A Lasso-based Sparse Knowledge Gradient Policy for Sequential Optimal Learning

Yan Li
Operations Research and
Financial Engineering,
Princeton University

Han Liu
Operations Research and
Financial Engineering,
Princeton University

Warren B. Powell
Operations Research and
Financial Engineering,
Princeton University

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 sparse knowledge gradient (SpKG) decision policy based on the ℓ_1-penalized regression Lasso to identify the sparsity pattern before our budget is exhausted. This policy is a unique and novel hybrid of Bayesian R&S with a frequentist learning approach. Theoretically, we provide the error bound of the posterior mean estimate, which has shown to be at the minimax optimal $\sqrt{s \log p/n}$ rate. Controlled experiments on both synthetic data and real application for automatically designing experiments to identify the structure of an RNA molecule show that the algorithm efficiently learns the correct set of nonzero parameters. It also outperforms several other learning policies.

1 INTRODUCTION

The sequential optimal learning 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. Furthermore, our experiments are time-consuming and expensive, forcing us to learn quickly within a finite budget. Specifically, in this paper, we are maximizing an unknown function $\mu_x : x \in \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X} = \{1, \ldots, M\}$ is a finite set with M alternatives. Our problem is to carefully identify which experiment to perform next so that we can identify the best alternative when our budget is exhausted. Also we assume that the objective function μ cannot be written in closed form and does not have easily available derivatives. These problems have been studied in different communities, which refer to the problem under names such as: Bayesian optimization (Brochu et al., 2010b), experimental design, bandits (Robbins, 1985), and optimal learning (Powell and Ryzhov, 2012).

In the artificial intelligence community, we have witnessed many important advances in sequential learning in applications such as interactive animation (Brochu et al., 2010a), autonomous robots (Martinez-Cantin et al., 2009), and automatic algorithm configuration (Snoek et al., 2012), but these applications are typically restricted to problems of moderate dimension (e.g. up to a few dozen). By contrast, our work is motivated by an important application to discover the structure of an RNA molecule (Vazquez-Anderson and Contreras, 2013; Sowa et al., 2014), where the dimension of the problem is equal to the length of an RNA molecule (~ 400). Our objective is to identify regions of the RNA molecule that are accessible to chemical interactions, which is evaluated by the fluorescence of a marker molecule. We need to learn coefficients that capture accessibility; since there are relatively few of these regions, our model will be sparse.

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

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors.

or even millions of alternatives using a low-dimensional model. Let $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_M]^T \in \mathbb{R}^M$ be the vector representing values of all alternatives. Linear beliefs assume the truth $\boldsymbol{\mu}$ can be represented as a linear combination of a set of parameters, that is, $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\alpha}$. Here $\boldsymbol{\alpha} \in \mathbb{R}^p$ is the underlying coefficient, and $\mathbf{X} \in \mathbb{R}^{M \times p}$ is the design matrix, where each row is a feature vector corresponding to a particular experiment.

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 zeros. Sparsity is a property that appears in a plethora of natural as well as man-made systems. In such 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 using just a few dozen experiments. 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 sparse knowledge gradient (SpKG) policy from the knowledge gradient (KG) policy proposed by Frazier et al. (2008). KG 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. Second, for the learning procedure, our algorithm adopts the frequentist homotopy recursive Lasso approach (Garrigues and El Ghaoui, 2008) to update the Lasso estimates. Then we directly use these 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.

Our contributions are the following:

- We provide an algorithm, called SpKG that mixes Lasso and KG. We prove that the mean of the posterior coefficient estimate converges to the truth at the minimax optimal $\sqrt{s \log p/n}$ rate (Raskutti et al., 2011). This also guarantees global convergence to the optimal design.
- We show how SpKG can be used for automatic experimental design by applying it to an important application to discover the structure of an RNA molecule, which is a high-dimensional learning problem. Empirically, SpKG performs significantly better than several other policies, especially when the measurement budget is much smaller than either the number of alternatives to be tested, as well as being much smaller than the number of parameters in the model.

There is another line of research on multi-armed bandits where the objective is to maximize the cumulative rewards (online learning), whereas our work addresses offline learning that occurs in a laboratory setting. For sparse linear bandits, Carpentier and Munos (2012) combine compressed sensing with the UCB (Upper Confidence Bound) policy to attack problems with a high degree of sparsity. Djolonga et al. (2013) propose an algorithm, leveraging low-rank matrix recovery techniques to learn the underlying low-dimensional space and applying UCB to optimize the function. For offline learning, Chen et al. (2012) propose a two stage strategy for high-dimensional Gaussian Process (GP). In the first stage, a hierarchical diagonal sampling (HDS) approach based on likelihood ratio tests is used to select relevant dimensions. Then the GP-UCB policy (Srinivas et al., 2010) is applied to optimize over the variables deemed relevant. Wang et al. (2013) assume noise-free function evaluations and uses random embeddings in Bayesian optimization to optimize high dimensional sparse functions. However, both the UCB based policies and the random embeddings ideas require hundreds or even thousands of measurements to achieve satisfactory convergence results, thus they may not be realistic in our settings where we are usually given dozens of measurement budgets.

2 BACKGROUND

We start with the following 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 let $\boldsymbol{v}_I$ be the subvector of $\boldsymbol{v}$ whose entries are indexed by I. We also denote $\mathbf{M}_{IJ}$ 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. For $0 < q < \infty$, we define the ℓ_0, ℓ_q vector norms as $\|v\|_0 := \text{card}(\text{supp}(v))$ and $\|v\|_q := (\sum_{i=1}^d |v_i|^q)^{1/q}$.

