High Dimensional Bayesian Optimization via Supervised Dimension Reduction

Miao Zhang[1,2*], **Huiqi Li**[1†] and **Steven Su**[2]
[1]School of Information and Electronics, Beijing Institute of Technology, China
[2]Faculty of Engineering and Information Technology, University of Technology Sydney, Australia
Miao.Zhang-2@student.uts.edu.au,huiqili@bit.edu.cn, Steven.Su@uts.edu.au

Abstract

Bayesian optimization (BO) has been broadly applied to computational expensive problems, but it is still challenging to extend BO to high dimensions. Existing works are usually under strict assumption of an additive or a linear embedding structure for objective functions. This paper directly introduces a supervised dimension reduction method, Sliced Inverse Regression (SIR), to high dimensional Bayesian optimization, which could effectively learn the intrinsic sub-structure of objective function during the optimization. Furthermore, a kernel trick is developed to reduce computational complexity and learn nonlinear subset of the unknowing function when applying SIR to extremely high dimensional BO. We present several computational benefits and derive theoretical regret bounds of our algorithm. Extensive experiments on synthetic examples and two real applications demonstrate the superiority of our algorithms for high dimensional Bayesian optimization.

1 Introduction

Many machine learning tasks are zeroth-order optimization problems with expensive evaluation function, where the optimal points could be reached only by limited querying objective function. Bayesian Optimization (BO) has shown its superiority in noisy and expensive black-box optimization problems, which utilizes surrogate function (acquisition function) based on a computationally cheap probabilistic model to balance the exploration and exploitation during optimizing process, and Gaussian Process (GP) [Rasmussen and Williams, 2004] is the most popular probabilistic model in Bayesian Optimization. The iteration of Bayesian Optimization is described as: 1) BO learns the probabilistic Gaussian Process model and construct the acquisition function based on queried points; 2) Maximize acquisition function to find the most promising point to be queried; 3) Evaluate $\mathbf{x}_t^*$ and update queried points.

*Contact Author
†Corresponding Author

There are many successful applications of Bayesian Optimization in artificial intelligence community, including hyper-parameter tuning [Snoek *et al.*, 2012; Bergstra *et al.*, 2011], neural architecture search [Kandasamy *et al.*, 2018], robotics [Deisenroth *et al.*, 2015] and more. Although BO works well in these applications with low-dimension or moderate-dimension, its applicability to high-dimensional problems is still challenging due to the statistical and computational challenges [Wang *et al.*, 2013]. Current theoretical research works suggest that Gaussian Process based Bayesian Optimization is exponentially difficult with dimension [Srinivas *et al.*, 2012], as the theoretical efficiency of statistical estimation of function depends exponentially on dimension. Several research works have been proposed to tackle the two problems in high-dimensional Bayesian Optimization (HDBO) by assuming the function varies along an underlying low-dimensional subspace [Chen *et al.*, 2012; Djolonga *et al.*, 2013; Wang *et al.*, 2013; Ulmasov *et al.*, 2016; Li *et al.*, 2017]. Another popular method to handle high-dimensional BO is assuming an additive form of objective function [Kandasamy *et al.*, 2015; Li *et al.*, 2016; Wang *et al.*, 2017; Rolland *et al.*, 2018].

In this paper, we introduce a supervised dimension reduction method (Sliced Inverse Regression, SIR) [Li, 1991] to high-dimensional Bayesian optimization, which is used as projection matrix and could automatically learn the intrinsic structure of objective function during the optimization. Different with [Ulmasov *et al.*, 2016], which uses unsupervised dimension reduction method (PCA), our method takes the statistical relationship between the input and response into consideration and could handle the situation that the number of observations is less than dimension $N \ll D$. Furthermore, a kernel trick is developed to reduce the computational complexity of SIR when it is applied to handle extreme-large dimensional data, and also provide a nonlinear generalization to make SIR to learn a nonlinear function. The contributions of the paper are summarized in the following.

- Firstly, we formulize a novel SIR-BO algorithm which directly introduces a supervised dimension reduction method SIR to high-dimensional Bayesian Optimization to automatically learn the intrinsic structure of objective function during the optimization.
- Secondly, the dimension reduction method in our

Bayesian Optimization is extended to nonlinear setting through a kernel trick, which not only greatly reduces the complexity of SIR to make it possible to apply SIR to extremely-high dimensional Bayesian optimization, but also extracts nonlinear components simultaneously.

- Thirdly, we theoretically analyze the algorithm and its regret bounds, and extensively conduct experiments on synthetic data and real-world applications. Experimental results illustrate the superiority of our methods and imply that automatically learning the intrinsic low-dimensional structure of objective functions may be a more appropriate way for high dimensional Bayesian optimization than setting a strict prior assumption.

2 Background

2.1 Bayesian Optimization

Bayesian optimization is aimed to maximize an unknown function $f : \mathcal{X} \to \mathbb{R}$ based on observations, that $\mathcal{X} \subseteq \mathbb{R}^D$. BO is usually applied to black-box optimization problems, where there is no prior information about the exact function but only able to obtain noisy function evaluations $y = f(\mathbf{x}) + \epsilon$ by querying several exact points $\mathbf{x} \in \mathcal{X}$. There are two key components for Bayesian Optimization: prior and acquisition function. In this paper, we investigate Gaussian Process as the prior [Rasmussen and Williams, 2004], that given t pairs of observation points $\mathbf{x}_{1:t}$ and corresponding evaluations $y_{1:t}$, we have a joint distribution:

$$\mathbf{f}(\mathbf{x}_{1:t}) \sim \boldsymbol{N}(\mathbf{m}(\mathbf{x}_{1:t}), \mathbf{K}(\mathbf{x}_{1:t}, \mathbf{x}_{1:t})) \tag{1}$$

where $\mathbf{K}(\mathbf{x}_{1:t}, \mathbf{x}_{1:t})_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ is the covariance matrix based on kernel function κ. Given a new observation point $\mathbf{x}^*$, the poster predictive distribution is calculated as:

$$f(\mathbf{x}^*)|\mathcal{D}_t, \mathbf{x}^* \sim \boldsymbol{N}(\mu(\mathbf{x}^*), \sigma(\mathbf{x}^*)) \tag{2}$$

where $\mathcal{D}_t = \{\mathbf{x}_{1:t}, y_{1:t}\}$, $\mu(\mathbf{x}^*) = \kappa(\mathbf{x}^*, \mathbf{x}_{1:t})(\mathbf{K}(\mathbf{x}_{1:t}, \mathbf{x}_{1:t}) + \theta_c^2 I)^{-1} y_{i:t}$, $\sigma(\mathbf{x}^*) = \kappa(\mathbf{x}^*, \mathbf{x}^*) - \kappa(\mathbf{x}^*, \mathbf{x}_{1:t})(\mathbf{K}(\mathbf{x}_{1:t}, \mathbf{x}_{1:t}) + \theta_c^2 I)^{-1} \kappa(\mathbf{x}^*, \mathbf{x}_{1:t})^T$, θ_c is a noise parameter. θ_c and the hyperparameters of kernel κ could be learned by maximizing the negative logarithm of the marginal likelihood function:

$$l = -\frac{1}{2} log(\mathbf{det}(\mathbf{K} + \theta_c^2 I)) - \frac{1}{2} y^T (\mathbf{K} + \theta_c^2 I)^{-1} y - \frac{t}{2} log 2\pi \tag{3}$$

Acquisition function is a surrogate function based on the prior model which is used to determine the most promising sampling point for evaluation. In this paper we investigate Gaussian processes with Upper Confidence Bound (GP-UCB), which is defined as:

$$\alpha_{\mathbf{UCB}}(\mathbf{x}) = \mu(\mathbf{x}) + \sqrt{\beta}\sigma(\mathbf{x}) \tag{4}$$

with a tunable β to balance exploitation against exploration.

2.2 Sliced Inverse Regression

There are many popular approaches for dimension reduction in machine learning community, including PCA, ISOMAP, LLE, and Laplacian Eigenmaps [Cunningham and Ghahramani, 2015], to find the manifold and intrinsic dimension of the input, but all of them are unsupervised without the consideration of the statistical relationship between the input $\mathbf{x}$ and response y. In contrast, sliced inverse regression (SIR) tries to find an effective subspace based on inversely regressing input $\mathbf{x}$ from response y, which has shown its superiority on dimension reduction in machine learning applications. SIR is based on the assumption of existing low-dimensional and effective subspace for a regression model:

$$y = f(\beta_1' \mathbf{x}, ..., \beta_d' \mathbf{x}, \epsilon), \quad \beta_k, \mathbf{x} \in \mathbb{R}^D \tag{5}$$

where $d \ll D$ and $\{\beta_1, ..., \beta_d\}$ are orthogonal vectors forming a basis of the subspace, and ϵ is a noise. $\{\beta_1' \mathbf{x}, ..., \beta_d' \mathbf{x}\}$ is the dimension reduction projection operator from D-high-dimensional space to d-low-effective space with sufficient relevant information in $\mathbf{x}$ about y. SIR utilizes the inverse regression to find effective dimension reduction (e.d.r) subspace who considers the conditional expectation $\mathbb{E}(\mathbf{x}|\mathbf{y} = y)$ where $\mathbf{y}$ varies as the opposite to classical regression setting $\mathbb{E}(\mathbf{y}|\mathbf{x} = x)$. Li [1991] has shown that the regression curve of $\mathbb{E}(\mathbf{x}|\mathbf{y} = y)$ hides in e.d.r subspace B, where the dimension reduction directions $\{\beta_1, ..., \beta_L\}$ could be found by solving following eigen decomposition problem:

$$\Gamma \beta = \lambda \Sigma \beta \tag{6}$$

where $\Gamma = \mathrm{Cov}(E(\mathbf{x}|\mathbf{y}))$ is between-slice covariance matrix and Σ is the covariance matrix of $\mathbf{x}$.

3 Supervised dimension reduction based Bayesian optimization

The supervised dimension reduction based Bayesian optimization usually contains two main stages [Djolonga *et al.*, 2013]: **subspace learning**, which is to find e.d.r directions to project inputs into low-dimension data; and executing **normal Bayesian optimization** on the learned subspace. We first utilize SIR to learn towards the actual low-dimensional subspace A that: $f(\mathbf{x}) = g(A\mathbf{x})$. Defining $\hat{A}$ as the learned approximated dimension reduction matrix that: $\hat{f}(\mathbf{x}) = \hat{g}(\hat{A}\mathbf{x}) = \hat{g}(\mathbf{u})$, we optimize $\hat{g}(\mathbf{u})$ based on BO in the effective subspace and update the approximate projection matrix $\hat{A}$ based on SIR in certain iterations to make it aligned closer with the actual matrix A.

3.1 Subspace learning and computational efficiency

We learn the projection matrix based on SIR by solving the eigen decomposition problem:

$$\hat{\Gamma} \beta = \lambda \hat{\Sigma} \beta \tag{7}$$

and the first d eigenvectors are composed as the approximate projection matrix. $\hat{\Sigma}$ and $\hat{\Gamma}$ are $D \times D$ matrix, and the computational complexity of solving Eq.(7) is $\mathcal{O}(2D^3)$, that we need to calculate the inverse of covariance matrix Σ^{-1} and then eigenvectors of $\Sigma^{-1}\hat{\Gamma}$. Although SIR has been applied to several dimension reduction applications, it is unrealistic to apply standard SIR to extremely-high dimensional Bayesian optimization, e.g. $D \geq 100000$. In this paper, we utilize two tricks to reduce the computation complexity when applying SIR to etremly-high dimensional Bayesian optimization: reduced SVD and kernelizing input which extends SIR into reproducing kernel Hilbert space (RKHS) in terms of kernels according to [Wu, 2008].

