$ produces satisfying results in terms of model fitting. However, it is not computationally efficient to calculate $\tau_T$ at each $T$ in SLANTS. Nevertheless without computing $\tau_T$, we can determine if $\tau_T < \bar{\tau}$ by checking the EM convergence. If $\tau_T$ exceeds $\bar{\tau}$, the EM would diverge and coefficients go to infinity exponentially fast. This can be proved via a similar argument to that of proof of Theorem 1. This motivates a lazy update of $\tau_T$ with shrinkage only if EM starts to diverge.

The choice of $\lambda_T$: On the choice of regularization parameter $\lambda_T$, different methods have been proposed in the literature. The common way is to estimate the batch data for a range of different $\lambda_T$’s, and select the one with minimum cross-validation error. To reduce the underlying massive computation required for such an approach, in the context of Bayesian LASSO [24, 25] proposed an sequential Monte Carlo (SMC) based strategy to efficiently implement cross-validation. The main proposal is to treat the posterior distributions educed by an ordered sequence of $\lambda_T$ as $\pi_t, t = 0, 1, \ldots$, the target distributions in SMC, and thus avoid the massive computation of applying Markov chain Monte Carlo (MCMC) for each $\lambda$ independently. Another method is to estimate the hyper-parameter $\lambda_T$ via empirical Bayes method [24]. In our context, however, it is not clear whether the Bayesian setting with MCMC strategy can be efficient, as the dimension $Lv$ can be very large. An effective implementation technique is to run three channels of our sequential modeling, corresponding to $\lambda_T^* = \lambda_T / \delta, \lambda_T^* = \lambda_T^* / \delta$, where $\delta > 1$ is a small step size. The one with minimum average prediction error over the latest window of data was chosen as the new $\lambda_T$. For example, if $\lambda_T^*$ gives better performance, let the three channels be $\lambda_T^* / \delta, \lambda_T^* / \delta, \lambda_T^* / \delta$. If there is an underlying optimal $\lambda^*$ which does not depend on $T$, we would like our channels to converge to the optimal $\lambda^*$ by gradually shrinking the stepsize $\delta$. Specifically in case that the forgetting factor $\gamma_t = 1/t$, we let $\delta_T = 1 + \lambda_T \tau_T - 1$ so that the step size $\delta_T \rightarrow 1$ at the same speed as weight of new data.

The choice of knots: The main difficulty in applying spline approximation is in determining the number of the knots to use and where they should be placed. Jupp [26] has shown that the data can be fit better with splines if the knots are free variables. de Boor suggests the spacing between knots is decreased in proportion to the curvature (second derivative) of the data. It has been shown that for a wide class of stationary process, the number of knots should be of the order of $O(T^\zeta)$ for available sample size $T$ and some positive constant $\zeta$ to achieve a satisfying rate of convergence of the estimated nonlinear function to the underlying truth (if it exists) [27]. Nevertheless, under some assumptions, we will show in Theorem 2 that the prediction error can be upper bounded by an arbitrarily small number (which depends on the specified number of knots). It is therefore possible to identify the correct nonzero additive components in the sequential setting. On the other hand, using a fixed number of knots is computationally desirable because sharp selection of significant spline basis/support in a potentially varying environment is computationally intensive. It has been observed in our synthetic data experiments that the variable selection results are not very sensitive to the number of knots as long as this number is moderately large (e.g. around $v = 10$).
III. Theoretical Results

Consider the harmonic step size $\gamma_t = 1/t$. For now assume that the sequential update at each time $t$ produces $\beta_t$ that is the same as the penalized least squares estimator given batch data. We are interested in two questions. First, how to extend the current algorithm in order to take into account an ever-increasing number of dimensions? Second, is it possible to select the “correct” nonzero components as sample size increases?

The first question is important in practice as any prescribed finite number of dimensions/time series may not contain the data-generating process, and it is natural to consider more candidates whenever more samples are obtained. It is directly related to the widely studied high-dimensional regression for batch data. In the second question, we are not only interested in optimizing the prediction error but also to obtain a consistent selection of the true nonzero components. Moreover, in order to maintain low complexity of the algorithm, we aim to achieve the above goal by using a fixed number of spline basis. We thus consider the following setup. Recall the predictive model \[ (1) \] and its alternative form \[ (2) \]. We assume that $L$ is fixed while $D$ is increasing with sample size $T$ at certain rate.

