The Knowledge Gradient Policy Using A Sparse Additive Belief Model

Yan Li   YANLI@PRINCETON.EDU

Han Liu   HANLIU@PRINCETON.EDU

Warren B. Powell   POWELL@PRINCETON.EDU

Department of Operations Research and Financial Engineering

Princeton University

Princeton, NJ 08544, USA

Editor:

Abstract

We propose a sequential learning policy for noisy discrete global optimization and ranking and selection (R&S) problems with high dimensional sparse belief functions, where there are hundreds or even thousands of features, but only a small portion of these features contain explanatory power. We aim to identify the sparsity pattern and select the best alternative before the finite budget is exhausted. We derive a knowledge gradient policy for sparse linear models (KGSpLin) with group Lasso penalty. This policy is a unique and novel hybrid of Bayesian R&S with frequentist learning. Particularly, our method naturally combines B-spline basis expansion and generalizes to the nonparametric additive model (KGSpAM) and functional ANOVA model. Theoretically, we provide the estimation error bounds of the posterior mean estimate and the functional estimate. Controlled experiments show that the algorithm efficiently learns the correct set of nonzero parameters even when the model is imbedded with hundreds of dummy parameters. Also it outperforms the knowledge gradient for a linear model.

Keywords: sequential decision analysis, sparse additive model, ranking and selection, knowledge gradient, functional ANOVA model

1. Introduction

The ranking and selection (R&S) problem arises when we are trying to find the best of a set of competing alternatives through a process of sequentially testing different choices, which we have to evaluate using noisy measurements. In specific, we are maximizing an unknown function $\mu(x) : \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X} \subset \mathbb{R}^m$ is a finite set with M alternatives. We have the ability to sequentially choose a set of measurement to estimate. Our goal is to select the best alternative when the finite budget is exhausted. We assume that experiments are time consuming and expensive. This problem arises in applications such as simulation optimization, medical diagnostics and the design of business processes. In such applications, the number of underlying parameters might be quite large; for example, we might have to choose a series of parameters to design a new material which might involve temperature, pressure, concentration and choice of component materials such as catalysts.

The early R&S literature assumes a lookup table belief model (Frazier et al., 2008, 2009), but recent research has used a parametric belief model, making it possible to represent

many thousands or even millions of alternatives using a low-dimensional model. Let $\boldsymbol{\mu}$ be the vector representing values of all alternatives. Linear beliefs assume the truth $\boldsymbol{\mu}$ can be represented as a linear combination of a set of parameters, that is, $\boldsymbol{\mu} = \widetilde{\mathbf{X}}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is the underlying coefficient and $\widetilde{\mathbf{X}}$ represent the alternative matrix, that is, each row of $\widetilde{\mathbf{X}}$ is a vector representing an alternative.

The problem is that there are many applications that is high dimensional, that is, the coefficient $\boldsymbol{\alpha}$ can potentially have hundreds or even thousands of coponents. For example, in learning the accessibility profile of a large RNA molecule, the underlying weight coefficient describing the accessibility of each site is high dimensional due to the large size of RNA molecule. However, it is typically the case that only a small portion of these coefficients contain explanatory power (Reyes et al., 2014).

More generally, for these applications, we propose a sparse additive model which offers considerably more flexibility than a linear model, while recognizing that the final model will be relatively low dimensional. Sparse additive model assumes the truth takes the form

$$\mu_i = f_1(\widetilde{X}_{i1}) + f_2(\widetilde{X}_{i2}) + \cdots + f_p(\widetilde{X}_{ip}) + \varsigma_i, \quad \text{for } i = 1, \ldots, M,$$

where the f_js are one-dimensional smooth functions, ς_i is some Gaussian noise and M is the number of competing alternatives. In high dimensional settings, we assume that most of the f_js are zeros. If each f_j is a linear function, then the sparse additive belief reduces to linear belief. In this model, we are working on a model with a potentially large number of features, most of which do not contribute significant explanatory power. Our challenge is not only to design an efficient search algorithm for identifying the best alternative, but also identify the underlying sparsity structure.

In this paper, we study high dimensional optimal learning with sparse beliefs. We first derive a knowledge gradient policy for linear models (KGSpLin) with $\ell_{1,\infty}$ group Lasso penalty. More generally, we can assume the belief function takes an additive model, which is a summation of unknown smooth functions of each feature, where only a few components are nonzero. If we approximate each smooth function by B-splines basis, the sparse additive model can be fitted using group Lasso. Therefore, KGSpLin can be naturally generalized to the knowledge gradient policy for sparse additive models (KGSpAM). Here we introduce a random indicator variable and maintain a Beta-Bernoulli conjugate prior to model our belief about which variables should be included in or dropped from the model. Additionally, in the broader class of models known as multivariate splines functional ANOVA model, tensor product B-splines can be adopted. KGSpAM can also be used in this model.

The remainder of the paper is organized as follows. Section 3 formulates the ranking and selection model in a Bayesian setting and establishes the notation used in this paper. It also highlights the knowledge gradient using both lookup table and a low dimensional linear, parametric belief model and introduces the homotopy algorithm for recursive $\ell_{1,\infty}$ group Lasso. Section 4 is devoted to a detailed description of the KGSpLin policy for high dimensional linear models with $\ell_{1,\infty}$ group Lasso. Section 5 generalizes the algorithm to nonparametric sparse additive belief model (KGSpAM) and also SS-ANOVA. Theoretical results are presented in Section 6, which shows the estimation error bounds for both posterior mean and functional estimate. In Section 7, we test the algorithm in the context of a series of controlled experiments.

2. Literature

There has been a substantial literature on the general problem of finding the maximum of an unknown function where we rely on making noisy measurements to actively make searching decisions. Spall (2005) provides a thorough review of the literature that traces its roots to stochastic approximation methods. However, these methods require lots of measurements to find maxima precisely, which is unrealistic when measurements are very expensive.

Our problem originates from the R&S problem, which has been considered by many authors, under four distinct mathematical formulations. We specifically consider the Bayesian formulation, for which early work dates to Raiffa and Schlaifer (1968). The other mathematical formulations are the indif-ference-zone formulation (Bechhofer et al., 1995); the optimal computing budget allocation, or OCBA (Chen, 2010; Chen et al., 2012); and the large-deviations approach (Glynn and Juneja, 2004).

In the Bayesian formulation, this R&S problem has received considerable attention under the umbrella of optimal learning (Powell and Ryzhov, 2012). In this work, there are three major classes of function approximation methods: look-up tables, parametric models and nonparametric models. Gupta and Miescke (1996) introduces the idea of selecting an alternative based on the marginal value of information. Frazier et al. (2008) extends the idea under the name knowledge gradient using a Bayesian approach which estimates the value of measuring an alternative by the predictive distributions of the means, where it was shown that the policy is myopically optimal by construction and asymptotically optimal. The knowledge gradient using a lookup table belief model approximates the function in a discrete way, without any underlying explicit structural assumption, for both uncorrelated and correlated alternatives (Frazier et al., 2008, 2009). Another closely related idea can be found in Chick et al. (2001), where samples are allocated to maximize an approximation to the expected value of information. Negoescu et al. (2011) introduces the use of a parametric belief model, making it possible to solve problems with thousands of alternatives. For nonparametric beliefs, Mes et al. (2011) proposes a hierarchical aggregation technique using the common features shared by alternatives to learn about many alternatives from even a single measurement, while Barut and Powell (2013) estimates the belief function using kernel regression and aggregation of kernels. However, all the methods above assume low dimensional belief models, where the number of features is relatively small. There are applications with hundreds or even thousands of features, but where only a few features are relevant. In such settings, previous algorithms may require a lot of tedious computation on the overall features.

Additionally, outside of the Bayesian framework, there is another line of research on sparse online learning, in which an algorithm is faced with a collection of noisy options of unknown value, and has the opportunity to engage these options sequentially. In the online learning literature, an algorithm is measured according to the cumulative value of the options engaged, while in our problem we only need to select the best one at the end of experiments. Another difference is that, rather than value, researchers often consider the regret, which is the loss of our option compared with the optimal decision in hindsight. Cumulative value/regret is appropriate in dynamic settings such as maximizing the cumulative rewards (learning while doing), while terminal value/regret fits in settings such as finding the best route in a transportation network (learn then do). Moreover, most of the

algorithms in online learning are based on stochastic gradient/subgradient descent method. The key idea to induce sparsity is to introduce some regularizer in the gradient mapping (Duchi and Singer, 2009; Langford et al., 2009; Xiao, 2010; Lin et al., 2011; Chen et al., 2012; Ghadimi and Lan, 2012). However, a major problem with these methods is that while the intermediate solutions are sparse, the final solution may not be exactly sparse because it is usually obtained by taking the average of the intermediate solutions.

Additive models were first proposed by Friedman and Stuetzle (1981) as a class of non-parametric regression models and has received more attention over the decades (Hastie and Tibshirani, 1990). In high dimensional statistics, there has been much work on estimation, prediction and model selection for penalized methods on additive model (Zhang et al., 2004; Lin and Zhang, 2006; Ravikumar et al., 2009; Fan et al., 2011; Guedj and Alquier, 2013). Sparsity is a feature present in a plethora of natural as well as manmade systems. In optimal learning problems, it is also natural to consider sparsity structure not only because nature itself is parsimonious but also because simple models and processing with minimal degrees of freedom are attractive from an implementation perspective. Most of the previous work on sparse additive models study it in a batch setting, but here we study it in an active learning setting, where not only observations come in recursively, but also we get to actively choose which alternative to measure.

3. Notation and Preliminaries

In this section, we briefly review some results from Bayesian models for R&S and the recursive algorithm for $\ell_{1,\infty}$ group Lasso. We start with an introduction of notation: Let $\mathbf{M} = [M_{ij}] \in \mathbb{R}^{a \times d}$ and $\boldsymbol{v} = [v_1, \ldots, v_d]^T \in \mathbb{R}^d$. We denote $\boldsymbol{v}_I$ to be the subvector of $\boldsymbol{v}$ whose entries are indexed by a set I. We also denote $\mathbf{M}_{I,J}$ to be the submatrix of $\mathbf{M}$ whose rows are indexed by I and columns are indexed by J. For $I = J$, we simply denote it by $\mathbf{M}_I$ or $\mathbf{M}_J$. Let $\mathbf{M}_{I*}$ and $\mathbf{M}_{*J}$ be the submatrix of $\mathbf{M}$ with rows indexed by I, and the submatrix of $\mathbf{M}$ with columns indexed by J. Let $\mathrm{supp}(v) := \{j : v_j \neq 0\}$. For $0 < p < \infty$, we define the $\ell_0, \ell_p, \ell_\infty$ vector norms as

$$\|v\|_0 := \mathrm{card}(\mathrm{supp}(v)), \|v\|_p := (\sum_{i=1}^{d} |v_i|^p)^{1/p}, \text{and} \|v\|_\infty := \max_{1 \leq i \leq d} |v_i|.$$

For a matrix $\mathbf{M}$, we define the Frobenius norm as: $\|\mathbf{M}\|_F := (\sum_{i=1}^{a} \sum_{j=1}^{d} |M_{ij}|^2)^{1/2}$ and the ℓ_p norm to be: $\|\mathbf{M}\|_p = \max_{\|\boldsymbol{v}\|_p=1} \|\mathbf{M}\boldsymbol{v}\|_p$. For any square matrix $\mathbf{M}$, let $\Lambda_{\max}(\mathbf{M})$ and $\Lambda_{\min}(\mathbf{M})$ be the largest and smallest eigenvalue of $\mathbf{M}$. For a summary of most symbols we use, please refer to Table 2 in Appendix A.

3.1 The Bayesian Model for Ranking and Selection

We denote the unknown function $\mu(x) : \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X} \subset \mathbb{R}^m$ is a finite set with M alternatives. In addition, if we have a measurement budget of N, our goal is to sequentially decide which alternatives to measure so that when we exhaust our budget, we have maximized our ability to find the best alternative using our estimated belief model. Here we use $\boldsymbol{x}$ to denote the vector and x to denote the corresponding alternative index, that is, $x \in \{1, \ldots, M\}$. We also use μ_x for $\mu(x)$. Let $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_M]^T$. Under this setting, the

number of alternatives M can be extremely large relative to the measurement budget N. In a Bayesian setting, we assume $\boldsymbol{\mu}$ takes multinormal distribution

$$\boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma}). \tag{1}$$

Now suppose we have a sequence of measurement decisions, $\boldsymbol{x}^0, \boldsymbol{x}^1, \ldots, \boldsymbol{x}^{N-1}$ to learn about these alternatives. Here $\boldsymbol{x}^i \in \mathcal{X}$, for $i = 0, \ldots, N-1$. At time n, if we measure alternative x, we observe

$$y_x^{n+1} = \mu_x + \epsilon_x^{n+1},$$

where $\epsilon_x^{n+1} \sim \mathcal{N}(0, \sigma_x^2)$ and σ_x is known.

Initially, assume we have a multivariate normal prior distribution on $\boldsymbol{\mu}$,

$$\boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\theta}^0, \boldsymbol{\Sigma}^0).$$

Additionally, because decisions are made sequentially, $\boldsymbol{x}^n$ is only allowed to depend on the outcomes of the sampling decisions $\boldsymbol{x}^0, \boldsymbol{x}^1, \ldots, \boldsymbol{x}^{n-1}$. In the remainder of the paper, a random variable indexed by n means it is measurable with respect to $\mathcal{F}^n$, which is defined as the σ-algebra generated by $\{(\boldsymbol{x}^0, y_{x^0}^1), (\boldsymbol{x}^1, y_{x^1}^2), \ldots, (\boldsymbol{x}^{n-1}, y_{x^{n-1}}^n)\}$. Following this definition, we denote $\boldsymbol{\theta}^n := \mathbb{E}[\boldsymbol{\mu}|\mathcal{F}^n]$, and $\boldsymbol{\Sigma}^n := \mathrm{Var}[\boldsymbol{\mu}|\mathcal{F}^n]$. It means conditionally on $\mathcal{F}^n$, our posterior belief distribution on $\boldsymbol{\mu}$ is multivariate normal with mean $\boldsymbol{\theta}^n$ and covariance matrix $\boldsymbol{\Sigma}^n$. When the measurement budget of N is exhausted, our goal is to find the optimal alternative, so the final decision is

$$x^N = \operatorname*{argmax}_{\boldsymbol{x} \in \mathcal{X}} \theta_x^N.$$

We define Π to be the set of all possible policies satisfying our sequential requirement; that is, $\Pi := \{[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{N-1}] : \boldsymbol{x}^n \in \mathcal{F}^n\}$. Let $\mathbb{E}^\pi$ indicate the expectation with respect to the prior over both the noisy outcomes and the truth $\boldsymbol{\mu}$ while the sampling policy is fixed to $\pi \in \Pi$. After exhausting the budget of N measurements, we select the alternative with the highest posterior mean. Our goal is to choose a measurement policy maximizing expected reward, which can be written as

$$\sup_{\pi \in \Pi} \mathbb{E}^\pi \left[\max_{\boldsymbol{x} \in \mathcal{X}} \theta_x^N \right].$$

We work in the Bayesian setting to sequentially update the estimates of the alternatives. At time n, suppose we select $\boldsymbol{x}^n = \boldsymbol{x}$ and observe $y_{x^{n+1}}$; we can compute the $n + 1$ time posterior distribution with the following Bayesian updating equations (Gelman et al., 2003):

$$
\begin{aligned}
\boldsymbol{\theta}^{n+1} &= \boldsymbol{\theta}^n + \frac{y_x^{n+1} - \theta_x^n}{\sigma_x^2 + \Sigma_{xx}^n} \boldsymbol{\Sigma}^n \boldsymbol{e}_x, \\
\boldsymbol{\Sigma}^{n+1} &= \boldsymbol{\Sigma}^n - \frac{\boldsymbol{\Sigma}^n \boldsymbol{e}_x \boldsymbol{e}_x^T \boldsymbol{\Sigma}^n}{\sigma_x^2 + \Sigma_{xx}^n},
\end{aligned} \tag{2}
$$

where $\boldsymbol{e}_x$ is the standard basis vector with one indexed by x and zeros elsewhere.

