Journal of Machine Learning Research () Submitted ; Published

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. Our problem setting, motivated by the experimental sciences, arises where we have to choose which experiment to run next. Here the experiments are time-consuming and expensive. 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 a B-spline basis of finite order and approximates the nonparametric additive model and functional ANOVA model. Theoretically, we provide the estimation error bounds of the posterior mean estimate and the functional estimate. Controlled experiments on both synthetic and real data for identifying the accessibility of an RNA molecule show that the algorithm efficiently learns the correct set of nonzero parameters. Also it outperforms several other policies.

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. Specifically, we are maximizing an unknown function $\mu_x : x \in \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X}$ is a finite set with $M < \infty$ alternatives. We have the ability to sequentially choose a set of measurements to estimate. Our goal is to select the best alternative when the finite budget is exhausted. The experiments are time-consuming and expensive. Also, we assume that the objective function μ cannot be written in closed form and does not have easily available derivatives. 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.

Our work is motivated by learning the accessibility patterns of an RNA molecule known as the *Tetrahymena Group I intron* (gI intron), which has been widely used as an RNA folding model (Cech et al., 1981). Experimentally, such accessibility patterns can be inferred from fluorescence measurements obtained from the iRS3 by using various complementary probes designed a priori to target a region within the gI intron (Sowa et al., 2014). Here the dimension of the problem is equal to the length of the RNA molecule (~ 400). The objective function μ_x captures the amount of accessibility (or fluorescence) of each targeted probe sequence. We use a thermo-kinetic model (Reyes et al., 2014; Li et al., 2015) which represents μ_x as a linear function of the coefficients representing the accessibility of each nucleotide. However, not all sites are accessible, so we believe our model will be relatively sparse. A more detailed description of the thermo-kinetic model and more algorithms are included in our paper Li et al. (2015).

For these applications, we propose the following sparse linear model:

$$\mu_x = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_m x_m, \tag{1}$$

where $\boldsymbol{x} := [x_1, \ldots, x_m]^T$ is the design vector, and $\boldsymbol{\alpha} := [\alpha_1, \ldots, \alpha_m]^T$ is the linear coefficient vector. Our problem setting assumes that m can be several hundred or in the thousands, but only a small portion of the components of $\boldsymbol{\alpha}$ are nonzero. More generally, we could consider group sparsity structure; that is, the coefficients can be divided into several known groups, and those within each group are either all zero or all nonzero.

The early R&S literature assumes a lookup table belief model (Frazier et al., 2008, 2009). Recent research has used a linear belief model, making it possible to represent many thousands or even millions of alternatives using a low-dimensional model (Negoescu et al., 2011). However, these problems typically involve learning models characterized by low-dimensional parameter vectors (for example, up to a few dozen parameters). Also, there is no sparsity structure assumption on the coefficient $\boldsymbol{\alpha}$.

In our work we consider problems where the coefficient vector $\boldsymbol{\alpha}$ can have hundreds or even thousands of components. However, we assume that most of the components of $\boldsymbol{\alpha}$ are zero. Sparsity is a feature present in a plethora of natural as well as man-made systems. In such optimal learning problems, we are confronted with two challenges. First, we need to design an efficient experimental policy to search for the best alternative to maximize μ_x based on the belief model. Second, learning the underlying sparsity structure will produce a more parsimonious model which will streamline the experimental work and simplify the ultimate design problem.

This paper tackles the two challenges by first deriving a *knowledge gradient* policy for sparse linear models (KGSpLin). The knowledge gradient (KG), first proposed by Frazier et al. (2008), is a learning policy that maximizes the marginal value of information from each expensive experiment. In the sparse belief setting, we introduce a random indicator variable $\boldsymbol{\zeta}$ and maintain a Beta-Bernoulli conjugate prior to model our belief about which variables should be included in or dropped from the model. Based on this, we show later in the paper that our KGSpLin algorithm can be approximately computed by a weighted sum over the KG values of all possible low-dimensional beliefs. KGSpLin then naturally generalizes to the KG for sparse additive models (KGSpAM). Here $\mu_x = \sum_{j=1}^{p} f_j(x_j)$. The f_js are one-dimensional scalar functions, many of which are zero. After approximating the individual f_j with B-splines of finite order, this belief model also results in the same form

as (1). Additionally, in the broader class of models known as multivariate splines functional ANOVA models (Wahba, 1990; Wahba et al., 1995; Gu, 2002), tensor product B-splines can be adopted. KGSpAM can also be used in this model.

Second, for the learning procedure, our algorithm adopts the frequentist homotopy recursive approach for group Lasso with $\ell_{1,\infty}$ penalty (Chen and Hero, 2012). Then we directly use these sequential estimates to update the Bayesian model, not only learning the values of the linear coefficients, but also the probabilities of whether each feature is in or not. In a nutshell, our work is a novel and unique hybrid of Bayesian R&S with the frequentist learning approach.

Theoretically, we prove the estimation consistency. That is, the estimate converges to the truth when given enough measurements, under some appropriate assumptions. Specifically, we show that the mean of the posterior coefficient estimate converges to the truth at the same rate as that of the Lasso estimates. Besides, we also show that the recovered sparsity set is *rate consistent*. That is, the cardinality of the recovered sparsity pattern can be bounded by the true sparse cardinality with a constant factor.

The remainder of the paper is organized as follows. In Section 2 we give a brief overview of the relevant literature. Section 3 formulates the R&S model in a Bayesian setting and establishes the notation used in this paper. It also highlights the knowledge gradient using both a lookup table and a linear, non-sparse belief model. Section 4 is devoted to a detailed description of the Bayesian sparse linear model and the KGSpLin policy. Section 5 generalizes the algorithm to a nonparametric sparse additive belief model (KGSpAM) and also SS-ANOVA. Theoretical results are presented in Section 6, which shows the estimation error bounds and the recovered sparsity set bounds for both the coefficient estimates and the functional estimates. In Section 7, we test the algorithms in a series of controlled experiments, including the experiments on an actual dataset drawn from identifying the accessibility of the RNA molecule gI intron.

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 experimental decisions. These problems have been studied in different communities, which refer to the problems under names such as: Bayesian optimization (Brochu et al., 2010), experimental design (Robbins, 1985), multi-armed bandits (Auer et al., 2002), optimal learning (Powell and Ryzhov, 2012), and reinforcement learning (Sutton and Barto, 1998).

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 literature, 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 indifference-zone formulation (Bechhofer et al., 1995); the optimal computing budget allocation, or OCBA (Chen, 2010; Chen et al., 2012b); 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) introduce the idea of selecting an alternative based on the marginal value of information. Frazier et al. (2008) extend 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 shows 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 and Inoue (2001), where samples are allocated to maximize an approximation of the expected value of information. Negoescu et al. (2011) introduce the use of a parametric belief model, making it possible to solve problems with thousands of alternatives. For nonparametric beliefs, Mes et al. (2011) propose 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) estimate the belief function using kernel regression and aggregation of kernels.

However, all the methods above are restricted to problems of moderate dimension, typically up to about 10. There are applications with hundreds or even thousands of features. In such settings, the above methods become inefficient or even infeasible, since the efficiency often depends exponentially on the dimension of the domain. This "curse of dimensionality" is notoriously hard and is regarded as one of the holy grails of the field. To advance the state of the art, there have been several other efforts to scale different algorithms to deal with high-dimensional models. For linear bandits, Carpentier and Munos (2012) propose a compressed sensing strategy to attack problems with a high degree of sparsity. Chen et al. (2012a) use a two stage strategy for optimization and variable selection of high-dimensional Gaussian processes. Djolonga et al. (2013) propose an algorithm, leveraging low-rank matrix recovery techniques to learn the underlying low-dimensional space and applying Gaussian process upper confidence sampling for optimization of the function. Wang et al. (2013) adopt random embeddings to optimize high-dimensional functions with low intrinsic dimensionality.

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 values, and has the opportunity to test 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 the alternative identified as best, compared with the optimal decision given perfect information. 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., 2012b; 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 have 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). 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 studies them 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 the Bayesian R&S and the KG policy with a lookup table belief model and a linear, non-sparse belief model. We start by introducing some 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

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

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 eigenvalue and the smallest eigenvalue of $\mathbf{M}$. For a summary of most symbols we use, please refer to Table 3 in Appendix A.

3.1 The Bayesian Model for Ranking and Selection

We first review the Bayesian R&S with both a lookup table belief model and a non-sparse, linear belief model. The unknown function is denoted by $\mu_x : x \in \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X}$ is a finite set with M alternatives. We have a finite 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. 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.

For a lookup table belief model (Frazier et al., 2008, 2009), we assume $\boldsymbol{\mu}$ follows a multivariate normal distribution:

$$\boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma}). \tag{2}$$

Now suppose we have a sequence of measurement decisions, $x^0, x^1, \ldots, x^{N-1}$ to learn about these alternatives, where $x^i \in \mathcal{X}$, 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 ϵ_x^{n+1} is the random measurement noise and $\epsilon_x^{n+1} \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Here we assume σ_ϵ 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, x^n is only allowed to depend on the outcomes of the sampling decisions $x^0, x^1, \ldots, x^{n-1}$. Throughout the paper, the random variable indexed by n in the superscript is measurable with respect to the filtration $\mathcal{F}^n$, which is defined as the σ-algebra generated by all of the observations up to time n, $\{(x^0, y^1_{x^0}), (x^1, y^2_{x^1}), \ldots, (x^{n-1}, y^n_{x^{n-1}})\}$. 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

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

We define Π to be the set of all possible policies satisfying our sequential requirement; that is, $\Pi := \{[x^0, \ldots, x^{N-1}] : x^n \in \mathcal{F}^n\}$. Let $\mathbb{E}^\pi$ indicate the expectation 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 the expected reward, which can be written as

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

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

$$\begin{aligned} \boldsymbol{\theta}^{n+1} &= \boldsymbol{\theta}^n + \frac{y_x^{n+1} - \theta_x^n}{\sigma_\epsilon^2 + \Sigma_{xx}^n} \boldsymbol{\Sigma}^n \boldsymbol{e}_x, \\ \boldsymbol{\Sigma}^{n+1} &= \boldsymbol{\Sigma}^n - \frac{\boldsymbol{\Sigma}^n \boldsymbol{e}_x \boldsymbol{e}_x^T \boldsymbol{\Sigma}^n}{\sigma_\epsilon^2 + \Sigma_{xx}^n}, \end{aligned} \tag{3}$$

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

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 μ 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.

Based on this idea, Negoescu et al. (2011) further extend this nonparametric, lookup table belief to a parametric belief 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, $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\alpha}$. Here $\boldsymbol{\alpha} \in \mathbb{R}^m$ is the coefficient vector, and $\mathbf{X} \in \mathbb{R}^{M\times m}$ is the design matrix, where each row is a feature vector corresponding to a particular experiment (or an alternative). (Notice that we use $\mathbf{X}$ to denote the design matrix including all the possible finite alternatives. Later in the paper the design matrix $\mathbf{X}^n$ includes all the sequential decisions up to time n.) If we assume $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\vartheta}, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}})$, this induces a normal distribution on $\boldsymbol{\mu}$ via the linear transformation,

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

At time n, if we measure alternative $x^n = x$, we can update $\boldsymbol{\vartheta}^{n+1}$ and $\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n+1}$ recursively via recursive least squares (see Powell and Ryzhov, 2012, P.187),

$$\begin{aligned}
\boldsymbol{\vartheta}^{n+1} &= \boldsymbol{\vartheta}^n + \frac{\widehat{\epsilon}^{n+1}}{\gamma^n}\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}\boldsymbol{x}^n, \\
\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n+1} &= \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n} - \frac{1}{\gamma^n}(\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}\boldsymbol{x}^n(\boldsymbol{x}^n)^T\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}),
\end{aligned}$$

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

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. For example, if we have a problem with thousands of alternatives (which easily happens if $\boldsymbol{x}$ is multidimensional), then $\boldsymbol{\Sigma}^n$ would have thousands of rows and columns, which can be very cumbersome. By contrast, the linear model allows us to maintain the parameter covariance matrix $\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}$, which is dimensioned by the size of the parameter vector $\boldsymbol{\vartheta}$.

3.2 The Knowledge Gradient Policy

In this section, we briefly review the knowledge gradient (KG) for both the lookup table belief model and the linear, non-sparse belief model. KG is a fully sequential sampling policy for learning the values of the alternatives. Each time it chooses the alternative that can maximize the expected incremental value. If we represent the state of knowledge at time n as: $S^n := (\boldsymbol{\theta}^n, \boldsymbol{\Sigma}^n)$, then the corresponding value of being in state S^n at time n is

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

The knowledge gradient policy is to choose the alternative that can maximize the KG value, which is defined as:

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

and

$$x^{\mathrm{KG},n} = \underset{x' \in \mathcal{X}}{\operatorname{argmax}}\, v_{x'}^{\mathrm{KG},n}.$$

Here the calculation of the expectation can generally be computationally intractable. However, for the lookup table and linear belief models described in Section 3.1, Frazier et al. (2009) propose an algorithm to exactly compute the KG values. We briefly describe the algorithm in the following.

We can rearrange equation (3) as the time n conditional distribution of $\boldsymbol{\theta}^{n+1}$, namely,

$$\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n + \widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, x^n) Z^{n+1}, \tag{4}$$

where

$$\begin{aligned} \widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, x) &= \frac{\boldsymbol{\Sigma}^n \boldsymbol{e}_x}{\sqrt{\sigma_\epsilon^2 + \Sigma_{xx}^n}}, \\ Z^{n+1} &= \frac{(y_x^{n+1} - \theta_x^n)}{\sqrt{\mathrm{Var}[y_x^{n+1} - \theta_x^n | \mathcal{F}^n]}}. \end{aligned} \tag{5}$$

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 (4) into the KG formula,

$$\begin{aligned} v_x^{\mathrm{KG},n} &= \mathbb{E}(\max_{x' \in \mathcal{X}} \theta_{x'}^n + \widetilde{\boldsymbol{\sigma}}_{x'}(\boldsymbol{\Sigma}^n, x^n) Z^{n+1} | S^n, x^n = x) - \max_{x' \in \mathcal{X}} \theta_{x'}^n \\ &= h(\boldsymbol{\theta}^n, \widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, x)), \end{aligned} \tag{6}$$

where $\widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, x)$ is a vector-valued function defined in (5), and $\widetilde{\sigma}_{x'}(\boldsymbol{\Sigma}^n, x^n)$ indicates the component $\boldsymbol{e}_{x'}^T \widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, x^n)$ of the vector $\widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, x^n)$. Here $h(\boldsymbol{a}, \boldsymbol{b}) = \mathbb{E}[\max_i a_i + b_i Z] - \max_i a_i$ is a generic function of any vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ of the same dimension, and Z is a standard normal random variable.

The expectation can be computed as the point-wise maximum of the 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 are in nondecreasing order, and ties in b_is are broken by removing the pair (a_i, b_i) when $b_i = b_{i+1}$ and $a_i \leq a_{i+1}$. Next, all pairs (a_i, b_i) that are dominated by the other pairs, that is, $a_i + b_i Z \leq \max_{j \neq i} a_j + b_j Z$ for all values of Z, are removed. Thus the knowledge gradient can be computed using

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

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 $\widetilde{M}$.

For the knowledge gradient with linear models (KGLin), we can substitute $(\boldsymbol{\theta}^n, \boldsymbol{\Sigma}^n)$ with $(\mathbf{X}\boldsymbol{\vartheta}, \mathbf{X}\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}}\mathbf{X}^T)$ into (6) and compute the KGLin value. In addition, we never need to compute the full matrix $\mathbf{X}\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}}\mathbf{X}^T$. We only need to compute a row of this matrix.