Following the setup of \[ (25) \], we suppose that each $X_d$ takes values from a compact interval $[a, b]$. Let $[a, b]$ be partitioned into $J$ equal-sized intervals $\{I_j\}_{j=1}^J$, and let $\mathcal{G}$ denote the space of polynomial splines of degree $\ell \geq 1$ consisting of functions $g(\cdot)$ satisfying 1) the restriction of $g(\cdot)$ to each interval is a polynomial of degree $\ell$, and 2) $g(\cdot) \in C^{\ell-1}[a, b]$ ($\ell$-times continuously differentiable). Typically, splines are called linear, quadratic or cubic splines accordingly as $\ell = 1, 2, \text{or}$ 3. There exists a normalized B-spline basis $\{b_j\}_{j=1}^B$ for $\mathcal{G}$, where $v = J + \ell$, and any $f_j(x) \in \mathcal{G}$ can be written in the form of \[ (5) \]. Let $k \leq \ell$ be a nonnegative integer, $\beta \in (0, 1)$ that $p = k + \beta > 0.5$, and $M > 0$. Suppose each considered (non)linear function $f$ has $k$th derivative, $f^{(k)}$, and satisfies the Holder condition with exponent $\beta$: $|f^{(k)}(x) - f^{(k)}(x')| \leq M|x - x'|^\beta$ for $x, x' \in [a, b]$. Define the norm $\|f\|_2 = \sqrt{\int_a^b f(x)^2 dx}$. Let $f^* \in \mathcal{G}$ be the best $L_2$ spline approximation of $f$. Standard results on splines imply that $\|f - f^*\|_2 = O(v^{-p})$ for each $d$. The spline approximation is usually an estimation under a mis-specified model class (unless the data-generating function is low-degree polynomials), and large $v$ narrows the distance to the true model. We will show that for large enough $v$, it is possible to achieve the aforementioned two goals. To make the problem concrete, we need the following assumptions on the data-generating procedure.

Assumption 1: The number of additive components is finite and will be included into the candidate set in finite time steps. In other words, there exists a “significant” variable set $S_0 = \{1, \ldots, d_{0,0}\}$ such that 1) $f_d(x) \neq 0$ for each $d \in S_0$, 2) $f_d(x) \equiv 0$ for $d \notin S_0$, and 3) both $D_0$ and $i_{D_0}$ are finite integers that do not depend on sample size $T$.

We propose two steps for a practitioner targeting two goals given below.

**Step 1. (unbiasedness)** This step aims to discover the significant variable set with probability close to one as more data is collected. The approach is to minimize the objective function in \[ (10) \], and it can be efficiently implemented using the proposed sequential algorithm in Section II-B with negligible error (Theorem \[ (1) \]). In the case of equal weights $w_{T,d} = 1/T$, it can be rewritten as

$$\|Y_T - Z_T \beta_T\|_2^2 + \lambda_T \sum_{i=1}^D \|\beta_{T,i}\|_2$$

where $\lambda_T = 2T \lambda_T$. Due to Assumption \[ (1) \] the significant variable set $S_0$ is included in the candidate set $\{1, \ldots, D\}$ for sufficiently large $T$. Our selected variables are those whose group coefficients are nonzero, i.e. $S_1 = \{d : 1 \leq d \leq D, \beta_{T,d} \neq 0\}$. We are going to prove that all the significant variables will be selected by minimizing \[ (20) \] with appropriately chosen $\lambda_T$, i.e., $S_0 \subseteq S_1$.

**Step 2. (minimal variance)** The second step is optional and it is applied only when a practitioner’s goal is to avoid selecting any redundant variables outside $S_0$. Suppose that we obtain a candidate set of $\tilde{D}$ variables $S_1$ (satisfying $S_0 \subseteq S_1$ from the previous step). Since a thorough search over all subsets of variables is computationally demanding, we use a backward stepwise procedure. We start with the set of selected variables $S_1$, delete one variable at a time by minimizing the MSE of a spline model with $v_T = T^c$ number of equally spaced knots. We note that $v_T$ in the optional Step 2 can be different from the $v$ in SLANTS. Specifically, suppose that at step $k (k = 1, 2, \ldots)$, the survived candidate models are indexed by $S(k)$. We solve the least-squares problem for each $\tilde{d} \in S(k)$

$$\hat{e}^{(k)}_d = \min_{\mu, c_d,j} \sum_{t=1}^T (Y_i - \mu - \sum_{d' \in S(k)} c_{d',j} b_{d',j}(X_{d,t}))^2$$

where $S(k) = S(k-1) \setminus \{\tilde{d}\}$, and select $\tilde{d} = \hat{d}^{(k)}_d$ that minimize $\hat{e}^{(k)}_d$ with minimum denoted by $\hat{e}^{(k)}_d$. Here $A - B$ denotes the set of elements that are in a set $A$ but not in a set $B$. We let $S(k) = S(k-1) \setminus \{d_k\}$. By default, we let $S(0) = S_1$ and use $e^{(0)}$ to denote the minimum of \[ (21) \] with $S = S_1$. If $\hat{e}^{(k-1)} - \hat{e}^{(k)} < (v_T \log T)/T$, i.e., the gain of goodness of fit is less than the incremented Bayesian information criterion (BIC) penalty \[ (29) \], then we stop the procedure and output $S_2 = S(k-1)$; otherwise we proceed to the $(k + 1)$th iteration. We prove that the finally selected subset $S_2$ satisfies $\lim_{T \to \infty} \text{pr}(S_2 = S_0) = 1$.