3.2 Knowledge Gradient for Linear Belief

In this section, we briefly review the knowledge gradient for correlated normal beliefs (KGCB), which is a fully sequential sampling policy for learning correlated alternatives (Frazier et al., 2008). Here correlated alternatives mean that the performances of different alternatives may have correlations as described in (1). We also review the knowledge gradient for a linear belief model (KGLin). It means that the belief model is linear in terms of a set of known basis functions. In this case, Bayesian updating is performed using recursive least squares (Frazier et al., 2009). We represent the state of knowledge at time n as: $S^n := (\boldsymbol{\theta}^n, \boldsymbol{\Sigma}^n)$. The corresponding value of being in state S^n at time n is

$$V^n(S^n) = \max_{\boldsymbol{x}' \in \mathcal{X}} \theta_{x'}^n.$$

The knowledge gradient policy is to choose the alternative that can maximize the expected incremental value,

$$
\begin{aligned}
v_x^{KG,n} &= \mathbb{E}(V^{n+1}(S^{n+1}(x)) - V^n(S^n)|S^n, \boldsymbol{x}^n = \boldsymbol{x}) \\
&= \mathbb{E}(\max_{\boldsymbol{x}' \in \mathcal{X}} \theta_{x'}^{n+1}|S^n, \boldsymbol{x}^n = \boldsymbol{x}) - \max_{\boldsymbol{x}' \in \mathcal{X}} \theta_{x'}^n
\end{aligned}
$$

and

$$\boldsymbol{x}^{KG,n} = \operatorname*{argmax}_{\boldsymbol{x} \in \mathcal{X}} v_x^{KG,n}.$$

Frazier et al. (2009) proposes an algorithm to compute the KG values for alternatives with correlated beliefs. First we can further rearrange equation (2) as the time n conditional distribution of $\boldsymbol{\theta}^{n+1}$, namely,

$$\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n + \widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, \boldsymbol{x}^n)Z^{n+1}, \tag{3}$$

where

$$
\begin{aligned}
\widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, \boldsymbol{x}) &= \frac{\boldsymbol{\Sigma}^n \boldsymbol{e}_x}{\sqrt{\sigma_x^2 + \Sigma_{xx}^n}}, \tag{4} \\
Z^{n+1} &= \frac{(y_x^{n+1} - \theta_x^n)}{\sqrt{\operatorname{Var}[y_x^{n+1} - \theta_x^n | \mathcal{F}^n]}}.
\end{aligned}
$$

It is easy to see that Z^{n+1} is standard normal when conditioned on $\mathcal{F}^n$ (Frazier et al., 2008). Then we substitute equation (3) into the KG formula,

$$
\begin{aligned}
v_x^{KG,n} &= \mathbb{E}(\max_{\boldsymbol{x}' \in \mathcal{X}} \theta_{x'}^n + \widetilde{\boldsymbol{\sigma}}_{x'}(\boldsymbol{\Sigma}^n, \boldsymbol{x}^n)Z^{n+1}|S^n, \boldsymbol{x}^n = \boldsymbol{x}) - \max_{\boldsymbol{x}' \in \mathcal{X}} \theta_{x'}^n \\
&= h(\boldsymbol{\theta}^n, \widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, \boldsymbol{x})),
\end{aligned}
$$

where $\widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, \boldsymbol{x})$ is a vector-valued function defined in (4) and $\widetilde{\sigma}_{x'}(\boldsymbol{\Sigma}^n, \boldsymbol{x}^n)$ indicates the component $\boldsymbol{e}_{x'}^T \widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, \boldsymbol{x}^n)$ of the vector $\widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, \boldsymbol{x}^n)$ and $h(\boldsymbol{a}, \boldsymbol{b}) = \mathbb{E}[\max_i a_i + b_i Z] - \max_i a_i$ is a generic function of any vectors of the same dimension, Z is a standard normal random variable.

The expectation can be computed as the point-wise maximum of affine functions $a_i + b_i Z$ with an algorithm of complexity $O(M^2 log(M))$. It works as follows. First the algorithm sorts the sequence of pairs (a_i, b_i) such that the b_is in nondecreasing order and ties in b are broken by removing the pair (a_i, b_i) when $b_i = b_{i+1}$ and $a_i \le a_{i+1}$. Next, all pairs (a_i, b_i) that are dominated by the other pairs, that is, $a_i + b_i Z \le \max_{j \ne i} a_j + b_j Z$ for all values of Z, are removed. Thus the knowledge gradient can be computed using

$$v_x^{KG} = h(\boldsymbol{a}, \boldsymbol{b}) = \sum_{i=1,\ldots,\widetilde{M}} (\widetilde{b}_{i+1} - \widetilde{b}_i) f\left(-\left|\frac{\widetilde{a}_i - \widetilde{a}_{i+1}}{\widetilde{b}_{i+1} - \widetilde{b}_i}\right|\right),$$

where $f(z) = \phi(z) + z\Phi(z)$. Here $\phi(z)$ and $\Phi(z)$ are the normal density and cumulative distribution functions respectively. $\widetilde{\boldsymbol{a}}$ and $\widetilde{\boldsymbol{b}}$ are the new vectors after sorting a and b and dropping off the redundant components and are of dimension $\widetilde{M}$.

If the number of alternatives is quite large, the above representation becomes clumsy. Thus if the underlying belief model has some structure, then we could take advantage of this structure to represent the model and simplify the computation. In a simple case, if f has a linear form or can be written as a basis expansion, we can make it easier by maintaining a belief on the coefficients instead of the alternatives.

Negoescu et al. (2011) further extends KGCB to parametric beliefs using a linear model. Now we assume the truth $\boldsymbol{\mu}$ can be represented as a linear combination of a set of parameters, that is, $\boldsymbol{\mu} = \widetilde{\mathbf{X}}\boldsymbol{\alpha}$, where $\boldsymbol{\mu} \in \mathbb{R}^M$ and $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m]^T \in \mathbb{R}^m$ are random variables, $\widetilde{\mathbf{X}} \in \mathbb{R}^{M \times m}$ represent the alternative matrix, that is, each row of $\widetilde{\mathbf{X}}$ is a vector representing an alternative. If we assume $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\vartheta}, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}})$, this induces a normal distribution on $\boldsymbol{\mu}$ via linear transformation,

$$\boldsymbol{\mu} \sim \mathcal{N}(\widetilde{\mathbf{X}}\boldsymbol{\vartheta}, \widetilde{\mathbf{X}}\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}}\widetilde{\mathbf{X}}^T).$$

At time n, if we measure alternative $\boldsymbol{x}^n = \boldsymbol{x}$, we can update $\boldsymbol{\vartheta}^{n+1}$ and $\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n+1}$ recursively via Recursive Least Squares (Powell and Ryzhov, 2012),

$$\boldsymbol{\vartheta}^{n+1} = \boldsymbol{\vartheta}^n + \frac{\widehat{\epsilon}^{n+1}}{\gamma^n}\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}\boldsymbol{x}^n,$$

$$\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n+1} = \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n} - \frac{1}{\gamma^n}(\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}\boldsymbol{x}^n(\boldsymbol{x}^n)^T\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}),$$

where $\widehat{\epsilon}^{n+1} = y^{n+1} - (\boldsymbol{\vartheta}^n)^T\boldsymbol{x}^n$ and $\gamma^n = \sigma_x^2 + (\boldsymbol{x}^n)^T\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}\boldsymbol{x}^n$.

The linear model allows us to represent the alternatives in a compact format since the dimension of the parameters is usually much smaller than the number of the alternatives. Suppose we have tens of thousands of alternatives, without the linear model, we would need to create and update the covariance matrix $\boldsymbol{\Sigma}^n$ with tens of thousands of rows and columns. With the linear model, we only need to maintain the parameter covariance matrix $\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}$, the size of which is equal to the dimension of the parameter vector $\boldsymbol{\vartheta}$. In addition, we never need to compute the full matrix $\widetilde{\mathbf{X}}\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}}\widetilde{\mathbf{X}}^T$. We only have to compute a row of this matrix.

3.3 A Homotopy Algorithm for Recursive $\ell_{1,\infty}$ Group Lasso

In the Bayesian updating scheme described in Section 4 and 5, a recursive $\ell_{1,\infty}$ group Lasso is required, which we review in this section. When the regularization takes the ℓ_1

norm, this regularized version with least squares loss is Lasso (least absolute shrinkage and selection operator)(Tibshirani, 1996). It is well known that Lasso leads to solutions that are sparse and therefore achieves model selection. If we consider a more general group sparsity system, which is composed of a few nonoverlapping clusters of nonzero coefficients, $\ell_{1,\infty}$ group Lasso penalty can be used to encourage correlations within groups and achieve sparsity at a group level. Here we briefly describe the recursive homotopy algorithm for $\ell_{1,\infty}$ group Lasso proposed in Chen and Hero (2012). For the recursive homotopy algorithm for Lasso, one can refer to Garrigues and El Ghaoui (2008). This algorithm computes an exact update of the optimal $\ell_{1,\infty}$ penalized recursive least squares predictor. Each update minimizes a convex but nondifferentiable function optimization problem. This algorithm has been demonstrated to have lower implementation complexity than direct group Lasso solvers. It also fits the recursive setting in optimal learning.

The $\ell_{1,\infty}$ group Lasso estimator after n observations is given by

$$\widehat{\boldsymbol{\beta}}^n = \operatorname*{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^m} \frac{1}{2}\sum_{i=1}^{n}\left[(\boldsymbol{x}^{i-1})^T\boldsymbol{\beta} - y^i\right]^2 + \lambda^n\|\boldsymbol{\beta}\|_{1,\infty}, \tag{5}$$

where $(y^i, \boldsymbol{x}^{i-1}) \in \mathbb{R}\times\mathbb{R}^m, i = 1,\ldots,n$ are the n observations. λ^n is the regularization parameter, and $\|\boldsymbol{\beta}\|_{1,\infty} := \sum_{j=1}^{p}\|\boldsymbol{\beta}_{\mathcal{G}_j}\|_{\infty}$. $\{\mathcal{G}_j\}_{j=1}^{p}$ is the group partition of the index set $\mathcal{G} = \{1,\ldots,m\}$, that is,

$$\cup_{j=1}^{p}\mathcal{G}_j = \mathcal{G}, \quad \mathcal{G}_j\cap\mathcal{G}_j' = \emptyset \quad\text{if}\quad j\neq j',$$

and $\boldsymbol{\beta}_{\mathcal{G}_j}$ is a subvector of $\boldsymbol{\beta}$ indexed by $\mathcal{G}_j$. Let $d_j = |\mathcal{G}_j|$ be the number of features in the jth group, and $m = \sum_{j=1}^{p}d_j$. Group Lasso reduces to Lasso when each group contains only one coefficient.

At time n, suppose we have $\widehat{\boldsymbol{\beta}}^n$ to the Lasso with n observation and we are given the next observation $(y^{n+1}, \boldsymbol{x}^n) \in \mathbb{R}\times\mathbb{R}^m$. The algorithm computes the next estimate $\widehat{\boldsymbol{\beta}}^{n+1}$ via the following optimization problem. Let $\mathbf{R}^{n-1} = \sum_{i=1}^{n}\boldsymbol{x}^{i-1}(\boldsymbol{x}^{i-1})^T$, $\boldsymbol{r}^n = \sum_{i=1}^{n}\boldsymbol{x}^{i-1}y^i$. Let us define a function

$$u(t,\lambda) = \operatorname*{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^m}\frac{1}{2}\boldsymbol{\beta}^T(\mathbf{R}^{n-1} + t\boldsymbol{x}^n(\boldsymbol{x}^n)^T)\boldsymbol{\beta} - \boldsymbol{\beta}^T(\boldsymbol{r}^n + t\boldsymbol{x}^n y^{n+1}) + \lambda\|\boldsymbol{\beta}\|_{1,\infty}.$$

We have $\widehat{\boldsymbol{\beta}}^n = u(0,\lambda^n)$ and $\widehat{\boldsymbol{\beta}}^{n+1} = u(1,\lambda^{n+1})$. The homotopy algorithm that computes a path from $\widehat{\boldsymbol{\beta}}^n$ to $\widehat{\boldsymbol{\beta}}^{n+1}$ in two steps:

1 Fix $t = 0$, vary the regularization parameter from λ^n to λ^{n+1} with $t = 0$. This amounts to computing the regularization path between λ^n and λ^{n+1} using homotopy methods as the iCap algorithm done in Zhao et al. (2009). This solution path is piecewise linear.

2 Fix λ and calculate the solution path between $u(0,\lambda^{n+1})$ and $u(1,\lambda^{n+1})$ using the homotopy approach. This is derived by proving that the solution path is piecewise smooth in t. The algorithm computes the next "transition point" at which active groups and solution signs change, and updates the solution until t reaches 1.