Algorithm 1 SIR-BO

Input: n, T, kernel κ

1: $D_1 = \{\mathbf{x}_{1:n}, y_{1:n}\} \leftarrow$ randomly generate n points and evaluations.
2: $\beta \leftarrow$ calculate dimension reduction directions based on Eq.(10).
3: **for** $i = 1, 2, ...$ **do**
4: Built Gaussian process prior based on $\mathbf{u} = \beta D$;
5: Find $\mathbf{x}^* \in \mathcal{X}$ with maximal acquisition function value: $\mathbf{x}_t^*$: $\mathbf{x}^* = \mathbf{argmax}_{\mathbf{x}\in\mathcal{X}}\mathbf{UCB}(\mathbf{u}), \mathbf{u} = \beta\mathbf{x}$;
6: $\mathcal{D}_{i+1} = \mathcal{D}_i \cup \{\mathbf{x}^*, y^*\}$;
7: **end for**

Numerical method based on reduced SVD

As suggested by [Yeh *et al.*, 2009], the rank of slice covariance $\hat{\Gamma}$ is $(J-1)$, as:

$$\hat{\Gamma} = WW^T \tag{8}$$

where the jth column of W is defined as $w_j = \sqrt{\frac{n_j}{n}}E(\mathbf{x}_i | i \in S_j)$, and J is number of slices in SIR. We set UMU^T as the eigenvalue decomposition of $W^T\hat{\Sigma}^{-1}W$, and U is $J \times (J-1)$, M is $(J-1)\times(J-1)$ when we only consider eigenvectors with nonzero eigenvalues. Eq.(7) could be transformed as :

$$\hat{\Sigma}^{-\frac{1}{2}}WW^T\hat{\Sigma}^{-\frac{1}{2}}z = \lambda z \tag{9}$$

where $z = \hat{\Sigma}^{\frac{1}{2}}\beta$. We assume $ZM^{\frac{1}{2}}U^T$ is the SVD of $\hat{\Sigma}^{-\frac{1}{2}}W$, and each column of $Z = \hat{\Sigma}^{-\frac{1}{2}}WUM^{-\frac{1}{2}}$ is thus a solution of Eq.(9) when we set $\Lambda = M$, so that each column of

$$V = \hat{\Sigma}^{-\frac{1}{2}}Z = \hat{\Sigma}^{-1}WUM^{-\frac{1}{2}} \tag{10}$$

is a solution of Eq.(7), and the e.d.r matrix can be constructed by the combination of the columns of V. In this way, the computational complexity to find an e.d.r direction contains: $\mathcal{O}(J^3)$ for the eigenvalue decomposition of $W^T\hat{\Sigma}^{-1}W$ and $\mathcal{O}(D^3)$ for the inverse of $\hat{\Sigma}$. Normally, we set $J = d + 1$ in this paper, and all non-zero eigenvectors are composed as the dimension reduction matrix, where d is the assumed low-dimension.

SIR with Kernelized Input

Even though we could efficiently reduce the computational complexity of SIR to $\mathcal{O}(J^3) + \mathcal{O}(D^3)$ by SVD, it is still impossible to directly apply SIR to extremely-high Bayesian optimization. Furthermore, SIR is only able to find the effective linear subspace, while hard to extract nonlinear features. The kernel trick is a popular way to provide a nonlinear transformation for algorithms, where the input data are represented by kernel function of pairwise comparisons [Wu *et al.*, 2007]. In this paper, we proposed Kernelized Input SIR (KISIR) to handle this situations, which utilizes kernel trick to extend SIR to nonlinear dimension reduction, and reduce the computational complexity greatly.

Given a kernel κ and its spectrum: $\kappa(\mathbf{x}, \mathbf{v}) = \sum_{q=1}^{Q}\lambda_q\phi(\mathbf{x})\phi(\mathbf{v})$, $Q \leq \infty$, $\mathbf{x}, \mathbf{v} \in \mathcal{X}$, the input space is transformed into spectrum-based feature space $\mathcal{H}_\kappa$: $\mathbf{x} \mapsto \mathbf{z} : \Phi(\mathbf{x}) := (\sqrt{\lambda_1}\phi_1(\mathbf{x}), \sqrt{\lambda_2}\phi_2(\mathbf{x}), ...)^T$, and $\langle\Phi(\mathbf{x}), \Phi(\mathbf{v})\rangle_{\mathcal{H}_\kappa} = \kappa(\mathbf{x}, \mathbf{v})$. So, the regression model with low-dimensional subspace as in Eq.(5) could be described as:

$$y = f(\beta_1^{\mathcal{H}_\kappa}\mathbf{z}, ..., \beta_d^{\mathcal{H}_\kappa}\mathbf{z}, \epsilon), \quad \beta_k^{\mathcal{H}_\kappa} \in \mathbb{R}^Q, \ Q \leq \infty \tag{11}$$

in the feature space $\mathcal{H}_\kappa$. Consider $m_\mathbf{z} = \sum_{i=1}^{n}\Phi(\mathbf{x}_i) = 0$ (otherwise center them according [Wu, 2008]), we define $\hat{\Sigma}_\mathbf{z} = \frac{1}{n}\sum_{i=1}^{n}\Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i)^T$ as the sample covariance matrix of $\mathbf{z} := \Phi(\mathbf{x})$, and $\hat{\Gamma}_\mathbf{z} = \sum_{j=1}^{J}\frac{n_j}{n}m_\mathbf{z}^j m_\mathbf{z}^{jT}$ as the sample between-slice covariance matrix, where $m_\mathbf{z}^j = \frac{1}{n_j}\sum_{s\in S_j}\Phi(\mathbf{x}_s)$ is the the mean for slice S_j with n_j size. We could rephrase the eigenvalue decomposition of Eq.(7) in feature space $\mathcal{H}_\kappa$ as:

$$\hat{\Gamma}_\mathbf{z}\gamma = \lambda\hat{\Sigma}_\mathbf{z}\gamma \tag{12}$$