Before we proceed to the theoretical result, we introduce some necessary assumptions and their interpretations.

**Assumption 2:** There is a positive constant $c_0$ such that $\min_{d \in S_0} \|f_d\|_2 \geq c_0$.

**Assumption 3:** The noises $e_t$ are sub-Gaussian distributed, i.e., $E(e^{\epsilon e_t}) \leq e^{\epsilon^2 \sigma^2/2}$ for a constant $\sigma > 0$ and any $\epsilon \in \mathbb{R}$.

**Assumption 4:** Suppose that $S_1$ is a finite subset of $\{1, \ldots, D\}$. In addition, the “design matrix” $Z_{S_1}$ satisfies $Z_{S_1}^T Z_{S_1}/T \geq \kappa$ for a positive constant $\kappa$ that depend only on $v$ (the number of splines).

We use $o_p(1)$ and $O_p(1)$ to denote a sequence of random variables that converges in probability to zero, and that is stochastically bounded, respectively. We use $O(1)$ to denote a bounded deterministic sequence.
Theorem 2: Suppose that Assumptions 1-4 hold. Then for any given \( v \) it holds that
\[
\|\beta_{S_1} - \beta_{S_1}\|^2 \leq 8c_2v^{-2p}/\kappa + O_p(T^{-1}\log D) + O_p(T^{-1}) + O(T^{-2}\lambda^2)
\]  
for some positive constant \( c_2 \). If we further assume that \( \log D = o(T) \), \( \lambda = o(T) \), then there exists a constant \( c_1 > 0 \) such that for all \( v > c_1c_0^{-1/p}\max\{1, c_0^{-1/p}\} \), \( \lim_{T\to\infty} \Pr(S_0 \subseteq S_1) = 1 \).

Remark 3: Theorem 2 gives an error bound between the estimated spline coefficients with the oracle, where the first term is dominating. As a result, if \( v \) is sufficiently large, then it is guaranteed that \( S_0 \) will be selected with probability close to one. We note that the constant \( c_1 \) depends only on the true nonlinear function and the selected spline basis function. In proving Theorem 2, Assumptions 2-3 serve as standard conditions to ensure that a significant variable is distinguishable, and that any tail probability could be well bounded. Assumption 4 is needed to guarantee that if the estimated coefficients \( \beta \) produces low prediction errors, then it is also close to the true (oracle) coefficients. This assumption is generally required by requiring \( \lambda > c\sqrt{T}\log D \). See for example [5], [30].

To prove the consistency in Step 2, we also need the following assumption (which further requires that the joint process is strictly stationary and strongly mixing).

Assumption 5: \( \sup_x \{\E(|Y_i|^r | X_i = x)\} < \infty \) for some \( r > 2 \).

The \( \alpha \)-mixing coefficient is defined as \( \alpha_S(j) = \sup\{P(E_y \cap E_x) - P(E_y)P(E_x) : E_y \in \sigma(\{Y_{i, t}, X_{d, t} ; d \in S ; t \leq n\}), E_x \in \sigma(\{Y_{i, t}, X_{d, t} ; d \in S ; t \geq n+j\})\} \), where \( \sigma(\cdot) \) denotes the \( \sigma \)-field generated by the random variables inside the parenthesis.

Assumption 6: The process \( \{(X_{d, t} ; d \in S_1)\} \) is strictly stationary, and the joint process \( \{(Y_{i, t}, X_{d, t} ; d \in S_1)\} \) is \( \alpha \)-mixing with coefficient
\[
\alpha_S(j) \leq \min\{O(j^{-2.5}\zeta/(1-\zeta)), O(j^{-2r/(r-2)})\},
\]
where \( \zeta \) has been defined in Step 2.

Theorem 3: Suppose that Assumptions 1-6 hold, then the \( S_2 \) produced by the above step 2 satisfies \( \lim_{T \to \infty} \Pr(S_2 = S_0) = 1 \).

IV. NUMERICAL RESULTS

In this section, we present experimental results to demonstrate the theoretical results and the advantages of SLANTS on both synthetic and real-world datasets. The synthetic experiments include cases where the data-generating model is fixed over time, is varying over time, or involves large dimensionality.

A. Synthetic data experiment: modeling nonlinear relation in stationary environment

The purpose of this experiment is to show the performance of SLANTS in stationary environment where the data-generating model is fixed over time. We generated synthetic data using the following nonlinear model
\[
X_{1,t} = \epsilon_{1,t} \\
X_{2,t} = 0.5X_{1,t-1}^2 - 0.8X_{1,t-7} + 0.2\epsilon_{2,t}, \ t = 1, \ldots, 500
\]
where \( \epsilon_{1,t} \) and \( \epsilon_{2,t} \) are i.i.d. standard Gaussian. The goal is to model/forecast the series \( X_{2,t} \). We choose \( L = 8 \), and place \( v = 10 \) quadratic splines in each dimension. The knots are equally spaced between the 0.01 and 0.99 quantiles of observed data. The initial \( L \) values of \( X_{2,t} \) are set to zeros. We choose the step size \( \gamma_t = 1/t \) to ensure convergence.