2.1 Bayesian Ranking and Selection

We are maximizing an unknown performance metric $\mu_x : x \in \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X}$ is a finite set with M alternatives (which may be quite large). We have to learn μ_x using a relatively small budget of N measurements. We assume we have a Bayesian prior $\boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma})$. Now we need to choose a measurement *policy* $(x^0, x^1, \ldots, x^{N-1})$ to learn about these alternatives, where $x^i \in \mathcal{X}$. At time n, if we measure alternative x, we observe $y_x^{n+1} = \mu_x + \epsilon_x^{n+1}$, where

$\epsilon_x^{n+1} \sim \mathcal{N}(0, \sigma_\epsilon^2)$, and σ_ϵ is known.

For a lookup table belief model (Frazier et al., 2009), let $\mathcal{N}(\boldsymbol{\theta}^n, \boldsymbol{\Sigma}^n)$ be the prior distribution on $\boldsymbol{\mu}$ at time n. When the measurement budget of N is exhausted, our goal is to find the optimal alternative $x^N = \text{argmax}_{x \in \mathcal{X}} \theta_x^N$. We define Π to be the set of all admissible policies. Let $\mathbb{E}^\pi$ indicate the expectation over both the noisy outcomes and the truth $\boldsymbol{\mu}$ while the sampling policy is fixed to $\pi \in \Pi$. Our goal is to choose a measurement policy maximizing the expected reward, which can be written as $\sup_{\pi \in \Pi} \mathbb{E}^\pi \left[\max_{x \in \mathcal{X}} \theta_x^N\right]$. In the Bayesian setting, we can sequentially update the mean and covariance estimates of the alternatives by the following Bayesian updating equations (Gelman et al., 2003):

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

$$\boldsymbol{\Sigma}^{n+1} = \boldsymbol{\Sigma}^n - \frac{\boldsymbol{\Sigma}^n \boldsymbol{e}_x \boldsymbol{e}_x^T \boldsymbol{\Sigma}^n}{\sigma_\epsilon^2 + \Sigma_{xx}^n}, \tag{2}$$

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

For the linear belief model (Negoescu et al., 2011), we assume $\boldsymbol{\mu}$ can be represented as a linear combination of a set of parameters, that is, $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\alpha}$, where $\boldsymbol{\mu} \in \mathbb{R}^M$ and $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_p]^T \in \mathbb{R}^p$ are random variables. Recall that $\mathbf{X} \in \mathbb{R}^{M \times p}$ is the design matrix, where each row is a feature vector corresponding to an alternative. Now we assume $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\vartheta}, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}})$. It induces a normal distribution on $\boldsymbol{\mu}$ via linear transformation, that is $\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\vartheta}, \mathbf{X}\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}}\mathbf{X}^T)$. Then the belief on $\boldsymbol{\alpha}$ can also be recursively updated via recursive least squares (see Powell and Ryzhov, 2012, p. 187). Linear belief is a more compact representation since we only need to maintain beliefs on the parameter space.

2.2 Knowledge Gradient Policy

The knowledge gradient policy is a myopic one-step lookahead policy (Frazier et al., 2008). 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 KG value of x is defined as:

$$v_x^{\text{KG},n} = \mathbb{E}(\max_{x' \in \mathcal{X}} \theta_{x'}^{n+1} | S^n, x^n = x) - \max_{x' \in \mathcal{X}} \theta_{x'}^n. \tag{3}$$

At each time, the KG policy chooses the one with the maximum KG value. The computation of the expectation in (3) can be computed using the methods described in Frazier et al. (2009). First notice that equation (1) can be rearranged as the time n conditional distribution of $\boldsymbol{\theta}^{n+1}$, namely, $\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n + \widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, x^n)Z$, where $\widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, x)$ is a vector-valued function defined as $\widetilde{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^n, x) := \boldsymbol{\Sigma}^n \boldsymbol{e}_x / \sqrt{\sigma_\epsilon^2 + \Sigma_{xx}^n}$, and Z follows standard normal distribution.

Then we substitute this equation into the KG formula and get that

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

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. In light of this, the expectation in the KG formula can be computed as the point-wise maximum of the affine functions $a_i + b_i Z$. Frazier et al. (2009) provide an algorithm to compute the value of function h with complexity of $O(M^2 log(M))$. Notice that for the linear belief, we can replace $(\boldsymbol{\theta}^n, \boldsymbol{\Sigma}^n)$ with $(\mathbf{X}\boldsymbol{\vartheta}, \mathbf{X}\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}}\mathbf{X}^T)$.

3 LASSO-BASED SPARSE KNOWLEDGE GRADIENT

In this section we propose an adaptation of the knowledge gradient (SpKG) to handle sparse additive belief models. We respectively cover the searching and learning components as described in the introduction, specifically, the computation of SpKG and the Bayesian updating rules. We begin by presenting the Bayesian belief model to handle sparsity.

