High-dimensional Bayesian optimization using low-dimensional feature spaces

Riccardo Moriconi
Department of Computing
Imperial College London
London SW7 2AZ, UK
r.moriconi16@imperial.ac.uk

Marc Peter Deisenroth
University College London
London WC1V 6LJ, UK
m.deisenroth@ucl.ac.uk

K. S. Sesh Kumar
Imperial College London
London SW7 2AZ, UK
seshkumar@gmail.com

Abstract

Bayesian optimization (BO) is a powerful approach for seeking the global optimum of expensive black-box functions and has proven successful for fine tuning hyperparameters of machine learning models. However, in practice, BO is typically limited to optimizing 10–20 parameters. To scale BO to high dimensions, we normally make structural assumptions on the decomposition of the objective and/or exploit the intrinsic lower dimensionality of the problem, e.g. by using linear projections. The limitation of aforementioned approaches is the assumption of a linear subspace. We could achieve a higher compression rate with nonlinear projections, but learning these nonlinear embeddings typically requires much data. This contradicts the BO objective of a relatively small evaluation budget. To address this challenge, we propose to learn a low-dimensional feature space jointly with (a) the response surface and (b) a reconstruction mapping. Our approach allows for optimization of the acquisition function in the lower-dimensional subspace. We reconstruct the original parameter space from the lower-dimensional subspace for evaluating the black-box function. For meaningful exploration, we solve a constrained optimization problem.

1 Introduction

Bayesian optimization (BO) is a useful model-based approach to global optimization of black-box functions that are expensive to evaluate [25, 28, 38]. This sample-efficient technique for optimization has proven effective in experimental design of machine learning algorithms [5], robotics applications [9] and medical therapies [50] for optimization of spinal-cord electro-stimulation. Despite its great success, BO is practically limited to optimizing 10–20 parameters, and a large body of literature has been devoted to address scalability issues to elevate BO to high-dimensional optimization problems, such as discovery of chemical compounds [16] or automatic software configuration [23].

The standard BO routine consists of two key steps: (i) estimating the black-box function from data through a probabilistic surrogate model, usually a Gaussian process (GP), referred to as the *response surface*; (ii) maximizing an *acquisition function* that computes a score that trades off exploration and exploitation according to uncertainty and optimality of the response surface. As the dimensionality of the input space increases, these two steps become challenging. The sample complexity to ensure good coverage of inputs for learning the response surface is exponential in the number of dimensions [48]. With only a small evaluation budget the learned response surface and the resulting acquisition function are characterized by vast flat regions interspersed with highly non convex landscapes [41]. This renders the maximization of the acquisition inherently hard [14].

However, high-dimensional data often possesses a lower intrinsic dimensionality, which can also be exploited for optimization. A *feature mapping* can then used to map the original D-dimensional data

4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.

onto a $d \ll D$-dimensional manifold. For example, in [54], the authors used random linear mappings in the context of BO to reduce dimensionality. Similar approaches use linear dimensionality reduction drive exploration in BO to actively learn this linear embedding [14]. While these methods perform well in practice they are restricted to linear subspaces of the original domain. With nonlinear embeddings higher compression rates would be possible.

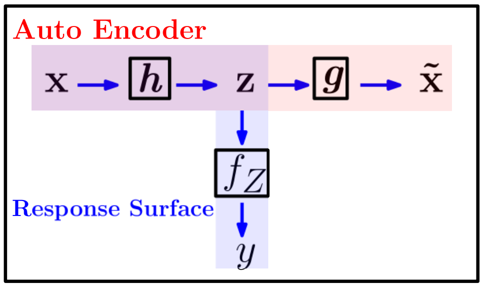


Figure 1: Model for Bayesian optimization on data manifolds, jointly solving two distinct tasks: (i) a regression from feature space to observations (in blue) and (ii) a reconstruction mapping from feature space to high-dimensional space (in red).

A generalization to nonlinear subspaces was proposed in [16, 17, 29] and [20]. In [16], a low-dimensional data representation is learned with variational autoencoders (VAEs). A characteristic of this approach is that the required amount of data necessary for learning a meaningful representation substantially exceeds small evaluation budgets that often constrain BO. However, in the specific application of automatic discovery of molecules, where libraries of existing compounds are available prior to optimization, this approach makes much sense. VAE models [35] were used to propagate uncertainty of latent space representations through the response surface model with *Gaussian process latent variable models* [31, 30, 51, 32]. However, in [35], the latent space representation is not learned specifically for the regression task. Gradient-based methods [1] have been used to learn a lower-dimensional Riemannian manifold for optimization and sampling.

Nonlinear embeddings also allow for modeling non-stationary objective functions. In this context, a composition of GPs, referred to as *deep GPs* [12, 46, 10, 11, 22] is especially useful when the response surface is characterized by abrupt changes or has constraints. An extensive investigation on the employment of deep GP models in BO is presented in [10, 21]. In our work, we also exploit the idea of learning highly nonlinear functions through the composition of simpler ones [33], but we rather elaborate on deterministic dimensionality reduction and optimization in feature space.