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

In this section, we derive an extension of KG policy with a high-dimensional sparse linear model called the knowledge gradient with sparse linear models (KGSpLin). We begin by establishing the Bayesian framework. Then we derive the new KG policy and describe the algorithm which combines the sparse estimation with group Lasso penalty.

Following the notation for linear models, we have $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \mathbb{R}^m, \boldsymbol{\mu} \in \mathbb{R}^M$ are random variables, and $\mathbf{X} \in \mathbb{R}^{M\times m}$ is the design matrix. Our problem setting is that m can become relatively large, and $\boldsymbol{\alpha}$ is sparse in the sense that only a few components are nonzero. However, unlike the sparsity assumption in classical frequentist statistics, we assume the sparsity structure is random; that is, the indicator variable of which one is selected or not is a random vector. More generally, we consider a group-wise sparse pattern. We now assume there exists some known group structure in $\boldsymbol{\alpha}$. Let $\{\mathcal{G}_j\}_{j=1}^p$ be 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 $\boldsymbol{\alpha}_{\mathcal{G}_j}$ is a subvector of $\boldsymbol{\alpha}$ indexed by $\mathcal{G}_j$. Let $\boldsymbol{\zeta} = [\zeta_1, \ldots, \zeta_p]^T \in \mathbb{R}^p$ be the 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,

$$\boldsymbol{\alpha}|\boldsymbol{\zeta} \sim \mathcal{N}(\boldsymbol{\vartheta}, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}}). \tag{7}$$

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

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

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 covariance $\boldsymbol{\Sigma}^{\boldsymbol{\theta}}_{\mathcal{S}'}$. 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. Therefore, through all the updatings, we just need to maintain the mean and covariance matrix on $\boldsymbol{\zeta} = \mathbf{1}$.

Now we briefly recall and summarize the random variables in this Bayesian model. The underlying unknown value of alternative x is denoted by μ_x and parametrized by $\boldsymbol{\alpha}$. Here $\boldsymbol{\alpha}$ follows a "mixture" normal distribution by (7) and ζ_j follows a Bernoulli distribution. 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. One may think of $\boldsymbol{\zeta}$ and $\boldsymbol{\alpha}$ as fixed and of $\boldsymbol{\zeta}^n$ and $\boldsymbol{\vartheta}_{\mathcal{S}}^n$ as converging toward $\boldsymbol{\zeta}$ and $\boldsymbol{\alpha}$, while some norm of the precision matrix $(\boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},n})^{-1}$ is 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.

4.1 Knowledge Gradient Policy for Sparse Linear Models

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 $j, j' = 1, \ldots, p$,

$$\boldsymbol{\alpha}|\boldsymbol{\zeta}^n = \mathbf{1} \sim \mathcal{N}(\boldsymbol{\vartheta}^n, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}), \tag{8}$$

$$\zeta_j^n|p_j^n \sim \text{Bernoulli}(p_j^n), \tag{9}$$

$$\zeta_j^n \perp \zeta_{j'}^n, \quad \text{for} \quad j \neq j', \tag{10}$$

$$p_j^n|\xi_j^n, \eta_j^n \sim \text{Beta}(\xi_j^n, \eta_j^n), . \tag{11}$$

where p_j^n is the probability of the jth group of features being in the model, and (ξ_j^n, η_j^n) are the shape parameters for the Beta distribution of p_j^n. For different groups j and j', we assume that ζ_j^n and $\zeta_{j'}^n$ are independent. At time n, the prior $\boldsymbol{\zeta}^n$ is a discrete random variable. Let $\boldsymbol{\zeta}^{n,1}, \ldots, \boldsymbol{\zeta}^{n,N_{\boldsymbol{\zeta}}}$ be all the possible realizations of $\boldsymbol{\zeta}^n$, and $\mathbb{P}(\boldsymbol{\zeta}^n = \boldsymbol{\zeta}^{n,k}) = p^{n,k}, k = 1, \ldots, N_{\boldsymbol{\zeta}}$.

For the following computation of the expectation in KGSpLin, we need to make two approximations. First, we need to approximate the distribution of $(\boldsymbol{\zeta}^{n+1}, \boldsymbol{p}^{n+1})$ by that of $(\boldsymbol{\zeta}^n, \boldsymbol{p}^n)$. This is because the change of the sparsity belief depends on the next observation and the Lasso algorithm, and thus can be very complicated to model. Therefore, by the Law of Total Expectation, the KGSpLin value can be computed by:

$$\begin{aligned}
v_x^{\text{KG},n} &= \mathbb{E}(V^{n+1}(S^{n+1}(x)) - V^n(S^n)|S^n, x^n = x) \\
&= \mathbb{E}_{\boldsymbol{\alpha},\epsilon,\boldsymbol{\zeta}^{n+1},\boldsymbol{p}^{n+1}}(\max_{x' \in \mathcal{X}} \theta_{x'}^{n+1}|S^n, x^n = x) - \max_{x' \in \mathcal{X}} \theta_{x'}^n \\
&\approx \mathbb{E}_{\boldsymbol{p}^n}\mathbb{E}_{\boldsymbol{\zeta}^n|\boldsymbol{p}^n}\mathbb{E}_{\boldsymbol{\alpha},\epsilon|\boldsymbol{\zeta}^n,\boldsymbol{p}^n}(\max_{x' \in \mathcal{X}} \theta_{x'}^{n+1}|S^n, x^n = x, \boldsymbol{\zeta}^n, \boldsymbol{p}^n) - \max_{x' \in \mathcal{X}} \theta_{x'}^n \\
&= \sum_{k=1}^{N_{\boldsymbol{\zeta}}} \mathbb{E}_{\boldsymbol{p}^n}(p^{n,k}) h(\boldsymbol{a}^{n,k}, \boldsymbol{b}^{n,k}) \\
&= \sum_{k=1}^{N_{\boldsymbol{\zeta}}} \prod_{\{j:\zeta_j^{n,k}=1\}} \frac{\xi_j^n}{\xi_j^n + \eta_j^n} \prod_{\{j:\zeta_j^{n,k}=0\}} \frac{\eta_j^n}{\xi_j^n + \eta_j^n} h(\boldsymbol{a}^{n,k}, \boldsymbol{b}^{n,k})
\end{aligned} \tag{12}$$

where

$$\begin{aligned}
\boldsymbol{a}^{n,k} &= \mathbf{X}_{*\boldsymbol{\zeta}^{n,k}} \boldsymbol{\vartheta}_{\boldsymbol{\zeta}^{n,k}}^n, \\
\boldsymbol{b}^{n,k} &= \widetilde{\boldsymbol{\sigma}}(\mathbf{X}_{*\boldsymbol{\zeta}^{n,k}} \boldsymbol{\Sigma}_{\boldsymbol{\zeta}^{n,k}}^{n,\boldsymbol{\vartheta}} (\mathbf{X}_{*\boldsymbol{\zeta}^{n,k}})^T, x),
\end{aligned}$$

and

$$h(\boldsymbol{a}, \boldsymbol{b}) \quad := \quad \mathbb{E}[\max_i a_i + b_i Z] - \max_i a_i,$$

is the function defined in Section 3.2 and thus can be computed.

The second approximation is required to assist with computing the expectation over $\boldsymbol{\zeta}$. Note that conditioning on each sample realization of $\boldsymbol{\zeta}^n$, the KGSpLin calculation is identical with that of KGLin. Therefore we have shown that the KGSpLin value is a weighted summation over all the possible sample realizations of $\boldsymbol{\zeta}^n$. The weights $\mathbb{E}_{p^n}(p^{n,k})$ are computed by the independent Beta distributions on all the p_j^n's. Additionally, if $N_{\boldsymbol{\zeta}}$ takes its largest possible value, that is $N_{\boldsymbol{\zeta}} = 2^p$, we can re-sort the weights and approximate the knowledge gradient value by only computing the ones with the highest probabilities. In Section 7, we can see that that we do not lose much by making these approximations. The KGSpLin value still serves as a reasonable sampling criterion based on the value of information.

4.2 Bayesian Update

At time n we have the Bayesian model described in (8)-(11). Parallel with that, we use Lasso as a "solver" to generate estimates of linear coefficients as well as the sparsity pattern. The $\ell_{1,\infty}$ group Lasso estimator after n observations is given by

$$\widehat{\boldsymbol{\beta}}^n = \underset{\boldsymbol{\beta} \in \mathbb{R}^m}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} [(\boldsymbol{x}^{i-1})^T \boldsymbol{\beta} - y^i]^2 + \lambda^n \|\boldsymbol{\beta}\|_{1,\infty}, \tag{13}$$

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 $\|\boldsymbol{\beta}\|_{1,\infty} := \sum_{j=1}^{p} \|\boldsymbol{\beta}_{\mathcal{G}_j}\|_\infty$. When each group contains only one coefficient, the regularization takes the ℓ_1 norm. Then 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 when we get a new measurement, we recursively solve the Lasso problem based on the homotopy algorithm proposed in Chen and Hero (2012). This algorithm is an exact update of the $\ell_{1,\infty}$ group Lasso solutions when one additional observation is achieved. (For the recursive homotopy algorithm for Lasso, one can refer to Garrigues and El Ghaoui (2008).) Each update minimizes a convex but nondifferentiable function optimization problem. This algorithm has been demonstrated to have lower implementation complexity than the direct group Lasso solvers. It also fits the recursive setting in optimal learning. Refer to Appendix B for a more detailed description of this algorithm.

At time n, when KGSpLin gives us the current measurement decision x^n, we sample the value of μ at x^n and get a noisy measurement y^{n+1}. If we let $\widehat{\boldsymbol{\vartheta}}^n$ be the Lasso solution at time n, then we use this new sample (x^n, y^{n+1}) to update the Lasso estimate from $\widehat{\boldsymbol{\vartheta}}^n$ to $\widehat{\boldsymbol{\vartheta}}^{n+1}$. Next, we need to sample a covariance matrix $\widehat{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},n}$ corresponding to this Lasso estimate to represent our uncertainty of the Lasso estimate. This can be Monte Carlo

simulated from the first order optimality condition of the optimization problem (13), for which we present the details later in this Section. Let $\widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^{n+1}$ be the nonzero part of $\widehat{\boldsymbol{\vartheta}}^{n+1}$. Once we have the updated Lasso estimates of $\widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^{n+1}$ and $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1}$, we can use the following heuristic updating scheme for a Beta-Bernoulli model and a Gaussian-Gaussian model. Let $\mathcal{P}^n := \{j : \widehat{\boldsymbol{\vartheta}}_{\mathcal{G}_j}^n \neq 0\}$. The updating equations are given by:

$$\boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1} = \left[(\boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},n})^{-1} + (\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1})^{-1}\right]^{-1}, \tag{14}$$

$$\boldsymbol{\vartheta}_{\mathcal{S}}^{n+1} = \boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1}\left[(\boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta},n})^{-1}\boldsymbol{\vartheta}_{\mathcal{S}}^{n} + (\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1})^{-1}\widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^{n+1}\right], \tag{15}$$

$$\xi_j^{n+1} = \xi_j^n + 1, \eta_j^{n+1} = \eta_j^n, \quad \text{for} \quad j \in \mathcal{P}^{n+1}, \tag{16}$$

$$\xi_j^{n+1} = \xi_j^n, \eta_j^{n+1} = \eta_j^n + 1, \quad \text{for} \quad j \notin \mathcal{P}^{n+1}. \tag{17}$$

Here (14)(15) are the updating equations for a Gaussian-Gaussian model, and (16)(17) are the updating equations for a Beta-Bernoulli model. The frequencies of "in" and "out" are essentially denoted by (ξ_j, η_j) and updated recursively via Lasso estimates. In order to better clarify this Bayesian model and the updating scheme, we illustrate the updating (14)-(17) in Figure 1.

Now we present the technique to approximately sample the covariance matrix $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1}$ from the first order optimality condition in problem (13). We begin with a series of set definitions. Figure 2 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

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

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 (13) can also be written as

$$\boldsymbol{\beta}^n = \underset{\boldsymbol{\beta} \in \mathbb{R}^m}{\operatorname{argmin}} \frac{1}{2}\boldsymbol{\beta}^T \mathbf{R}^{n-1}\boldsymbol{\beta} - \boldsymbol{\beta}^T \boldsymbol{r}^n + \lambda^n \|\boldsymbol{\beta}\|_{1,\infty}, \tag{18}$$

where $\mathbf{R}^{n-1} = \sum_{i=1}^n \boldsymbol{x}^{i-1}(\boldsymbol{x}^{i-1})^T$, $\boldsymbol{r}^n = \sum_{i=1}^n \boldsymbol{x}^{i-1} y^i$. 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 (18) are given by

$$\mathbf{R}^{n-1}\boldsymbol{\beta} - \boldsymbol{r}^n + \lambda^n \boldsymbol{z} = \mathbf{0}, \quad \boldsymbol{z} \in \partial\|\boldsymbol{\beta}\|_{1,\infty}. \tag{19}$$

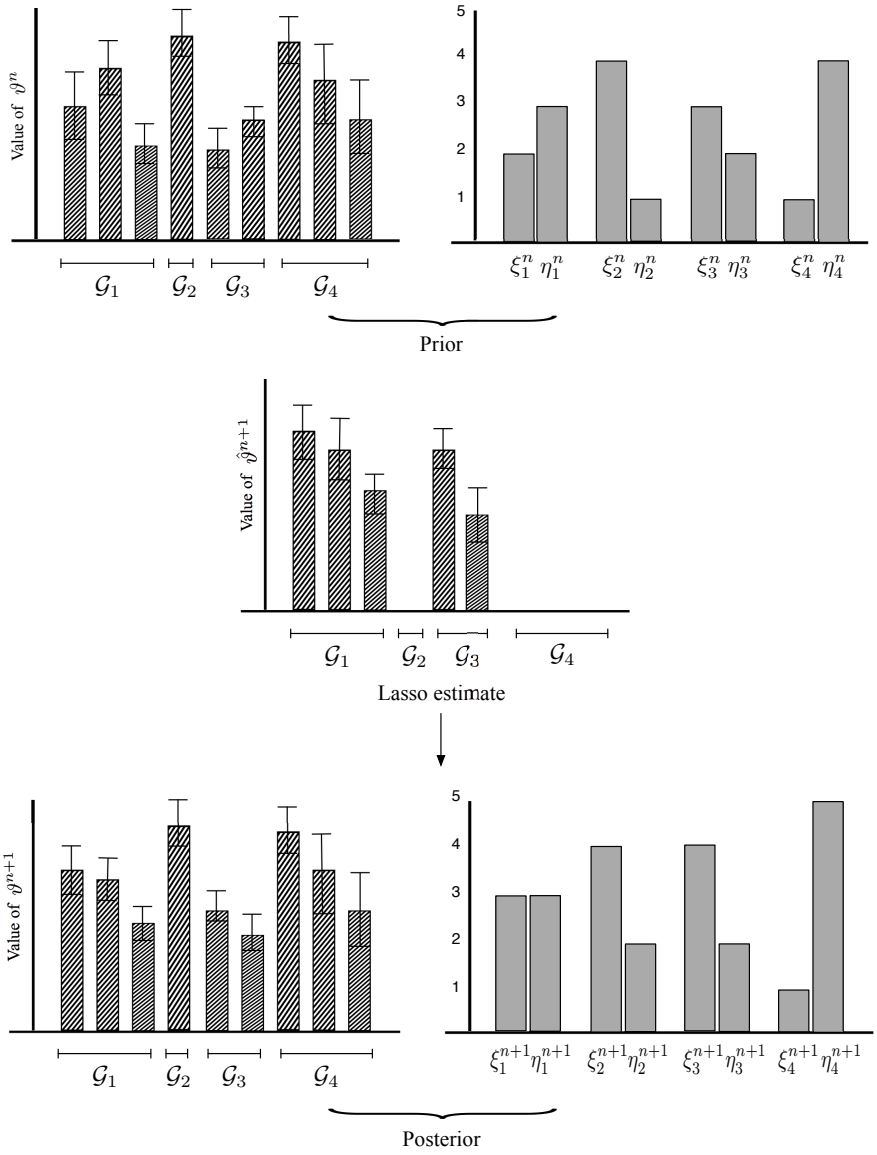


Figure 1: Illustration of the Bayesian model and the heuristic updating scheme for a Beta-Bernoulli model and a Gaussian-Gaussian model. A nine-element coefficient vector $\boldsymbol{\alpha}$ are divided into four groups. The prior at time n includes the mean estimate $\boldsymbol{\vartheta}^n$ with bar plots representing the standard deviations and the frequencies estimates (ξ_j^n, η_j^n) of "in" and "out". Combining with the Lasso estimate $\widehat{\vartheta}^{n+1}$ results in the posterior. On active sets $\mathcal{G}_1$ and $\mathcal{G}_3$, the coefficients are updated according to (14)-(15), and ξ_j^n are added by one. On inactive sets $\mathcal{G}_2$ and $\mathcal{G}_4$, the coefficients remain unchanged, and η_j^n are added by one.