4. Knowledge Gradient for Linear Model with $\ell_{1,\infty}$ Group Lasso

In Section 3.2, we review knowledge gradient policy for linear belief models in low dimensional settings. In this section, we derive the KG policy in a high dimensional linear model. We have $\boldsymbol{\mu} = \widetilde{\mathbf{X}}\boldsymbol{\alpha}$, where $\widetilde{\mathbf{X}} \in \mathbb{R}^{M \times m}$ is the alternative matrix and $\boldsymbol{\alpha} \in \mathbb{R}^m, \boldsymbol{\mu} \in \mathbb{R}^M$ are random variables. Here m can become relatively large and $\boldsymbol{\alpha}$ is sparse in the sense that only a few components of $\boldsymbol{\alpha}$ contribute to $\boldsymbol{\mu}$. However, unlike the sparsity assumption in classical frequentist statistics, we assume the sparsity structure is random; that is, the group indicator variable of which is selected or not is a random vector. Specifically, we now assume there exists some known group structure in $\boldsymbol{\alpha}$, let $\boldsymbol{\zeta} = [\zeta_1, \ldots, \zeta_p] \in \mathbb{R}^p$ be a group indicator random variable of $\boldsymbol{\alpha}$,

$$\zeta_j = \begin{cases} 1 & \text{if } \boldsymbol{\alpha}_{\mathcal{G}_j} \neq \mathbf{0} \\ 0 & \text{if } \boldsymbol{\alpha}_{\mathcal{G}_j} = \mathbf{0} \end{cases}, \quad \text{for } j = 1, \ldots, p.$$

Additionally, $\boldsymbol{\alpha}$ is assumed to be sparse in the following sense,

$$\boldsymbol{\alpha}|\boldsymbol{\zeta} \sim \mathcal{N}(\boldsymbol{\vartheta}, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}}). \tag{6}$$

Let $\mathcal{S} = \{j : \zeta_j = 1\}$. Thus, without loss of generality, conditioning on $\boldsymbol{\zeta}$, we can permute the elements of $\boldsymbol{\alpha}$ to create the following partition,

$$\boldsymbol{\alpha}^T = [(\boldsymbol{\alpha}_{\mathcal{S}})^T, \mathbf{0}],$$

where $\boldsymbol{\alpha}_{\mathcal{S}} \sim \mathcal{N}(\boldsymbol{\vartheta}_{\mathcal{S}}, \boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta}})$. So $\boldsymbol{\vartheta}$ and $\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}}$ can be correspondingly partitioned

$$\boldsymbol{\vartheta} = \begin{bmatrix} \boldsymbol{\vartheta}_{\mathcal{S}} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Here we make a critical assumption on the distribution of $\boldsymbol{\alpha}$. Let us assume that conditioning on $\boldsymbol{\zeta} = \mathbf{1}$, $\boldsymbol{\alpha}$ has the following distribution: $\boldsymbol{\alpha}|\boldsymbol{\zeta} = \mathbf{1} \sim \mathcal{N}(\boldsymbol{\vartheta}, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}})$. Then for any other $\boldsymbol{\zeta}'$, the conditional distribution of $\boldsymbol{\alpha}$ on $\boldsymbol{\zeta}'$ is normal with mean $\boldsymbol{\theta}_{\mathcal{S}'}$ and variance $\boldsymbol{\Sigma}_{\mathcal{S}'}^{\boldsymbol{\theta}}$. Here $\mathcal{S}' = \{j : \zeta'_j = 1\}$. This means that we can write all the conditional distributions of $\boldsymbol{\alpha}$ through an index set $\mathcal{S}$ characterized by $\boldsymbol{\zeta}$. So in the following we use both $\boldsymbol{\zeta}$ and $\mathcal{S}$ as indices. We also omit the time dependent variable n to simplify notations. Furthermore, as we are updating the mean and covariance matrix of a certain conditional distribution, we also update all the elements with the same index in the other distributions. That means, through all the updatings, we just need to maintain the mean and covariance matrix on $\boldsymbol{\zeta} = \mathbf{1}$.

4.1 Knowledge Gradient Policy for Sparse Linear Model

Before deriving the sparse knowledge gradient algorithm, let us describe the Bayesian model at time n. To get a Bayesian update, we can maintain Beta-Bernoulli conjugate priors on each component of $\boldsymbol{\zeta}$. At time n, we have the following Bayesian model, for $j, j' = 1, \ldots, p$,

$$\boldsymbol{\alpha}|\boldsymbol{\zeta}^n = \mathbf{1} \sim \mathcal{N}(\boldsymbol{\vartheta}^n, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}, n}), \tag{7}$$

$$\zeta_j^n|p_j^n \sim \text{Bernoulli}(p_j^n), \tag{8}$$

$$\zeta_j^n \perp \zeta_{j'}^n, \quad \text{for} \quad j \neq j', \tag{9}$$

$$p_j^n|\xi_j^n, \eta_j^n \sim \text{Beta}(\xi_j^n, \eta_j^n). \tag{10}$$

9

At time n, the prior $\boldsymbol{\zeta}^n$ is a discrete random variable. Let $\boldsymbol{\zeta}^{n,1}, \ldots, \boldsymbol{\zeta}^{n,N_\zeta}$ be all the possible realizations of $\boldsymbol{\zeta}^n$, and $\mathbb{P}(\boldsymbol{\zeta}^n = \boldsymbol{\zeta}^{n,k}) = p^{n,k}, k = 1, \ldots, N_\zeta$. So by the Law of Total Expectation, the KG value can be computed by

$$
\begin{aligned}
v_x^{KG,n} &= \mathbb{E}(V^{n+1}(S^{n+1}(x)) - V^n(S^n)|S^n, \boldsymbol{x}^n = \boldsymbol{x}) \\
&= \mathbb{E}(\max_{\boldsymbol{x}' \in \mathcal{X}} \theta_{x'}^{n+1}|S^n, \boldsymbol{x}^n = \boldsymbol{x}) - \max_{\boldsymbol{x}' \in \mathcal{X}} \theta_{x'}^n \\
&= \mathbb{E}_{p^n} \mathbb{E}_{\boldsymbol{\zeta}^n|p^n} \mathbb{E}_{\boldsymbol{\alpha},\boldsymbol{\epsilon}|\boldsymbol{\zeta}^n,p^n} (\max_{\boldsymbol{x}' \in \mathcal{X}} \theta_{x'}^{n+1}|S^n, \boldsymbol{x}^n = \boldsymbol{x}, \boldsymbol{\zeta}^n, p^n) - \max_{\boldsymbol{x}' \in \mathcal{X}} \theta_{x'}^n \\
&= \sum_{k=1}^{N_\zeta} \mathbb{E}_{p^n}(p^{n,k}) h(\boldsymbol{a}^{n,k}, \boldsymbol{b}^{n,k}) \\
&= \sum_{k=1}^{N_\zeta} \prod_{\{j:\zeta_j^{n,k}=1\}} \frac{\xi_j^n}{\xi_j^n + \eta_j^n} \prod_{\{j:\zeta_j^{n,k}=0\}} \frac{\eta_j^n}{\xi_j^n + \eta_j^n} h(\boldsymbol{a}^{n,k}, \boldsymbol{b}^{n,k}),
\end{aligned}
$$

where

$$
\begin{aligned}
h(\boldsymbol{a}, \boldsymbol{b}) &:= \mathbb{E}[\max_i a_i + b_i Z] - \max_i a_i, \\
\boldsymbol{a}^{n,k} &= \widetilde{\mathbf{X}}_{\boldsymbol{\zeta}^{n,k}}^n \boldsymbol{\vartheta}_{\boldsymbol{\zeta}^{n,k}}^n, \\
\boldsymbol{b}^{n,k} &= = \widetilde{\boldsymbol{\sigma}}(\widetilde{\mathbf{X}}_{\boldsymbol{\zeta}^{n,k}}^n \boldsymbol{\Sigma}_{\boldsymbol{\zeta}^{n,k}}^{n,\boldsymbol{\vartheta}} (\widetilde{\mathbf{X}}_{\boldsymbol{\zeta}^{n,k}}^n)^T, \boldsymbol{x}).
\end{aligned}
$$

Note that conditioning on each sample realization of $\boldsymbol{\zeta}^n$, the KG calculation is identical with KGLin. The KG value for a sparse linear model is a weighted summation over all the possible sample realization of $\boldsymbol{\zeta}^n$, of which the weight $\mathbb{E}_{p^n}(p^{n,k})$ is computed by the independent Beta distributions on all the p_j^n's. Also, if N_ζ takes its largest possible value, that is $N_\zeta = 2^p$, we can re-sort the weights and approximate the knowledge gradient value by only computing ones with several top largest probabilities.

4.2 Bayesian Update

At time n we have the Bayesian model described in (7)-(10). Parallel with that, we also have the current Lasso estimate, denoted as $\widehat{\boldsymbol{\vartheta}}^n$. The nonzero part is $\widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^n$. The covariance matrix corresponding to $\widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^n$ is denoted as $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n}$, which is Monte Carlo simulated from the first order optimality condition of the optimization problem (5) (details described in Section 4.3). After we get the new observation, we can update to the next Lasso estimate recursively by the algorithm described in Section 3.3. Thus we have the updated Lasso estimate $\widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^{n+1}$ and $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1}$. Let $\mathcal{P}^n := \{j : \widehat{\boldsymbol{\vartheta}}_{\mathcal{G}_j}^n \neq 0\}$. The Bayesian updating equations are given by Gelman et al. (2003):

$$
\boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1} = \left[(\boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},n})^{-1} + (\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1})^{-1} \right]^{-1}, \tag{11}
$$

$$
\boldsymbol{\vartheta}_{\mathcal{S}}^{n+1} = \boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1} \left[(\boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},n})^{-1} \boldsymbol{\vartheta}_{\mathcal{S}}^n + (\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1})^{-1} \widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^{n+1} \right], \tag{12}
$$

$$
\xi_j^{n+1} = \xi_j^n + 1, \eta_j^{n+1} = \eta_j^n, \quad \text{for} \quad j \in \mathcal{P}^{n+1},
$$

$$
\xi_j^{n+1} = \xi_j^n, \eta_j^{n+1} = \eta_j^n + 1, \quad \text{for} \quad j \notin \mathcal{P}^{n+1}.
$$

Now we briefly recall and summarize the random variables which play a role in the measurement process. The underlying and unknown value of alternative x is denoted μ_x and parametrized by $\boldsymbol{\alpha}$. Here $\boldsymbol{\alpha}$ follows a "mixture" normal distribution by (6) and ζ_j follows a conditional Bernoulli distribution with the frequency of "in" and "out" denoted by ξ_j and η_j. Both $\boldsymbol{\alpha}$ and $\boldsymbol{\zeta}$ are randomly fixed at the beginning of the measurement process. At time n, $\boldsymbol{\zeta}^n$ and $\boldsymbol{\vartheta}^n$ give us the best estimate of $\boldsymbol{\alpha}$. $(\boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},n})^{-1}$ is the precision with which we make this estimate. The result of our time n measurement causes us to first update the Lasso solution from $\widehat{\boldsymbol{\vartheta}}^n$ to $\widehat{\boldsymbol{\vartheta}}^{n+1}$ and then update the mean estimate from $\boldsymbol{\vartheta}_{\mathcal{S}}^n$ to $\boldsymbol{\vartheta}_{\mathcal{S}}^{n+1}$, which we now know with precision $(\boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1})^{-1}$.

One may think of $\boldsymbol{\zeta}$ and $\boldsymbol{\alpha}$ as fixed and of $\boldsymbol{\zeta}^n$ as converging toward $\boldsymbol{\zeta}$ and $\boldsymbol{\vartheta}_{\mathcal{S}}^n$ as converging toward $\boldsymbol{\alpha}$ while some norm of the precision matrix $(\boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},n})^{-1}$ converging to infinity under some appropriate sampling strategy. It is also appropriate, however, to fix $\boldsymbol{\zeta}^n$ and $\boldsymbol{\vartheta}_{\mathcal{S}}^n$ and think of $\boldsymbol{\zeta}$ and $\boldsymbol{\alpha}$ as unknown quantities. Furthermore, from this perspective, the randomness of $\boldsymbol{\zeta}$ and $\boldsymbol{\alpha}$ does not imply they must be chosen from Bernoulli and mixture normal distribution respectively, but instead it only quantifies our uncertain knowledge of $\boldsymbol{\zeta}$ and $\boldsymbol{\alpha}$ adopted when they were first chosen.

4.3 Knowledge Gradient with Recursive $\ell_{1,\infty}$ Group Lasso

In this section, we first provide a technique to approximately sample the covariance matrix $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1}$ from the first order optimality condition in problem (5). Then we outline the knowledge gradient policy for sparse linear models in Algorithm 1,

We begin with a series of set definitions. Figure 1 provides an illustrative example. Let us divide the entire group index into $\mathcal{P}$ and $\mathcal{Q}$ respectively, where $\mathcal{P}$ contains active groups and $\mathcal{Q}$ is the complement. For each active group $j \in \mathcal{P}$, we partition the group into two parts: $\mathcal{A}_j$ with maximum absolute values and $\mathcal{B}_j$ with the rest of the values. That is

$$\mathcal{A}_j = \operatorname*{argmax}_{k \in \mathcal{G}_j} |\beta_k|, \quad \mathcal{B}_j = \mathcal{G}_j - \mathcal{A}_j, \quad j \in \mathcal{P}.$$
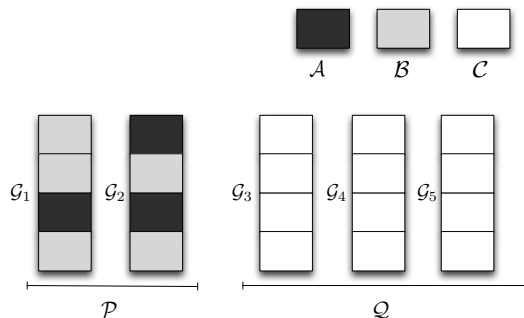

Figure 1: Illustration of the partitioning of a 20 element coefficient vector $\boldsymbol{\beta}$ into five groups of four indices. The sets $\mathcal{P}$ and $\mathcal{Q}$ contains the active groups and the inactive groups, respectively. Within each of the two active groups the coefficients with maximal absolute values are denoted by the black color.