As before, let us assume $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\alpha}$, where $\mathbf{X} \in \mathbb{R}^{M \times p}$ is the design matrix, and $\boldsymbol{\alpha} \in \mathbb{R}^p, \boldsymbol{\mu} \in \mathbb{R}^M$ are random variables. Our problem setting is that p 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 in the Bayesian setting; that is, the indicator variable of which dimension is selected or not is a random vector. Specifically, let $\boldsymbol{\zeta} = [\zeta_1, \ldots, \zeta_p]^T \in \mathbb{R}^p$ be the indicator random variable of $\boldsymbol{\alpha}$. That is, let $\zeta_j = 1$ for $\alpha_j \neq 0$, and $\zeta_j = 0$ for $\alpha_j = 0$. Additionally, we assume that $\boldsymbol{\alpha} | \boldsymbol{\zeta} \sim \mathcal{N}(\boldsymbol{\vartheta}, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta}})$. (We leave the dependence of $\boldsymbol{\vartheta}$ and $\boldsymbol{\Sigma}^{\boldsymbol{\vartheta}}$ on $\boldsymbol{\zeta}$ implicit to simplify notation.) Without loss of generality, conditioning on $\boldsymbol{\zeta}$, we can permute the elements of $\boldsymbol{\alpha}$ to create the partition $\boldsymbol{\alpha}^T = [(\boldsymbol{\alpha}_{\mathcal{S}})^T, \mathbf{0}]$, where $\boldsymbol{\alpha}_{\mathcal{S}} \sim \mathcal{N}(\boldsymbol{\vartheta}_{\mathcal{S}}, \boldsymbol{\Sigma}_{\mathcal{S}}^{\boldsymbol{\vartheta}})$, and $\mathcal{S} = \{j : \zeta_j = 1\}$. The mean and covariance of $\boldsymbol{\alpha}$ have zeros indicated by $\boldsymbol{\zeta}$. 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 $\boldsymbol{\zeta}'$ and $\mathcal{S}' = \{j : \zeta_j' = 1\}$, the conditional distribution of $\boldsymbol{\alpha}$ on $\boldsymbol{\zeta}'$ is normal with mean $\boldsymbol{\vartheta}_{\mathcal{S}'}$ and covariance $\boldsymbol{\Sigma}_{\mathcal{S}'}^{\boldsymbol{\vartheta}}$. This means that we can write all the conditional distributions of $\boldsymbol{\alpha}$ through an index set $\mathcal{S}$ characterized by $\boldsymbol{\zeta}$. Through all the updatings, we just need to maintain the mean and covariance matrix on the coefficients where $\boldsymbol{\zeta} = \mathbf{1}$.

One may think of $\boldsymbol{\zeta}$ and $\boldsymbol{\alpha}$ as fixed and of $\boldsymbol{\zeta}^n$ and $\boldsymbol{\vartheta}^n_{\mathcal{S}}$ as converging toward $\boldsymbol{\zeta}$ and $\boldsymbol{\alpha}$, while some norm of the precision matrix $(\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}_{\mathcal{S}})^{-1}$ goes to infinity under some appropriate sampling strategy. It is also appropriate, however, to fix $\boldsymbol{\zeta}^n$ and $\boldsymbol{\vartheta}^n_{\mathcal{S}}$ 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 that 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.

3.1 Knowledge Gradient Policy for Sparse Linear Model

We begin by describing the Bayesian model at time n. We can maintain Beta-Bernoulli conjugate priors on each component of $\boldsymbol{\zeta}$. At time n, we have the following Bayesian model, for $j, j' = 1, \ldots, p$,

$$\boldsymbol{\alpha} | \boldsymbol{\zeta}^n = \mathbf{1} \sim \mathcal{N}(\boldsymbol{\vartheta}^n, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}), \tag{4}$$

$$\zeta^n_j | p^n_j \sim \text{Bernoulli}(p^n_j), \tag{5}$$

$$\zeta^n_j \perp \zeta^n_{j'}, \text{ for } j \neq j', \tag{6}$$

$$p^n_j | \xi^n_j, \eta^n_j \sim \text{Beta}(\xi^n_j, \eta^n_j), \tag{7}$$

where p_j is the probability of the jth feature being in the model, and (ξ_j, η_j) are the shape parameters for the corresponding Beta distribution. 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 SpKG, 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 SpKG value can be computed by:

$$\begin{aligned} & v^{\text{KG},n}_x \\ &= \mathbb{E}_{\boldsymbol{\alpha},\epsilon,\boldsymbol{\zeta}^{n+1},\boldsymbol{p}^{n+1}}(\max_{x' \in \mathcal{X}} \theta^{n+1}_{x'} | S^n, x^n = x, \boldsymbol{\zeta}^n, \boldsymbol{p}^n) - \max_{x' \in \mathcal{X}} \theta^n_{x'} \\ &\approx \mathbb{E}_{\boldsymbol{p}^n} \mathbb{E}_{\boldsymbol{\zeta}^n | \boldsymbol{p}^n} \mathbb{E}_{\boldsymbol{\alpha},\epsilon | \boldsymbol{\zeta}^n, \boldsymbol{p}^n}(\max_{x' \in \mathcal{X}} \theta^{n+1}_{x'} | S^n, x^n = x, \boldsymbol{\zeta}^n, \boldsymbol{p}^n) \\ &\qquad - \max_{x' \in \mathcal{X}} \theta^n_{x'} \\ &= \sum_{k=1}^{N_{\boldsymbol{\zeta}}} \mathbb{E}_{\boldsymbol{p}^n}(p^{n,k}) h(\boldsymbol{a}^{n,k}, \boldsymbol{b}^{n,k}) \\ &= \sum_{k=1}^{N_{\boldsymbol{\zeta}}} \prod_{\{j:\zeta^{n,k}_j = 1\}} \frac{\xi^n_j}{\xi^n_j + \eta^n_j} \prod_{\{j:\zeta^{n,k}_j = 0\}} \frac{\eta^n_j}{\xi^n_j + \eta^n_j} h(\boldsymbol{a}^{n,k}, \boldsymbol{b}^{n,k}), \end{aligned} \tag{8}$$

where h is the function defined in Section 2.2 and thus can be computed, $\boldsymbol{a}^{n,k} = \mathbf{X}^n_{\boldsymbol{\zeta}^{n,k}} \boldsymbol{\vartheta}^n_{\boldsymbol{\zeta}^{n,k}}$, and $\boldsymbol{b}^{n,k} = \widetilde{\boldsymbol{\sigma}}(\mathbf{X}^n_{\boldsymbol{\zeta}^{n,k}} \boldsymbol{\Sigma}^{n,\boldsymbol{\vartheta}}_{\boldsymbol{\zeta}^{n,k}} (\mathbf{X}^n_{\boldsymbol{\zeta}^{n,k}})^T, x)$.