Simulation results are summarized in Fig. 1. The top-left plot shows the convergence of all the \( 2 \times 8 \times 10 = 160 \) spline coefficients. The right-top plot shows how the eight nonlinear components \( f_d, d = 1, \ldots, 8 \) evolve, where the number 1-8 indicate each additive component (splines). The values of each function are centralized to zero for identifiability. The remaining two plots show the optimal choice of control parameters \( \lambda_t \) and \( \tau_t \) that have been automatically tuned over time. In the experiment, the active components \( f_1 \) and \( f_2 \) are correctly selected and well estimated. It is remarkable that the convergence is mostly achieved after only a few incoming points (less than the number of coefficients 160).

![Fig. 1. Four subplots show the estimated coefficients of splines, nonlinear functions, and trace plots of automatically-tuned regularization parameter \( \lambda_t \) and innovation parameter \( \tau_t \). A demo video is available in the supplement.](image-url)
where $\epsilon_{1,t}$ and $\epsilon_{2,t}$ are i.i.d. standard Gaussian. $u_{1,t}$ are i.i.d. uniform on $[-1, 1]$. The goal is to model the series $X_{2,t}$. Compared with the previous experiment, the only difference is that the forgetting factor is set to $\gamma = 0.99$ in order to track potential changes in the underlying true model. Fig. 2 shows that SLANTS successfully tracked a change after the change point $t = 500$. The top plot in Fig. 2 shows the inference results right before the change. It successfully recovers the quadratic pattern of lag 1 and linear effect of lag 7. The bottom plot in Fig. 2 shows the inference results at $t = 1000$. It successfully finds the exponential curve of lag 7 and reversed sign of the quadratic curve of lag 1. From the bottom left subplot we can see how the autotuning regularization parameter decreases since the change point $t = 500$.

Granger-type of causality) among multi-dimensional time series. We have generated a 9-dimensional time series using the following nonlinear network model,

$$X_{1,t} = \epsilon_{1,t}$$
$$X_{2,t} = 0.6X_{3,t-1} + \epsilon_{2,t}$$
$$X_{3,t} = 0.3X_{4,t-2} + \epsilon_{3,t}$$
$$X_{4,t} = 0.7X_{5,t-1} - 0.2X_{5,t-2} + \epsilon_{4,t}$$
$$X_{5,t} = -0.2X_{2,t-1} + \epsilon_{5,t}$$
$$X_{6,t} = 0.5X_{6,t-2} + 1 + \epsilon_{6,t}$$
$$X_{7,t} = 2\exp(-X_{7,t-2}^2) + \epsilon_{7,t}$$
$$X_{8,t} = 6X_{7,t-1} - 5X_{9,t-2} + \epsilon_{8,t}$$
$$X_{9,t} = -X_{6,t-1} + 0.9X_{7,t-2} + \epsilon_{9,t}$$

where $\epsilon_{1,t}$ and $\epsilon_{2,t}$ are i.i.d. standard Gaussian. The initial $L$ values are set to zero. The goal is to model each dimension and draw sequential causality graph based on the estimation. We choose $L = 2$, $v = 10$ and $\gamma_t = 1/t$. For illustration purpose, we only show the estimation for $X_{9,t}$. The left-top plot shows the 9 dimensional raw data that are sequentially obtained. The right-top plot shows the convergence of the $DLv = 9 \times 2 \times 10 = 180$ coefficients in modeling $X_{9,t}$. The right-bottom plot shows how the nonlinear components $f : X_{6,t-1} \rightarrow X_{9,t}$ and $f : X_{7,t-2} \rightarrow X_{9,t}$ evolve. Similar as before, the values of each function are centralized to zero for identifiability. The left-bottom plot shows the causality graph, which is the digraph with black directed edges and edge labels indicating functional relations. For example, in modeling $X_{9,t}$, if the function component corresponding to $X_{6,t-1}$ is nonzero, then we draw a directed edge from 6 to 9 with label 1; if the function components corresponding to both $X_{6,t-1}$ and $X_{6,t-2}$ are nonzero, then we draw a directed edge from 6 to 9 with label 12. The true causality graph (determined by the above data generating process) is drawn as well, in red thick edges. From the simulation, the discovered causality graph quickly gets close to the truth.

C. Synthetic data experiment: causal discovery for multi-dimensional time series

The purpose of this experiment is to show the performance of SLANTS in identifying nonlinear functional relation (thus
D. Synthetic data experiment: computational cost

The purpose of this experiment is to show that SLANTS is computationally efficient by comparing it with standard batch group LASSO algorithm. We use the same data generating process in the first synthetic data experiment, and let the size of data be $T = 100, 200, \ldots, 1000$.