The set $\mathcal{A}$ and $\mathcal{B}$ are defined as the union of the $\mathcal{A}_j$ and $\mathcal{B}_j$ sets, respectively,

$$\mathcal{A} = \cup_{j \in \mathcal{P}} \mathcal{A}_j, \quad \mathcal{B} = \cup_{j \in \mathcal{P}} \mathcal{B}_j.$$

Finally, we define

$$\mathcal{C} = \cup_{j \in \mathcal{Q}} \mathcal{G}_j, \quad \mathcal{C}_j = \mathcal{G}_j \cap \mathcal{C}.$$

The $\ell_{1,\infty}$ group Lasso problem (5) can also be written as

$$\boldsymbol{\beta}^n = \operatorname*{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^m} \frac{1}{2} \boldsymbol{\beta}^T \mathbf{R}^{n-1} \boldsymbol{\beta} - \boldsymbol{\beta}^T \boldsymbol{r}^n + \lambda^n \|\boldsymbol{\beta}\|_{1,\infty}. \tag{13}$$

This optimization problem is convex and nonsmooth since the $\ell_{1,\infty}$ norm is nondifferentiable. Here there is a global minimum at $\boldsymbol{\beta}$ if and only if the subdifferential of the objective function at $\boldsymbol{\beta}$ contains the $\mathbf{0}$-vector. The optimality conditions for (13) are given by

$$\mathbf{R}^{n-1}\boldsymbol{\beta} - \boldsymbol{r}^n + \lambda^n \boldsymbol{z} = \mathbf{0}, \quad \boldsymbol{z} \in \partial\|\boldsymbol{\beta}\|_{1,\infty}. \tag{14}$$

We also have that $\boldsymbol{z} \in \partial\|\boldsymbol{\beta}\|_{1,\infty}$ if and only if $\boldsymbol{z}$ satisfies the following conditions,

$$
\begin{aligned}
\|\boldsymbol{z}_{\mathcal{A}_j}\|_1 &= 1, \quad j \in \mathcal{P}, \tag{15}\\
\operatorname{sgn}(\boldsymbol{z}_{\mathcal{A}_j}) &= \operatorname{sgn}(\boldsymbol{\beta}_{\mathcal{A}_j}), \quad j \in \mathcal{P}, \tag{16}\\
\boldsymbol{z}_{\mathcal{B}} &= \mathbf{0}, \tag{17}\\
\|\boldsymbol{z}_{\mathcal{C}_j}\|_1 &\leq 1, \quad j \in \mathcal{Q},
\end{aligned}
$$

where $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{P}$ and $\mathcal{Q}$ are $\boldsymbol{\beta}$-dependent sets defined above. For notational convenience we leave out the time variable n in the set notation. As $\boldsymbol{\beta}_{\mathcal{C}} = \mathbf{0}$, (14) implies that

$$
\begin{aligned}
\mathbf{R}_{\mathcal{S}}^{n-1}\boldsymbol{\beta}_{\mathcal{S}} - \boldsymbol{r}_{\mathcal{S}}^n + \lambda^n \boldsymbol{z}_{\mathcal{S}} &= \mathbf{0}, \tag{18}\\
\mathbf{R}_{\mathcal{CS}}^{n-1}\boldsymbol{\beta}_{\mathcal{S}} - \boldsymbol{r}_{\mathcal{C}}^n + \lambda^n \boldsymbol{z}_{\mathcal{C}} &= \mathbf{0}.
\end{aligned}
$$

If $\mathbf{R}_{\mathcal{S}}^{n-1}$ is invertible, then the solution is unique and we can rewrite (18) as

$$\boldsymbol{\beta}_{\mathcal{S}} = (\mathbf{R}_{\mathcal{S}}^{n-1})^{-1}(\boldsymbol{r}_{\mathcal{S}}^n - \lambda^n \boldsymbol{z}_{\mathcal{S}}). \tag{19}$$

Let $\mathbf{X}^{n-1} \in \mathbb{R}^{n \times m}$ be the design matrix at time n defined as

$$(\mathbf{X}^{n-1})^T := [\boldsymbol{x}^0, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^{n-1}],$$

and

$$\mathbf{Y}^n := [y^1, \ldots, y^n]^T.$$

Then (19) is equivalent to

$$\boldsymbol{\beta}_{\mathcal{S}} = \left[(\mathbf{X}_{*\mathcal{S}}^{n-1})^T \mathbf{X}_{*\mathcal{S}}^{n-1} \right]^{-1} \left[(\mathbf{X}_{*\mathcal{S}}^{n-1})^T \mathbf{Y}^n - \lambda^n \boldsymbol{z}_{\mathcal{S}} \right]. \tag{20}$$

Let $\mathbf{M}_{\mathcal{S}}^{n-1} = \left[(\mathbf{X}_{*\mathcal{S}}^{n-1})^T \mathbf{X}_{*\mathcal{S}}^{n-1}\right]^{-1}$. Since the elements of $\mathbf{Y}^n$ are independent and $\mathrm{Cov}(\mathbf{Y}^n) = \sigma_\epsilon^2 \mathbf{I}$, (20) gives us

$$\mathrm{Cov}(\boldsymbol{\beta}_{\mathcal{S}})^{(n)} = \mathbf{M}_{\mathcal{S}}^{n-1}\sigma_\epsilon^2 + (\lambda^n)^2 \mathbf{M}_{\mathcal{S}}^{n-1}\mathrm{Cov}(\boldsymbol{z}_{\mathcal{S}})^{(n)}\mathbf{M}_{\mathcal{S}}^{n-1}. \tag{21}$$

By definition, $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n} := \mathrm{Cov}(\boldsymbol{\beta}_{\mathcal{S}})^{(n)}$. If we replace n with $n+1$, (21) provides us with the equation

$$\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1} = \mathbf{M}_{\mathcal{S}}^{n}\sigma_\epsilon^2 + (\lambda^{n+1})^2 \mathbf{M}_{\mathcal{S}}^{n}\mathrm{Cov}(\boldsymbol{z}_{\mathcal{S}})^{(n+1)}\mathbf{M}_{\mathcal{S}}^{n}. \tag{22}$$

One should note that we can not directly compute $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1}$ from the right hand side of (22), since $\boldsymbol{z}_{\mathcal{S}}$ is also a random variable dependent on $\widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^{n+1}$. But assuming that $\widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^{n+1}$ should not be far from $\boldsymbol{\vartheta}_{\mathcal{S}}^{n}$, one can sample a set of random variables from the distribution $\mathcal{N}(\boldsymbol{\vartheta}_{\mathcal{S}}^{n}, \boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},n})$ and then sample the subgradients according to the equations (15), (16) and (17), so $\mathrm{Cov}(\boldsymbol{z}_{\mathcal{S}})^{(n+1)}$ can be estimated from the sample covariance matrix estimator $\widehat{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}})^{(n+1)}$. Additionally, to make this estimator stable in theory, we need to make sure that all the eigenvalues of $\widehat{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}})^{(n+1)}$ are bounded away from 0 and infinity. Heuristically, we first define a matrix space $\mathcal{M}(C_{\min}, C_{\max})$ as

$$\mathcal{M}(C_{\min}, C_{\max}) = \{\mathbf{M} : C_{\min} \leq \Lambda_{\min}(\mathbf{M}) \leq \Lambda_{\max}(\mathbf{M}) \leq C_{\max}\}.$$

Then we can project $\widehat{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}})^{(n+1)}$ into $\mathcal{M}(C_{\min}, C_{\max})$ and find a solution $\widetilde{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}})^{(n+1)}$ to the following convex optimization problem

$$\widetilde{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}})^{(n+1)} = \operatorname*{argmin}_{\mathbf{M} \in \mathcal{M}(C_{\min}, C_{\max})} \|\widehat{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}})^{(n+1)} - \mathbf{M}\|_F. \tag{23}$$

Empirically we can use a surrogate projection procedure that computes a singular value decomposition of $\widehat{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}})^{(n+1)}$ and truncates all the eigenvalues to be within interval $[C_{\min}, C_{\max}]$. Therefore we can approximately estimate $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1}$ by

$$\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1} = \mathbf{M}_{\mathcal{S}}^{n}\sigma_\epsilon^2 + (\lambda^{n+1})^2 \mathbf{M}_{\mathcal{S}}^{n}\widetilde{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}})^{n+1}\mathbf{M}_{\mathcal{S}}^{n}. \tag{24}$$

Now we have all the ingredients for the knowledge gradient policy for sparse linear model (KGSpLin) and we outline it in Algorithm 1.

Algorithm 1 Sparse Knowledge Gradient Algorithm

Input: $\boldsymbol{\vartheta}^0, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},0}, \{\xi_j^0, \eta_j^0\}_{j=1}^p, \widetilde{\mathbf{X}}, \{\lambda^i\}_{i=1}^N$.
Output: $\boldsymbol{\vartheta}^N, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},N}, \{\xi_j^N, \eta_j^N\}_{j=1}^p$.
for $n = 0 : N-1$ **do**
> 1. KG: $\boldsymbol{x}^n = \operatorname{argmax} v_x^{KG,n}$;
> 2. Lasso homotopy update:[1] $\widehat{\boldsymbol{\vartheta}}^n, (\boldsymbol{x}^n, y^{n+1}) \in \mathbb{R}^m \times \mathbb{R}, \lambda^n, \lambda^{n+1} \to \widehat{\boldsymbol{\vartheta}}^{n+1}$;
> 3. Monte Carlo Simulation: approximately simulate $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1}$ from (24);
> 4. Bayesian update to: $\boldsymbol{\vartheta}^{n+1}, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n+1}, \{\xi_j^{n+1}, \eta_j^{n+1}\}_{j=1}^p$.

end

---

1. In practice, we often begin with some historical observations. Thus in the first iteration the Lasso estimator can be obtained from the historical dataset.

5. Knowledge Gradient for Sparse Additive Model

As we have the sparse knowledge gradient algorithm for $\ell_{1,\infty}$ group Lasso, we can generalize the knowlege gradient for sparse linear model to a nonparametric sparse additive model. In this section, we first describe the knowledge gradient for a sparse additive model, then we generalize it to the multivariate functional ANOVA model through tensor product splines.

5.1 Sparse Additive Modeling

In the additive model, $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_M]^T \in \mathbb{R}^M$, $\widetilde{\mathbf{X}} = [\widetilde{X}_{ij}] \in \mathbb{R}^{M \times p}$ is the alternative matrix and

$$\mu_i = f(\widetilde{\mathbf{X}}_{i*}) = \varsigma_i + \sum_{j=1}^{p} f_j(\widetilde{X}_{ij}), \quad \text{for } i = 1, \ldots, M, \tag{25}$$

where the f_js are one-dimensional smooth component functions, one for each covariate and $\boldsymbol{\varsigma} = [\varsigma_1, \ldots, \varsigma_M]^T$ is the residual term. For simplicity and identification purposes, we assume $\boldsymbol{\varsigma} = \mathbf{0}$ and $\int f_j(x_j)\,\mathrm{d}x_j = 0$ for each j. When $f_j(x) = \alpha_j x$, this simply reduces to the linear model in Section 4. In a high dimensional setting, where p may be relatively large, we assume most of the f_js are zeros.

If the truth takes the nonparametric additive form as in (25), similarly, we let the choice of which f_j is selected or not be random. Let $\boldsymbol{\zeta} = [\zeta_1, \ldots, \zeta_p]^T \in \mathbb{R}^p$ be the indicator random variable of f_j's, that is,

$$\zeta_j = \begin{cases} 1 & \text{if } f_j \neq 0 \\ 0 & \text{if } f_j = 0 \end{cases}, \quad \text{for } j = 1, \ldots, p.$$

Firstly, let us approximate each functional component in (25) through one-dimensional splines. Without loss of generality, suppose that all elements of $\widetilde{\mathbf{X}}$ take values in $[0, 1]$. Let $0 = \tau_0 < \tau_1 < \cdots < \tau_K < \tau_{K+1} = 1$ be a partition of $[0, 1]$ into $K + 1$ subintervals. Let $\mathcal{S}_l$ be the space of polynomial splines of order l (or degree $l - 1$) consisting of functions h satisfying

1 the restriction of h to each subinterval is a polynomial of degree $l - 1$;

2 for $l \geq 2$ and $0 \leq l' \leq l - 2$, h is l' times continuously differentiable on $[0, 1]$.

This definition is phrased after Stone (1985), which is a descriptive version of Definition 4.1 in Schumaker (1981, p. 108). Under suitable smoothness assumptions, the f_j's can be well approximated by functions in $\mathcal{S}_{l_j}$. Specifically, let $\widetilde{f}_j \in \mathcal{S}_{l_j}$ be the estimate of f_j. Furthermore, for each $\widetilde{f}_j$, there exists a normalized B-spline basis $\{\phi_{jk}(x), 1 \leq k \leq d_j\}$ for $\mathcal{S}_{l_j}$, where $d_j = K + l_j$ (Schumaker, 1981). If we let $\boldsymbol{\alpha}_{j\bullet} = [\alpha_{j1}, \ldots, \alpha_{jd_j}]$ be the coefficients of $\widetilde{f}_j$ projected onto $\mathcal{S}_{l_j}$, then for any $\widetilde{f}_j \in \mathcal{S}_{l_j}$, we can write

$$\widetilde{f}_j(x) = \sum_{k=1}^{d_j} \alpha_{jk}\phi_{jk}(x), \quad \text{for } 1 \leq j \leq p. \tag{26}$$

14

Algorithm 2 Knowledge Gradient Algorithm for Sparse Additive Models

---

Input:[2] $\boldsymbol{\vartheta}^0, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},0}, \{\xi_j^0, \eta_j^0\}_{j=1}^p, \widetilde{\mathbf{X}}, \{\lambda^i\}_{i=1}^N, \{\phi_{jk}\}_{k=1,j=1}^{d_j,p}, \{\tau_j\}_{j=0}^{K+1}$

Output: $\{f_j^N\}_{j=1}^p, \boldsymbol{\vartheta}^N, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},N}, \{\xi_j^N, \eta_j^N\}_{j=1}^p.$

for $n = 0 : N - 1$ **do**

    1. KG: $\boldsymbol{x}^n = \operatorname{argmax} v_x^{KG,n}$;

    2. Lasso homotopy update: $\widehat{\boldsymbol{\vartheta}}^n, (\phi_{jk}(x_j^n), y^{n+1}) \in \mathbb{R}^m \times \mathbb{R}, \lambda^n, \lambda^{n+1} \to \widehat{\boldsymbol{\vartheta}}^{n+1}$;

    3. Monte Carlo Simulation: approximately simulate $\widehat{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},n+1}$ from (24);