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 SpKG calculation is identical with KG. Therefore we have shown that the SpKG value is a weighted summation over all the possible sample realizations of $\boldsymbol{\zeta}^n$. The weights $\mathbb{E}_{\boldsymbol{p}^n}(p^{n,k})$ are computed by the independent Beta distributions on all the p^n_j's. Besides, 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. Figure 4 in Section 5 shows that we do not lose much by making these approximations. The SpKG value still serves as a reasonable sampling criterion based on value of information.

3.2 Bayesian Update

At time n we have the Bayesian model described in (4)-(7). Parallel with that, we use Lasso as a "solver" to generate estimates of linear coefficients as well as the sparsity pattern. The Lasso estimator after n observations is given by:

$$\widehat{\boldsymbol{\vartheta}}^n = \operatorname*{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^p} \frac{1}{2} \sum_{i=1}^{n} \left[(\boldsymbol{x}^{i-1})^T \boldsymbol{\beta} - y^i \right]^2 + \lambda^n \|\boldsymbol{\beta}\|_1, \tag{9}$$

where $(y^i, \boldsymbol{x}^{i-1}) \in \mathbb{R} \times \mathbb{R}^p, i = 1, \ldots, n$ are the n observations, and λ^n is the regularization parameter. Here we recursively solve the Lasso problem based on the homotopy algorithm proposed by Garrigues and El Ghaoui (2008). The algorithm is an exact update of the Lasso solution when one additional observation is achieved. It satisfies the sequential setting here and could reduce the computational complexity from $O(p^3)$ to $O(p \log(p))$ compared to the direct solution of (9).

Next we need to sample a covariance matrix $\widehat{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},n}$ corresponding to the Lasso estimate. Let $(\mathbf{X}^n)^T := [\boldsymbol{x}^0, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^n]$, $\mathbf{Y}^{n+1} := [y^1, \ldots, y^{n+1}]^T$, and $\boldsymbol{z} \in \partial \|\widehat{\boldsymbol{\vartheta}}\|_1$. By the KKT conditions, we know that $\widehat{\boldsymbol{\vartheta}}^{n+1}_{\mathcal{S}} = [(\mathbf{X}^n_{*\mathcal{S}})^T \mathbf{X}^n_{*\mathcal{S}}]^{-1} [(\mathbf{X}^n_{*\mathcal{S}})^T \mathbf{Y}^{n+1} - \lambda^{n+1} \boldsymbol{z}^{n+1}_{\mathcal{S}}]$. We let $\mathbf{M}^n_{\mathcal{S}} := [(\mathbf{X}^n_{*\mathcal{S}})^T \mathbf{X}^n_{*\mathcal{S}}]^{-1}$. Since $\text{Cov}(\mathbf{Y}^{n+1}) = \sigma^2_\epsilon \mathbf{I}$, we have

$$\widehat{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},n+1}_{\mathcal{S}} = \mathbf{M}^n_{\mathcal{S}} \sigma^2_\epsilon + (\lambda^{n+1})^2 \mathbf{M}^n_{\mathcal{S}} \text{Cov}(\boldsymbol{z}_{\mathcal{S}})^{n+1} \mathbf{M}^n_{\mathcal{S}}. \tag{10}$$

Note that we cannot directly compute $\widehat{\boldsymbol{\Sigma}}^{\boldsymbol{\vartheta},n+1}_{\mathcal{S}}$ from the right hand side of (10), since $\boldsymbol{z}_{\mathcal{S}}$ is also a random variable dependent on $\widehat{\boldsymbol{\vartheta}}^{n+1}_{\mathcal{S}}$. But assuming that $\widehat{\boldsymbol{\vartheta}}^{n+1}_{\mathcal{S}}$ should not be far from $\boldsymbol{\vartheta}^n_{\mathcal{S}}$, one can sample a set of random variables from the distribution $\mathcal{N}(\boldsymbol{\vartheta}^n_{\mathcal{S}}, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n}_{\mathcal{S}})$

and then sample the subgradients to get $\widehat{\mathrm{Cov}}(\boldsymbol{z}_{\mathcal{S}})^{(n+1)}$. Additionally, to make this estimator stable in theory (see Appendix B for detailed proof), 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. 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 $[C_{\min}, C_{\max}]$.

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 $\mathcal{P}^n := \{j : \widehat{\vartheta}_j^n \neq 0\}$. The updating equations are given by:

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

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

$$\xi_j^{n+1} = \xi_j^n + 1, \eta_j^{n+1} = \eta_j^n, \quad \text{for} \quad j \in \mathcal{P}^{n+1}, \tag{13}$$

$$\xi_j^{n+1} = \xi_j^n, \eta_j^{n+1} = \eta_j^n + 1, \quad \text{for} \quad j \notin \mathcal{P}^{n+1}. \tag{14}$$

Here (11)(12) are the updating equations for a Gaussian-Gaussian model, and (13)(14) 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 (11)-(14) in Figure 1. An outline for the SpKG algorithm is listed below in Algorithm 1.