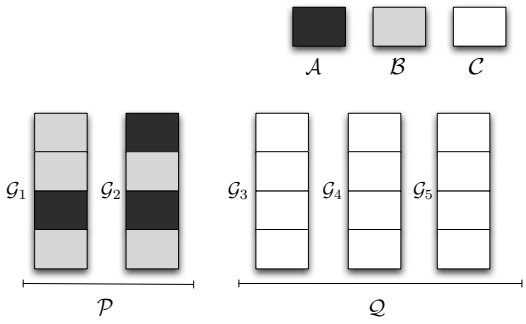


Figure 2: Illustration of the partitioning of a 20 element coefficient vector $\boldsymbol{\beta}$ into five groups of four indices. The sets $\mathcal{P}$ and $\mathcal{Q}$ contains the active groups and the inactive groups, respectively. Within each of the two active groups the coefficients with maximal absolute values are denoted by the black color.

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

$$\begin{aligned} \|\boldsymbol{z}_{\mathcal{A}_j}\|_1 &= 1, \quad j \in \mathcal{P}, && (20)\\ \operatorname{sgn}(\boldsymbol{z}_{\mathcal{A}_j}) &= \operatorname{sgn}(\boldsymbol{\beta}_{\mathcal{A}_j}), \quad j \in \mathcal{P}, && (21)\\ \boldsymbol{z}_{\mathcal{B}} &= \mathbf{0}, && (22)\\ \|\boldsymbol{z}_{\mathcal{C}_j}\|_1 &\leq 1, \quad j \in \mathcal{Q}, \end{aligned}$$

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 $\boldsymbol{\beta}_{\mathcal{C}} = \mathbf{0}$, (19) implies that

$$\begin{aligned} \mathbf{R}^{n-1}_{\mathcal{S}}\boldsymbol{\beta}_{\mathcal{S}} - \boldsymbol{r}^n_{\mathcal{S}} + \lambda^n \boldsymbol{z}_{\mathcal{S}} = \mathbf{0}, \\ \mathbf{R}^{n-1}_{\mathcal{CS}}\boldsymbol{\beta}_{\mathcal{S}} - \boldsymbol{r}^n_{\mathcal{C}} + \lambda^n \boldsymbol{z}_{\mathcal{C}} = \mathbf{0}. \end{aligned} \tag{23}$$

If $\mathbf{R}^{n-1}_{\mathcal{S}}$ is invertible, then the solution is unique, and we can rewrite (23) as

$$\boldsymbol{\beta}_{\mathcal{S}} = (\mathbf{R}^{n-1}_{\mathcal{S}})^{-1}(\boldsymbol{r}^n_{\mathcal{S}} - \lambda^n \boldsymbol{z}_{\mathcal{S}}). \tag{24}$$

Let $\mathbf{X}^{n-1} \in \mathbb{R}^{n\times m}$ be the design matrix containing all the historical decisions up to time $n-1$, which is 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 (24) is equivalent to

$$\boldsymbol{\beta}_{\mathcal{S}} = \left[(\mathbf{X}^{n-1}_{*\mathcal{S}})^T\mathbf{X}^{n-1}_{*\mathcal{S}}\right]^{-1}\left[(\mathbf{X}^{n-1}_{*\mathcal{S}})^T\mathbf{Y}^n - \lambda^n \boldsymbol{z}_{\mathcal{S}}\right]. \tag{25}$$

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 $\mathrm{Cov}(\mathbf{Y}^n) = \sigma_\epsilon^2 \mathbf{I}$, (25) gives us

$$\mathrm{Cov}(\boldsymbol{\beta}_{\mathcal{S}})^{(n)} = \mathbf{M}_{\mathcal{S}}^{n-1}\sigma_\epsilon^2 + (\lambda^n)^2 \mathbf{M}_{\mathcal{S}}^{n-1}\mathrm{Cov}(\boldsymbol{z}_{\mathcal{S}})^{(n)}\mathbf{M}_{\mathcal{S}}^{n-1}. \tag{26}$$

By definition, $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n} := \mathrm{Cov}(\boldsymbol{\beta}_{\mathcal{S}})^{(n)}$. If we replace n with $n+1$, (26) provides us with the equation

$$\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1} = \mathbf{M}_{\mathcal{S}}^{n}\sigma_\epsilon^2 + (\lambda^{n+1})^2 \mathbf{M}_{\mathcal{S}}^{n}\mathrm{Cov}(\boldsymbol{z}_{\mathcal{S}})^{(n+1)}\mathbf{M}_{\mathcal{S}}^{n}. \tag{27}$$

One should note that we can not directly compute $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1}$ from the right hand side of (27), 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 (20), (21), and (22), 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

$$\widetilde{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}})^{(n+1)} = \underset{\mathbf{M}\in\mathcal{M}(C_{\min},C_{\max})}{\operatorname{argmin}} \|\widehat{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}})^{(n+1)} - \mathbf{M}\|_F. \tag{28}$$

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 the interval $[C_{\min}, C_{\max}]$. Therefore we can approximately estimate $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1}$ by

$$\widetilde{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1} = \mathbf{M}_{\mathcal{S}}^{n}\sigma_\epsilon^2 + (\lambda^{n+1})^2 \mathbf{M}_{\mathcal{S}}^{n}\widetilde{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}})^{n+1}\mathbf{M}_{\mathcal{S}}^{n}. \tag{29}$$

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

Algorithm 1 The Knowledge Gradient Algorithm for Sparse Linear Models (KGSpLin)

Input: $\boldsymbol{\vartheta}^0, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},0}, \{\xi_j^0, \eta_j^0\}_{j=1}^p, \mathbf{X}, N, \sigma_\epsilon, \{\lambda^i\}_{i=1}^N$.

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. Compute KGSpLin by (12): $x^n = \operatorname{argmax} v_x^{KG,n}$;
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 estimate $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1}$ by $\widetilde{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1}$ in (29);
4. Bayesian update to: $\boldsymbol{\vartheta}^{n+1}, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n+1}, \{\xi_j^{n+1}, \eta_j^{n+1}\}_{j=1}^p$ by (14)-(17).

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 Models

As we have the sparse knowledge gradient algorithm with $\ell_{1,\infty}$ group Lasso, we can generalize the knowlege gradient for sparse linear model to a nonparametric sparse additive model. This can be done by approximating the nonparametric smooth function by finite order spline Basis. 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, $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_M]^T \in \mathbb{R}^M$, $\mathbf{X} = [X_{ij}] \in \mathbb{R}^{M\times p}$ is the design matrix, and

$$\mu_i = f(\mathbf{X}_{i*}) = \varsigma_i + \sum_{j=1}^{p} f_j(X_{ij}), \quad \text{for } i = 1, \ldots, M, \tag{30}$$

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 zero.

If the truth μ takes the nonparametric additive form as in (30), 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 random indicator variable of f_j's, that is,

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

First, let us approximate each functional component in (30) through one-dimensional splines. Without loss of generality, suppose that all elements of $\mathbf{X}$ 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

$$\widetilde{f}_j(x) = \sum_{k=1}^{d_j} \alpha_{jk}\phi_{jk}(x), \quad \text{for } 1 \leq j \leq p. \tag{31}$$

Then 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 the 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 $(\boldsymbol{x}^{i-1}, y^i) \in \mathbb{R}^p \times \mathbb{R}, i = 1, \ldots, n$, one can get $\widehat{\boldsymbol{\vartheta}}^n$ by solving the following penalized least squares problem

$$\widehat{\boldsymbol{\vartheta}}^n = \underset{\boldsymbol{\vartheta} \in \mathbb{R}^m}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} \left[y^i - \sum_{j=1}^{p} \sum_{k=1}^{d_j} \vartheta_{jk} \phi_{jk}(x_j^{i-1}) \right]^2 + \lambda^n \sum_{j=1}^{p} \|\boldsymbol{\vartheta}_{j\bullet}\|_\infty, \tag{32}$$

where λ^n is the tuning parameter. Optimization problem (32) 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. Therefore, we can also derive the knowledge gradient policy and Bayesian updating formulae as in Section 4. Here we let f_j^n be the Bayesian estimate of f_j at time n, that is,

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

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

Algorithm 2 The Knowledge Gradient Algorithm for Sparse Additive Models (KGSpAM)

Input:[2] $\boldsymbol{\vartheta}^0, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},0}, \{\xi_j^0, \eta_j^0\}_{j=1}^p, \mathbf{X}, N, \sigma_\epsilon, \{\lambda^i\}_{i=1}^N, \{\phi_{jk}\}_{k=1,j=1}^{d_j,p}, \{\tau_j\}_{j=0}^{K+1}$
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. Compute KGSpAM by (12):: $x^n = \operatorname{argmax} v_x^{KG,n}$;
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 estimate $\widehat{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},n+1}$ by $\widetilde{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1}$ in (29);
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

5.2 Tensor Product Smoothing Splines Functional ANOVA

If the regression functions in (30) 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

$$\mu_i = f(\mathbf{X}_{i*}) = \varsigma_i + \sum_{j=1}^{p} f_j(X_{ij}) + \sum_{j<k} f_{jk}(X_{ij}, X_{ik}) + \cdots, \tag{33}$$

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 $\varsigma = \mathbf{0}$, $\int f_j(x_j)\,\mathrm{d}x_j = 0$ for each j, $\iint f_{jk}(x_j, x_k)\,\mathrm{d}x_j\mathrm{d}x_k = 0$ 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

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

Let

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

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 (33) as basis expansion forms. Then we can generalize the sparse knowledge gradient algorithm to SS-ANOVA models.

6. Theoretical Results

In this section we show the convergence results of the Bayesian posterior mean estimate $\boldsymbol{\vartheta}^n$ in Algorithm 1 as well as of the functional estimate f^n in Algorithm 2. First, in Lemma 3, we present the asymptotic selection and estimation properties of $\ell_{1,\infty}$ group Lasso in high-dimensional settings when the number of groups p exceeds the sample size n. Specifically, we provide sufficient conditions under which the group Lasso is *rate consistent*, which means the cardinality of the selected sparsity pattern is on the same order as that of the true sparsity pattern. Also, we show the estimation error bound of group Lasso.

Based on these results, we assume that we begin with some historical observations, and the initial fixed design matrix satisfies the *sparse Riesz condition* (SRC) (Zhang and Huang, 2008), which is a form of *restricted eigenvalue* (RE) condition that limits the range of the eigenvalues of the covariance matrices of all subsets of a fixed number of covariates. (We refer to Van De Geer et al. (2009) for an extensive discussion of different types of restricted eigenvalue conditions.) If we have such a "warm" start, we can show that the Bayesian posterior estimation error is bounded as in Theorem 7. 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 10. Note that these error bounds are proved on the intersection $\bar{\mathcal{S}}$ of the support set $\mathcal{S}^n$ from the group Lasso estimator. But we can also show that $\bar{\mathcal{S}}$ is on the same order with the true support set $\mathcal{S}^*$. Additionally, all these theorems establish asymptotic bounds on estimation errors. Since the KG policy is proved to be myopically optimal in Frazier et al. (2009), this lends a strong guarantee that the algorithm will work well for finite budgets.

6.1 Linear Coefficient Error Bound

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 $\bar{d} := \max_{j=1,\ldots,p} d_j$. Recall that $m = \sum_{j=1}^p d_j$. Let $\mathcal{S}^n = \{j : \widehat{\boldsymbol{\vartheta}}^n_{\mathcal{G}_j} \neq 0\}$ be the estimated group support from the current Lasso estimator. Let $\mathcal{S}^*$ be the true support, that is $\mathcal{S}^* = \{j : \boldsymbol{\vartheta}_{\mathcal{G}_j} \neq 0\}$. 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 in Lemma 3. Our presentation needs the following assumptions.

Assumption 1 *For any n, the random noise errors $\epsilon^1, \ldots, \epsilon^n$ are independent and identically distributed as $\mathcal{N}(0, \sigma^2_\epsilon)$, that is, $\epsilon^1, \ldots, \epsilon^n \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2_\epsilon)$.*

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*

$$c_*\|\boldsymbol{\nu}\|_2^2 \leq \frac{\|\mathbf{X}^{n-1}_{*\mathcal{S}}\boldsymbol{\nu}\|_2^2}{n} \leq c^*\|\boldsymbol{\nu}\|_2^2, \quad \forall \mathcal{S} \text{ with } r = |\mathcal{S}| \text{ and } \boldsymbol{\nu} \in \mathbb{R}^{\sum_{j\in\mathcal{S}} d_j}. \tag{34}$$

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

Both assumptions can be reasonably expected to hold in practice. Assumption 1 is on the distribution of random noise. We let $\boldsymbol{\Sigma}^{\mathbf{X},n-1} := \frac{1}{n}(\mathbf{X}^{n-1})^T\mathbf{X}^{n-1}$ be the sample covariance matrix of the historical n observations. The SRC in Assumption 2 assumes the eigenvalues of the sample covariance matrix $\boldsymbol{\Sigma}^{\mathbf{X},n-1}_{\mathcal{S}} = \frac{1}{n}(\mathbf{X}^{n-1}_{*\mathcal{S}})^T\mathbf{X}^{n-1}_{*\mathcal{S}}$ are inside the interval $[c_*, c^*]$ when the size of $\mathcal{S}$ is no greater than r. The quantities c_* and c^* are considered as *sparse minimum and maximum eigenvalues* (Donoho, 2006; Meinshausen and Yu, 2009). When the number of groups exceeds the number of observations ($p > n$), there are potentially many models fitting the same data. However there is a certain uniqueness among such models under sparsity constraints. Under the SRC, all sets of r design vectors are linearly independent for a certain given rank r. One can refer to Zhang and Huang (2008) to see some sufficient conditions for the sparse Riesz condition to hold for both deterministic and random design matrices $\mathbf{X}$. We can show the selection and estimation consistency of $\ell_{1,\infty}$ group Lasso under these conditions if the penalty level λ^n is set to the following asymptotic order.

Additionally, we define $\widehat{c} = c^*/c_*$. We consider the Lasso path for

$$\lambda_* \equiv 2\sigma_\epsilon\sqrt{8(1+c_0)r\widehat{c}c^*\bar{d}^2 n\log(m \vee a_n)}, \tag{35}$$

with $c_0 \geq 0$ and $a_n \geq 0$ satisfying $p\bar{d}/(m \vee a_n)^{1+c_0} \approx 0$. For large p, this means that $\lambda_* \sim O(\sqrt{\bar{d}^2 n\log m})$ with $a_n = 0$. Then we can prove the following lemma. The technical details of the proof are based on Zhang and Huang (2008) and Wei and Huang (2010) and can be found in Appendix C.