    4. Bayesian update to: $\{f_j^{n+1}\}_{j=1}^p, \boldsymbol{\vartheta}^{n+1}, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n+1}, \{\xi_j^{n+1}, \eta_j^{n+1}\}_{j=1}^p.$

end

Let $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_{1\bullet}, \ldots, \boldsymbol{\alpha}_{p\bullet}]$. We assume that $\boldsymbol{\alpha}$ takes the conditional distribution

$$\boldsymbol{\alpha}|\boldsymbol{\zeta} \sim \mathcal{N}(\boldsymbol{\vartheta}, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}}),$$

and also has the sparsity structure as described in Section 4. Then at time n, we also have estimate $\widehat{f}_j^n$ from group Lasso based on one-dimensional splines. More Specifically, for each $\widehat{f}_j^n \in \mathcal{S}_{l_j}$, let $\widehat{\boldsymbol{\vartheta}}_{j\bullet}^n = [\widehat{\vartheta}_{j1}^n, \ldots, \widehat{\vartheta}_{jd_j}^n]$ be the coefficients of $\widehat{f}_j^n$ and let $\widehat{\boldsymbol{\vartheta}}^n = [\widehat{\boldsymbol{\vartheta}}_{1\bullet}^n, \ldots, \widehat{\boldsymbol{\vartheta}}_{p\bullet}^n]$. Accordingly, in the batch setting, where we already have n samples $(\boldsymbol{x}^{i-1}, y^i) \in \mathbb{R}^m \times \mathbb{R}, i = 1, \ldots, n$, one can get $\widehat{\boldsymbol{\vartheta}}^n$ by solving the following penalized least squares problem

$$\widehat{\boldsymbol{\vartheta}}^n = \operatorname*{argmin}_{\boldsymbol{\vartheta} \in \mathbb{R}^m} \frac{1}{2} \sum_{i=1}^n \left[y^i - \sum_{j=1}^p \sum_{k=1}^{d_j} \vartheta_{jk} \phi_{jk}(x_j^{i-1}) \right]^2 + \lambda \sum_{j=1}^p \|\boldsymbol{\vartheta}_{j\bullet}\|_\infty, \tag{27}$$

where λ is the tuning parameter. Optimization problem (27) is essentially an $\ell_{1,\infty}$ group Lasso optimization problem. The parameter p is the number of groups and the group sparse solution on $\widehat{\boldsymbol{\vartheta}}$ would lead to a sparse solution on f_j's. Accordingly, we can also derive the knowledge gradient policy and Bayesian updating formulas as in Section 4. Here we let f_j^n be the Bayesian estimate of f_j at time n, that is,

$$f_j^n(x) = \sum_{k=1}^{d_j} \vartheta_{jk}^n \phi_{jk}(x), \quad \text{for } 1 \leq j \leq p.$$

We outline the knowledge gradient algorithm for sparse additive models (KGSpAM) in Algorithm 2.

5.2 Tensor Product Smoothing Splines Functional ANOVA

If the regression functions in (25) can also take bivariate or even multivariate functions, this model is known as the smoothing spline analysis of variance (SS-ANOVA) model (Wahba, 1990; Wahba et al., 1995; Gu, 2002). In SS-ANOVA, we write

$$\mu_i = f(\widetilde{\mathbf{X}}_{i*}) = \varsigma_i + \sum_{j=1}^p f_j(\widetilde{X}_{ij}) + \sum_{j<k} f_{jk}(\widetilde{X}_{ij}, \widetilde{X}_{ik}) + \cdots, \tag{28}$$

---

2. The prior mean and covariance matrix can also be obtained by some priors on f_j's.

where f_j's are the main effects components, f_{jk}'s are the two-factor interaction components, and so on. ς is the residual term. Similar as before, we assume $\varsigma = \mathbf{0}$, $\int f_j(x_j)\,dx_j = 0$ for each j, $\iint f_{jk}(x_j, x_k)\,dx_j dx_k = 0$ for each j, k and so on. This model is also called functional ANOVA. The sequence is usually truncated somewhere to enhance interpretability. This SS-ANOVA generalizes the popular additive model in Section 5.1 and provides a general framework for nonparametric multivariate function estimation, thus has been widely studied in the past decades.

As we approximate each f_j by $\mathcal{S}_{l_j}$, under certain smoothness assumptions, f_{jk} can be well approximated by the tensor product space $\mathcal{S}_{l_j} \otimes \mathcal{S}_{l_k}$ defined by

$$
\begin{aligned}
\mathcal{S}_{l_j} \otimes \mathcal{S}_{l_k} : \quad &= \quad \{h_j h_k : \text{for all } h_j \in \mathcal{S}_{l_j}, h_k \in \mathcal{S}_{l_k}\} \\
&= \quad \{\sum_{r=1}^{d_j} \sum_{q=1}^{d_k} c_{rq} \phi_{jr} \phi_{kq} : \text{for all } c_{rq} \in \mathbb{R}\}.
\end{aligned}
$$

Let

$$
\phi_{jrkq}(x_j, x_k) := \phi_{jr}(x_j)\phi_{kq}(x_k), \quad \text{for } 1 \leq r \leq d_j, 1 \leq q \leq d_k,
$$

then these are the basis functions for $d_j d_k$ dimensional tensor product space $\mathcal{S}_{l_j} \otimes \mathcal{S}_{l_k}$. This can also be generalized to multi-factor interaction components. Therefore, similarly, we can write all the functional components in (28) as basis expansion forms. Then we can generalize a knowledge gradient algorithm for SS-ANOVA model.

6. Theoretical Results

In this section we provide the estimation error bounds of the Bayesian posterior mean estimate in Algorithm 1 as well as of the functional estimate in Algorithm 2. We first state the selection and estimation properties of $\ell_{1,\infty}$ group Lasso in high dimensional settings when the number of groups exceeds the sample size. We show the estimation error bound of group Lasso. We also provide the sufficient conditions under which the group Lasso selects a model whose dimension is comparable with the underlying model with high probability. Based on these results, we assume that we begin with some historical observations and the Lasso estimator from the historical dataset has good initial property. If we have such a "warm" start, we can show that the Bayesian posterior estimation error is bounded as in Theorem 10. The theorem actually shows that the posterior can converge to the truth at the same rate as that of group Lasso. Besides, based on this error bound, we can also show the estimation error bound of the functional estimate as in Theorem 12. Note that these error bounds are proved on the intersection $\bar{\mathcal{S}}$ of the support set $\mathcal{S}^n$ from group Lasso estimator. But we can also prove that $\bar{\mathcal{S}}$ is comparable with the true support set $\mathcal{S}^*$. Additionally, all these theorems show the estimation error bounds as large enough measurements are made. Since our policy is also myopically optimal by construction, this lends a strong theoretical guarantee that the algorithm will work well for finite budgets.

6.1 Bayesian Posterior Mean Estimation Error Bound

In addition to the aforementioned notation, let $\boldsymbol{\epsilon}^n = [\epsilon^1, \ldots, \epsilon^n]^T$ be the measurement noise vector, so we have $\mathbf{Y}^n := \mathbf{X}^{n-1}\boldsymbol{\vartheta} + \boldsymbol{\epsilon}^n$. Then, we define the maximum group size

$\bar{d} := \max_{j=1,\dots,p} d_j$ and the minimum group size $\underline{d} := \min_{j=1,\dots,p} d_j$. Let $d = \bar{d}/\underline{d}$. Let $\mathcal{S}^n = \{j : \widehat{\boldsymbol{\vartheta}}_{\mathcal{G}_j}^n \neq 0\}$ be the estimated group support from current Lasso estimator. Let $\mathcal{S}^*$ be the true support. Also, let $s^* = |\mathcal{S}^*|$ be the cardinality of $\mathcal{S}^*$.

Before proving the estimation error bound, let us first introduce the selection and estimation properties of $\ell_{1,\infty}$ group Lasso. Our presentation will need the following assumptions.

Assumption 1 *For any n, the random noise errors $\epsilon^1, \dots, \epsilon^n$ are independent and identically distributed as $\mathcal{N}(0, \sigma_\epsilon^2)$.*

Assumption 2 *The design matrix $\mathbf{X}^{n-1}$ satisfies the sparse Riesz condition (SRC) with rank r and spectrum bounds $0 < c_* < c^* < \infty$ if*

$$c_* \|\boldsymbol{\nu}\|_2^2 \leq \frac{\|\mathbf{X}_{*\mathcal{S}}^{n-1}\boldsymbol{\nu}\|_2^2}{n} \leq c^* \|\boldsymbol{\nu}\|_2^2, \quad \forall \mathcal{S} \text{ with } r = |\mathcal{S}| \text{ and } \boldsymbol{\nu} \in \mathbb{R}^{\sum_{j\in\mathcal{S}} d_j}.$$

We refer to this condition as SRC (r, c_*, c^*).

Assumption 3 *For a given group $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_p\}$. We say $\mathbf{X}^{n-1}$ is block normalized if*

$$\frac{\|\mathbf{X}_{*\mathcal{G}_j}^{n-1}\|_2}{\sqrt{n}} \leq 1, \quad \text{for all } j = 1, 2, \dots, p.$$

Remark 4 *In Assumption 3, we set the upper bound to one in order to simplify notation. This particular choice entails no loss of generality. Note that this assumption is a natural generalization of the column normalization condition. Specifically, if we have $m = p$ groups, each of size one, the matrix norm reduces to the vector norm on every column of $\mathbf{X}^{n-1}$.*

All three assumptions can be reasonably expected to hold in practice. Assumption 1 is on the distribution of random noise. The SRC in Assumption 2 assumes the eigenvalues of the sample covariance matrix $\boldsymbol{\Sigma}_{\mathcal{S}}^{\mathbf{X},n-1} := \frac{1}{n}(\mathbf{X}_{*\mathcal{S}}^{n-1})^T \mathbf{X}_{*\mathcal{S}}^{n-1}$ are bounded below from zero and above from infinity when the size of $\mathcal{S}$ is no greater than r. It is natural to ask whether such condition also holds for general matrices. In fact,

Proposition 5 *Suppose $\mathbf{X}^{n-1}$ is column standardized with $\|\mathbf{X}_{*j}^{n-1}\|_2^2/n = 1$. Let $\rho_{jk} = (\mathbf{X}_{*j}^{n-1})^T \mathbf{X}_{*k}^{n-1}/n$ be the correlation. If*

$$\max_{|\mathcal{S}|=r} \inf_{\kappa \geq 1} \left\{ \sum_{j\in\mathcal{G}_\mathcal{S}} \left(\sum_{k\in\mathcal{G}_\mathcal{S}, k\neq j} |\rho_{jk}|^{\kappa/(\kappa-1)} \right)^{\kappa-1} \right\}^{1/\kappa} \leq \delta < 1,$$

then the sparse Riesz condition in Assumption 2 holds with rank r and spectrum bounds $c_* = 1 - \delta$ and $c^* = 1 + \delta$. In particular, Assumption 2 holds with $c_* = 1 - \delta$ and $c^* = 1 + \delta$ if

$$\max_{1\leq j<k\leq m} |\rho_{jk}| \leq \frac{\delta}{r-1}, \quad \delta < 1.$$

Based on these assumptions, we can combine the results in Wei and Huang (2010) and Negahban et al. (2012) and get the estimation error bound for $\ell_{1,\infty}$ group Lasso estimator as given in Lemma 6.

Lemma 6 *Under Assumption 1, 2 and 3, if we solve the group Lasso given in (5) with*

$$\lambda^n = O(\bar{d}\sqrt{n \log p}),$$

then the following properties hold with probability converging to 1:

(1) $|\mathcal{S}^n| \leq C_1|\mathcal{S}^*|$ *for some finite positive constant* C_1. *In specific,* $C_1 = 2 + 4d\bar{c}$, *where* $\bar{c} := c^*/c_*$.

(2) *Any optimal solution* $\widehat{\boldsymbol{\beta}}^n$ *to (5) satisfies the following error bound*

$$\|\widehat{\boldsymbol{\beta}}^n - \boldsymbol{\beta}\|_2^2 \leq \frac{C_2 \sigma_\epsilon^2 s^* \bar{d}^2 \log p}{n},$$

for some positive constant C_2.

As one can see from the updating equations in (13) and (12), the posterior mean estimate $\boldsymbol{\vartheta}_{\mathcal{S}}^{n+1}$ is the weighted sum of prior $\boldsymbol{\vartheta}_{\mathcal{S}}^n$ and the current Lasso estimate $\widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^{n+1}$. If the Lasso estimate has ℓ_2 estimation bound as described in Lemma 6, the posterior estimate should also have a similar bound under certain conditions of the weighted covariance matrix. One should note that both the mean and covariance are updated on some support $\mathcal{S}$ from the current Lasso estimate. Thus we will work on a sequence of Lasso solutions and prove the bound on the intersection support set as large enough samples are made. Also note that in order to use the bound in Lemma 6, we need to make sure that assumptions 1, 2 and 3 are satisfied for every Lasso problem in such a sequence. Assumptions 1 and 3 are easy to satisfy. To show all the sequential Lasso problems satisfy Assumption 2, we work from a "warm" start at time N'. The following proposition actually verifies that if the design matrix at time N' satisfies Assumption 2, then the following ones should also satisfy this assumption, only with a slight loose on the constant.

Proposition 7 *If for any n, there exists some constant $B > 0$ such that $\|\boldsymbol{x}^n\|_2^2 \leq B$. Besides, assume for some large enough N', the design matrix $\mathbf{X}^{N'-1}$ satisfies condition SRC (r, c_*, c^*). Then, for all $N' < n' \leq cN'$, of which $c > 1$ is some constant, the design matrix $\mathbf{X}^{n'-1}$ can satisfy condition SRC $(r, c_*/c, \max(c^*, B))$.*

Thus we have all the ingredients to complete the proof of ℓ_2 error bound of the Bayesian posterior mean estimator. Before that, let us state some assumptions for this result.

Assumption 8 *For any n, there exists some constant $B > 0$ such that $\|\boldsymbol{x}^n\|_2^2 \leq B$.*

Assumption 9 *For some large enough n, suppose for some constant $c > 1$ and $n \leq cN'$, the design matrix $\mathbf{X}^{N'-1}$ satisfies the block normalization condition 3 and condition SRC $(C_3 s^*, c_*, c^*)$, where $C_3 := 2 + 4dc \max(c^*, B)/c_*$.*

Under these assumptions, we have the following theorem of the ℓ_2 mean posterior estimation error bound.

Theorem 10 *Under Assumption 1, 8 and 9, if we solve the group Lasso given in* (5) *with*

$$\lambda^n = O(\bar{d}\sqrt{n \log p})$$

and let $\bar{\mathcal{S}} := \bigcap_{n'=N'}^{n} \mathcal{S}^{n'}$, *then the following properties hold with probability converging to 1:*

(1) $|\bar{\mathcal{S}}| \leq C_3 |\mathcal{S}^*|$ *for some finite positive constant* C_3 *defined in Assumption 9.*

(2) *Any posterior estimate* $\boldsymbol{\vartheta}^n$ *from Algorithm 1 satisfies*

$$\|\boldsymbol{\vartheta}^n_{\bar{\mathcal{S}}} - \boldsymbol{\vartheta}_{\bar{\mathcal{S}}}\|_2^2 \leq \frac{C_4 \sigma_\epsilon^2 s^* \bar{d}^2 \log p}{n},$$

for some positive constant C_4.