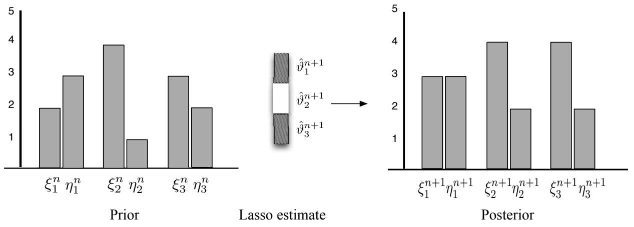


Figure 1: Illustration of the Bayesian model and the heuristic updating scheme for a Beta-Bernoulli model and a Gaussian-Gaussian model. Let $\boldsymbol{\alpha}$ be a three-element coefficient vector. The prior at time n includes the frequencies estimates (ξ_j^n, η_j^n) of "in" and "out." Combining with the Lasso estimate $\widehat{\vartheta}^{n+1}$ results in the posterior. On active sets $\{1, 3\}$, ξ_j^n are incremented by one. On inactive sets $\{2\}$, η_j^n are incremented by one.

Algorithm 1 Sparse knowledge gradient algorithm

```
Require: $\boldsymbol{\vartheta}^0, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},0}, \{\xi_j^0, \eta_j^0\}_{j=1}^p, \mathbf{X}, N, \sigma_\epsilon, \{\lambda^i\}_{i=0}^N$.
for $n = 0$ to $N-1$ do
    Compute SpKG by (8) $x^n = \operatorname{argmax} v_x^{\mathrm{KG},n}$;
    Lasso homotopy update:[1]
        $\widehat{\boldsymbol{\vartheta}}^n, (\boldsymbol{x}^n, y^{n+1}) \in \mathbb{R}^p \times \mathbb{R}, \lambda^n, \lambda^{n+1} \to \widehat{\boldsymbol{\vartheta}}^{n+1}$;
    Approximately simulate $\widehat{\boldsymbol{\Sigma}}_{\mathcal{S}}^{\boldsymbol{\vartheta},n+1}$ by (10);
    Bayesian update to $\{\xi_j^{n+1}, \eta_j^{n+1}\}_{j=1}^p$, $\boldsymbol{\vartheta}^{n+1}$, $\boldsymbol{\Sigma}^{\boldsymbol{\vartheta},n+1}$ by (11)-(14).
end for
return $\boldsymbol{\vartheta}^N, \boldsymbol{\Sigma}^{\boldsymbol{\vartheta},N}, \{\xi_j^N, \eta_j^N\}_{j=1}^p$.
```

4 THEORETICAL PROPERTIES

In this section we show the asymptotic convergence result of the Bayesian posterior mean estimate $\boldsymbol{\vartheta}^n$ in Algorithm 1. Since the knowledge gradient policy is myopically optimal by construction, this lends a strong guarantee that the algorithm will work well for finite budgets, as long as appropriate corrections are taken when the value of information is nonconcave. The proof is based on the assumption that we begin with some historical observations, and the initial design matrix satisfies the sparse Riesz condition (SRC), which is a form of Restricted Eigenvalue (RE) condition. (Zhang and Huang, 2008, provide the sufficient conditions for SRC to be satisfied.) If we have such a "warm" start, we can show that: (1) Algorithm 1 selects a model whose dimension is comparable with the underlying true model with probability converging to 1; (2) The Bayesian posterior mean converges to the truth with the same rate as that of Lasso (Zhang and Huang, 2008), which is the minimax optimal rate.

In addition to the aforementioned notations, 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, let $\mathcal{S}^n = \{j : \widehat{\vartheta}_j^n \neq 0\}$ be the estimated support from the current Lasso estimator. Let $\mathcal{S}^*$ be the true support, that is $\mathcal{S}^* = \{j : \vartheta_j \neq 0\}$. Also, let $s^* = |\mathcal{S}^*|$ be the cardinality of $\mathcal{S}^*$. Our presentation needs the following assumptions.

Assumption 4.1. *For any* n*, the random noise errors* $\epsilon^1, \ldots, \epsilon^n \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_\epsilon^2)$.

Assumption 4.2. *The design matrix* $\mathbf{X}^{n-1}$ *satisfies the sparse Riesz condition (SRC) with rank* s *and the spectrum bounds* $0 < c_* < c^* < \infty$*, if* $c_*\|\boldsymbol{\nu}\|_2^2 \leq \|\mathbf{X}_{*\mathcal{S}}^{n-1}\boldsymbol{\nu}\|_2^2/n \leq c^*\|\boldsymbol{\nu}\|_2^2$*,* $\forall \mathcal{S}$ *with* $s = |\mathcal{S}|$ *and* $\boldsymbol{\nu} \in \mathbb{R}^s$*. We refer to this condition as SRC* (s, c_*, c^*).

Assumption 4.3. *For any* n*, there exists some constant* $B > 0$ *such that* $\|\boldsymbol{x}^n\|_2^2 \leq B$.

Define $\widehat{c} = c^*/c_*$. Consider the Lasso path for $\lambda(n, p) \equiv 2\sigma_\epsilon\sqrt{2(1 + c_0)c^* n \log(p \vee a_n)}$ with $c_0 \geq 0$ and $a_n \geq 0$ satisfying $p/(p \vee a_n)^{1+c_0} \approx 0$. For large

[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.

p, this means that $\lambda(n,p) \sim O(\sqrt{n \log p})$ with $a_n = 0$. Then we have the following theorem of the ℓ_2 estimation error bound. The detailed proof can be found in Appendix B.