Lemma 3 *Suppose Assumptions 1 and 2 are satisfied. Let $\{c^*, c_*, r, c_0\}$ be fixed. Let $1 \leq n \leq p \to \infty$. If we solve the group Lasso given in (13) with $\lambda^n = \lambda_*$ defined as (35), 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 *defined as* $C_1 := 2 + 4\widehat{c}$*;*

(2) Any optimal solution $\widehat{\boldsymbol{\beta}}^n$ *to* (13) *satisfies the following error bound*

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

for some positive constant C_2 *depending only on* $\{c^*, c_*, r, c_0\}$*.*

Remark 4 *Oracle inequalities and variable selection properties for the Lasso have been established under a variety of different assumptions on the design matrix; see Yuan and Lin (2006); Bunea et al. (2007); Liu and Zhang (2008); Zhang and Huang (2008); Bickel et al. (2009); Lounici et al. (2011) and references therein. Zhao and Yu (2006); Zou (2006) show that the irrepresentable condition is almost necessary and sufficient for Lasso to exactly select the true model. However, the irrepresentable condition is somewhat restrictive. Lemma 3, which is based on the result of Zhang and Huang (2008) and Wei and Huang (2010), relies on the SRC and proves the rate consistency of* $\ell_{1,\infty}$ *group Lasso, that is the selected model is of the correct order of sparsity. Such similar results are also shown in Belloni and Chernozhukov (2011, 2013); Lounici et al. (2011) based on other types of RE conditions.*

As one can see from the updating equations in (14) and (15), the posterior mean estimate $\boldsymbol{\vartheta}_{\mathcal{S}}^{n+1}$ is the weighted sum of the prior $\boldsymbol{\vartheta}_{\mathcal{S}}^{n}$ and the current Lasso estimate $\widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^{n+1}$. If the Lasso estimate has the ℓ_2 estimation bound as described in Lemma 3, 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 3, we need to make sure that assumptions 1 and 2 are satisfied for every Lasso problem in such a sequence. Assumption 1 is easy to satisfy. To show all the design matrices of 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 the SRC, only with different sparse minimum and maximum eigenvalues. To verify this, we need the following assumption.

Assumption 5 *For any* n*, there exists some constant* $B > 0$ *such that* $\|\boldsymbol{x}^n\|_2^2 \leq B$*.*

This assumption requires that each design $\boldsymbol{x}^n$ is chosen within an l_2-ball in $\mathbb{R}^m$. Note that in the SRC, it is easy to prove that $\|\mathbf{X}_{*\mathcal{S}}^{n-1}\boldsymbol{\nu}\|_2^2/(n\|\boldsymbol{\nu}\|_2^2)$ is automatically bounded above by B under Assumption 5. So without loss of generality, we assume that $c^* \leq B$, which would result in a sharper bound in Assumption 2. Then the following proposition proves that if the intial design matrix $\mathbf{X}^{N'-1}$ satisfies the SRC, then all the following design matrices satisfy the SRC with looser spectrum bounds.

Proposition 6 *Let Assumption 5 be satisfied. In addition, assume for some large enough* N'*, the design matrix* $\mathbf{X}^{N'-1}$ *satisfies the 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 the SRC* $(r, c_*/c, B)$*.*

Thus we have all the ingredients to prove the following theorem of the ℓ_2 error bound of the Bayesian posterior mean estimator.

Theorem 7 *Assume that Assumptions 1 and 5 are satisfied. Suppose we begin with N' historical observations and the fixed design matrix $\mathbf{X}^{N'-1}$ satisfies the SRC $(C_3 s^*, c_*, c^*)$, where C_3 is a positive constant defined below. Let c_*, c^*, c_0, s^*, and B be fixed, and $\bar{\mathcal{S}} := \bigcap_{n'=N'}^{n} \mathcal{S}^{n'}$. If we solve the Lasso given in (13) with $\lambda^n = \lambda_*$, then for some large enough n satisfying $\underline{c}N' \le n \le \bar{c}N'$ and $1 < \underline{c} \le \bar{c}$ being fixed constants, the following properties hold with probability converging to 1 as $n \to \infty$:*

(1) $|\bar{\mathcal{S}}| \le C_3 |\mathcal{S}^*|$ for some finite positive constant $C_3 := 2 + 4\bar{c}B/c_*$;

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

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

for some positive constant C_4 depending only on $c_*, c^*, c_0, \underline{c}, \bar{c}, B$, and $[C_{\min}, C_{\max}]$.

To prove the selection and estimation consistency results, we need to assume that we have a "warm" start of N' historical observations. We believe that this assumption is valid in some applications we have seen. For example, in the RNA problem (illustrated in Section 7.2), we do have some initial samples to give us a sense of how sparse the model is. Based on this, we can prove that the posterior can converge to the truth at the same rate as that of $\ell_{1,\infty}$ group Lasso as shown in Lemma 3. This result is satisfied for some large n in the interval $[\underline{c}N', \bar{c}N']$ with $1 < \underline{c} \le \bar{c}$ being fixed constants, and with high probability. Here this probability can converge to 1 as $n \to \infty$ as one can see from the proof in Appendix C. Additionally, similar to Lemma 3, we only prove that the posterior is rate consistent. It is unknown if $\bar{\mathcal{S}} \subseteq \mathcal{S}^*$ or $\bar{\mathcal{S}} \supseteq \mathcal{S}^*$.

Remark 8 *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 $\|\widehat{\boldsymbol{\beta}}^n - \boldsymbol{\beta}\|_2^2 \precsim \frac{s^* \bar{d} \log m}{n}$, which is minimax optimal. As one can see, the error term for $\ell_{1,\infty}$ group Lasso $\frac{s^* \bar{d}^2 \log m}{n}$ 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.*

6.2 Functional Estimate Error Bound

Based on the results in Section 6.1, we can also get the error bound for the 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, and 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,

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

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 $\mathcal{S} \subseteq \{1,\ldots,p\}$, 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 9 *$f_j \in \mathcal{H}(2, L)$ for $1 \le j \le p$.*

Also note here we have the new design matrix $\mathbf{X}^{n-1}$ on the basis ϕ_{jk}. Let $\mathbf{\Psi}_j^{n-1}$ be the $n \times d_j$ matrix $\mathbf{\Psi}_j(i,k) = \psi_{jk}(x_j^{i-1})$, where ψ_{jk} is the orthonormal B-spline basis. Let $\mathbf{\Psi}^{n-1} := [\mathbf{\Psi}_1^{n-1}, \ldots, \mathbf{\Psi}_p^{n-1}]$. Based on this and Theorem 7, we have the following theorem of the functional estimation error bound.

Theorem 10 *Assume that Assumptions 1 and 9 are satisfied. Similar to Assumption 5, let $\sum_{j,k}\psi_{jk}^2(x_j^n) \le B$ for any n,. Suppose we begin with N' historical observations and the fixed design matrix $\mathbf{\Psi}^{N'-1}$ satisfies the SRC $(C_3 s^*, c_*, c^*)$. Let c_*, c^*, c_0, s^*, and B be fixed, $\bar{d} = O(n^{1/6})$, and $\bar{\mathcal{S}} := \bigcap_{n'=N'}^{n} \mathcal{S}^{n'}$. If we solve the Lasso given in (32) with $\lambda^n = \lambda_*$, then for some large enough n satisfying $\underline{c}N' \le n \le \bar{c}N'$ and $1 < \underline{c} \le \bar{c}$ being fixed constants, the following properties hold with probability converging to 1 as $n \to \infty$:*

(1) $|\bar{\mathcal{S}}| \le C_3|\mathcal{S}^*|$ for some finite positive constant $C_3 := 2 + 4\bar{c}B/c_*$;

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

$$\|f_{\bar{\mathcal{S}}}^n - f_{\bar{\mathcal{S}}}\|_2^2 \le \frac{C_5\sigma_\epsilon^2 s^* \log m}{n^{2/3}},$$

for some positive constant C_5 depending only on $c_*, c^*, c_0, \underline{c}, \bar{c}, B$, and $[C_{\min}, C_{\max}]$.

Remark 11 *Note that Assumption 2 is usually assumed to be valid in the settings with i.i.d. samples. However, in our problem setting, the sampling decisions are chosen by the KG algorithm, and thus can generally be highly dependent. To this end, in Theorems 7 and 10, we assume that we begin with N' historical samples and only assume the fixed design matrix $\mathbf{X}^{N'-1}$ satisfies Assumption 2. Therefore, our results are only based on this "warm" start, without putting any i.i.d. assumptions on the samples.*

7. Simulations

In this section we present the results of the experiments. In Section 7.1, we investigate the empirical performance of KGSpLin and KGSpAM on several different experimental settings. In Section 7.2, the application problem for identifying the accessibility region of the RNA molecule gI intron is briefly described. We illustrate how the KGSpLin policy is used to guide the experiments and present its empirical performance.

7.1 Controlled Experiments

In this section, we test KGSpLin and KGSpAM in controlled experiments. Three other policies are compared against our policies: *exploration*, where an alternative is chosen randomly at each time; *exploitation*, which chooses an alternative that has the maximum value in μ according to the current belief distribution; and *KGLin*, which is the KG policy with linear belief, using recursive least squares for updating the estimates (Negoescu et al., 2011). To better compare different policies, the updating scheme for both exploration and exploitation is the same as that of KGSpLin and KGSpAM presented in Section 4.2.

In all the experiments presented in this paper, we repeatedly sample the truth $\boldsymbol{\alpha}$ from some Gaussian distribution, while the sparsity pattern is known and fixed. We compare different policies to see how well we are discovering the values of $\boldsymbol{\alpha}$ and the underlying sparsity patterns. Also, throughout all the simulations, we always start with non-informative priors of the sparsity structures. That is, $\xi_j^0 = \eta_j^0 = 1$, for $j = 1, \ldots, p$.

In the first set of experiments, we focus on the comparison of KGSpLin with other policies using a relatively large measurement budget $N = 200$. We generate a linear model with $m = 100$ 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 each standard deviation of 30% of the mean. Specifically, we let $\mu = \sum_{j=1}^m \alpha_j x_j + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. For $j = 1, \ldots, 20$, let α_j be independently drawn from $\mathcal{N}(\vartheta_j, \Sigma_{jj}^{\boldsymbol{\vartheta}})$, where $\vartheta_j = j + 10$, and $\Sigma_{jj}^{\boldsymbol{\vartheta}} = (0.3\vartheta_j)^2$. For $j = 21, \ldots, 100$, let $\alpha_j = 0$. The prior is also independently sampled. For the nonzero parts, the prior is sampled from the same distribution as $\boldsymbol{\alpha}$; for the zero parts, the prior is sampled with $\vartheta_j^0 = \text{mean}(\boldsymbol{\vartheta})$ and $\Sigma_{jj}^{\boldsymbol{\vartheta},0} = (0.3\text{mean}(\boldsymbol{\vartheta}))^2$. Then we uniformly sample $M = 100$ alternatives from $[0, 1]^m$.

To quantitatively measure the performance of different policies, we consider a quantitative metric called opportunity cost (OC), which is defined as the difference in the true value between the best option and the option chosen according to the policy's posterior belief distribution, that is

$$\text{OC}(n) = \mu(x^*) - \mu(x^{n,*}).$$

For illustrative purposes, we compare the percentage OC with respect to the optimal value, defined as

$$\text{OC\%}(n) = \frac{\mu(x^*) - \mu(x^{n,*})}{\mu(x^*)}.$$

This normalization better illustrates how far in percentage we are from the optimal and provides a unit-free representation of a policy's performance. By taking the average percentage OC over several replications, we can estimate the policy's average performance in practice. As to the tunable parameter λ^n, theoretical results in Section 6 show that λ should be increasing with iteration number. For simplification, we carefully tune λ^n to be piecewise linearly increasing with respect to measurement n.

Figure 3 shows the log of the averaged OC% and the normalized estimation error of $\boldsymbol{\vartheta}$ (the ℓ_2 error divided by m) over 300 replications using well chosen tuning parameter sequences with low and high measurement noises. The standard deviations of the measurement noise are respectively 5% and 30% of the expected range of the truth.

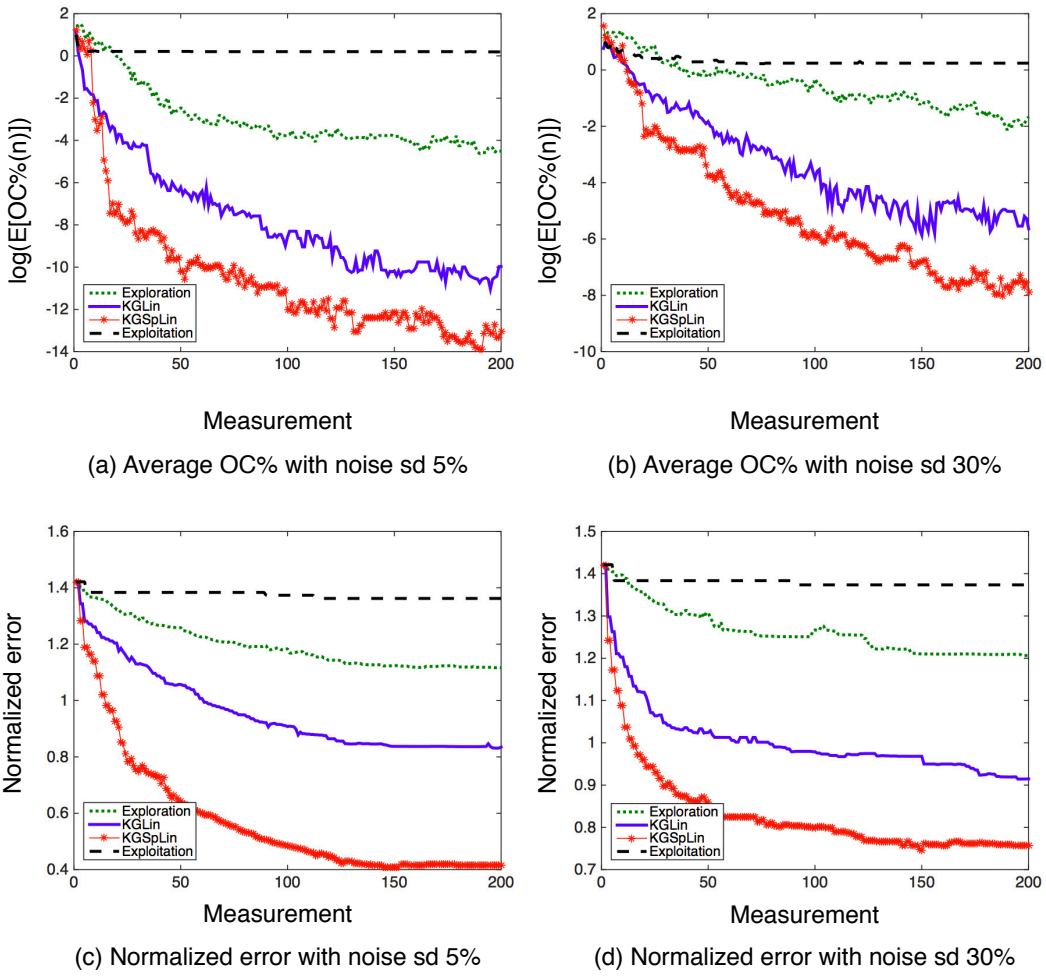


Figure 3: (a)(b) and (c)(d) compares exploration, exploitation, KGLin, and KGSpLin by showing the averaged OC% and normalized estimation errors over 300 runs under low measurement noise (5% range of the truth) and high measurement noise (30% range of the truth).