6.2 Functional Estimation Error Bound

Based on the results in Section 6.1, we can also get the error bound for functional estimate of Algorithm 2 in Section 5.1. To show this error bound, let us introduce more definitions and assumptions.

Let β be a nonnegative integer, let $\delta \in [0, 1]$ be such that $q = \beta + \delta > 0.5$, and $L \in (0, \infty)$. Let $\mathcal{H}(q, L)$ denote the collection of functions h on [0,1] whose βth derivative, $h^{(\beta)}$, exists and satisfies the Hölder condition with exponent δ,

$$|h^{(\beta)}(t') - h^{(\beta)}(t)| \leq L|t' - t|^\delta, \quad \text{for } 0 \leq t, t' \leq 1.$$

Whenever the integral exists, for a function h on $[0, 1]$, denote its $\|\cdot\|_2$ norm by

$$\|h\|_2 := \sqrt{\int_0^1 h^2(x)\mathrm{d}x},$$

Additionally, for any $\mathcal{S} \subset \{1, \ldots, p\}$, we define

$$\|h_{\mathcal{S}}\|_2^2 := \sum_{j \in \mathcal{S}} \|h_j\|_2^2.$$

To prove the functional estimation error bound, we assume the true functions belong to this function class with smoothness parameter $q = 2$.

Assumption 11 $f_j \in \mathcal{H}(2, L)$ *for* $1 \leq j \leq p$.

Also note here we have the new design matrix $\mathbf{X}^{n-1}$ on the basis ϕ_{jk}. Let Ψ_j^{n-1} be the $n \times d_j$ matrix $\Psi_j(i, k) = \psi_{jk}(x_j^{i-1})$, where ψ_{jk} is the orthonormal B-spline basis. Let $\Psi^{n-1} := [\Psi_1^{n-1}, \ldots, \Psi_p^{n-1}]$. Based on this and Theorem 10, we have the following theorem of the functional estimation error bound.

Theorem 12 *Under assumptions 1 and 11, if the design matrix $\Psi^{N'-1}$ satisfies Assumption 9 and 8, the group Lasso is solved with some λ^n satisfying*

$$\lambda^n = O(\bar{d}\sqrt{n \log p}),$$

let $\bar{\mathcal{S}} := \bigcap_{n'=N'}^{n} \mathcal{S}^{n'}$, $\bar{d} = O(n^{1/6})$, $s^* = O(1)$, then the following properties hold with probability converging to 1:

(1) $|\bar{\mathcal{S}}| \leq C_3 |\mathcal{S}^*|$ for some finite positive constant C_3.

(2) Any posterior estimate f^n from Algorithm 2 satisfies

$$\|f^n_{\bar{\mathcal{S}}} - f_{\bar{\mathcal{S}}}\|_2^2 \leq \frac{C_5 \sigma_\epsilon^2 \log p}{n^{2/3}},$$

where C_5 is some positive constant.

Remark 13 *Note here we use $\ell_{1,\infty}$ group Lasso instead of $\ell_{1,2}$ group Lasso, this is because the homotopy algorithm for recursive $\ell_{1,\infty}$ group Lasso largely reduces the computational complexity, but we do not have such results for $\ell_{1,2}$ group Lasso. However for $\ell_{1,2}$ group Lasso, the bound takes the form $\|\widehat{\boldsymbol{\beta}}^n - \boldsymbol{\beta}\|_2^2 \precsim \frac{s^* \bar{d} \log p}{n}$. As one can see, the error term for $\ell_{1,\infty}$ group Lasso $\frac{s^* \bar{d}^2 \log p}{n}$ is larger by a factor of $\bar{d}$, which corresponds to the amount by which an ℓ_∞-ball in $\bar{d}$ dimensions is larger than the corresponding ℓ_2-ball. Therefore, we do not achieve the minimax optimal rate as in $\ell_{1,2}$ group Lasso. Thus using $\ell_{1,\infty}$ group Lasso instead of $\ell_{1,2}$ group Lasso is actually a tradeoff between computational complexity and statistical estimation.*

7. Experimental Testing

In this section, we investigate the performance of KGSpLin and KGSpAM in controlled experiments. In these experiments, we repeatedly sample the truth from some distribution and compare different policies to see how well we are discovering the truth.

We first test the KGSpLin by generating a linear model with $p = 100$ predictors, in ten groups of ten. The last 80 predictors all have coefficients of zero. The coefficients of the first 2 groups, that is 20 predictors, are randomly sampled from a normal distribution with means from 11 to 30 respectively, with standard deviation of 30% of the mean. We randomly choose $M = 100$ alternatives from some Gaussian distribution. Finally, normal measurement noise with standard deviation ϵ is added to each observation. In our first experiment, we focus on the comparison with KGLin and exploration policies using a relatively large measurement budget $N = 200$.

Furthermore, of all the experiments in this paper, to make a fair comparison of KG and exploration, the updating scheme when using 〈tb〉 exploration policy is as described in Section 4.2. The only difference is that at each iteration, exploration randomly measures each alternative with the same probability, while KG chooses the one with maximum KG value. Also, we assume that we do not have any prior information on the sparsity structures. That is, $\xi_j^0 = \eta_j^0 = 1$, for $j = 1, \ldots, p$.

Figure 2(a) shows the corresponding misclassification groups for KGSpLin and KGLin as the regularization parameter λ is varied. (A misclassified group is one with at least one

nonzero coefficient whose estimated coefficients are all set to zero, or vice versa.) Figure 2(b) and (c) show the log of the averaged opportunity cost over 300 replications using a well chosen tuning parameter with low and high measurement noise (the standard deviations of the measurement noises are respectively 5% and 30% of the expected range of the truth). Here the opportunity cost (OC) is defined as the difference in true value between the best option and the option chosen by the policy, that is

$$OC = \max_i \mu_i - \mu_{i*}.$$
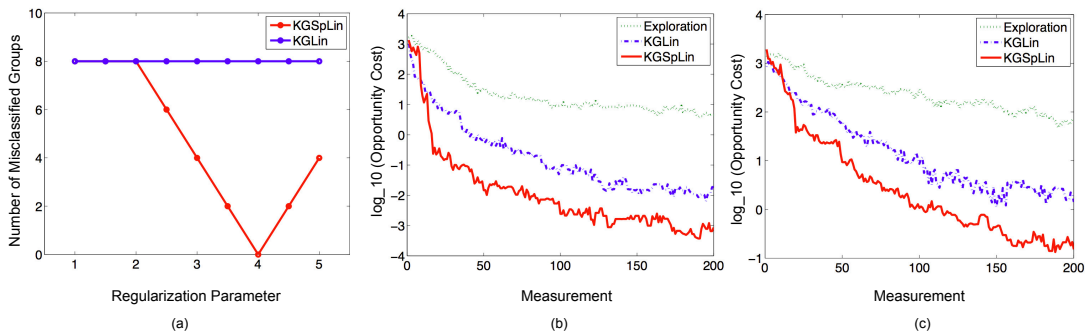
Figure 2: (a) shows the misclassified groups for KGSpLin and KGLin as the regularization parameter λ is varied. (b)(c) shows the averaged opportunity cost over 300 runs under low measurement noise (5% range of the truth) and high measurement noise (30 % range of the truth).

From Figure 2 we can see that during the first several iterations, KGSpLin behaves comparable with pure exploration, because Lasso takes several iterations to identify the key features. However, after several initial samples, KGSpLin far outperforms both KGLin and pure exploration. This is because Lasso gives a rather precise estimate of the sparse linear coefficients given enough samples. So the algorithm mainly updates the beliefs on the key features based on these Lasso estimators, leading to more precise estimates of the model.

To further compare the KGSpLin policy with KGLin from Negoescu et al. (2011) for high dimensional sparse belief functions, we take several standard low dimensional test functions and hide them in a $p = 200$ dimensional space. These functions were designed to be minimized, so both policies were applied to the negative of the functions. Table 1 shows the performance on the different functions. Each policy was run 500 times with the specified amount of observation noise. Table 1 gives the sample mean and standard deviation of the mean of the opportunity cost after $N = 50$ iterations of each policy. Here each function is scaled to have a range of 100, so that the measurement noises are given on the same scale.

Furthermore, we now test the KGSpAM policy on the following SS-ANOVA model with $p = 100$ and four relevant variables,

$$\mu_i = f_{12}(X_{i1}, X_{i2}) + \sum_{j=3}^{5} f_j(X_{ij}) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, 1);$$

Test function	σ	KGSpLin			KGLin		
		$\mathbb{E}(OC)$	$\sigma(OC)$	Med	$\mathbb{E}(OC)$	$\sigma(OC)$	Med
Matyas	1	.0104	.0256	.0071	.0284	.0157	.0244
$\mathcal{X} = [-10, 10]^2$	10	.2772	.1960	.0125	.3451	.1166	0.3781
	20	.7658	.8423	.3997	1.7155	.3208	1.5627
Trid	1	2.1422	1.4011	1.1843	2.7092	1.5331	1.3036
$d = 6, \mathcal{X} = [-36, 36]^6$	10	9.8196	3.8757	8.9874	9.9787	4.2098	8.2282
	20	15.7164	4.0201	14.9040	16.8911	4.5881	15.4959
Bohachevsky	1	.0746	.0249	.0035	.0853	.0370	.0013
$\mathcal{X} = [-100, 100]^2$	10	.3585	2.5349	.2876	.5611	2.7056	.2993
	20	1.8224	3.230	1.5578	1.9668	3.696	1.7008
Six-hump Camel	1	.0023	.0019	.0000	.0117	.8097	.0000
$\mathcal{X} = [-3, 3] \times [-2, 2]$	10	.0895	.6332	.0000	.1293	.6098	.0000
	20	.4922	.2159	.0215	.6183	.2696	0.0306

Table 1: Quantitative comparison for KGSpLin and KGLin on standard test functions. Each row summarizes 500 runs of each policy on the specified test function. We compute the mean, standard deviation and median of OC. Each function is scaled to have a range of 100 and the results are given for different levels of noise standard deviation.

the relevant component functions are given by

$$f_{12}(x_1, x_2) \ = \ 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1 x_2 + x_2^2, \tag{29}$$

$$f_3(x) \ = \ 2\sin(2\pi x), \tag{30}$$

$$f_4(x) \ = \ 8(x - 0.5)^2, \tag{31}$$

$$f_5(x) \ = \ 2\exp(-3x), \tag{32}$$

where the first component function f_{12} in (29) is known as the Three-hump camel function. We plot the true Three-hump camel function in Figure 3(a), while the key part is shown in Figure 3(b). For f_{12}, we use B-splines tensor product space $\mathcal{S}_4 \otimes \mathcal{S}_4$ to approximate it. The knot sequence is equally spaced on $[-5, 5]^2$ with $K = 4$ (the number of subintervals for each dimension is $K + 1 = 5$). The remaining three relevant components are approximated using B-splines with order $l = 4$ and equally spaced knot sequence on $[0, 1]$ with $K = 4$. The alternatives are uniformly sampled on the domain with $M = 400$ and the measurement budget N is 30. The standard deviation of measurement noise is set to 20% of the expected range of the truth.

Then we run the KGSpAM policy on a $p = 100$-dimensional space. To better visualize its performance, we plot the starting prior and estimated function of negative f_{12} on its key region after the initial 10 and 30 observations as shown in Figure 4. Comparing these estimates with the true function shown in Figure 3, we visually see that the policy has done a good job estimating the lower key regions of the functions as desired after 10 observations
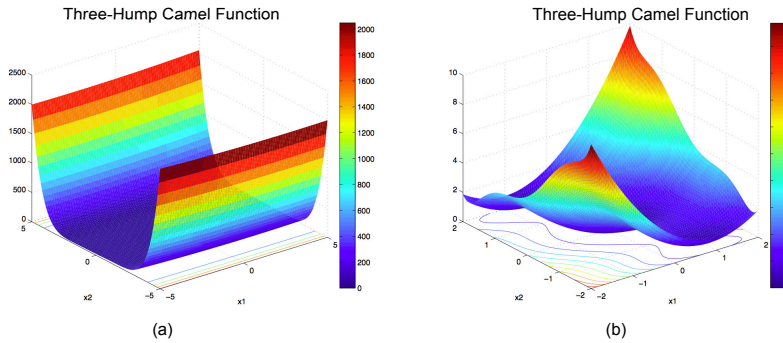
Figure 3: (a) shows the negative Three-hump camel function on its recommended input domain, and (b) shows only a portion of this domain, to allow for easier viewing of the function's key characteristics. The function has one global maximum and two other local maxima.

and it identifies the areas of the three maxima after 30 observations. For the remaining three relevant functional components in (30), (31) and (32), we plot the prior, truth and final estimates of KGLin and KGSpAM in Figure 5.
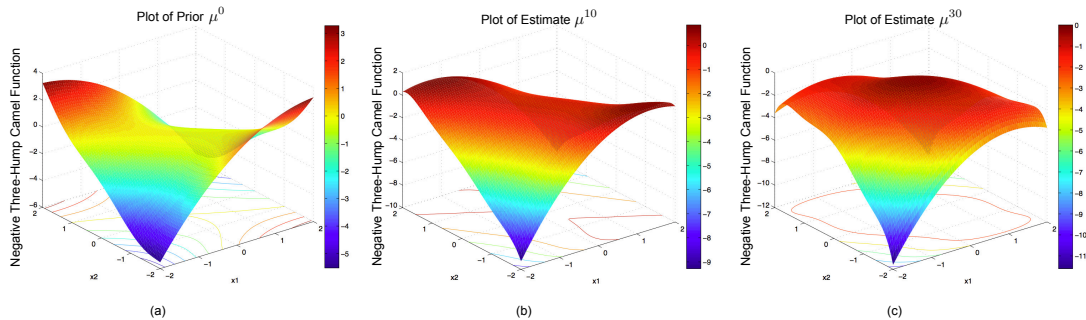

Figure 4: (a) shows the prior of negative Three-hump camel function on its key region. (b) and (c) show the estimates of negative Three-hump camel function on its key region after 10 and 30 observations respectively.