Theorem 4.1. *Assume that Assumptions 4.1 and 4.3 are satisfied. Suppose we begin with N' historical observations and the fixed design matrix $\mathbf{X}^{N'-1}$ satisfies SRC $(C_1 s^*, c_*, c^*)$, where C_1 is a positive constant defined below. Let $c_*, c^*, c_0, \sigma_\epsilon, s^*$, and B be fixed, $p \to \infty$, and $\bar{\mathcal{S}} := \bigcap_{n'=N'}^{n} \mathcal{S}^{n'}$. If we solve the Lasso given in (9) with $\lambda^n = \lambda(n,p)$, then for some large enough n with $\underline{c}N' \leq n \leq \bar{c}N'$ and $1 < \underline{c} \leq \bar{c}$ being fixed constants, the following properties hold with probability converging to 1 as $n \to \infty$:*

(1) $|\bar{\mathcal{S}}| \leq C_1 |\mathcal{S}^*|$ for some finite positive constant $C_1 := 2 + 4\bar{c}B/c_*$;

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

$$\|\boldsymbol{\vartheta}^n_{\bar{\mathcal{S}}} - \boldsymbol{\vartheta}_{\bar{\mathcal{S}}}\|_2^2 \leq \frac{C_2 \sigma_\epsilon^2 s^* \log p}{n},$$

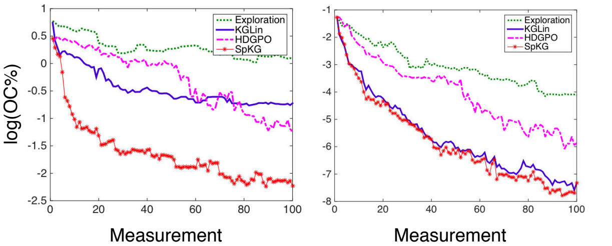
for some positive constant C_2 depending only on $c_*, c^*, c_0, \underline{c}, \bar{c}, B$, and $[C_{\min}, C_{\max}]$.

Theorem 4.1 proves the selection and estimation consistency, where the major assumption is that we have a "warm" start of N' historical observations. We believe that this assumption is valid in some applications we have seen. Based on this, we can prove that the posterior mean estimate converges to the truth at the same rate as that of Lasso. This result is satisfied for some large n in the interval $[\underline{c}N', \bar{c}N']$ with $1 < \underline{c} \leq \bar{c}$ being fixed constants, and with high probability. Here the probability converges to 1 as $n \to \infty$.

5 EXPERIMENTS

In this section we investigate the empirical performance of the SpKG algorithm proposed in this paper on both synthetic and the RNA data for identifying the accessibility region of an RNA molecule. We compare *SpKG* against the following baseline policies:

- *Pure exploration*, where an alternative is chosen randomly at each time and the updating scheme is the same as SpKG in Section 3.2;
- *KGLin* (Negoescu et al., 2011), which allows all the coefficients to be nonzero;
- *Hierarchical diagonal Gaussian process optimization (HDGPO)* (Chen et al., 2012), where the first step uses hierarchical diagonal sampling (HDS) to conduct variable selection and the second step applies GP-UCB (Srinivas et al., 2010) to optimize the function over the identified relevant dimensions.



(a) Average OC% for sparse belief (b) Average OC% for non-sparse belief

Figure 2: Average OC% comparing pure exploration, KGLin, HDGPO, and SpKG for sparse and non-sparse linear belief models.

5.1 Simulation Study

In these experiments, we repeatedly sample the truth $\boldsymbol{\alpha}$ from some normal distribution and compare different policies to see how well we are discovering the truth. Throughout all the simulations, we assume that we do not have any prior information on the sparsity structures. That is, $\xi_j^0 = \eta_j^0 = 1$, for $j = 1, \ldots, p$. In the first experiment, we compare SpKG with other policies by generating a linear model with $p = 200$ predictors and 40 relevant variables, using a relatively large measurement budget $N = 100$. Here $\mu = \sum_{j=1}^{p} \alpha_j x_j + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Specifically, for $j = 1, \ldots, 40$, 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 $j = 41, \ldots, 200$, let $\alpha_j = 0$. The prior is independently sampled with $\vartheta_j^0 = 30$ and $\Sigma_{jj}^{\boldsymbol{\vartheta},0} = 9^2$. Then we uniformly sample $M = 200$ alternatives from $[0,1]^p$. To quantitatively measure the performance of different policies, we consider the percentage opportunity cost (OC) with respect to the optimal value, defined as $\text{OC}\%(n) = \frac{\mu_{x^\star} - \mu_{x^{n,\star}}}{\mu_{x^\star}}$, where $x^\star$ is the true optimal alternative and $x^{n,\star}$ is the estimated optimal alternative at time n. This normalization better illustrates how far in percentage we are from the optimal. Figure 2(a) shows the log of the averaged OC% over 300 replications using well chosen tuning parameters with σ_ϵ being 30% of the expected range of μ.