We compare SLANTS with the standard R package “grplasso” [31] and “gglasso” [32] which implement widely used group LASSO algorithms. The package “gglasso” implements the efficient active-set algorithm proposed in [33]. For the two packages, at each time $t$, solution paths on a fixed grid of 100 penalties are calculated. To provide fair comparisons, we run SLANTS in two ways. The first is the proposed algorithm with adaptive tuned penalties. In the table, it is denoted as SLANTS(a). The second is SLANTS without adaptive tuning but also run on a fixed grid of 100 equivalent penalties as in “grplasso” and “gglasso”, denoted as SLANTS(b). In computing solution paths, we adopted the techniques suggested in [33]. The results are shown in Table I.

Table I shows the time in seconds for SLANTS(a), SLANTS(b), gglasso, and grplasso to run through a dataset sequentially with different size $T$. Each run is repeated 30 times and the standard error of running time is shown in parenthesis. From Table I, the computational cost of SLANTS grows linearly with $T$ while gglasso and grplasso grow much faster. Moreover, the prediction error is very similar for SLANTS(b), gglasso and grplasso on the grid of penalties. This is understandable as they calculate the solution to the same optimization problem. SLANTS(a) approaches the optimal prediction error as the penalty parameter is stabilized. But SLANTS(b) is faster than R. However, the growth of computational cost of SLANTS is much slower than that of grplasso, and thus SLANTS is faster for large $T$.

E. Real data experiment: Boston weather data from 1980 to 1986

In this experiment, we study the daily Boston weather data from 1980 Jan to 1986 Dec. with $T = 2557$ points in total. The data is a six-dimensional time series, with each dimension corresponding respectively to temperature (K), relative humidity (%), east-west wind (m/s), north-south wind (m/s), sea level pressure (Pa), and precipitation (mm/day). In other words, the raw data is in the form of $X_{d,t}$, $d = 1, \ldots, 6$, $t = 1, \ldots, T$. We plot the raw data corresponding to year 1980 (i.e. $X_{d,t}$, $d = 1, \ldots, 6$, $t = 1, \ldots, 365$) in Fig. 4.

We compare the predictive performance of SLANTS with that of a linear model. For brevity, suppose that we are going to predict the east-west wind. We chose the autoregressive model of order 3 (denoted by AR(3)) as the representative linear model. The order was chosen by applying Bridge criterion [34] to the batch data of $T$ observations. We started processing the data from $t_0 = 10$, and for each $t = t_0 + 1, \ldots, T$ the one-step ahead prediction error $\hat{e}_t$ was made by applying AR(3) and SLANTS to the currently available $t - 1$ observations. The cumulated average prediction error at time step $t$ is computed to be \[ \sum_{i=t_0+1}^{t} \frac{\hat{e}_i}{t - t_0}, \] where $\hat{e}_i$ is the squared difference between the true observation and our prediction at time step $i$. The results are shown in Fig. 5(a). At the last time step, the significant (nonzero) functional components are the third, fourth, and sixth dimension, corresponding to EW wind, NS wind, precipitation, have been plotted in Fig. 5(b), (c), (d), respectively. From the plot, the marginal effect of $X_{4,t}$ on $X_{3,t+1}$ is clearly nonlinear. It seems that the correlation is low for $X_{4,t} < 0$ and high for $X_{4,t} > 0$. In fact, if we let $\mathcal{I} = \{ t : X_{4,t} > 0 \}$, the correlation of $\{ X_{4,t} : t \in \mathcal{I} \}$ with $\{ X_{3,t+1} : t \not\in \mathcal{I} \}$ is 0.25 (with $p$ value $1.4 \times 10^{-8}$) while $\{ X_{4,t} : t \not\in \mathcal{I} \}$ with $\{ X_{3,t+1} : t \not\in \mathcal{I} \}$ is $-0.05$ (with $p$ value 0.24).

F. Real data experiment: the weekly unemployment data from 1996 to 2015

In this experiment, we study the US weekly unemployment initial claims from Jan 1996 to Dec 2015. The data is a one-dimensional time series with $T = 1043$ points in total. We plot the raw data in Fig. 6.

Though the data exhibits strong cyclic pattern, it may be difficult to perform cycle-trend decomposition in a sequential setting. We explore the power of SLANTS to do lag selection to compensate the lack of such tools.

We compare three models. The first model, AR(5), is linear autoregression with lag order 5. The lag order was chosen by applying Bridge criterion [34] to the batch data. The second and third are SLANTS(1) with linear spline and SLANTS(2) with quadratic splines. SLANTS(1) have 1 spline per dimension, which is exactly LASSO with auto-tuned penalty parameter in SLANTS. SLANTS(2) have 8 splines per dimension. We allow SLANTS to select from a maximum lag of 55, which is roughly the size of annual cycle of 52 weeks.