This problem could be solved by equivalent system:

$$\left\langle\Phi(\mathbf{x}_i), \hat{\Gamma}_\mathbf{z}\gamma\right\rangle_{\mathcal{H}_\kappa} = \lambda\left\langle\Phi(\mathbf{x}_i), \hat{\Sigma}_\mathbf{z}\gamma\right\rangle_{\mathcal{H}_\kappa} \tag{13}$$

where the solution is in the form of $\gamma = \sum_{i=1}^{N}\alpha_i\Phi(\mathbf{x}_i)$. Based on previous definition, we could kernelize the input data as: $\mathbf{K} = \{\kappa_{i,j} = \langle\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle_{\mathcal{H}_k} = \kappa(\mathbf{x}_i, \mathbf{x}_j)\}_{N\times N}$, $\mathbf{K} = \mathbf{K}^T$. Define $\mathcal{H}_\kappa$ as the reproducing kernel Hilbert space based on κ, and the eigenvalue decomposition problem of SIR in $\mathcal{H}_k$ could be rephrased as:

$$\hat{\Gamma}_\mathbf{K}\gamma = \lambda\hat{\Sigma}_\mathbf{K}\gamma \tag{14}$$

where $\hat{\Sigma}_\mathbf{K} = \mathbf{K} \times \mathbf{K}^T$, $\hat{\Gamma}_\mathbf{K} = \sum_{j=1}^{J}m_\mathbf{K}^j \times m_\mathbf{K}^{jT}$, and $m_\mathbf{K}^j = \frac{1}{n_j}\sum_{s\in S_j}\mathbf{K}(:, s)$ is the mean for slice S_j with n_j size. The computational complexity of solving Eq.(14) is $\mathcal{O}(2N^3)$, where $N \ll D$ in extremely-high dimension reduction problem. Combining with the proof in Eq.(10), the computational complexity could be reduced into $\mathcal{O}(J^3) + \mathcal{O}(N^3)$.

Algorithm 2 KISIR-BO

Input: n, T, kernel κ

1: $D_1^\mathbf{K} = \{\mathbf{k}_{1:n}, y_{1:n}\} \leftarrow$ kernelize randomly generated points D_1 through kernel trick.
2: $V \leftarrow$ calculate e.d.r directions based on Eq.(14).
3: Transform the kernel data to subspace by V: $D_1^\mathbf{K} \xrightarrow{V} D_{V1}^\mathbf{K}$, and perform **GP-UCB** on this subspace.
4: **for** $i = 1, 2, ...$ **do**
5: Built Gaussian process prior based on $D_{Vi}^\mathbf{K}$;
6: Find $\mathbf{x}^* \in \mathcal{X}$ with maximal acquisition function value: $\mathbf{x}_t^*$: $\mathbf{x}^* = \mathbf{argmax}_{\mathbf{x}\in\mathcal{X}}\mathbf{UCB}(\mathbf{u}), \mathbf{u} = V\mathbf{k}, \mathbf{k} = \kappa(\mathbf{x}, :)$;
7: $\mathcal{D}_{i+1} = \mathcal{D}_i \cup \{\mathbf{x}^*, y^*\}$, kernelize D_{i+1} to get $D_{i+1}^\mathbf{K}$, update V and get $D_{Vi+1}^\mathbf{K}$;
8: Update κ for certain iterations;
9: **end for**

3.2 Bayesian optimization in subspace

The general process of SIR-BO is very similar to SI-BO in [Djolonga *et al.*, 2013], and we also apply GP-UCB [Srinivas *et al.*, 2012] as the Bayesian optimizing method to theoretically analyze the regret bounds with ease, and perform

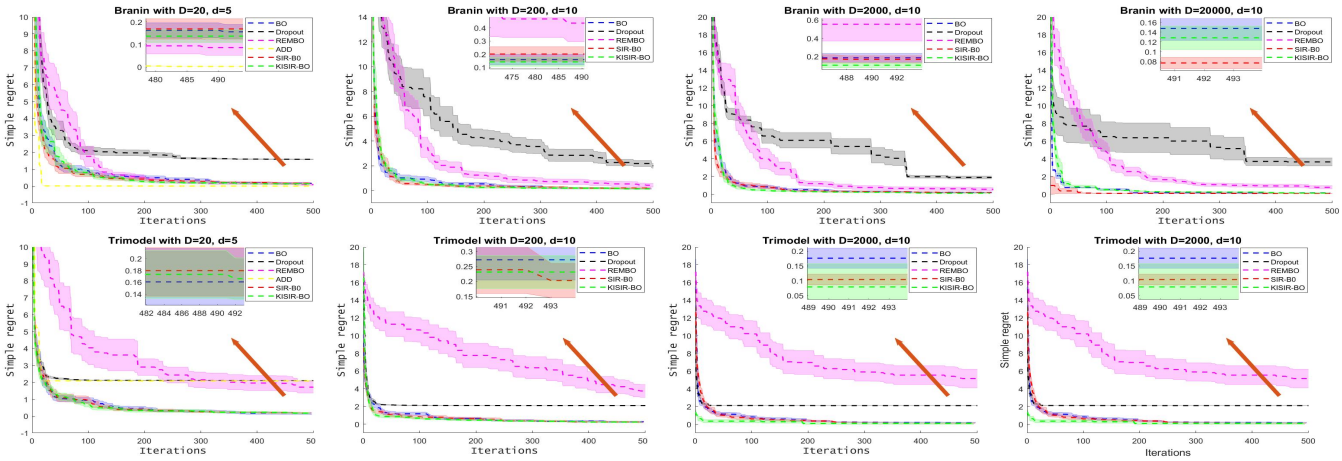

Figure 1: Simple regrets over iterations of our SIR-BO and KISIR-BO with compared approaches on two synthetic functions under different dimensions. We plot means with $1/4$ standard errors across 20 independent runs.

it in an effective subspace. Algorithm 1 presents a simple implementation of SIR-BO. The theoretical regret bounds of SIR-BO could also be derived based on Lemma 1, Theorem 4 and Theorem 5 in [Djolonga *et al.*, 2013]. However, after we kernelize input data through kernel trick, Bayesian optimization is supposed to be performed on a subspace of RKHS in our KISIR-BO, so we need to analyze transforming the input data into RKHS space trough kernel function κ: $\mathcal{X} \mapsto \mathcal{H}_\kappa$.