From Figure 3(a)(b) 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 $10 \sim 20$ measurements when the true sparsity pattern becomes detectable, KGSpLin far outperforms KGLin, exploitation, 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. Figure 3(c)(d) show that KGSpLin outperforms the other polices in estimating the linear coefficients. Even if at the initial stage, Lasso still finds low-dimensional models that can better approximate the true model compared with other techniques. In light of this, it is interesting to see how

KGSpLin compares with KGLin on data which is *not* sparse. Refer to Appendix D to see more empirical results.

In the second set of experiments, to further compare KGSpLin with KGLin, we use test functions that have higher degrees of sparsity (that is, more irrelevant dimensions in the feature space), with a relatively small measurement budget $N = 50$. We take several standard low-dimensional test functions and hide them in a $m = 200$-dimensional space. Note all the test functions chosen here can be written as linear expansions with respect to basis functions of x. This means the test functions of $3, 6, 4$, and 18 dimensions are embeded in a 200-dimensional space. These functions were designed to be minimized, so both policies are applied to the negative of the functions. We uniformly sample $M = 400$ alternatives from the feasible regions. The detailed configurations of these test functions are shown in Table 1 below. We include the mathematical forms of these four test functions. We also illustrate the distributions to sample the truths $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\vartheta}, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}})$ and the distributions of the priors $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\vartheta}^0, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},0})$. For all the four test functions, the covariance matrices $\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}}$ are sampled as: $\Sigma^{\boldsymbol{\vartheta}}_{jj} = (0.3\vartheta_j)^2$ for $\vartheta_j \neq 0$, and $\Sigma^{\boldsymbol{\vartheta}}_{jk} = 0$ otherwise. The prior covariance matrices $\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},0}$ are sampled as: $\Sigma^{\boldsymbol{\vartheta},0}_{jj} = (0.3\vartheta^0_j)^2$ for $\vartheta_j \neq 0$, $\Sigma^{\boldsymbol{\vartheta},0}_{jj} = (0.3\text{mean}(\boldsymbol{\vartheta}^0))^2$ for $\vartheta_j = 0$, and $\Sigma^{\boldsymbol{\vartheta},0}_{jk} = 0$ otherwise.

Test function	**Mean**
Matyas $\mu(\boldsymbol{x}) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$, $\mathcal{X} = [-10, 10]^2$	$\boldsymbol{\vartheta} = [-0.26, -0.26, 0.48, 0, \ldots, 0]$ $\boldsymbol{\vartheta}^0 = [-0.18, -0.34, 0.3, 0, \ldots, 0]$
Six-hump Camel $\mu(\boldsymbol{x}) = (4 - 2.1x_1^2 + x_1^4/3)x_1^2$ $+x_1x_2 + (-4 + 4x_2^2)x_2^2$, $\mathcal{X} = [-3, 3] \times [-2, 2]$	$\boldsymbol{\vartheta} = [-4, 2.1, -1/3, -1, 4, -4, 0, \ldots, 0]$ $\boldsymbol{\vartheta}^0 = [-3.2, 1.5, -0.1, -1.5, 4.5, -3.6, 0, \ldots, 0]$
Bohachevsky $\mu(\boldsymbol{x}) = x_1^2 + 2x_2^2 + 0.7-$ $0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2)$, $\mathcal{X} = [-100, 100]^2$	$\boldsymbol{\vartheta} = [-1, -2, 0.3, 0.4, 0, \ldots, 0]$ $\boldsymbol{\vartheta}^0 = [-0.6, -2.4, 0.1, 0.8, 0, \ldots, 0]$
Trid $\mu(\boldsymbol{x}) = \sum_{i=1}^d (x_i - 1)^2 - \sum_{i=2}^d x_i x_{i-1}$, $d = 6, \mathcal{X} = [-36, 36]^6$	$\boldsymbol{\vartheta} = [\underbrace{-1, \ldots, -1}_{6}, \underbrace{2, \ldots, 2}_{6}, \underbrace{1, \ldots, 1}_{5}, -6, 0, \ldots, 0]$ $\boldsymbol{\vartheta}^0 = [\underbrace{-0.6, \ldots, -0.6}_{6}, \underbrace{2.3, \ldots, 2.3}_{6}, \underbrace{1.5, \ldots, 1.5}_{5}, -4, 0, \ldots, 0]$

Table 1: Detailed configurations for test functions.

We compare the performance of KGLin and KGSpLin on these four different test functions. Each policy is run 500 times with the specified amount of observation noise. Table 2 gives the sample means, medians, and standard deviations of the opportunity cost after $N = 50$ iterations of each policy. We bold and underline the smaller values. The results are given for different levels of noise. The standard deviation of the normally distributed noise σ_ϵ is chosen to be $1\%, 10\%$, and 20% of the range of μ.

		KGSpLin			KGLin		
Test function	σ_ϵ	$\mathbb{E}$(OC)	σ(OC)	Median	$\mathbb{E}$(OC)	σ(OC)	Median
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
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
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**	3.2300	**1.5578**	1.9668	3.696	1.7008
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

Table 2: Quantitative comparison for KGSpLin and KGLin on standard test functions.

From Table 2, we can see that for all the four test functions, KGSpLin outperforms KGLin in having smaller means of opportunity cost. However, in some cases, KGLin performs better or competitively by having smaller medians. This is because in some of these simulations, it takes longer for Lasso to identify the true support, which results in higher opportunity costs at the initial samplings. Additionally, because of this, the margins of these two algorithms for Bohachevsky and Trid functions are relatively small, especially considering we are using a relatively small number of budget. However, if we compare the performance of KGSpLin and KGLin excluding the initial 10 measurements, we can see that KGSpLin does significantly better than KGLin (See Table 4 in Appendix D).

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

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

the relevant component functions are given by

$$f_{12}(x_1, x_2) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2, \tag{36}$$

$$f_3(x) = 2\sin(2\pi x), \tag{37}$$

$$f_4(x) = 8(x - 0.5)^2, \tag{38}$$

$$f_5(x) = 2\exp(-3x), \tag{39}$$

where the first component function f_{12} in (36) is known as the Three-hump camel function. We plot the true Three-hump camel function in Figure 4(a), while the key part is shown in Figure 4(b). For f_{12}, we use B-splines tensor product space $\mathcal{S}_4 \otimes \mathcal{S}_4$ to approximate it. The knot sequences are 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 sequences 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 σ_ϵ is set to 20% of the expected range of the truth μ.

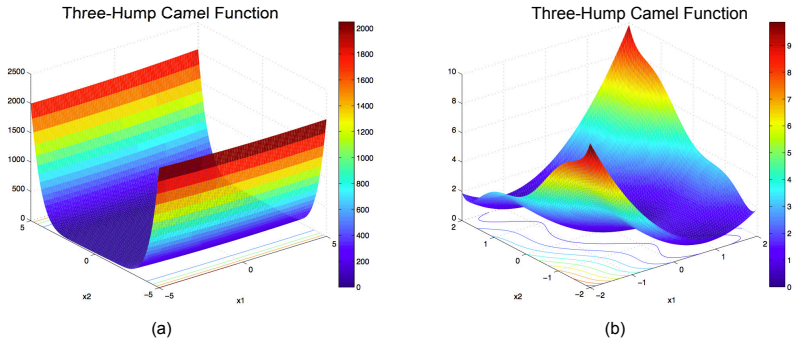


Figure 4: (a) shows the negative Three-hump camel function on its recommended input domain. (b) shows only a portion of this domain, to allow for easier viewing of the function's key characteristics. The function has one global maximum and two other local maxima.

Then we run 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 5. Comparing these estimates with the true function shown in Figure 4, we can see see that the policy has done a good job estimating the lower key regions of the function as desired after 10 observations, and it identifies the areas of the three maxima after 30 observations. For the remaining three relevant functional components in (37), (38), and (39), we plot the prior, truth, and final estimates of KGLin and KGSpAM in Figure 6. Finally, we run 300 replications and plot the averaged OC% and the normalized estimation error in Figure 7.

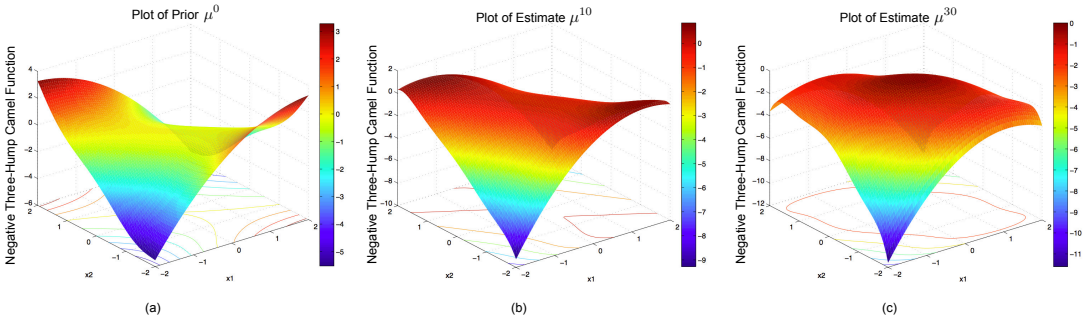


Figure 5: (a) shows the prior of negative Three-hump camel function on its key region. (b) and (c) show the estimates of negative Three-hump camel function on its key region after 10 and 30 observations respectively.

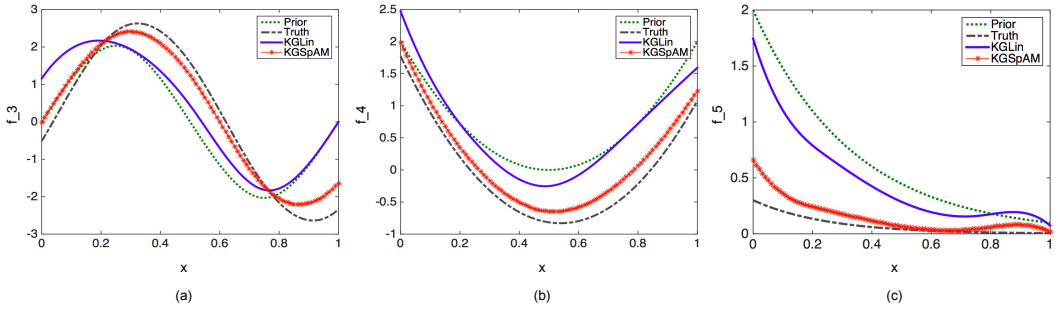


Figure 6: (a)(b)(c) The prior, truth, and final estimate of the sparse additive model in (37)-(39) comparing KGLin and KGSpAM after $N = 30$ observations. The standard deviation of measurement noise 20% of the expected range of the truth.

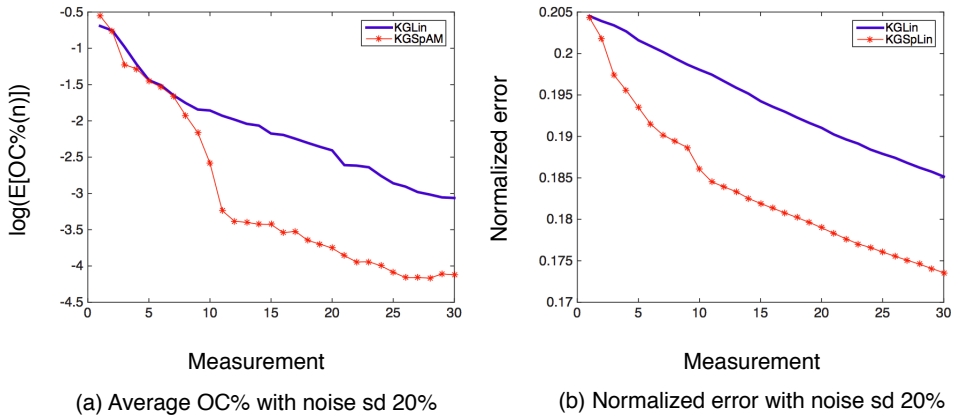


Figure 7: (a)(b) compares KGSpAM and KGLin by showing the averaged OC% and normalized estimation errors over 300 runs under 20% measurement noise level.

7.2 Application to RNA Data

An important step in health research requires learning the structure of RNA molecules to improve our understanding of how different drugs might behave in humans. This application addresses the problem of determining the accessibility patterns of an RNA molecule known as the *Tetrahymena Group I intron* (gI intron). Determining these accessibility patterns is difficult to do in silico, as they depend on the complicated folding of the molecule known as the intron's tertiary structure (Vazquez-Anderson and Contreras, 2013). Experimentally, such accessibility patterns can be inferred from fluorescence measurements obtained from the iRS3 by using various complementary probes designed a priori to target a region within the gI intron (Sowa et al., 2014). By fixing the size of the probe, we can view the selection of the probe and the target region that maximize the fluorescent signal as a key step in identifying the accessibility patterns of the molecule. See our parallel paper Li et al. (2015) for a more detailed description of the problem and more simulation results.

Followed by the thermo-kinetic model, the amount of accessibility (fluorescence signal) μ has a linear relationship with respect to the coefficients representing the accessibility of each nucleotides. Also, most of the coefficients are zero, thus we have a sparse linear model. Alternatives (testing probe sequences) with a number of $M = 91$ are selected by the domain experts. The number of dimensions is the length of the molecule, which is $m = 414$. In the following experiments, we take a subsequence of the molecule (from site 95 to 251) with $m = 157$ to better visualize the results. The prior data is the in-vitro DMS footprinting data published in Russell et al. (2006). The true coefficient vector is simulated by both vertically perturbing (normally deviated with sd = 20%) and horizontally shifting (uniformly shifted $20 \sim 50$ sites) the prior in-vitro DMS footprinting data.

First, we illustrate how KGSpLin policy works under a measurement noise of 30%. For one such simulated truth, we depict the KGSpLin value initially, after one and two measurements, respectively in Figure 8. For these figures, we only include those probes with KGSpLin values above the mean to better visualize the KGSpLin scores. As indicated by the arrows, for the probes with the largest KGSpLin scores, the KGSpLin scores drop after they have been measured. As we only plot those with KGSpLin scores above average, some probes with high KGSpLin scores in Figure 8(a) have the scores dropped below average after being measured and are therefore not shown in Figure 8(b). This observation is consistent with our intuition of KGSpLin as a measure of the value of information, and thus we can use this policy as a guideline to pick the next experiments.

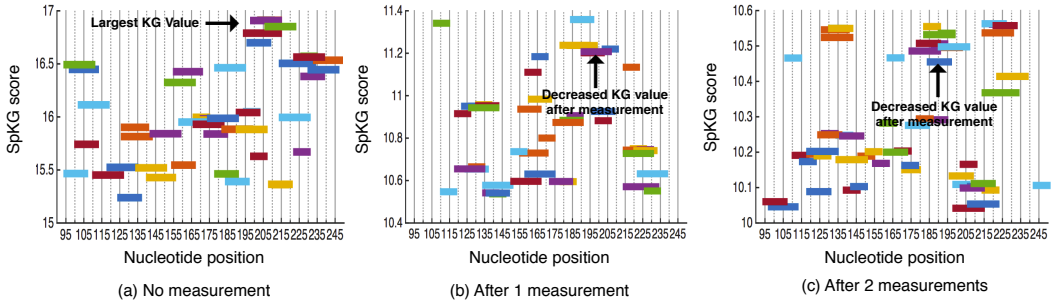


Figure 8: KGSpLin values before and after 1 and 2 measurements with noise ratio of 30%. (A subsequence of the RNA molecule is selected from site 95 to 251. Each bar is a potential range of a probe.)

Finally, for one simulated truth, we also plot the estimates of the accessibility profiles (coefficients) after $20, 30, 40, 50$ measurements with a noise ratio of 30% in Figure 9. As one can see, after 20 measurements, the estimate is still closer to the prior than the truth. After 30 measurements, we have discovered many of the accessible regions. After 40 measurements, we have not only discovered the location of the accessible regions, but obtained good estimates for the actual accessibility value. And after 50 measurements, our estimate closely matches the truth.