<table>
<thead>
<tr>
<th>$T$</th>
<th>SLANTS(a)</th>
<th>SLANTS(b)</th>
<th>gglasso</th>
<th>grplasso</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>4.8(0.1)</td>
<td>35.5(2.2)</td>
<td>32.6(2.5)</td>
<td>9.9(3.6)</td>
</tr>
<tr>
<td>200</td>
<td>11.1(0.3)</td>
<td>82.5(3.8)</td>
<td>110.8(7.9)</td>
<td>98.9(9.3)</td>
</tr>
<tr>
<td>300</td>
<td>15.4(0.7)</td>
<td>131.4(5.6)</td>
<td>204.3(10.8)</td>
<td>238.6(16.7)</td>
</tr>
<tr>
<td>400</td>
<td>21.4(0.7)</td>
<td>180.3(7.2)</td>
<td>296.2(10.7)</td>
<td>392.3(21.4)</td>
</tr>
<tr>
<td>500</td>
<td>26.0(0.9)</td>
<td>228.8(9.0)</td>
<td>386.8(12.1)</td>
<td>563.2(26.3)</td>
</tr>
<tr>
<td>600</td>
<td>31.3(1.1)</td>
<td>277.0(10.8)</td>
<td>477.5(13.4)</td>
<td>753.3(30.6)</td>
</tr>
<tr>
<td>700</td>
<td>37.1(1.2)</td>
<td>324.8(12.7)</td>
<td>569.4(15.0)</td>
<td>961.3(34.6)</td>
</tr>
<tr>
<td>800</td>
<td>42.1(1.4)</td>
<td>372.3(14.5)</td>
<td>663.0(19.1)</td>
<td>1189.0(38.5)</td>
</tr>
<tr>
<td>900</td>
<td>46.3(1.6)</td>
<td>419.4(16.3)</td>
<td>758.6(20.4)</td>
<td>1435.7(43.3)</td>
</tr>
<tr>
<td>1000</td>
<td>53.3(1.8)</td>
<td>466.3(18.1)</td>
<td>856.5(21.3)</td>
<td>1702.5(46.8)</td>
</tr>
</tbody>
</table>
Fig. 4. A graph showing the raw data of (a) temperature (K), (b) relative humidity (%), (c) east-west wind (m/s), (d) north-south wind (m/s), (e) sea level pressure (Pa), and (f) precipitation (mm/day).

Fig. 5. A graph showing (a) the cumulated average one-step ahead prediction error of east-west wind (m/s) produced by two approaches, and east-west wind decomposed into nonlinear functions of lagged values of (b) east-west wind, (c) north-south wind (m/s), and (c) precipitation (mm/day). The functions were output from SLANTS at the last time step \( t = T \).

Fig. 6. A graph showing the raw data of the number of unemployment initial claims.

Fig. 7. A graph showing the cumulated average one-step ahead prediction error at each time step produced by three approaches: linear autoregressive model, SLANTS with linear splines, and SLANTS with quadratic splines. Quadratic splines from SLANTS(2) are almost linear, which means the data has little nonlinearity. So SLANTS(1) performs best overall.

V. CONCLUDING REMARKS

to address several challenges in time series prediction that arises from environmental science, economics, and finance, we proposed a new method to model nonlinear and high dimensional time series data in a sequential and adaptive manner. The performance of our method was demonstrated by both
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSP.2017.2716898, IEEE Transactions on Signal Processing

synthetic and real data experiments. We also provided rigorous theoretical analysis of the rate of convergence, estimation error, and consistency in variable selection of our method.

Future work may include modeling and joint prediction of \(X_{1,T}, \ldots, X_{D,T}\). Currently, the prediction is separated into \(D\) individual problems. The performance may be further enhanced by considering potential correlations of innovations in each series. Adaptive placement of knots is another direction for future work. The knot sequence should adequately cover the range of data. In this paper, we assumed that the range of data is known. In some practical applications, however, the range may vary over time. In such case, it would be helpful to add a rejuvenation step that routinely updates the empirical domain of the data (and thus the knot placement).

**Appendix**

We prove Theorems 1-3 in the appendix. For any real-valued column vector \(x = [x_1, \ldots, x_m]\), we let \(\|x\|_2 = (\sum_{i=1}^{m} x_i^2)^{1/2}\) and \(\|x\|_A = x^T A x\). \(A\) denote respectively the \(\ell_2\) norm and matrix norm (with respect to \(A\), a positive semidefinite matrix).