As defined in [Tan and Mukherjee, 2018], we consider the function f is randomly drawn from a distribution over the functions in the RKHS: $f = \sum_i \beta_i \Phi(\mathbf{x}_i)$, where β_i is Gaussian random variable, $\mathbf{x}_i \in \mathcal{X} := \{\mathbf{x}_1, \mathbf{x}_2, ...\}$ and the feature map $\Phi : \mathcal{X} \mapsto \mathcal{H}_\kappa$ is induced by kernel function κ. The function with kernerlized input $g(\mathbf{k})$ is defined as:

$$\{g(\mathbf{k}) = f(\mathbf{x}) | \mathbf{x} \in \mathcal{X}, \mathbf{k} \in \mathcal{H}_\kappa, f = \sum_i \beta_i \Phi(\mathbf{x}_i), \boldsymbol{\beta} \sim \mathcal{N}(\mathbf{b}, \Sigma_\beta)\} \tag{15}$$

where $\mathbf{k} = \{\kappa(\mathbf{x}, \mathbf{x}_1), \kappa(\mathbf{x}, \mathbf{x}_2), ...\} = \{\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_1) \rangle_{\mathcal{H}_\kappa}, \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_2) \rangle_{\mathcal{H}_\kappa}, ...\}$, and the mean and covariance of $g(\mathbf{k})$ could be defined as:

$$\begin{aligned} m(g(\mathbf{k})) &= m(f(\mathbf{x})) = \sum_i b_i \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_i) \rangle_{\mathcal{H}_\kappa} \\ &= \sum_i b_i \kappa(\mathbf{x}, \mathbf{x}_i) = \sum_i b_i \mathbf{k}(i) \\ cov(g(\mathbf{s}), g(\mathbf{t})) &= \sum_i \sum_j (\Sigma_\beta)_{ij} \langle \Phi(\mathbf{s}), \Phi(\mathbf{x}_i) \rangle_{\mathcal{H}_\kappa} \times \\ & \quad \langle \Phi(\mathbf{t}), \Phi(\mathbf{x}_i) \rangle_{\mathcal{H}_\kappa} = \sum_i \sum_j (\Sigma_\beta)_{ij} \mathbf{s}(i) \mathbf{t}(j) \end{aligned} \tag{16}$$

$g(\mathbf{k})$ are Gaussian random variables indexed by $\mathcal{X}$ which could be formulated as a Gaussian process [Tan and Mukherjee, 2018], and it is called as Kernelized Input Gaussian processes (KIGP) in this paper. The obvious difference between KIGP and GP is that the response difference between $\mathbf{x}_i$ and $\mathbf{x}_j$ is not a function of $\mathbf{x}_i - \mathbf{x}_j$ but of $\mathbf{K}(\mathbf{x}_i) - \mathbf{K}(\mathbf{x}_j)$, where $\mathbf{K}(\mathbf{z}) = \{\kappa(\mathbf{z}, \mathbf{x}_1), \kappa(\mathbf{z}, \mathbf{x}_2), ...\}$. It has been proved that KIGP is a strict superset of functions than GP with a same trace-class covariance kernel function (see Theorem 1 in [Tan and Mukherjee, 2018]).

After proving $g(\mathbf{k})$ is a Gaussian distribution and learning a dimension reduction matrix V through SIR based on kernelized input data, we perform Bayesian optimization on an estimated subspace of $g(\mathbf{k})$: $\hat{h}(\mathbf{u})$, which is also similar to [Djolonga *et al.*, 2013]. We give the cumulative regret bounds of KISIR-BO based on [Djolonga *et al.*, 2013] as follow:

Theorem 1 *Assume computational budget allows T evaluations, and first spend $0 < n \leq T$ sample evaluations to learn e.d.r. matrix V which updates during the optimization. We also assume that with more evaluations, the learned e.d.r matrix V is closer to actual matrix A, and $\|g - \hat{g}\|_\infty \leq \eta$, $\|\hat{h}\| \leq B$ where the error is bounded by $\hat{\sigma}$, each with probability at least $1 - \delta/4$.*

*If perform **GP-UCB** with $T - n$ steps on $\hat{h}(\boldsymbol{u})$, with probability at least $1 - \delta$, then the cumulative regret for KISRI-BO is bounded by:*

$$R_T \leq \underbrace{n}_{d.r.m\ learning} + \underbrace{(T-n)\eta}_{approx.\ error} + \underbrace{\mathcal{O}^*(\sqrt{TB}(\sqrt{\gamma_t} + \gamma_t))}_{R_{UCB}(T, \hat{h}, k)} \tag{17}$$

where $R_{UCB}(T, \hat{h}, \kappa)$ is the regret w.r.t. $\hat{h}$.

Proof: As discussed above, $g(\mathbf{k})$ could be formulated as a Gaussian process, and our KISIR-BO is also performed on the subspace of kernelized input space $\mathcal{H}_\kappa$. As defined in Lemma 1 in [Djolonga *et al.*, 2013], the cumulative regret bounds for **GP-UCB** performed on the subspace is $n + T\eta + \mathcal{O}^*(\sqrt{TB}(\sqrt{\gamma_t} + \gamma_t))$, and based on the assumption that, with more evaluations, the learned e.d.r matrix V is closer to actual matrix A, so the $approx.\ error \leq (T-n)\eta$, and the regret bounds for our KISIR-BO could be obtained. □

Based on Theorem 3 in [Srinivas *et al.*, 2012] and Theorem 4 and 5 in [Djolonga *et al.*, 2013], the regret bound could be rephrased as:

$$\begin{aligned} R_T &\leq \mathcal{O}(k^3 d^2 \mathbf{log}^2(1/\delta)) + 2\mathcal{O}(\sqrt{TB}(\sqrt{\gamma_t} + \gamma_t)) \\ R_T &\leq \mathcal{O}(\delta^2 k^{11.5} d^7 T^{4/5} \mathbf{log}^3(1/\delta)) + 2\mathcal{O}(\sqrt{TB}(\sqrt{\gamma_t} + \gamma_t)) \end{aligned} \tag{18}$$

for noiseless observation and for noisy observations, respectively. [Srinivas *et al.*, 2012] gives the bounds γ_t for different commonly used kernels.