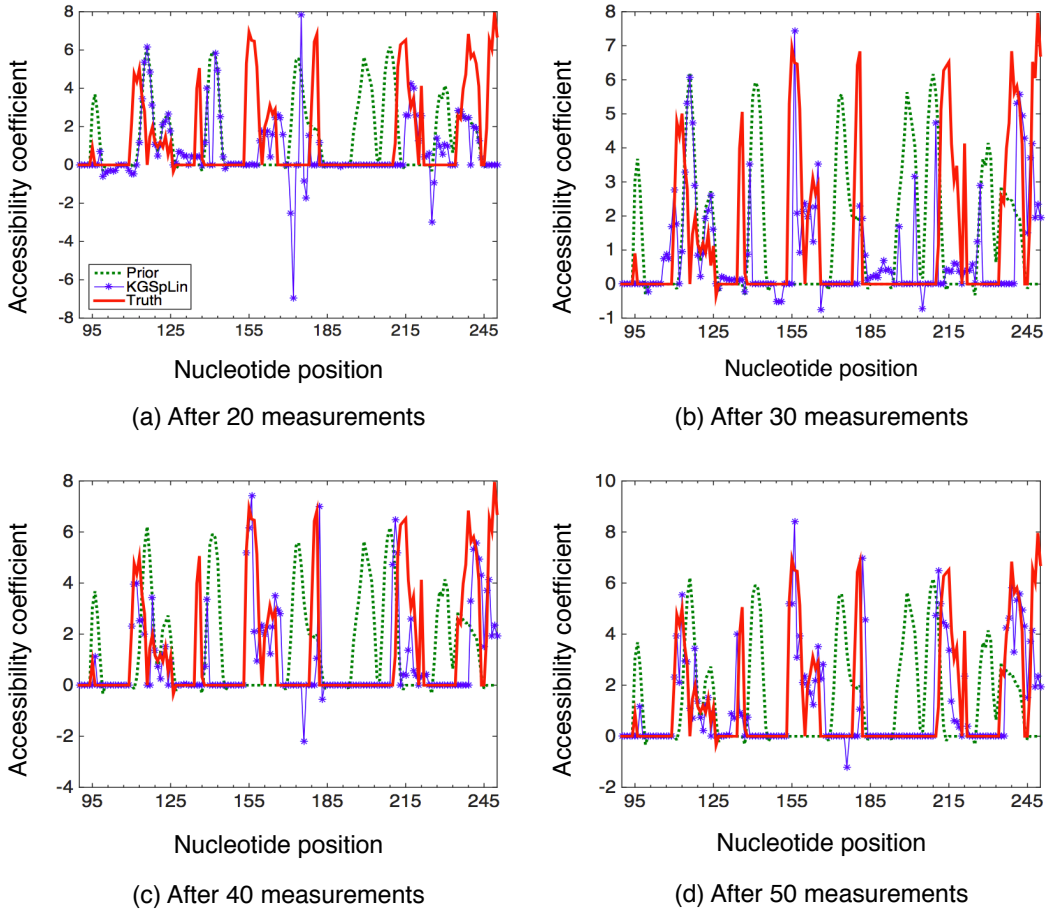


Figure 9: Accessibility profile estimate by the KGSpLin algorithm after $20, 30, 40$, and 50 measurements with noise ratio of 30%.

8. Conclusion

In this paper, we extend the KG policy to high-dimensional linear belief. Then this can be naturally generalized to the nonparametric additive beliefs, if we approximate each individual smooth function with B-splines of finite order. It is a novel hybrid of Bayesian R&S with the frequentist learning approach. Parallel with the Bayesian model, the policies use the 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 settings. 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 also guarantees the convergence to the global optimal alternative. All these advantages make them reasonable alternatives to other policies for high-dimensional applications with sparse structure. Despite the advances, the conver-

gence 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.

Acknowledgements

We are grateful to the editor and all the anonymous referees for their valuable comments, which lead to several improvements in the paper. We also thank Dr. Rick Russell for kindly providing the raw in-vitro DMS footprinting data used in this work and previously published in Russell et al. (2006).

Appendix A.

Refer to Table 3.

Appendix B. The Homotopy Algorithm for Recursive $\ell_{1,\infty}$ Group Lasso

In this Appendix, we briefly describe the recursive homotopy algorithm to exactly update the $\ell_{1,\infty}$ Lasso solutions, which is used in Algorithms 1 and 2. The homotopy algorithm for recursive Lasso is proposed in Garrigues and El Ghaoui (2008). Based on this result, Chen and Hero (2012) propose the following homotopy algorithm for recursive $\ell_{1,\infty}$ group Lasso.

Recall that we let $\widehat{\boldsymbol{\beta}}^n$ be the solution to the Lasso with n observations, that is

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

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. Recall that $\mathbf{R}^{n-1} := \sum_{i=1}^{n} \boldsymbol{x}^{i-1}(\boldsymbol{x}^{i-1})^T$, $\boldsymbol{r}^n := \sum_{i=1}^{n} \boldsymbol{x}^{i-1}y^i$. Let us define a function

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

It is easy to see that $\widehat{\boldsymbol{\beta}}^n = u(0,\lambda^n)$, and $\widehat{\boldsymbol{\beta}}^{n+1} = u(1,\lambda^{n+1})$. The homotopy algorithm 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 the homotopy methods such 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.

Variable	Description
$\mathcal{X}$	Set of alternatives
M	Number of alternatives
N	Number of measurements budget
μ_x	Unknown mean of alternative x
σ_ϵ	Known standard deviation of measurement noise
$\boldsymbol{\mu}$	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 x^n
y^{n+1}	Sampling observation from measuring alternative x^n
$\boldsymbol{\theta}^n$, $\boldsymbol{\Sigma}^n$	Mean and Covariance of prior distribution on $\boldsymbol{\mu}$ at time n
S^n	State variable, defined as the pair $(\boldsymbol{\theta}^n, \boldsymbol{\Sigma}^n)$
$v_x^{\mathrm{KG},n}$	Knowledge gradient value for alternative x at time n
$\boldsymbol{\alpha}$	Vector of linear coefficients
m	Number of features
$\mathbf{X}$	Design matrix including all the possible finite experimental designs
$\boldsymbol{\vartheta}^n, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}$	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, $d_j = \lvert\mathcal{G}_j\rvert$
$\boldsymbol{\zeta}^n$	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 shape 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 the 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, $s^* = \lvert\mathcal{S}^*\rvert$

Table 3: Table of Notation

Appendix C. Proofs

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

C.1 Proof of Lemma 3

The basic idea of the proof in Lemma 3 follows that in the proof of the rate consistency of the Lasso in Zhang and Huang (2008) and the rate consistency of the $\ell_{1,2}$ group Lasso in Wei and Huang (2010). However, there are some differences in the characterization of the solution via the KKT conditions and in the constraint needed for the penalty level. In the following we provide a sketch of the proof, especially highlighting the technical differences.

Let us begin by introducing some notation which will be used in the proof. For now let us leave out the superscript n to simplify notation. Let $\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_j\mathcal{S}_k} = \mathbf{X}^T_{\mathcal{S}_j}\mathbf{X}_{\mathcal{S}_k}/n$. Let $\{k : \|\widehat{\boldsymbol{\beta}}_{\mathcal{G}_k}\|_\infty > 0\} \subseteq \mathcal{S}_1 \subset \{k : \mathbf{X}^T_{*\mathcal{G}_k}(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = \lambda \boldsymbol{z}_{\mathcal{G}_k}\} \cup \mathcal{S}^*$, where $\boldsymbol{z}_{\mathcal{G}_k} \in \partial\|\widehat{\boldsymbol{\beta}}_{\mathcal{G}_k}\|_{1,\infty}$. Set $\mathcal{S}_2 = \{1,\ldots,p\} \setminus \mathcal{S}_1$, $\mathcal{S}_0 = \{1,\ldots,p\} \setminus \mathcal{S}^*$, $\mathcal{S}_3 = \mathcal{S}_1 \setminus \mathcal{S}_0$, $\mathcal{S}_4 = \mathcal{S}_1 \cap \mathcal{S}_0$, $\mathcal{S}_5 = \mathcal{S}_2 \setminus \mathcal{S}_0$, $\mathcal{S}_6 = \mathcal{S}_1 \cap \mathcal{S}_0$. Thus we have $\mathcal{S}_1 = \mathcal{S}_3 \cup \mathcal{S}_4$, $\mathcal{S}_3 \cap \mathcal{S}_4 = \emptyset$, $\mathcal{S}_2 = \mathcal{S}_5 \cup \mathcal{S}_6$, $\mathcal{S}_5 \cap \mathcal{S}_6 = \emptyset$. Let $|\mathcal{S}_i| = \sum_{k\in\mathcal{S}_i} d_k$, $N(\mathcal{S}_i) = \#\{k : k \in \mathcal{S}_i\}$, $i = 1,\ldots,6$, and $s_1 = N(\mathcal{S}_1)$. Recall that $\bar{d}$ is the maximum group size. Now we let $\underline{d}$ be the minimum group size, that is $\underline{d} := \min_{j=1,\ldots,p} d_j$.

(1) The proof of part (1) consists of three steps. Step 1 proves some inequalities related to s_1. Step 2 translates the results of Step 1 into upper bounds for $|\mathcal{S}|$. Step 3 completes the proof by showing the probability of the event in Step 2 converging to 1.

Since $\widehat{\boldsymbol{\beta}}$ is the solution of (13), by the KKT condition,

$$\begin{cases} \mathbf{X}^T_{*\mathcal{G}_k}(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = \lambda \boldsymbol{z}_{\mathcal{G}_k}, & \forall \|\widehat{\boldsymbol{\beta}}_{\mathcal{G}_k}\|_\infty > 0, \\ -\lambda \le \|\mathbf{X}^T_{*\mathcal{G}_k}(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}})\|_1 \le \lambda & \forall \|\widehat{\boldsymbol{\beta}}_{\mathcal{G}_k}\|_\infty = 0. \end{cases}$$

We then have $(\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1})^{-1}\mathbf{Q}_{\mathcal{S}_1}/n = (\boldsymbol{\beta}_{\mathcal{S}_1} - \widehat{\boldsymbol{\beta}}_{\mathcal{S}_1}) + (\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1})^{-1}\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1\mathcal{S}_2}\boldsymbol{\beta}_{\mathcal{S}_2} + (\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1})^{-1}\mathbf{X}^T_{\mathcal{S}_1}\boldsymbol{\epsilon}/n$, and $n\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_2\mathcal{S}_2}\boldsymbol{\beta}_{\mathcal{S}_2} - n\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_2\mathcal{S}_1}(\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1})^{-1}\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1\mathcal{S}_2}\boldsymbol{\beta}_{\mathcal{S}_2} \le \mathbf{C}_{\mathcal{S}_2} - \mathbf{X}^T_{\mathcal{S}_2}\boldsymbol{\epsilon} - \mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_2\mathcal{S}_1}(\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1})^{-1}\mathbf{Q}_{\mathcal{S}_1} + \mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_2\mathcal{S}_1}(\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1})^{-1}\mathbf{X}^T_{\mathcal{S}_1}\boldsymbol{\epsilon}$, where $\mathbf{Q}_{\mathcal{S}_i} = [\mathbf{Q}^T_{k_1},\ldots,\mathbf{Q}^T_{k_{s_i}}]^T \in \mathbb{R}^{|\mathcal{S}_i|}$, $\mathbf{Q}_{k_i} = \lambda \boldsymbol{q}_{k_i}$, $\boldsymbol{q}_k = \mathbf{X}^T_{*\mathcal{G}_k}(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}})/\lambda$, $\mathbf{C}_{\mathcal{S}_i} = [\mathbf{C}^T_{k_1},\ldots,\mathbf{C}^T_{k_{s_i}}]^T \in \mathbb{R}^{|\mathcal{S}_i|}$, $\mathbf{C}_{k_i} = I(\|\widehat{\boldsymbol{\beta}}_{k_i}\|_2 = 0)\boldsymbol{e}_{d_{k_i}\times 1}$, all the elements of matrix $\boldsymbol{e}_{d_{k_i}\times 1}$ equal 1 and $k_i \in \mathcal{S}_i$.

Step 1. Define

$$\mathbf{V}_{1j} = \frac{1}{\sqrt{n}}(\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1})^{-1/2}\mathbf{R}^T_{\mathcal{S}_j 1}\mathbf{Q}_{\mathcal{S}_j}, \text{for j=1,3,4}, \quad \boldsymbol{w}_k = (\mathbf{I} - \mathbf{P}_1)\mathbf{X}_{\mathcal{S}_k}\boldsymbol{\beta}_{\mathcal{S}_k}, \text{for } k = 2,\ldots,6, \quad (40)$$

where $\mathbf{R}_{\mathcal{S}_k j}$ is the matrix representing the selection of variables in $\mathcal{S}_k$ from $\mathcal{S}_j$. By the definition of $\mathbf{V}_{1j}$ in (40) and Lemma 1 in Zhang and Huang (2008) (It is easy to prove the Lemma is still true for group Lasso with $\ell_{1,\infty}$ penalty),

$$\|\mathbf{V}_{14}\|_2 = \frac{\|(\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1})^{-1/2}\mathbf{R}^T_{\mathcal{S}_4 1}\mathbf{Q}_{\mathcal{S}_4}\|_2}{\sqrt{n}} \ge \frac{\|\mathbf{R}^T_{\mathcal{S}_4 1}\mathbf{Q}_{\mathcal{S}_4}\|_2}{\sqrt{nc^*(|\mathcal{S}_1|)}} \ge \frac{\lambda\sum_{k\in\mathcal{S}_4}\|\boldsymbol{q}_k\|_1}{\sqrt{nc^*(|\mathcal{S}_1|)N(\mathcal{S}_4)}} \ge \frac{\lambda\sqrt{s_1 - s^*}}{\sqrt{nc^*(|\mathcal{S}_1|)}}.$$

That is $\|\mathbf{V}_{14}\|_2^2 \ge \lambda(s_1 - s^*)/nc^*(|\mathcal{S}_1|)$. . From the KKT conditions, we have

$$\begin{aligned} \mathbf{V}^T_{14}(\mathbf{V}_{13} + \mathbf{V}_{14}) &\le \mathbf{Q}^T_{\mathcal{S}_4}\mathbf{R}_{\mathcal{S}_4 1}(\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1})^{-1}\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1\mathcal{S}_2}\boldsymbol{\beta}_{\mathcal{S}_2} + \frac{\mathbf{Q}^T_{\mathcal{S}_4}\mathbf{R}_{\mathcal{S}_4 1}(\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1})^{-1}\mathbf{X}^T_{*\mathcal{S}^1}\boldsymbol{\epsilon}}{n} + \lambda\sum_{k\in\mathcal{S}_4}\|\boldsymbol{\beta}_k\|_1, \\ \|\boldsymbol{w}_2\|_2^2 &\le -\boldsymbol{w}_2^T\boldsymbol{\epsilon} - \mathbf{Q}^T_{\mathcal{S}_1}(\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1})^{-1}\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1\mathcal{S}_2}\boldsymbol{\beta}_{\mathcal{S}_2} + \boldsymbol{\beta}^T_{\mathcal{S}_2}\mathbf{C}_{\mathcal{S}_2}. \end{aligned}$$