From Figure 2(a) we can see that during the first several iterations, SpKG behaves comparable with pure exploration, because Lasso takes several iterations to identify the key features. However, after 10 to 20 measurements when the true sparsity pattern becomes detectable, SpKG far outperforms KGLin, HDGPO, 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. For this experiment, the default setting of HDGPO requires about 100 samples to find the relevant dimensions with high probability in its first step conducting HDS. To visualize its performance within the constraints of small experimental budgets, we use the initial 50 samples at the first step (exploration) and the other 50 samples to run GP-UCB algorithm at the second step (exploitation). We can see that during the first step, HDGPO performs slightly better than pure exploration and it behaves worse than SpKG in the second step. One of the reasons is that at every iteration, SpKG finds a low-dimensional model that can approximate the true model and this model is updated as new information is achieved. However, HDGPO is tailored towards exploration-exploitation tradeoffs by first aiming to find the relevant dimension and then identify the best alternative. *To the best of our knowledge, SpKG is the first work on sparse linear belief models which is well-suited to applications with small measurement budgets.*

Next, we compare SpKG with KGLin, pure exploration, and HDGPO on data which is *not* sparse. Here we consider a similar model as used in Figure 2(a). In the non-sparse setting, we only take the nonzero dimension of the function. For SpKG, 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 2(b) shows the log of the averaged OC% over 300 replications with the same level of measurement noise. As one can see, in the non-sparse setting, SpKG does not make erroneous conclusions of sparsity with a relatively small value of λ and performs competitively with KGLin. We can conclude that there is almost no loss even if we make approximations in the KG computation as well as the Bayesian update described in Section 3.2.

To further compare SpKG, KGLin, and HDGPO for a relatively small budget, we take several standard low dimensional test functions and hide them in a $p = 200$ dimensional space. These functions were designed to be minimized, so both policies are applied to the negative of the functions. We uniformly sample $M = 400$ alternatives from the feasible regions. Table 1 shows the quantitative results on the different functions with $N = 50$ over 500 replications. (Refer to Appendix A for detailed configurations and illustrations.)

Table 1: Quantitative comparison for SpKG, KGLin, and HDGPO on standard test functions over 500 runs. (Each function is scaled to have a range of 100.)

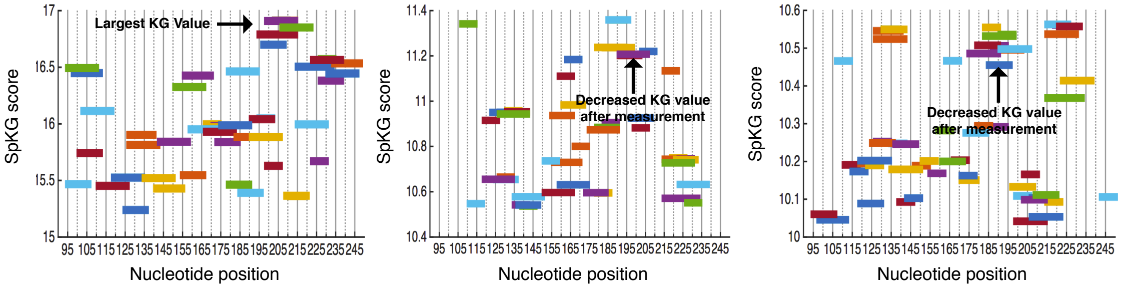
Function	σ_ϵ	**SpKG**	**KGLin**	**HDGPO**
Matyas	1	0.0064 ± 0.0022	0.0217 ± 0.0019	0.0167 ± 0.0025
	10	0.0526 ± 0.0243	0.1168 ± 0.0258	0.0914 ± 0.0392
	20	0.3581 ± 0.0578	0.8519 ± 0.0283	0.6390 ± 0.0464
Six-hump Camel	1	0.0011 ± 0.0031	0.0065 ± 0.0027	0.0043 ± 0.0036
	10	0.0258 ± 0.1279	0.0835 ± 0.1886	0.0607 ± 0.1643
	20	0.2260 ± 0.2644	0.3803 ± 0.2568	0.3304 ± 0.2165
Trid	1	1.0273 ± 0.0105	1.8367 ± 0.0089	1.6428 ± 0.0127
	10	4.8564 ± 0.1639	6.5181 ± 0.1565	5.9654 ± 0.1696
	20	8.4058 ± 0.3850	12.8532 ± 0.2964	10.9535 ± 0.4267

5.2 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* which has been widely used as an RNA folding model (Cech et al., 1981). Accessibility describes the ability of other molecules to attach to different segments of the RNA, which are affected by the folding of the molecule (portions may be simply inaccessible because they are buried within the folds, or because there are no sites for other compounds to attach to). Scientists can infer accessibility by using a probe which lights up when the probe successfully attaches to a region. These probes have to be designed for specific segments of the RNA, so a failure to attach indicates that the region for which the probe is designed is not accessible. Inaccessible regions are captured in our model with zero-valued coefficients; accessible regions have nonzero coefficients that capture the degree of accessibility. Our challenge is to identify the most accessible region while learning which coefficients are zero, and the values of the nonzero coefficients.

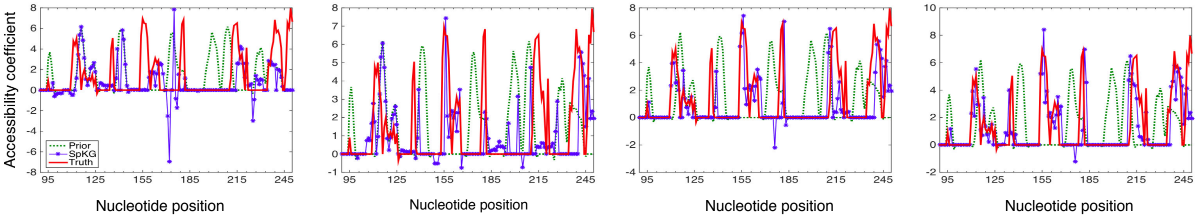
The molecular we are using for testing has $p = 414$ sites. Prior accessibility coefficients are drawn from experiments in Russell et al. (2006). We randomly generate a truth, which we have to discover using SpKG, by both vertically perturbing the prior and horizontally shifting the prior. Specifically, we vertically adjust the magnitude of the nonzero coefficients by adding a noise term with standard deviation of 20% and horizontally shift the prior by 20 to 50 sites.