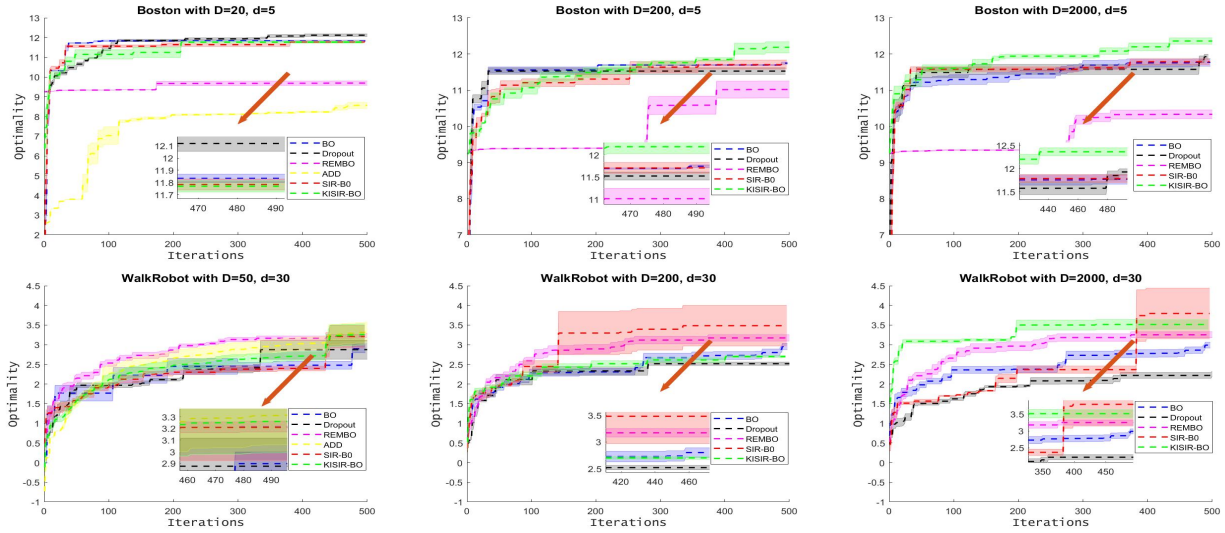

Figure 2: Performance for configuring neural network on Boston dataset and controlling a three-link walk robot. We plot the mean and $1/4$ standard deviation for all comparing algorithms under different dimensions.

Algorithm 2 presents the details of KISIR-BO. We firstly need to define computational budget with T evaluation and n initial sample evaluations. n randomly initial points with evaluation $D_1 = \{\mathbf{x}_{1:n}, y_{1:n}\}$, which are preprocessed by kernel trick to get $D_1^{\mathbf{K}} = \{\mathbf{k}_{1:n}, y_{1:n}\}$. We then calculate dimension reduction directions V based on Eq.(14), and transform the kernel data through V: $D_1^{\mathbf{K}} \xrightarrow{V} D_{V1}^{\mathbf{K}}$ and apply GP-UCB on the subspace $D_{V1}^{\mathbf{K}}$. The Bayesian optimization process is very similar to normal BO, while we need to notice that we could only infer $\mathbf{u}$ from $\mathbf{x}$ and it is impossible to recover $\mathbf{x}$ from $\mathbf{u}$ through V. Different from [Djolonga *et al.*, 2013] using a low-rank matrix recovery method to get an approximate $\hat{\mathbf{x}}$, we perform heuristic algorithm (CMA-ES) directly on input space $\mathcal{X}$, and find the point $\mathbf{x}_t^*$: $\mathbf{x}^* = \mathbf{argmax}_{\mathbf{x}\in\mathcal{X}}\mathbf{UCB}(\mathbf{u}), \mathbf{u} = V\mathbf{k}, \mathbf{k} = \kappa(\mathbf{x}, :)$. We update dimension reduction direction every iteration and update kernel hyperparameters for certain iterations.

4 Experiments

To evaluate the performance of our SIR-BO and KISIR-BO[1], we have conducted a set of experiments on two synthetic functions and two real datasets. We compare our approach against the standard BO, REMBO[2], ADD-GP-UCB[3], and Dropout-BO[4]. We adopt Gaussian kernel with lengthscale 0.1 for kernelizing input, and Gaussian kernel with adaptive lengthscale for Gaussian processes learned by maximizing marginal likelihood function through DIRECT, and optimize acquisition function by CMA-ES.

[1]Source code and used data are available at https://github.com/MiaoZhang0525

[2]The code of REMBO is available at https://github.com/ziyuw/rembo [Wang *et al.*, 2013].

[3]The code of ADD-BO is available at https://github.com/zi-w/Structural-Kernel-Learning-for-HDBBO [Kandasamy *et al.*, 2015].

[4]We implement Dropout-BO based on [Li *et al.*, 2017].

	Branin($d_e = 2$)					
	D=200			D=20000		
d	2	10	20	2	10	20
REMBO	**5e^{-7}±1.3e^{-3}**	0.43±0.56	0.35±0.45	**5e^{-7}±1.5e^{-7}**	0.77±0.69	0.80±1.10
SIR-BO	0.11± 0.12	0.20±0.22	0.24±0.19	0.18±0.07	**0.08±0.05**	0.16±0.08
KISIR-BO	0.25±0.23	**0.14±0.13**	**0.17±0.17**	0.14±0.09	0.13±0.10	**0.15±0.09**
	Trimodel($d_e = 2$)					
REMBO	**0.21±0.66**	3.70±2.90	10.0±5.40	0.20±0.57	4.20±1.90	5.90±4.20
SIR-BO	**0.22±0.15**	0.20±0.23	0.27±0.20	**0.13±0.12**	0.14±0.11	**0.12±0.14**
KISIR-BO	0.30±0.23	**0.23±0.22**	**0.18±0.19**	0.16±0.12	**0.04±0.03**	0.14±0.11

Table 1: Simple regrets under 500 evaluation budget for d_e=2 synthetic functions with different assumed dimension d.