8. Conclusion

In this paper, we extend the KG policy to high dimensional linear and nonparametric additive beliefs. It is a novel hybrid of Bayesian R&S with the frequentist learning approach. Parallel with the Bayesian model, the policies use frequentist recursive Lasso approach to generate estimates and update the Bayesian model. Empirically, both KGSpLin and KGSpAM greatly reduce the measurement budget effort and perform significantly better than several other policies in high dimensional setting. In addition, these policies are easy to implement and fast to compute. Theoretically, we prove that our policies are consistent. That is, the estimates can converge to the truth when given enough measurements. This
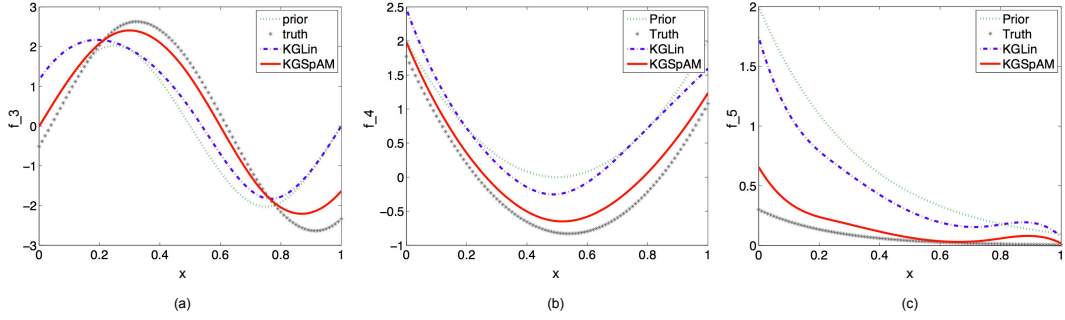
Figure 5: (a)(b)(c) The prior, truth and final estimate of sparse additive model in (30)-(32) comparing KGLin and KGSpAM after $N = 30$ observations. The standard deviation of measurement noise is 1, which is about 20% of the expected range of the truth.

also guarantees the convergence to global optimal alternative. All these advantages make them reasonable alternatives to other policies for high dimensional applications with sparse structure. Despite the advances, the convergence theory requires a number of structural assumptions, suggesting that future research should look to identify algorithms that work with more general model structures in high dimensions.

Appendix A.

Refer to Table 2.

Appendix B. Proofs

In the following, we present the detailed proofs of all the technical results.

B.1 Proof of Proposition 7

Let us define $\boldsymbol{\Sigma}^{\mathbf{X},n-1}$ be the sample covariance matrix, that is $\boldsymbol{\Sigma}^{\mathbf{X},n-1} = \frac{(\mathbf{X}^{n-1})^T \mathbf{X}^{n-1}}{n}$. For any $N' < n' \leq cN'$, let us divide the design matrix $\mathbf{X}^{n'-1}$,

$$\mathbf{X}^{n'-1} = \begin{bmatrix} \mathbf{X}^{N'-1} \\ \mathbf{X}^+ \end{bmatrix}.$$

We need to prove $\mathbf{X}^{n'-1}$ satisfies condition SRC $(r, c_*/c, \max(c^*, B))$. Note that $\mathbf{X}^{N'-1}$ satisfies SRC (r, c_*, c^*) is equivalent to

$$c_* \leq \Lambda_{\min}(\Sigma_{\mathcal{S}}^{\mathbf{X},N'-1}) \leq \Lambda_{\max}(\Sigma_{\mathcal{S}}^{\mathbf{X},N'-1}) \leq c^*, \quad \forall \mathcal{S} \text{ with } r = |\mathcal{S}| \text{ and } \boldsymbol{\nu} \in \mathbb{R}^{\sum_{j \in \mathcal{S}} d_j}.$$

24

Variable	Description
$\mathcal{X}$	Set of alternatives
M	Number of alternatives
N	Number of measurements budget
μ_x	Unknown mean of alternative x
σ_x	Known standard deviation of alternative x
$\boldsymbol{\mu}$	Column vector $(\mu_1, \ldots, \mu_M)^T$
$\boldsymbol{x}^i/x^i$	Sampling decision at time i (vector or scalar index)
ϵ_x^{n+1}	Measurement error of alternative $\boldsymbol{x}^n$
y^{n+1}	Sampling observation from measuring alternative $\boldsymbol{x}^n$
$\boldsymbol{\theta}^n, \boldsymbol{\Sigma}^n$	Mean and Covariance of prior distribution on μ at time n
S^n	State variable, defined as the pair $(\boldsymbol{\theta}^n, \boldsymbol{\Sigma}^n)$
$v_x^{KG,n}$	Knowledge gradient value for alternative x at time n
$\boldsymbol{\alpha}$	Vector of linear coefficients
m	Number of features
$\widetilde{\mathbf{X}}$	Alternative matrix
$\boldsymbol{\vartheta}^n, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}$	Mean of covariance of posterior distribution on α after n measurements
λ^n	Regularization parameter for Lasso at time n
p	Number of nonoverlapping groups for features
$\mathcal{G}, \mathcal{G}_j$	Group index
d_j	Number of features in the jth group, $d_j = |\mathcal{G}_j|$
$\boldsymbol{\zeta}^n$	Prior of $\boldsymbol{\zeta}$ at time n
p_j^n	Parameter of Bernoulli distribution on ζ_j^n
(ξ_j^n, η_j^n)	Set of parameters of Beta distribution on p_j^n
$\widehat{\boldsymbol{\vartheta}}^n$	Lasso estimate at time n
$(\widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^n, \widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n})$	Mean and covariance matrix estimator from Lasso solution at time n
$\mathcal{P}^n$	Index of selected groups from Lasso estimate at time n
$\mathcal{P}$	Active group index set
$\mathcal{Q}$	Inactive group index set
$\mathcal{A}_j$	Index set in the jth group with maximum absolute values
$\mathcal{B}_j$	Index set in the jth group except for $\mathcal{A}_j$
f_j	Smooth function of the jth feature
K	Number of interior knots for one dimensional splines
$\mathcal{S}_{l_j}$	Space of polynomial spline of order l_j
ϕ_{jk}	k-th B-spline basis function for $\mathcal{S}_{l_j}$
α_{jk}	Coefficient for f_j on basis function ϕ_{jk}
f_{jk}	Two-factor interaction component in SS-ANOVA model
ϕ_{jrkq}	rq-th B-spline basis function for $\mathcal{S}_{l_j} \otimes \mathcal{S}_{l_k}$
$\bar{d}$	Maximum group size
$\mathbf{X}^{n-1}$	Design matrix with rows of $\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{n-1}$
q	Smoothness parameter of the Hölder class $\mathcal{H}$
s^*	Cardinality of the true group set, $s^* = |\mathcal{S}^*|$

Table 2: Table of Notation

Then we have that for $\forall \mathcal{S}$ with $r = |\mathcal{S}|$

$$
\begin{aligned}
\mathbf{\Sigma}_{\mathcal{S}}^{\mathbf{X},n'-1} &= \frac{(\mathbf{X}_{*\mathcal{S}}^{n'-1})^T \mathbf{X}_{*\mathcal{S}}^{n'-1}}{n'} = \frac{(\mathbf{X}_{*\mathcal{S}}^{N'-1})^T \mathbf{X}_{*\mathcal{S}}^{N'-1} + (\mathbf{X}_{*\mathcal{S}}^{+})^T \mathbf{X}_{*\mathcal{S}}^{+}}{n'} \\
&= \frac{N' \mathbf{\Sigma}_{\mathcal{S}}^{\mathbf{X},N'-1} + (\mathbf{X}_{*\mathcal{S}}^{+})^T \mathbf{X}_{*\mathcal{S}}^{+}}{n'}
\end{aligned}
$$

This implies that

$$
\Lambda_{\min}(\Sigma_{\mathcal{S}}^{\mathbf{X},n'-1}) \geq \frac{N'}{n'} \Lambda_{\min}(\Sigma_{\mathcal{S}}^{\mathbf{X},N'-1}) \geq \frac{c_*}{c} \tag{33}
$$

and

$$
\Lambda_{\max}(\Sigma_{\mathcal{S}}^{\mathbf{X},n'-1}) \leq \frac{N'}{n'} \Lambda_{\max}(\Sigma_{\mathcal{S}}^{\mathbf{X},N'-1}) + \frac{1}{n'} \Lambda_{\max}[(\mathbf{X}_{*\mathcal{S}}^{+})^T \mathbf{X}_{*\mathcal{S}}^{+}].
$$

Since

$$
(\mathbf{X}_{*\mathcal{S}}^{+})^T \mathbf{X}_{*\mathcal{S}}^{+} = \boldsymbol{x}_{\mathcal{S}}^{N'}(\boldsymbol{x}_{\mathcal{S}}^{N'})^T + \boldsymbol{x}_{\mathcal{S}}^{N'+1}(\boldsymbol{x}_{\mathcal{S}}^{N'+1})^T + \cdots + \boldsymbol{x}_{\mathcal{S}}^{n'-1}(\boldsymbol{x}_{\mathcal{S}}^{n'-1})^T
$$

and

$$
\Lambda_{\max}[\boldsymbol{x}_{\mathcal{S}}^{n}(\boldsymbol{x}_{\mathcal{S}}^{n})^T] = \|\boldsymbol{x}_{\mathcal{S}}^{n}\|_2^2 \leq B, \quad \forall n,
$$

we can get that

$$
\Lambda_{\max}(\Sigma_{\mathcal{S}}^{\mathbf{X},n'-1}) \leq \frac{N'}{n'} c^* + \frac{n' - N'}{n'} B \leq \max(c^*, B). \tag{34}
$$

Combining (33) and (34) completes the proof. ∎

B.2 Proof of Theorem 10

The proof of part (1) directly follows Assumption 9, Proposition 7 and Lemma 6. We now proceed to prove part (2). If we let $\bar{\mathcal{S}} := \bigcap_{n'=N'}^{n} \mathcal{S}^{n'}$, then from updating formula in (13) and (12), we have

$$
\begin{aligned}
\boldsymbol{\vartheta}_{\bar{\mathcal{S}}}^{n} &= \mathbf{\Sigma}_{\bar{\mathcal{S}}}^{\boldsymbol{\vartheta},n} \left[(\mathbf{\Sigma}_{\bar{\mathcal{S}}}^{\boldsymbol{\vartheta},N'-1})^{-1} \boldsymbol{\vartheta}_{\bar{\mathcal{S}}}^{N'-1} + [(\widehat{\mathbf{\Sigma}}_{\mathcal{S}^{N'}}^{\boldsymbol{\vartheta},N'})^{-1}]_{\bar{\mathcal{S}}} \widehat{\boldsymbol{\vartheta}}_{\bar{\mathcal{S}}}^{N'} + \cdots + [(\widehat{\mathbf{\Sigma}}_{\mathcal{S}^{n}}^{\boldsymbol{\vartheta},n})^{-1}]_{\bar{\mathcal{S}}} \widehat{\boldsymbol{\vartheta}}_{\bar{\mathcal{S}}}^{n} \right], \\
\mathbf{\Sigma}_{\bar{\mathcal{S}}}^{\boldsymbol{\vartheta},n} &= \left[(\mathbf{\Sigma}_{\bar{\mathcal{S}}}^{\boldsymbol{\vartheta},N'-1})^{-1} + [(\widehat{\mathbf{\Sigma}}_{\mathcal{S}^{N'}}^{\boldsymbol{\vartheta},N'})^{-1}]_{\bar{\mathcal{S}}} + \cdots + [(\widehat{\mathbf{\Sigma}}_{\mathcal{S}^{n}}^{\boldsymbol{\vartheta},n})^{-1}]_{\bar{\mathcal{S}}} \right]^{-1}.
\end{aligned}
$$

Then if we define

$$
\begin{aligned}
\boldsymbol{\delta}_{\bar{\mathcal{S}}}^{n'} &:= \boldsymbol{\vartheta}_{\bar{\mathcal{S}}}^{n'} - \boldsymbol{\vartheta}_{\bar{\mathcal{S}}} \\
\widehat{\boldsymbol{\delta}}_{\bar{\mathcal{S}}}^{n'} &:= \widehat{\boldsymbol{\vartheta}}_{\bar{\mathcal{S}}}^{n'} - \boldsymbol{\vartheta}_{\bar{\mathcal{S}}},
\end{aligned}
$$

for all $N' - 1 \leq n' \leq n$ to simplify notation, we have

$$
\boldsymbol{\delta}_{\bar{\mathcal{S}}}^{n} = \mathbf{\Sigma}_{\bar{\mathcal{S}}}^{\boldsymbol{\vartheta},n} \left[(\mathbf{\Sigma}_{\bar{\mathcal{S}}}^{\boldsymbol{\vartheta},N'-1})^{-1} \boldsymbol{\delta}_{\bar{\mathcal{S}}}^{N'-1} + [(\widehat{\mathbf{\Sigma}}_{\mathcal{S}^{N'}}^{\boldsymbol{\vartheta},N'})^{-1}]_{\bar{\mathcal{S}}} \widehat{\boldsymbol{\delta}}_{\bar{\mathcal{S}}}^{N'} + \cdots + [(\widehat{\mathbf{\Sigma}}_{\mathcal{S}^{n}}^{\boldsymbol{\vartheta},n})^{-1}]_{\bar{\mathcal{S}}} \widehat{\boldsymbol{\delta}}_{\bar{\mathcal{S}}}^{n} \right].
$$

This gives us the following bound on $\boldsymbol{\delta}_{\mathcal{S}}^n$,

$$\|\boldsymbol{\delta}_{\mathcal{S}}^n\|_2 \leq \|\boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},n}\|_2 \left[\|(\boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},N'-1})^{-1}\|_2 \|\boldsymbol{\delta}_{\mathcal{S}}^{N'-1}\|_2 + \|[(\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}^{N'}}^{\boldsymbol{\vartheta},N'})^{-1}]_{\bar{\mathcal{S}}}\|_2 \|\widehat{\boldsymbol{\delta}}_{\mathcal{S}}^{N'}\|_2 + \right.$$
$$\left. \cdots + \|[(\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}^n}^{\boldsymbol{\vartheta},n})^{-1}]_{\bar{\mathcal{S}}}\|_2 \|\widehat{\boldsymbol{\delta}}_{\mathcal{S}}^n\|_2 \right].$$