First, we illustrate how SpKG policy works under a measurement noise of 30%. We take a subsequence of the molecule (from site 95 to 251) with $p = 157$ to better visualize the results. Alternatives (testing probe sequences) with a number of $M = 91$ are selected by the domain experts. For one experiment, we depict the SpKG value initially, after one and two measurements, respectively in Figure 3. For these figures, we only include those probes with SpKG values above the mean to better visualize the SpKG scores. As indicated by



(a) No measurement (b) After 1 measurement (c) After 2 measurements

Figure 3: SpKG values after 0, 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 probe.)



(a) After 20 measurements (b) After 30 measurements (c) After 40 measurements (d) After 50 measurements

Figure 4: Accessibility profile estimate by SpKG after 20, 30, 40, and 50 measurements with noise ratio of 30%.

the arrows, for probes with the largest SpKG scores, the SpKG scores drop after they have been measured. As we only plot those with SpKG scores above average, some probes with high KG scores in Figure 3(a) have the scores dropped below average after being measured and are therefore not shown in Figure 3(b). This observation is consistent with our intuition of SpKG as a measure of value of information, and thus we can use this policy as a guideline to pick the next experiments.

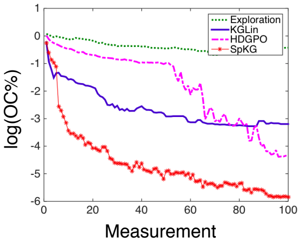


Figure 5: Average OC% comparing pure exploration, KGLin, HDGPO, and SpKG with the whole target molecule.

Next, for one simulated truth, we also plot the estimates of the accessibility profiles (coefficients) after 20, 30, 40, and 50 measurements with a noise ratio of 30% in Figure 4. As one can see, after 20 measurements, the estimate is still closer to the prior than 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. After 50 measurements, our estimate closely matches the truth.

Furthermore, we take the whole target molecule with $p = 393$. (Due to the nature of this problem, we are not able to identify the 21 nucleotides at one end of the molecule sequence.) The alternative probes are of length 10 with 3 overlaps for the adjacent ones. Figure 5 plots the average OC% over 100 runs for three different policies with a measurement noise of 30%.

6 Conclusion

We extend the KG policy to optimize high-dimensional sparse linear functions. The hybrid of Bayesian R&S with Lasso is novel. Our work is motivated by an important, high-impact application to discover the structure of an RNA molecule, which is a high-dimensional learning problem guided by a team of domain experts who participated in the research. Empirically, the SpKG algorithm allows us to quickly identify the best alternatives in experimental settings where measurement costs are quite high. We note at the same time that SpKG requires considerably more work computationally than its competitors, so it is best used in the setting where experiments are time-consuming and expensive. We feel this paper opens an entirely new line of research in high-dimensional active learning by combining the power of Bayesian priors with model structure.

References

Brochu, E., Brochu, T., and de Freitas, N. (2010a). A Bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 103–112. Eurographics Association.

Brochu, E., Cora, V. M., and De Freitas, N. (2010b). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

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

Cech, T. R., Zaug, A. J., and Grabowski, P. J. (1981). 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.

Chen, B., Castro, R., and Krause, A. M. (2012). 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.

Djolonga, J., Krause, A., and Cevher, V. (2013). High-dimensional Gaussian process bandits. In *Advances in Neural Information Processing Systems*, pages 1025–1033.

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

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

Garrigues, P. and El Ghaoui, L. (2008). An homotopy algorithm for the Lasso with online observations. In *Advances in Neural Information Processing Systems*, pages 489–496.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian data analysis*. CRC press.

Martinez-Cantin, R., de Freitas, N., Brochu, E., Castellanos, J., and Doucet, A. (2009). A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, 27(2):93–103.

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

Powell, W. B. and Ryzhov, I. O. (2012). *Optimal learning*. John Wiley and Sons, Hoboken, NJ.

Raskutti, G., Wainwright, M. J., and Yu, B. (2011). Minimax rates of estimation for high-dimensional linear regression over-balls. *Information Theory, IEEE Transactions on*, 57(10):6976–6994.

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

Russell, R., Das, R., Suh, H., Travers, K. J., Laederach, A., Engelhardt, M. A., and Herschlag, D. (2006). The paradoxical behavior of a highly structured misfolded intermediate in RNA folding. *Journal of molecular biology*, 363(2):531–544.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959.

Sowa, S. W., Vazquez-Anderson, J., Clark, C. A., De La Peña, R., Dunn, K., Fung, E. K., Khoury, M. J., and Contreras, L. M. (2014). Exploiting post-transcriptional regulation to probe RNA structures in vivo via fluorescence. *Nucleic acids research*, page gku1191.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Maching Learning (ICML-10)*, pages 1015–1022.

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

Wang, Z., Zoghi, M., Hutter, F., Matheson, D., and De Freitas, N. (2013). 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.

Zhang, C. H. and Huang, J. (2008). The sparsity and bias of the Lasso selection in high-dimensional linear regression. *The Annals of Statistics*, pages 1567–1594.