**Proof of Theorem 1**

At time \(T\) and iteration \(k\), we define the functions \(h(\cdot)\) and \(g(\cdot)\) that respectively map \(\beta^{(k)}_T\) to \(r^{(k)}_T\) and from \(r^{(k)}_T\) to \(\beta^{(k+1)}_T\), namely \(\beta^{(k)}_T \rightarrow r^{(k)}_T \rightarrow \beta^{(k+1)}_T\). Suppose that the largest eigenvalue of \(I - \tau^2 A_{T+1}\) in absolute value is \(\xi (\xi < 1)\). We shall prove that

\[
\|g(h(x_1)) - g(h(x_2))\|_2 \leq \xi \|x_1 - x_2\|_2. \tag{23}
\]

It suffices to prove that \(\|h(\alpha_1) - h(\alpha_2)\|_2 \leq \xi \|\alpha_1 - \alpha_2\|_2\) and \(\|g(\chi_1) - g(\chi_2)\|_2 \leq \xi \|\chi_1 - \chi_2\|_2\) for any vectors \(\alpha_1, \alpha_2, \chi_1, \chi_2\). The first inequality follows directly from the definition of \(r^{(k)}\) in the E step, and \(h(\alpha_1) - h(\alpha_2) = (I - \tau^2 A_t)(\alpha_1 - \alpha_2)\). To prove the second inequality, we prove

\[
\|g(h(x_1)) - g(h(x_2))\|_2 \leq \|x_1, i - x_2, i\|_2, \tag{24}
\]

where \(\chi_{k,i} (i = 1, \ldots, L)\) are subvectors (groups) of \(\chi^{(k)}_T\) corresponding to \(\beta^{(k)}_T\) for either \(k = 1\) or \(k = 2\). For brevity we define \(\tilde{\tau} = \lambda T \tau^2\). We prove (24) by considering three possible cases: 1) \(\|x_{1,i}\|_2, \|x_{2,i}\|_2 \geq \tilde{\tau};\) 2) one of \(\|x_{1,i}\|_2\) and \(\|x_{2,i}\|_2\) is less than \(\tilde{\tau}\) while the other is no less than \(\tilde{\tau}\); 3) \(\|x_{1,i}\|_2, \|x_{2,i}\|_2 < \tilde{\tau}\). For case 1), \(g(x_{1,i}) = g(x_{2,i}) = 0\) and (24) trivially holds. For case 2), assume without loss of generality that \(\|x_{2,i}\|_2 < \tilde{\tau}\). Then

\[
\|g(h(x_1)) - g(h(x_2))\|_2 \leq \|x_1, i - x_2, i\|_2. \tag{25}
\]

For case 3), we note that \(g(x_{k,i})\) is in the same direction of \(x_{k,i}\) for \(k = 1, 2\). We define the angle between \(x_{1,i}\) and \(x_{2,i}\) to be \(\theta\), and let \(a = \|x_{1,i}\|, b = \|x_{2,i}\|\). By the Law of Cosines, to prove \(\|g(x_1) - g(x_2)\|_2 \leq \|x_1 - x_2\|_2^2\) it suffices to prove that

\[
(a - \tilde{\tau})^2 + (b - \tilde{\tau})^2 - 2(a - \tilde{\tau})(b - \tilde{\tau}) \cos(\theta) \\
\leq a^2 + b^2 - 2ab \cos(\theta). \tag{26}
\]

By elementary calculations, Inequality (25) is equivalent to

\[
2(1 - \cos(\theta)) \left\{ (a + b)(\tilde{\tau} - \tilde{\tau}^2) \right\} \geq 0,
\]

which is straightforward.

Finally, Inequality (23) and Banach Fixed Point Theorem imply that there exists a unique fixed point \(\hat{\beta}_T\) and

\[
\|\hat{\beta}^{(k)}_T - \beta_T\|_2 \leq \frac{\xi^k}{1 - \xi} \|\hat{\beta}^{(1)}_T - \beta_T^{(0)}\|_2
\]

which decays exponentially in \(k\) for any given initial value \(\beta_T^{(0)}\).

Moreover, the fixed point \(\hat{\beta}_T\) is MAP, because each EM iteration increases the value in (10) implicitly by increasing the value in \(Q(\beta | \beta_T^{(k)})\) (see the justification of EM algorithm [35], [36]).

**Proof of Theorem 2**

The proof follows standard techniques in high-dimensional regression settings [5], [10]. We only sketch the proof below. For brevity, \(\beta_T\) and \(\beta_T,T\) are denoted as \(\beta\) and \(\beta_T\), respectively.

Let \(\tilde{S}_1 = S_0 \cup S_1\) be the set union of truly nonzero set of coefficients and the selected nonzero coefficients. By the definition of \(\hat{S}_1\), we have

\[
\|Y - Z_{\tilde{S}_1}\hat{\beta}_{\tilde{S}_1}\|_2^2 + \lambda \sum_{d \in \tilde{S}_1} \|\beta_d\|_2
\]

\[
\leq \|Y - Z_{\hat{S}_1}\hat{\beta}_{\hat{S}_1}\|_2^2 + \lambda \sum_{d \in \hat{S}_1} \|\beta_d\|_2. \tag{26}
\]