We now proceed to bound each of the quantities. Let us for now assume that $N' \leq n' \leq n$. As we suppose the design matrix for Lasso solution $\widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^{N'}$ satisfies Assumption 9, by Proposition 7 and Lemma 6, if we choose $\lambda^{n'}$ such that

$$\lambda^{n'} = O(\bar{d}\sqrt{n' \log p}), \tag{35}$$

then there exists some constant C_6 such that

$$\|\widehat{\boldsymbol{\delta}}_{\mathcal{S}}^{n'}\|_2 \leq C_6 \sigma_\epsilon \bar{d}\sqrt{\frac{s^* \log p}{n'}}, \quad \text{for all } N' \leq n' \leq n, \tag{36}$$

with probability converging to 1. We know from (24) that

$$\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}^{n'}}^{\boldsymbol{\vartheta},n'} = \mathbf{M}_{\mathcal{S}^{n'}}^{n'-1}\sigma_\epsilon^2 + (\lambda^{n'})^2 \mathbf{M}_{\mathcal{S}^{n'}}^{n'-1}\widetilde{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}^{n'}})^{(n')}\mathbf{M}_{\mathcal{S}^{n'}}^{n'-1},$$

where

$$\mathbf{M}_{\mathcal{S}^{n'}}^{n'-1} = \left[(\mathbf{X}_{*\mathcal{S}^{n'}}^{n'-1})^T \mathbf{X}_{*\mathcal{S}^{n'}}^{n'-1} \right]^{-1}.$$

Assumption 9 gives us

$$\Lambda_{\max}(\mathbf{M}_{\mathcal{S}}^{N'-1}) \leq \frac{1}{N'c_*} < \infty,$$
$$\Lambda_{\min}(\mathbf{M}_{\mathcal{S}}^{N'-1}) \geq \frac{1}{N'c^*} > 0,$$

for any $\mathcal{S}$ with $|\mathcal{S}| = C_3 s^*$. Therefore, since $|\mathcal{S}^{n'}| \leq C_3 s^*$, by Proposition 7, we can show that for all $N' \leq n' \leq n$, there exist positive constants C_7 and C_8, such that

$$\Lambda_{\max}(\mathbf{M}_{\mathcal{S}^{n'}}^{n'-1}) \leq \frac{C_7}{n'} < \infty, \tag{37}$$
$$\Lambda_{\min}(\mathbf{M}_{\mathcal{S}^{n'}}^{n'-1}) \geq \frac{C_8}{n'} > 0. \tag{38}$$

It is not hard to prove

$$\Lambda_{\min}(\mathbf{M}\mathbf{N}) \geq \Lambda_{\min}(\mathbf{M})\Lambda_{\min}(\mathbf{N})$$

for any positive semidefinite matrices $\mathbf{M}$ and $\mathbf{N}$, so using Weyl's inequality in matrix theory, (23) and (38), we have the following bound,

$$\begin{aligned} \|[(\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}^{n'}}^{\boldsymbol{\vartheta},n'})^{-1}]_{\bar{\mathcal{S}}}\|_2 &\leq \|(\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}^{n'}}^{\boldsymbol{\vartheta},n'})^{-1}\|_2 = \Lambda_{\min}^{-1}(\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}^{n'}}^{\boldsymbol{\vartheta},n'}) \\ &\leq \frac{1}{\Lambda_{\min}(\sigma_\epsilon^2 \mathbf{M}_{\mathcal{S}^{n'}}^{n'-1}) + (\lambda^{n'})^2 \Lambda_{\min}(\widehat{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}^{n'}}^{n'}))\Lambda_{\min}^2(\mathbf{M}_{\mathcal{S}^{n'}}^{n'-1})} \\ &\leq \frac{C_9 n'}{\sigma_\epsilon^2 \bar{d}^2 \log p}, \end{aligned} \tag{39}$$

for some constant C_9. Similarly, by (35), (37), and (23), we can also get

$$
\begin{aligned}
\|\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}^{n'}}^{\boldsymbol{\vartheta},n'}\|_2 &= \Lambda_{\max}(\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}^{n'}}^{\boldsymbol{\vartheta},n'}) \\
&\leq \sigma_\epsilon^2 \Lambda_{\max}(\mathbf{M}_{\mathcal{S}^{n'}}^{n'-1}) + (\lambda^{n'})^2 \Lambda_{\max}(\widehat{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}^{n'}}^{n'})) \Lambda_{\max}^2(\mathbf{M}_{\mathcal{S}^{n'}}^{n'-1}) \\
&\leq C_{10} \frac{\sigma_\epsilon^2 \bar{d}^2 \log p}{n'},
\end{aligned}
$$

for some constant C_{10}. Thus, for the posterior covariance matrix, we have

$$
\begin{aligned}
\|\boldsymbol{\Sigma}_{\bar{\mathcal{S}}}^{\boldsymbol{\vartheta},n}\|_2 &= \Lambda_{\min}^{-1}\left[(\boldsymbol{\Sigma}_{\bar{\mathcal{S}}}^{\boldsymbol{\vartheta},N'-1})^{-1} + [(\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}^{N'}}^{\boldsymbol{\vartheta},N'})^{-1}]_{\bar{\mathcal{S}}} + \cdots + [(\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}^n}^{\boldsymbol{\vartheta},n})^{-1}]_{\bar{\mathcal{S}}}\right] \\
&\leq \frac{1}{\Lambda_{\min}\left[[(\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}^{N'}}^{\boldsymbol{\vartheta},N'})^{-1}]_{\bar{\mathcal{S}}}\right] + \cdots \Lambda_{\min}\left[(\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}^n}^{\boldsymbol{\vartheta},n})^{-1}\right]_{\bar{\mathcal{S}}}} \\
&= \frac{1}{\Lambda_{\max}^{-1}(\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}^{N'}}^{\boldsymbol{\vartheta},N'}) + \cdots \Lambda_{\max}^{-1}(\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}^n}^{\boldsymbol{\vartheta},n})} \\
&\leq \frac{2C_{10}\sigma_\epsilon^2 \bar{d}^2 \log p}{(N'+n)(n-N'+1)} \\
&\leq \frac{C_{11}\sigma_\epsilon^2 \bar{d}^2 \log p}{n^2},
\end{aligned}
\tag{40}
$$

for some constant C_{11}. If we let

$$
\Delta_{\bar{\mathcal{S}}}(N') = \|(\boldsymbol{\Sigma}_{\bar{\mathcal{S}}}^{\boldsymbol{\vartheta},N'-1})^{-1}\|_2 \|\boldsymbol{\delta}_{\bar{\mathcal{S}}}^{N'-1}\|_2,
$$

then combining (36),(39) and (40) gives us the following bound on $\boldsymbol{\delta}_{\bar{\mathcal{S}}}^n$

$$
\begin{aligned}
\|\boldsymbol{\delta}_{\bar{\mathcal{S}}}^n\|_2 &\leq \frac{C_{11}\sigma_\epsilon^2 \bar{d}^2 \log p}{n^2}\left(\Delta_{\bar{\mathcal{S}}}(N') + \sum_{n'=N'}^{n} \frac{C_6 C_9 \sqrt{s^* n'}}{\sigma_\epsilon \bar{d}\sqrt{\log p}}\right) \\
&\leq \frac{C_{12}\sigma_\epsilon \bar{d}\sqrt{s^* \log p}}{\sqrt{n}} + \frac{C_{11}\sigma_\epsilon^2 \bar{d}^2 \log p \Delta_{\bar{\mathcal{S}}}(N')}{n^2},
\end{aligned}
\tag{41}
$$

for some constant C_{12}, which is equivalent to

$$
\|\boldsymbol{\vartheta}_{\bar{\mathcal{S}}}^n - \boldsymbol{\vartheta}_{\bar{\mathcal{S}}}\|_2^2 \leq \frac{C_4 \sigma_\epsilon^2 s^* \bar{d}^2 \log p}{n}
$$

and thus completes the proof. $\blacksquare$

B.3 Proof of Theorem 12

By definition of f_j, $1 \leq j \leq p$, part (1) follows from part (2) of Theorem 10 directly. Now consider part (2). We denote $\widetilde{f}_j^*$ as

$$
\widetilde{f}_j^*(x) = \sum_{k=1}^{d_j} \vartheta_{jk} \psi_{jk}(x), \quad \text{for } 1 \leq j \leq p.
$$

We also have

$$f_j^n(x) = \sum_{k=1}^{d_j} \vartheta_{jk}^n \psi_{jk}(x), \quad \text{for } 1 \leq j \leq p.$$

Since ψ_{jk} is the orthonormal basis, we have

$$\|f_j^n - \widetilde{f}_j^*\|_2^2 \leq \|\boldsymbol{\vartheta}_{j*}^n - \boldsymbol{\vartheta}_{j*}\|_2^2.$$

Also by Assumption 11 and Lemma 8 in Stone (1986), taking $q = 2$, we have

$$\|\widetilde{f}_j^* - f_j\|^2 = O(d_j^{-2q}) = O(d_j^{-4}).$$

Thus by the result of Theorem 10, we have

$$\|f_{\bar{\mathcal{S}}}^n - f_{\bar{\mathcal{S}}}\|^2 \leq \frac{C_4 \sigma_\epsilon^2 s^* \bar{d}^2 \log p}{n} + \frac{C_{13}}{\bar{d}^4}.$$

Note that choosing $\bar{d} = O(n^{1/6})$ and $s^* = O(1)$ would not change the rate in equation (41), so we have the following bound

$$\|f_{\bar{\mathcal{S}}}^n - f_{\bar{\mathcal{S}}}\|_2^2 \leq \frac{C_5 \sigma_\epsilon^2 \log p}{n^{2/3}}.$$

∎

References

Emre Barut and Warren B Powell. Optimal learning for sequential sampling with non-parametric beliefs. *Journal of Global Optimization*, pages 1–27, 2013.

Robert E Bechhofer, Thomas J Santner, and David M Goldsman. *Design and analysis of experiments for statistical selection, screening, and multiple comparisons*. Wiley New York, 1995.

Chun-hung Chen. *Stochastic simulation optimization: an optimal computing budget allocation*, volume 1. World Scientific, 2010.

Xi Chen, Qihang Lin, and Javier Pena. Optimal regularized dual averaging methods for stochastic optimization. In *NIPS*, volume 25, pages 404–412, 2012.

Yilun Chen and Alfred O Hero. Recursive group Lasso. *Signal Processing, IEEE Transactions on*, 60(8):3978–3987, 2012.

Stephen E Chick, Koichiro Inoue, Koichiro Inoue, and Koichiro Inoue. New two-stage and sequential procedures for selecting the best simulated system. *Operations Research*, 49 (5):732–743, 2001.

John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *The Journal of Machine Learning Research*, 10:2899–2934, 2009.

Jianqing Fan, Yang Feng, and Rui Song. Nonparametric independence screening in sparse ultra-high-dimensional additive models. *Journal of the American Statistical Association*, 106(494), 2011.

Peter I Frazier, Warren B Powell, and Savas Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5): 2410–2439, 2008.

Peter I Frazier, Warren B Powell, and Savas Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS journal on Computing*, 21(4):599–613, 2009.

Jerome H Friedman and Werner Stuetzle. Projection pursuit regression. *Journal of the American statistical Association*, 76(376):817–823, 1981.

Pierre Garrigues and Laurent El Ghaoui. An homotopy algorithm for the Lasso with online observations. In *NIPS*, pages 489–496, 2008.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2003.

Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic algorithmic framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012.

Peter Glynn and Sandeep Juneja. A large deviations perspective on ordinal optimization. In *Simulation Conference, 2004. Proceedings of the 2004 Winter*, volume 1. IEEE, 2004.

Chong Gu. *Smoothing Spline ANOVA Models*. Springer, New York, 2002.

Benjamin Guedj and Pierre Alquier. Pac-bayesian estimation and prediction in sparse additive models. *Electronic Journal of Statistics*, 7:264–291, 2013.

Shanti S Gupta and Klaus J Miescke. Bayesian look ahead one-stage sampling allocations for selection of the best population. *Journal of statistical planning and inference*, 54(2): 229–244, 1996.

Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*, volume 43. CRC Press, 1990.

John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10(777-801):65, 2009.

Qihang Lin, Xi Chen, and Javier Pena. A sparsity preserving stochastic gradient method for composite optimization. *Manuscript, Carnegie Mellon University, PA*, 15213, 2011.

Yi Lin and Hao Helen Zhang. Component selection and smoothing in multivariate non-parametric regression. *The Annals of Statistics*, 34(5):2272–2297, 2006.

Martijn RK Mes, Warren B Powell, and Peter I Frazier. Hierarchical knowledge gradient for sequential sampling. *The Journal of Machine Learning Research*, 12:2931–2974, 2011.

Sahand Negahban, Bin Yu, Martin J Wainwright, and Pradeep K Ravikumar. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

Diana M Negoescu, Peter I Frazier, and Warren B Powell. The knowledge-gradient algorithm for sequencing experiments in drug discovery. *INFORMS Journal on Computing*, 23(3): 346–363, 2011.

Warren B Powell and Ilya O Ryzhov. *Optimal learning*. John Wiley and Sons, Hoboken, NJ, 2012.

Howard Raiffa and Robert Schlaifer. *Applied Statistical Decision Theory*. The M.I.T. Press, 1968.

Pradeep Ravikumar, John Lafferty, Han Liu, and Larry Wasserman. Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):1009–1030, 2009.

Kristofer Reyes, Si Chen, Yan Li, and Warren B Powell. Quantifying the experimental choices in ensemble averaging and extrapolated estimation in the context of optimal learning and materials design. In *Supplemental UE: TMS 2015 Conference Proceedings*, 2014.

Larry Schumaker. *Spline Functions: Basic Theory*. Wiley, New York, 1981.

James C Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. Wiley. com, 2005.

Charles J. Stone. Additive regression and other nonparametric models. *The Annals of Statistics*, 13(2):689–705, 06 1985. doi: 10.1214/aos/1176349548.

Charles J Stone. The dimensionality reduction principle for generalized additive models. *The Annals of Statistics*, pages 590–606, 1986.

Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

Grace Wahba, Yuedong Wang, Chong Gu, Ronald Klein, and Barbara Klein. Smoothing spline anova for exponential families, with application to the wisconsin epidemiological study of diabetic retinopathy. *The Annals of Statistics*, pages 1865–1895, 1995.

Fengrong Wei and Jian Huang. Consistent group selection in high-dimensional linear regression. *Bernoulli: official journal of the Bernoulli Society for Mathematical Statistics and Probability*, 16(4):1369, 2010.

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(2543-2596):4, 2010.

Hao Helen Zhang, Grace Wahba, Yi Lin, Meta Voelker, Michael Ferris, Ronald Klein, and Barbara Klein. Variable selection and model building via likelihood basis pursuit. *Journal of the American Statistical Association*, 99(467):659–672, 2004.

Peng Zhao, Guilherme Rocha, and Bin Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, pages 3468–3497, 2009.