Define $\boldsymbol{u} = (\mathbf{X}_{\mathcal{S}_1}(\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1})^{-1}\mathbf{R}^T_{\mathcal{S}_{41}}\mathbf{Q}_{\mathcal{S}_4}/n - \boldsymbol{w}_2)/\|\mathbf{X}_{\mathcal{S}_1}(\mathbf{\Sigma}^{\mathbf{X}}_{\mathcal{S}_1})^{-1}\mathbf{R}^T_{\mathcal{S}_{41}}\mathbf{Q}_{\mathcal{S}_4}/n - \boldsymbol{w}_2\|_2$, it follows that

$$\begin{aligned}\|\mathbf{V}_{14}\|_2^2 + \|\boldsymbol{w}_2\|_2^2 \leq \quad & (\|\mathbf{V}_{14}\|_2 + \|\mathbf{P}_1\mathbf{X}_{\mathcal{S}_2}\boldsymbol{\beta}_{\mathcal{S}_2}\|_2)(\tfrac{\lambda^2 N(\mathcal{S}_3)}{nc_*(\|\mathcal{S}_1\|)})^{1/2} + \lambda\|\boldsymbol{\beta}_{\mathcal{S}_5}\|_2 \\ & + (\|\mathbf{V}_{14}\|_2^2 + \|\boldsymbol{w}_2\|_2^2)^{1/2}\boldsymbol{u}^T\boldsymbol{\epsilon}. \end{aligned} \tag{41}$$

Step 2. Define $B_1^2 = \lambda^2 s^*/(nc^*(|\mathcal{S}_1|))$ and $B_2^2 = \lambda^2 s^*/(nc_*(|\mathcal{S}_0| \vee |\mathcal{S}_1|))$. In this step, we consider the event $|\boldsymbol{u}^T\boldsymbol{\epsilon}|^2 \leq (|\mathcal{S}_1| \vee \underline{d})B_1^2/(4s^*\bar{d})$. Suppose that the set $\mathcal{S}_1$ contains all the large $\boldsymbol{\beta}_k \neq 0$. From (41), we have $\|\mathbf{V}_{14}\|_2^2 \leq B_1^2 + 4B_2^2$, so we have

$$(s_1 - s^*)^+ \leq \frac{s^*\|\mathbf{V}_{14}\|_2}{B_1^2} \leq s^* + \frac{4s^*c^*(|\mathcal{S}_1|)}{c_*(|\mathcal{S}_1|)}. \tag{42}$$

Step 3. Letting $c_*(\mathcal{S}_m) = c_*$, $c^*(\mathcal{S}_m) = c^*$, for $N(\mathcal{S}_m) \leq r$, we have

$$s_1 \leq N(\mathcal{S}_1 \cup \mathcal{S}_5) \leq r, \quad |\boldsymbol{u}^T\boldsymbol{\epsilon}|^2 \leq \frac{(|\mathcal{S}_1| \vee \underline{d})\lambda^2}{4\bar{d}nc^*(|\mathcal{S}_1|)} \tag{43}$$

Since $\widehat{c} = c^*/c_*$, (42) gives us that $s_1 \leq (2 + 4\widehat{c})s^*$ when $\lambda \geq \lambda_*$, which implies the result of part (1). Define

$$x_s^* \equiv \max_{|\mathcal{S}|=s} \max_{\|\mathbf{U}_{\mathcal{S}_k}\|_1=1, k=1,\ldots,s} \left| \boldsymbol{\epsilon}^T \frac{\mathbf{X}_{\mathcal{S}}(\mathbf{X}_{\mathcal{S}}^T\mathbf{X}_{\mathcal{S}})^{-1}\bar{\mathbf{Q}}_{\mathcal{S}} - (\mathbf{I} - \mathbf{P}_{\mathcal{S}})\mathbf{X}\boldsymbol{\beta}}{\|\mathbf{X}_{\mathcal{S}}(\mathbf{X}_{\mathcal{S}}^T\mathbf{X}_{\mathcal{S}})^{-1}\bar{\mathbf{Q}}_{\mathcal{S}} - (\mathbf{I} - \mathbf{P}_{\mathcal{S}})\mathbf{X}\boldsymbol{\beta}\|_2} \right|, \tag{44}$$

for $|\mathcal{S}| = s_1 = s \geq 0$, $\bar{\mathbf{Q}}_{\mathcal{S}} = [\bar{\mathbf{Q}}_{\mathcal{S}_1}^T, \ldots, \bar{\mathbf{Q}}_{\mathcal{S}_s}^T]^T$, where $\bar{\mathbf{Q}}_{\mathcal{S}_k}^T = \lambda\mathbf{U}_{\mathcal{S}_k}$, $\|\mathbf{U}_{\mathcal{S}_k}\|_1 = 1$. Let $\mathbf{A}_{\mathcal{S}} = \mathbf{X}_{\mathcal{S}}^*(\mathbf{X}_{\mathcal{S}}^T\mathbf{X}_{\mathcal{S}})^{-1}$, where $\mathbf{X}_k^* = \lambda\mathbf{X}_k$ for $k \in \mathcal{S}$. For a given $\mathcal{S}$, let $\mathbf{Z}_{lj} = (0, \ldots, 0, 1, 0, \ldots, 0)$ be the $|\mathcal{S}| \times 1$ vector with the jth element in the lth group being 1. Then, $\mathbf{U}_{\mathcal{S}} = \sum_{l\in\mathcal{S}}\sum_{j=1}^{d_l} u_{lj}Z_{lj}$ and $\sum_{j=1}^{d_l}|u_{lj}| = 1$. By the SRC, $\|\mathbf{A}_{\mathcal{S}}\mathbf{U}_{\mathcal{S}}\|_2^2 \geq \lambda^2 s/(nc^*(|\mathcal{S}|)\bar{d})$. Define $z = s\max_{l,j}\|A_{\mathcal{S}}\mathbf{Z}_{lj}\|_2/(\mathbf{A}_{\mathcal{S}}\mathbf{U}_{\mathcal{S}})$, then by the definition of $\mathbf{A}_{\mathcal{S}}$ and the SRC, we know $z \leq \sqrt{\bar{d}\widehat{c}s}$.

Thus by (44), we have

$$x_s^* \leq \max_{|\mathcal{S}|=s}\max_{l,j}\left\{ \left| \boldsymbol{\epsilon}^T \frac{\mathbf{A}_{\mathcal{S}}\mathbf{Z}_{lj}}{\|\mathbf{A}_{\mathcal{S}}\mathbf{Z}_{lj}\|_2} \right| \frac{s\|\mathbf{A}_{\mathcal{S}}\mathbf{Z}_{lj}\|_2}{\|\mathbf{A}_{\mathcal{S}}\mathbf{U}_{\mathcal{S}}\|_2} + \left| \frac{\boldsymbol{\epsilon}^T(\mathbf{I} - \mathbf{P}_{\mathcal{S}})\mathbf{X}\boldsymbol{\beta}}{\|(\mathbf{I} - \mathbf{P}_{\mathcal{S}})\mathbf{X}\boldsymbol{\beta}\|_2} \right| \right\}.$$

If we define $\Omega_{s'} = \{(\mathbf{U}, \boldsymbol{\epsilon}) : x_s^* \leq \sigma_\epsilon\sqrt{8(1 + c_0)z^2((s\underline{d}) \vee \underline{d})\log(m \vee a_n)}, \forall s \geq s'\}$, then

$$(\mathbf{X}, \boldsymbol{\epsilon}) \in \Omega_{s'} \Rightarrow |\boldsymbol{u}^T\boldsymbol{\epsilon}|^2 \leq (x_s^*)^2 \leq \frac{(|\mathcal{S}_1| \vee \underline{d})\lambda^2}{4\bar{d}nc^*}, \text{for } N(\mathcal{S}_1) \geq s' \geq 0.$$

By the definition of x_s^*, it is less than the maximum of $\binom{p}{s}\sum_{k\in\mathcal{S}} d_k$ normal variables with mean 0 and variance $\sigma_\epsilon^2 z_\epsilon^2$, plus the maximum of $\binom{p}{s}$ normal variables with mean 0 and variances σ_ϵ^2. It follows that $\mathcal{P}\{(\mathbf{X}, \boldsymbol{\epsilon}) \in \Omega_{s'}\} \to 1$ when (43) holds. This completes the sketch of the proof of Lemma 3 part (1).

(2) Consider the case when $\{c^*, c_*, c_0\}$ are fixed. Let $\mathcal{S}_1 = \{k : \|\widehat{\boldsymbol{\beta}}_k\|_\infty > 0 \text{ or } k \notin \mathcal{S}_0\}$. Define $\boldsymbol{v}_1 = \mathbf{X}_{\mathcal{S}_1}(\widehat{\boldsymbol{\beta}}_{\mathcal{S}_1} - \boldsymbol{\beta}_{\mathcal{S}_1})$ and $\boldsymbol{g}_1 = \mathbf{X}_{\mathcal{S}_1}^T(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}})$. We then have $\|\boldsymbol{v}_1\|_2^2 \geq c_*n\|\widehat{\boldsymbol{\beta}}_{\mathcal{S}_1} - \boldsymbol{\beta}_{\mathcal{S}_1}\|_2^2$, $(\widehat{\boldsymbol{\beta}}_{\mathcal{S}_1} - \boldsymbol{\beta}_{\mathcal{S}_1})^T\boldsymbol{g}_1 = \boldsymbol{v}_1^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{X}_{\mathcal{S}_1}\boldsymbol{\beta}_{\mathcal{S}_1} + \boldsymbol{\epsilon}) - \|\boldsymbol{v}_1\|_2^2$ and $\|\boldsymbol{g}_1\|_\infty \leq \max_{k,\|\widehat{\boldsymbol{\beta}}_k\|_\infty>0}\|\lambda\widehat{\boldsymbol{z}}_k\|_\infty = \lambda$. Therefore, $\|\boldsymbol{v}_1\|_2 \leq \|\mathbf{P}_{\mathcal{S}_1}\boldsymbol{\epsilon}\|_2 + \lambda\sqrt{N(\mathcal{S}_1)/(nc_*)}$. Since $\|\mathbf{P}_{\mathcal{S}_1}\boldsymbol{\epsilon}\|_2 \leq 2\sigma_\epsilon\sqrt{N(\mathcal{S}_1)\log m}$ with

probability converging to 1 under the normality assumption, $\|\mathbf{X}(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta})\|_2 \leq \|\mathbf{P}_{\mathcal{S}_1}\boldsymbol{\epsilon}\|_2 + \lambda\sqrt{N(\mathcal{S}_1)/(nc_*)}$. We then have

$$(\sum_{k\in\mathcal{S}_1}\|\widehat{\boldsymbol{\beta}}_k-\boldsymbol{\beta}_k\|_2^2)^{1/2} \leq \frac{\|\boldsymbol{v}_1\|_2}{\sqrt{nc_*}} \leq \frac{1}{\sqrt{nc_*}}(2\sigma_\epsilon\sqrt{N(\mathcal{S}_1)\log m}+\sqrt{C_1\widehat{c}}B_1).$$

It then follows that

$$\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_2 \leq \frac{1}{\sqrt{nc_*}}(2\sigma_\epsilon\sqrt{C_1 s^*\log m}+\sqrt{C_1\widehat{c}}B_1). \tag{45}$$

Then the result of part (2) follows by substituting the $B_1=\lambda\sqrt{s^*/(nc^*)}$ and $\lambda=\lambda_*$ into (45). This completes the sketch of the proof of Lemma 3 part (2). ■

C.2 Proof of Proposition 6

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

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

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

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

This implies that

$$\Lambda_{\min}(\Sigma_{\mathcal{S}}^{\mathbf{X},n'-1})\geq\frac{N'}{n'}\Lambda_{\min}(\Sigma_{\mathcal{S}}^{\mathbf{X},N'-1})\geq\frac{c_*}{c}, \tag{46}$$

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

$$\Lambda_{\max}(\Sigma_{\mathcal{S}}^{\mathbf{X},n'-1}) \leq \frac{N'}{n'}c^* + \frac{n'-N'}{n'}B \leq \max(c^*, B) = B. \tag{47}$$

Combining (46) and (47) completes the proof. ■

C.3 Proof of Theorem 7

We begin with the proof of part (1).

Theorem 7 assumes that $\mathbf{X}^{N'-1}$ satisfies the SRC$(C_3 s^*, c_*, c^*)$. By Proposition 6, we know that for all $\underline{c}N' \leq n' \leq \bar{c}N'$, the design matrix $\mathbf{X}^{n'-1}$ can satisfy the SRC$(C_3 s^*, c_*/\bar{c}, B)$. Thus the result of part (1) directly follows from part(1) of Lemma 3.

We now proceed to prove part (2). Throughout the proof, we let $c_*, c^*, c_0, \underline{c}, \bar{c}$, and B be fixed. We also let the bounds $[C_{\min}, C_{\max}]$ for truncating the eigenvalues of $\widehat{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}})^{(n+1)}$ be fixed positive constants, so in the following, the C_is are some positive constants depending only on these quantities. Let $\bar{\mathcal{S}} := \bigcap_{n'=N'}^{n} \mathcal{S}^{n'}$. In both Algorithm 1 and Algorithm 2, we approximately estimate $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n}$ by $\widetilde{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n}$, then from updating formulae in (15) and (14), we have

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

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

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

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

$$\begin{aligned}
\|\boldsymbol{\delta}_{\bar{\mathcal{S}}}^n\|_2 \leq \|\boldsymbol{\Sigma}_{\bar{\mathcal{S}}}^{\boldsymbol{\vartheta},n}\|_2 \Big[&\|(\boldsymbol{\Sigma}_{\bar{\mathcal{S}}}^{\boldsymbol{\vartheta},N'-1})^{-1}\|_2\|\boldsymbol{\delta}_{\bar{\mathcal{S}}}^{N'-1}\|_2 + \|[(\widetilde{\boldsymbol{\Sigma}}_{\mathcal{S}^{N'}}^{\boldsymbol{\vartheta},N'})^{-1}]_{\bar{\mathcal{S}}}\|_2\|\widehat{\boldsymbol{\delta}}_{\bar{\mathcal{S}}}^{N'}\|_2 + \\
&\cdots + \|[(\widetilde{\boldsymbol{\Sigma}}_{\mathcal{S}^n}^{\boldsymbol{\vartheta},n})^{-1}]_{\bar{\mathcal{S}}}\|_2\|\widehat{\boldsymbol{\delta}}_{\bar{\mathcal{S}}}^{n}\|_2\Big].
\end{aligned}$$

We now proceed to bound each of the quantities. For now let n' be an index satisfying $N' \leq n' \leq n$. As we suppose the design matrix for the Lasso solution $\widehat{\boldsymbol{\vartheta}}_{\mathcal{S}}^{N'}$ satisfies the SRC $(C_3 s^*, c_*, c^*)$, by Proposition 6 and Lemma 3, if we choose $\lambda^{n'} = \lambda_*$ and

$$\lambda^{n'} = O(\bar{d}\sqrt{n' \log m}), \tag{48}$$

then there exists some constant C_6 such that

$$\|\widehat{\boldsymbol{\delta}}^{n'}_{\bar{\mathcal{S}}}\|_2 \leq C_6\sigma_\epsilon\bar{d}\sqrt{\frac{s^*\log m}{n'}}, \quad \text{for all } \underline{c}N' \leq n' \leq n, \tag{49}$$

with probability converging to 1. We know from (29) that $\widetilde{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},n'}_{\mathcal{S}^{n'}}$ is computed by:

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

where

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

The $\mathrm{SRC}(C_3s^*, c_*, c^*)$ gives us

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

for any $\mathcal{S}$ with $|\mathcal{S}| = C_3s^*$. Therefore, since $|\mathcal{S}^{n'}| \leq C_3s^*$, which is proved in part (1), by Proposition 6, we can show that for all $N' \leq n' \leq n$, there exist positive constants C_7 and C_8, such that

$$\Lambda_{\max}(\mathbf{M}^{n'-1}_{\mathcal{S}^{n'}}) \leq \frac{C_7}{n'} < \infty, \tag{50}$$

$$\Lambda_{\min}(\mathbf{M}^{n'-1}_{\mathcal{S}^{n'}}) \geq \frac{C_8}{n'} > 0. \tag{51}$$