Define \(\rho = Y - Z\beta\), and \(\psi = Z_{\hat{S}_1}(\hat{\beta}_{\hat{S}_1} - \beta_{\hat{S}_1})\). We obtain

\[
\|\psi\|_2^2 \leq 2\psi^T \rho + \lambda \sum_{d \in \hat{S}_1} (\|\beta_d\|_2 - \|\beta_d\|_2)
\]

\[
\leq 2\psi^T \rho + \lambda \sum_{d \in \hat{S}_1} (\|\beta_d\|_2 - \|\beta_d\|_2)
\]

\[
\leq 2\psi^T \rho + \lambda \sqrt{|S_0|} \|\beta_{S_1} - \beta_{\hat{S}_1}\|_2
\]

\[
\leq 2\psi^T \rho + \lambda \sqrt{|S_1|} \|\beta_{S_1} - \beta_{\hat{S}_1}\|_2
\]

\[
\leq 2\|\psi\|_2 \|\rho\|_2 + \lambda \sqrt{|S_1|} \|\beta_{S_1} - \beta_{\hat{S}_1}\|_2
\]

where the first inequality is rewritten from (25), the second and fourth follow from (26), the third and fifth follow from Cauchy inequality. From the above equality and 2) \(\|\psi\|_2 \|\rho\| \leq \|\psi\|_2^2 + 2\|\rho\|_2^2\), we obtain

\[
\|\psi\|_2^2 \leq 4\|\rho\|_2^2 + 2\lambda \sqrt{|S_1|} \|\beta_{S_1} - \beta_{\hat{S}_1}\|_2. \tag{27}
\]

On the other hand, Assumption 4 gives \(\|\psi\|_2^2 \geq \kappa T \|\beta_{S_1} - \beta_{\hat{S}_1}\|_2^2\). Therefore,

\[
\kappa T \|\beta_{S_1} - \beta_{\hat{S}_1}\|_2^2 \leq 4\|\rho\|_2^2 + 2\lambda \sqrt{|S_1|} \|\beta_{S_1} - \beta_{\hat{S}_1}\|_2
\]

\[
\leq 4\|\rho\|_2^2 + \frac{2\lambda|S_1|}{\kappa T} + \frac{\kappa T}{2} \|\beta_{S_1} - \beta_{\hat{S}_1}\|_2^2
\]

which implies that

\[
\|\beta_{S_1} - \beta_{\hat{S}_1}\|_2^2 \leq 8|S_1|/(\kappa T) + 4\lambda^2 |S_1|/(\kappa^2 T^2). \tag{28}
\]

In order to bound \(\beta_{S_1} - \beta_{\hat{S}_1}\), it remains to bound \(\|\rho\|_2^2\). Since \(\rho_t\) can be written as

\[
\varepsilon_t + \sum_{d \in \hat{S}_1} \{f_d(X_{d,t}) - f_d'(X_{d,t})\} + (\mu - \hat{Y}),
\]
Thus, we have \( \hat{c}(k) - c \leq O_p(v_T T^{-1}) \) if \( S_0 \subseteq S(k-1) \), and \( \hat{c}(k) - c = c + o_p(1) \) for some constant \( c > 0 \) if \( S_0 \not\subseteq S(k-1) \). Note that the penalty increment \( (v_T \log T)/T \) is larger than \( O_p(v_T T^{-1}) \) and smaller than \( c + o_p(1) \) for large \( T \). By successive application of this fact finitely many times, we can prove that \( S_0 \subseteq S(k) \) for each \( k = 0, \ldots, K \), and that \( |S_k| = |S_0| \) with probability close to one.

**DERIVATION OF EQUATION (12) IN SLANTS**

We need to compute

\[
Q(\beta | \hat{\beta}_T) = E_{\theta_T | (\hat{\beta}_T, Y_T)} \log p(Y_T, \theta_T | \beta_T) - \lambda_T \sum_{i=1}^D \|\beta_i\|_2
\]

up to a constant (which does not depend on \( \beta \)). The complete log-likelihood is

\[
\log p(Y_T, \theta_T | \beta) = C_0 - \|\theta_T - \beta\|_2^2 / 2 \tau^2_T
\]

where \( C_1 \) and \( C_2 \) are constants that do not involve \( \beta \). So it remains to calculate \( E_{\theta_T | (\hat{\beta}_T, Y_T)} \psi_T \). Note that \( Y_T, \theta_T \sim N(Z_T \theta_T, W_T^{-1} - \tau^2_T Z_T^T Z_T) \), \( \theta_T | \hat{\beta}_T \sim N(\hat{\beta}_T, \tau^2_T I) \). Thus, \( \theta_T | (\hat{\beta}_T, Y_T) \) is Gaussian with mean

\[
E_{\theta_T | (\hat{\beta}_T, Y_T)} \theta_T = \hat{\beta}_T
\]

It follows that

\[
Q(\beta | \hat{\beta}_T) = -1 / 2 \tau^2_T \| \beta - \hat{\beta}_T \|_2^2 - \lambda_T \sum_{i=1}^D \|\beta_i\|_2
\]

**ACKNOWLEDGEMENT**

The authors thank Dr. Lu Shen for suggesting the Boston weather data. The authors also thank associate editor Dr. Morten Morup and three anonymous reviewers for their reviewing the paper and providing insightful comments.

**REFERENCES**