4.1 Optimization on synthetic functions

In this section, we demonstrate our approaches on two synthetic function: the first one is standard Branin function [Lizotte, 2008] with intrinsic dimension $d_e = 2$, embedded into D-dimensional space, defined as:

$$f(\mathbf{x}) = g(\mathbf{u}) = a(x_j - bx_i^2 + cx_i - r) + s(1-t)cos(x_i) + s \quad (19)$$

where i, j are two randomly selected dimensions, $a = 1, b = \frac{5.1}{4\pi^2}, c = \frac{5}{\pi}, r - 6, s = 10, t = \frac{1}{8\pi}$, and the global minimum is $g(\mathbf{u}^*) = 0.397887$ at $\mathbf{u}^* = (-\pi, 12, 275), (\pi, 2.275)$ and $(9.42478, 2.475)$. The second synthetic function is a trimodal function with $d_e = 2$:

$$\begin{aligned} f(\mathbf{x}) = g(\mathbf{u}) = \log(0.1 \times \text{mvnpdf}(\mathbf{u}, \mathbf{c}_1, \sigma^2) + \\ 0.8 \times \text{mvnpdf}(\mathbf{u}, \mathbf{c}_2, \sigma^2) + 0.1 \times \text{mvnpdf}(\mathbf{u}, \mathbf{c}_3, \sigma^2)) \end{aligned} \quad (20)$$

where $\sigma^2 = 0.01 d_e^{0.1}$ and $\text{mvnpdf}(\mathbf{x}, \mu, \sigma^2)$ is multivariates Gaussian distribution and c_i are fixed centers in $\mathbb{R}^{d_e}$. The global maximum is $f(\mathbf{x}^*) = g(\mathbf{u}^*) = g(\mathbf{c}_2) = 2.4748$ at the center $\mathbf{c}_2$ with highest probability 0.8.

We study the cases with different input dimensions and reduction dimensions: $\{(D = 20, d = 5), (D = 200, d = 10), (D = 2000, d = 10), (D = 20000, d = 10)\}$. One thing needed be noticed is that, ADD-GP-UCB is not efficient in extremely-high dimensional Bayesian optimization for the

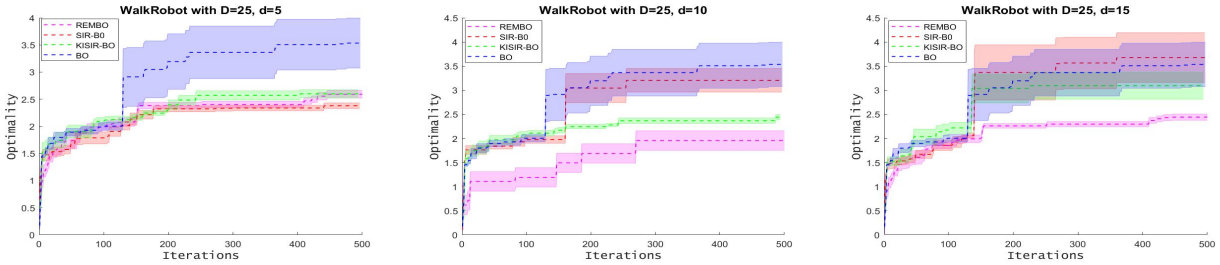

Figure 3: Performance for Walk Robot controlling with different assumed dimensions of subspace. We plot means and $1/4$ standard deviations for each algorithms with 20 independent runs, compared with standard BO optimizing full 25 parameters.

combinatorial explosion to find the optimal decomposition [Baptista and Poloczek, 2018] and we do not apply ADD-GP-UCB to cases with $D \geq 200$. Figure 1 demonstrates the results where our approach get 6 best performances in all 8 cases, and outperforms other baselines in all high dimensions ($D \geq 200$) cases. More specifically, ADD-BO performs best in Branin function with median dimension (D=20) and REMBO also performs excellent in this case and Branin with remaining dimensions. While they both perform poorly in the nonlinear Trimodel function which is not surprised due to the nonlinear function does not fit the prior linear embedding assumption of REMBO and additive assumption of ADD-BO. Standard BO obtains the best performance in low dimension (D=20) for Trimodel function, while it degrades among high dimensions. More interesting, we could find that several basline HDBO methods (REMBO, ADD-BO and Dropout) all perform worse than standard BO in Trimodel function in all dimensions, which again shows that those HDBO methods usually perform poorly when the prior assumption does not fit the objective functions. In the contrast,our SIR-BO and KISIR-BO perform well both in linear function and nonlinear function without the need of prior assumption of objective function, since it could automatically learn the intrinsic sub-structure of objective function during the optimization.

Although REMBO reaches surprised excellent performance in several situations, it is highly dependent on the assumed reduced dimension d. To investigate robustness and the effects of the assumed reduced dimension, we further conduct experiments to compare our SIR-BO and KSIR-BO with REMBO under different assumed reduced dimension d. Table 1 analyses the effect of the number of assumed effective dimension d, and we can find that REMBO is much dependent on the assumed dimension, where it performs poorly when the assumed dimension d differ from the intrinsic dimension d_e while perform excellently when the assumed dimension equals to intrinsic dimension d=d_e=2. In contrast, our approach is much more robust, where exists no dramatic degrades during the variation of d.