It is not hard to prove

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

for any positive semidefinite matrices $\mathbf{M}$ and $\mathbf{N}$, so using Weyl's inequality in matrix theory, (28), and (51), we have the following bound,

$$\begin{aligned}\|[(\widetilde{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},n'}_{\mathcal{S}^{n'}})^{-1}]_{\bar{\mathcal{S}}}\|_2 &\leq \|(\widetilde{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},n'}_{\mathcal{S}^{n'}})^{-1}\|_2 = \Lambda^{-1}_{\min}(\widetilde{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},n'}_{\mathcal{S}^{n'}})\\ &\leq \frac{1}{\Lambda_{\min}(\sigma^2_\epsilon\mathbf{M}^{n'-1}_{\mathcal{S}^{n'}}) + (\lambda^{n'})^2\Lambda_{\min}\widetilde{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}^{n'}})^{n'}\Lambda^2_{\min}(\mathbf{M}^{n'-1}_{\mathcal{S}^{n'}})}\\ &\leq \frac{C_9n'}{\sigma^2_\epsilon\bar{d}^2\log m},\end{aligned} \tag{52}$$

for some constant C_9. Similarly, by (48), (50), and (28), we can also get

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

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

$$\begin{aligned}
\|\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}_{\bar{\mathcal{S}}}\|_2 &= \Lambda^{-1}_{\min}\left[(\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},N'-1}_{\bar{\mathcal{S}}})^{-1} + [(\widetilde{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},N'}_{\mathcal{S}^{N'}})^{-1}]_{\bar{\mathcal{S}}} + \cdots + [(\widetilde{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},n}_{\mathcal{S}^{n}})^{-1}]_{\bar{\mathcal{S}}}\right] \\
&\leq \frac{1}{\Lambda_{\min}\left[[(\widetilde{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},N'}_{\mathcal{S}^{N'}})^{-1}]_{\bar{\mathcal{S}}}\right] + \cdots \Lambda_{\min}\left[(\widetilde{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},n}_{\mathcal{S}^{n}})^{-1}\right]_{\bar{\mathcal{S}}}} \\
&= \frac{1}{\Lambda^{-1}_{\max}(\widetilde{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},N'}_{\mathcal{S}^{N'}}) + \cdots \Lambda^{-1}_{\max}(\widetilde{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},n}_{\mathcal{S}^{n}})} \\
&\leq \frac{2C_{10}\sigma^2_\epsilon \bar{d}^2 \log m}{(N'+n)(n-N'+1)} \\
&\leq \frac{C_{11}\sigma^2_\epsilon \bar{d}^2 \log m}{n^2}, \qquad (53)
\end{aligned}$$

for some constant C_{11}. If we let

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

then combining (49),(52), and (53) gives us the following bound on $\boldsymbol{\delta}^n_{\bar{\mathcal{S}}}$

$$\begin{aligned}
\|\boldsymbol{\delta}^n_{\bar{\mathcal{S}}}\|_2 &\leq \frac{C_{11}\sigma^2_\epsilon \bar{d}^2 \log m}{n^2}\left(\Delta_{\bar{\mathcal{S}}}(N') + \sum_{n'=N'}^{n} \frac{C_6 C_9 \sqrt{s^* n'}}{\sigma_\epsilon \bar{d}\sqrt{\log m}}\right) \\
&\leq \frac{C_{12}\sigma_\epsilon \bar{d}\sqrt{s^* \log m}}{\sqrt{n}} + \frac{C_{11}\sigma^2_\epsilon \bar{d}^2 \log m \Delta_{\bar{\mathcal{S}}}(N')}{n^2}, \qquad (54)
\end{aligned}$$

for some constant C_{12}. After dropping off the higher order term, (54) is equivalent to

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

and thus completes the proof. ■

C.4 Proof of Theorem 10

By definition of f_j, $1 \leq j \leq p$, part (1) follows from part (2) of Theorem 7 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

$$f^n_j(x) = \sum_{k=1}^{d_j} \vartheta^n_{jk}\psi_{jk}(x), \quad \text{for } 1 \leq j \leq p.$$

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 9 and Lemma 8 in Stone (1986), taking $q = 2$, we have

$$\|\widetilde{f}_j^* - f_j\|^2 \leq C_{13} d_j^{-2q} = C_{13} d_j^{-4},$$

where C_{13} is some fixed positive constant. Thus by the result of Theorem 7, we have

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

Note that choosing $\bar{d} = O(n^{1/6})$ would not change the rate in equation (54), so we have the following bound

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

■

Appendix D. More Empirical Results

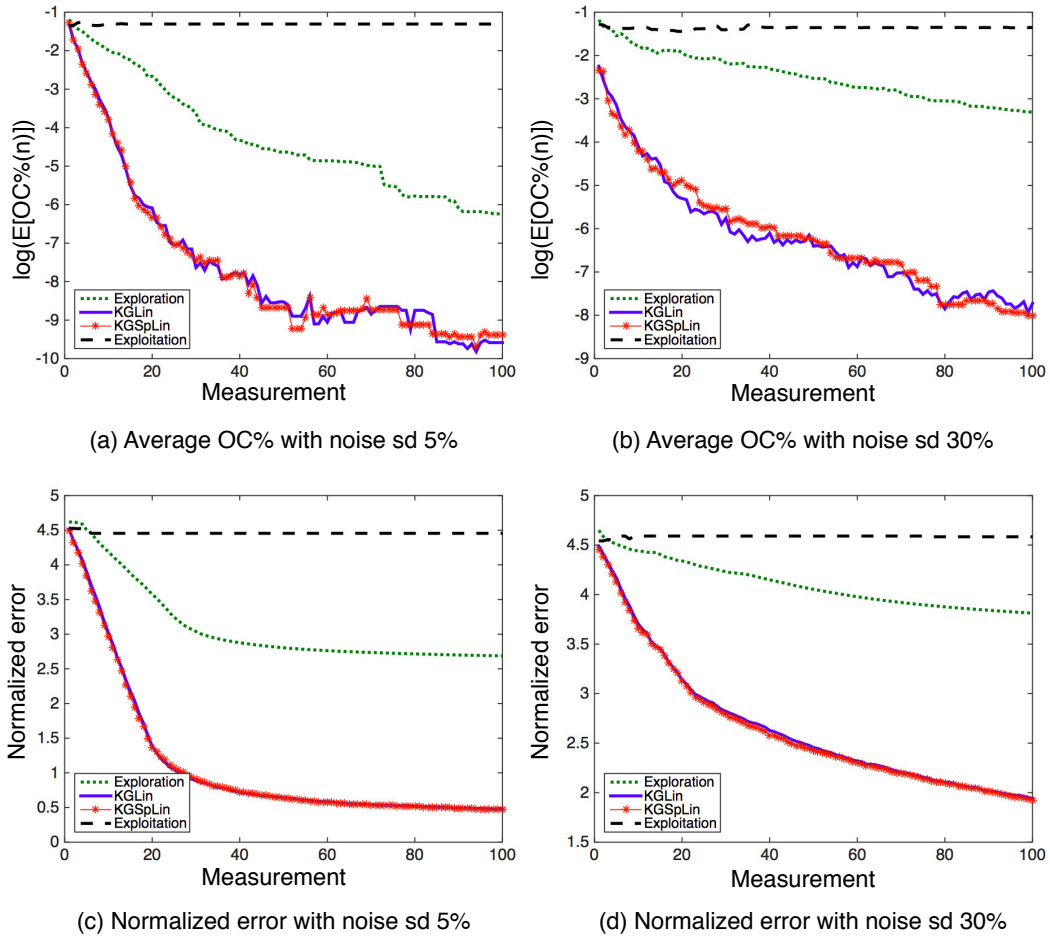
In this Section, we show some more detailed empirical results.

		KGSpLin			**KGLin**		
Test function	σ_ϵ	$\mathbb{E}$(OC)	σ(OC)	Median	$\mathbb{E}$(OC)	σ(OC)	Median
Matyas	1	**.0082**	.0231	**.0068**	.0263	.0146	.0238
$\mathcal{X} = [-10, 10]^2$	10	**.1764**	.1927	**.0103**	.3084	.1132	0.3636
	20	**.5893**	.8283	**.2530**	1.5475	.3102	1.2859
Six-hump Camel	1	**.0016**	.0017	**.0000**	.0106	.6775	**.0000**
$\mathcal{X} = [-3, 3] \times [-2, 2]$	10	**.0750**	.6283	**.0000**	.1138	.5273	**.0000**
	20	**.3282**	.2031	**.0187**	.5836	.2537	0.0285
Bohachevsky	1	**.0513**	.0187	**.0010**	.0787	.0341	.0012
$\mathcal{X} = [-100, 100]^2$	10	**.2359**	2.0965	**.2166**	.4174	2.5327	.2732
	20	**1.2406**	2.8345	**1.2084**	1.8940	3.4507	1.6287
Trid	1	**1.7328**	1.2806	**0.8685**	2.4271	1.4716	1.1243
$d = 6, \mathcal{X} = [-36, 36]^6$	10	**7.2585**	3.4074	**7.0487**	9.1764	4.0085	7.8952
	20	**10.1484**	3.8105	**9.9850**	14.1521	4.2179	13.9403

Table 4: Quantitative comparison for KGSpLin and KGLin on standard test functions excluding the initial 10 measurements.

First, we compare KGSpLin with KGLin on data which is *not* sparse. Here we consider a similar model as used in the first set of experiments in Section 7.1. In the non-sparse

setting, we only take the nonzero dimension of the function. That is we let $\mu = \sum_{j=1}^{m} \alpha_j x_j + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Here $m = 20$. For $j = 1, \ldots, m$, let α_j be independently drawn from $\mathcal{N}(\vartheta_j, \Sigma^{\boldsymbol{\vartheta}}_{jj})$, where $\vartheta_j = j + 10$, and $\Sigma^{\boldsymbol{\vartheta}}_{jj} = (2\vartheta_j)^2$. The prior is also independently sampled from the same distribution. The difference is that here we take larger standard deviations so that the sampled truth is significantly different from the prior. In this way, we can better visualize the performance of different algorithms as more observations are made. Then we uniformly sample $M = 100$ alternatives from $[0, 1]^m$. For KGSpLin, the tuning parameter λ^n is chosen to be a relatively small number 10^{-2} and remains fixed as n becomes large. As before, figure 10 shows the log of the averaged OC% and the normalized estimation error of $\boldsymbol{\vartheta}$ over 300 replications with low and high measurement noise. The standard deviations of the measurement noise are respectively 5% and 30% of the expected range of the truth.



(a) Average OC% with noise sd 5%

(b) Average OC% with noise sd 30%

(c) Normalized error with noise sd 5%

(d) Normalized error with noise sd 30%

Figure 10: (a)(b) and (c)(d) compares exploration, exploitation, KGLin, and KGSpLin for a non-sparse model by showing the averaged OC% and normalized estimation errors over 300 runs under low measurement noise (5% range of the truth) and high measurement noise (30% range of the truth).

As one can see from Figure 10, in the non-sparse setting, KGSpLin does not make erroneous conclusions of sparsity with a relatively small value of λ. KGSpLin performs competitively with KGLin in both the opportunity cost and estimation error. For this set of experiments, we can see that there is almost no loss even if we make approximations in the KG computation as well as the Bayesian update as describe in Section 4.

Second, for the experiments of comparing KGSpLin and KGLin on canonical test functions, Table 4 gives the sample means, sample standard deviations and sample medians of the opportunity cost without the first 10 iterations for each policy. One can see that given some initial samples to identify the true suppport, KGSpLin does significantly better than KGLin for all the test functions.

References

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

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.

Alexandre Belloni and Victor Chernozhukov. $\ell 1$-penalized quantile regression in high-dimensional sparse models. *The Annals of Statistics*, 39(1):82–130, 2011.

Alexandre Belloni and Victor Chernozhukov. Least squares after model selection in high-dimensional sparse models. *Bernoulli*, 19(2):521–547, 2013.

Peter J Bickel, Ya'acov Ritov, and Alexandre B Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, pages 1705–1732, 2009.

Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Florentina Bunea, Alexandre Tsybakov, and Marten Wegkamp. Sparsity oracle inequalities for the Lasso. *Electronic Journal of Statistics*, 1:169–194, 2007.

Alexandra Carpentier and Rémi Munos. Bandit theory meets compressed sensing for high dimensional linear bandit. *International Conference on Artificial Intelligence and Statistics*, pages 190–198, 2012.

Thomas R Cech, Arthur J Zaug, and Paula J Grabowski. In vitro splicing of the ribosomal RNA precursor of Tetrahymena: involvement of a guanosine nucleotide in the excision of the intervening sequence. *Cell*, 27(3):487–496, 1981.

Bo Chen, Rui Castro, and Andreas M Krause. Joint optimization and variable selection of high-dimensional Gaussian processes. In *Proceedings of the 29th International Conference on Maching Learning (ICML-12)*, pages 1423–1430, 2012a.

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 *Advances in Neural Information Processing Systems*, volume 25, pages 404–412, 2012b.

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

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

Josip Djolonga, Andreas Krause, and Volkan Cevher. High-dimensional Gaussian process bandits. In *Advances in Neural Information Processing Systems*, pages 1025–1033, 2013.

David L Donoho. For most large underdetermined systems of linear equations the minimal $\ell 1$-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59(6):797–829, 2006.

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 *Advances in Neural Information Processing Systems*, 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.

Yan Li, Jorge Vazquez-Anderson, Yingfei Wang, Lydia Contreras, and Warren B. Powell. A knowledge gradient policy for sequencing experiments to identify the structure of RNA molecules using a sparse additive belief model. 2015. in prepration.

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 nonparametric regression. *The Annals of Statistics*, 34(5):2272–2297, 2006.

Han Liu and Jian Zhang. On ℓ1- ℓq regularized regression. Technical report, Citeseer, 2008.

Karim Lounici, Massimiliano Pontil, Sara Van De Geer, and Alexandre B Tsybakov. Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics*, pages 2164–2204, 2011.

Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, pages 246–270, 2009.

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.

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.

Herbert Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer, 1985.

Rick Russell, Rhiju Das, Hyejean Suh, Kevin J Travers, Alain Laederach, Mark A Engelhardt, and Daniel Herschlag. The paradoxical behavior of a highly structured misfolded intermediate in RNA folding. *Journal of Molecular Biology*, 363(2):531–544, 2006.

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

Steven W Sowa, Jorge Vazquez-Anderson, Chelsea A Clark, Ricardo De La Peña, Kaitlin Dunn, Emily K Fung, Mark J Khoury, and Lydia M Contreras. Exploiting post-transcriptional regulation to probe RNA structures in vivo via fluorescence. *Nucleic Acids Research*, page gku1191, 2014.

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.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

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

Sara A Van De Geer, Peter Bühlmann, et al. On the conditions used to prove oracle results for the Lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009.

Jorge Vazquez-Anderson and Lydia M Contreras. Regulatory RNAs: charming gene management styles for synthetic biology applications. *RNA Biology*, 10(12):1778–1797, 2013.

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.

Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, and Nando De Freitas. Bayesian optimization in high dimensions via random embeddings. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1778–1784. AAAI Press, 2013.

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.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Cun-Hui Zhang and Jian Huang. The sparsity and bias of the Lasso selection in high-dimensional linear regression. *The Annals of Statistics*, pages 1567–1594, 2008.

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 and Bin Yu. On model selection consistency of Lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.

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.

Hui Zou. The adaptive Lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.