4.2 Optimization on real applications

Following [Wang *et al.*, 2017], we consider two real-word tasks: training a neural network for Boston dataset with 4 parameters, and controlling a three-link walk robot with 25 parameters [Westervelt *et al.*, 2007]. To emulate the practical fact that we may not know how many parameters the problem dependents on, we augment the number of parameter to $\{(D$=20,d=5),(D=200,d=10),(D=2000,d=10)$\}$ for Boston dataset and $\{(D$=50,d=30),(D=200,d=30),(D=2000,d=30)$\}$ for walk robot with dummy variables. Figure 2 illustrates the superiority of our SIR-BO and KISIR-BO which could find an effective low-dimensionality for high-dimension hyperparameter configurations. REMBO and ADD-BO both perform poorly in the two real-world applications when the prior assumptions are not in line with the objective functions, while SIR-BO and KISIR-BO perform well since it could learn the intrinsic structure during the optimization.

More interesting, [Hutter *et al.*, 2014] has demonstrated that hyperparameter optimization problem may lie in a low dimensionality of dependent parameters that some hyperparameters are truly unimportant while some hyperparameters are much important. We further conduct experiments on the robot control problem with 25 dependent parameters and optimize it on the subspace of the intrinsic space that $d < d_e$. We study this case with $\{(D$=d_e=25,d=5),(D=d_e=25,d=10),(D=d_e=25,d=15)$\}$, and the results in Figure 3 demonstrate that the hyperparameter optimization could be conducted in a subspace of dependent parameters (when d=15, SIR-BO outperform normal BO with optimizing all dependent parameters), which is in line with the hypothesis of [Hutter *et al.*, 2014], and suggests that perform a dimension reduction is necessary for hyperparameter optimization problem in machine learning.

5 Conclusion and future work

This paper proposed a novel model for high dimensional Bayesian optimization which introduces SIR to automatically learn the intrinsic structure of objective function. In particular, two tricks, reduced SVD and kernelizing input, are developed to reduce computational complexity and learn nonlinear subsets. We theoretically analyze our approach and derive regret bounds. Experimental results demonstrate that our approaches not only perform excellently on high dimension problems, but also are effective on extremely-high dimension problems with D=20000. However, our approach still works on high dimensional space to optimize acquisition functions, which is not efficient. In the future works, we try to find more efficient ways to optimize acquisition functions, and also focus on extending our approaches to hyperparameter tuning and network architecture search in Deep Meta-Learning.

References

[Baptista and Poloczek, 2018] Ricardo Baptista and Matthias Poloczek. Bayesian optimization of combinatorial structures. In *International Conference on Machine Learning*, pages 462–471, 2018.

[Bergstra *et al.*, 2011] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011.

[Chen *et al.*, 2012] Bo Chen, Rui M. Castro, and Andreas Krause. Joint optimization and variable selection of high-dimensional gaussian processes. pages 1423–1430, 2012.

[Cunningham and Ghahramani, 2015] John P Cunningham and Zoubin Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *The Journal of Machine Learning Research*, 16(1):2859–2900, 2015.

[Deisenroth *et al.*, 2015] Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):408–423, 2015.

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

[Hutter *et al.*, 2014] Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. An efficient approach for assessing hyperparameter importance. In *International Conference on Machine Learning*, pages 754–762, 2014.

[Kandasamy *et al.*, 2015] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning*, pages 295–304, 2015.

[Kandasamy *et al.*, 2018] Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric P Xing. Neural architecture search with bayesian optimisation and optimal transport. In *Advances in Neural Information Processing Systems*, pages 2020–2029, 2018.

[Li *et al.*, 2016] Chun-Liang Li, Kirthevasan Kandasamy, Barnabás Póczos, and Jeff Schneider. High dimensional bayesian optimization via restricted projection pursuit models. In *Artificial Intelligence and Statistics*, pages 884–892, 2016.

[Li *et al.*, 2017] Cheng Li, Sunil Gupta, Santu Rana, Vu Nguyen, Svetha Venkatesh, and Alistair Shilton. High dimensional bayesian optimization using dropout. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2096–2102. AAAI Press, 2017.

[Li, 1991] Ker-Chau Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

[Lizotte, 2008] Daniel James Lizotte. *Practical bayesian optimization*. PhD thesis, University of Alberta, 2008.

[Rasmussen and Williams, 2004] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, Massachusetts, 2004.

[Rolland *et al.*, 2018] Paul Rolland, Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. High-dimensional bayesian optimization via additive models with overlapping groups. In *International Conference on Artificial Intelligence and Statistics*, pages 298–307, 2018.

[Snoek *et al.*, 2012] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[Srinivas *et al.*, 2012] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[Tan and Mukherjee, 2018] Zilong Tan and Sayan Mukherjee. Subspace-induced gaussian processes. *arXiv preprint arXiv:1802.07528*, 2018.

[Ulmasov *et al.*, 2016] Doniyor Ulmasov, Caroline Baroukh, Benoit Chachuat, Marc Peter Deisenroth, and Ruth Misener. Bayesian optimization with dimension scheduling: Application to biological systems. In *Computer Aided Chemical Engineering*, volume 38, pages 1051–1056. Elsevier, 2016.

[Wang *et al.*, 2013] Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, Nando De Freitas, et al. Bayesian optimization in high dimensions via random embeddings. In *IJCAI*, pages 1778–1784, 2013.

[Wang *et al.*, 2017] Zi Wang, Chengtao Li, Stefanie Jegelka, and Pushmeet Kohli. Batched high-dimensional bayesian optimization via structural kernel learning. In *International Conference on Machine Learning*, pages 3656–3664, 2017.

[Westervelt *et al.*, 2007] Eric R Westervelt, Christine Chevallereau, Jun Ho Choi, Benjamin Morris, and Jessy W Grizzle. *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2007.

[Wu *et al.*, 2007] Qiang Wu, F Liang, and S Mukherjee. Regularized sliced inverse regression for kernel models. Technical report, Technical Report 07-25, ISDS, Duke Univ, 2007.

[Wu, 2008] Han-Ming Wu. Kernel sliced inverse regression with applications to classification. *Journal of Computational and Graphical Statistics*, 17(3):590–610, 2008.

[Yeh *et al.*, 2009] Yi-Ren Yeh, Su-Yun Huang, and Yuh-Jye Lee. Nonlinear dimension reduction with kernel sliced inverse regression. *IEEE Transactions on Knowledge and Data Engineering*, 21(11):1590–1603, 2009.