In this paper, we propose a BO algorithm for high-dimensional optimization, which learns a nonlinear feature mapping $\mathbf{h} : \mathbb{R}^D \to \mathbb{R}^d$ to reduce the dimensionality of the inputs, and a *reconstruction mapping* $\mathbf{g} \colon \mathbb{R}^d \to \mathbb{R}^D$ based on GPs to evaluate the true objective function, jointly, see Figure 1. This allows us to optimize the acquisition function in a lower-dimensional feature space, so that the overall BO routine scales to high-dimensional problems that possess an intrinsic lower dimensionality. Finally, we formulate a constrained maximization of the acquisition function in feature space to prevent meaningless reconstructions.

2 Bayesian optimization in low-dimensional feature spaces

We consider global minimization problems of the form

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathcal{X}} f_X(\mathbf{x}) \tag{1}$$

with a high-dimensional input space $\mathcal{X} = [0, 1]^D$, but where the objective $f_X \colon \mathcal{X} \to \mathbb{R}$ possesses an intrinsic lower dimensionality. In our setting, we consider functions f_X that are expensive to evaluate and for which we are allowed a small budget of evaluation queries to express our best guess of the optimum's location $\mathbf{x}^*$. We further assume we have access only to noisy evaluations of the objective $y = f_X + \varepsilon$, where the noise $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$ is i.i.d. Gaussian. We restrict ourselves to the typical setting, where neither gradients nor convexity properties of f_X are available.

```
Input: $\mathbf{X}_0 \in \mathbb{R}^{N_0 \times D}$
Observations: $\mathbf{y}_0 \in \mathbb{R}^{N_0}$
for $t = 0, 1, 2, ...$ do
    Response surface learning $f_X = f_Z \circ \mathbf{h}$
      Dimensionality reduction  $\mathbf{Z}_t = \mathbf{h}(\mathbf{X}_t)$
      Low-dimensional surface  $p(f_Z | \mathbf{Z}_t, \mathbf{y}_t)$
    Optimal input selection $\mathbf{x}_{t+1}$
      Acquisition  $\mathbf{z}_* = \operatorname{argmax}_{\mathbf{z} \in \mathcal{Z}} \alpha(\mathbf{z})$
      Input reconstruction  $\mathbf{x}_{t+1} := \tilde{\mathbf{x}}_* = \mathbf{g}(\mathbf{z}_*)$

    Evaluation
      $y_{t+1} = f_X(\mathbf{x}_{t+1}) + \varepsilon$
    $\mathbf{X}_t \cup \{\mathbf{x}_{t+1}\}$,    $\mathbf{y}_t \cup \{y_{t+1}\}$
end for
Return $\mathbf{x}^* = \arg\min \mathbf{y}_t$
```

Algorithm 1: Key steps of BO in feature space.

The main steps of a BO routine involve (i) *response surface learning*, (ii) *optimal input selection* $\mathbf{x}_*$ and (iii) *evaluation* of the objective function f_X at $\mathbf{x}_*$. The first step trains a probabilistic surrogate model $p(f_X)$, the response surface, which describes the black-box relationship between inputs $\mathbf{x}$ and observations y. In the $t+1$st iteration of BO, the optimal input selection step finds an input $\mathbf{x}_{t+1}$ that maximizes an *acquisition function* $\alpha(\cdot)$, which describes the added value of an input $\mathbf{x}$. The evaluation step observes the noise-corrupted outcome of the true objective function $f_X(\mathbf{x}_{t+1})+\varepsilon$ at the selected location. These steps are summarized in lines 4, 7 and 10 of Algorithm 1, respectively. In relatively high-dimensional settings ($D > 20$) both the response surface learning and optimal input selection become computationally challenging.

In our work, we exploit the effective low dimensionality of the objective function for BO in a lower-dimensional*feature space* $\mathcal{Z} \subset \mathbb{R}^d$, where $d \ll D$. In particular, we express the true objective function $f_X : \mathbb{R}^D \to \mathbb{R}$ as a composition of a feature mapping $\mathbf{h} : \mathbb{R}^D \to \mathbb{R}^d$ and a function $f_Z \colon \mathcal{Z} \to \mathbb{R}$ so that $f_X = f_Z \circ \mathbf{h}$. The lower-dimensional feature space allows for both learning the response surface f_X and maximizing an acquisition function α with domain $\mathcal{Z}$, which yields optimizer $\mathbf{z}_*$.

However, we cannot evaluate the true objective f_X directly at the low-dimensional features $\mathbf{z}_*$, but need to project $\mathbf{z}_*$ back into the D-dimensional data space $\mathcal{X}$. We therefore define a *reconstruction* mapping $\mathbf{g} \colon \mathcal{Z} \to \mathcal{X}$. We can think of this mapping as a decoder within an auto-encoder framework. We model both the composition $f_X = f_Z \circ \mathbf{h}$ and the reconstruction with Gaussian process models. The algorithm in Algorithm 1 summarizes the main steps of the feature space BO.

In the following, we detail the model (see Figure 1) for jointly learning the feature map $\mathbf{h}(\cdot)$, the low-dimensional response surface in feature space f_Z, and the reconstruction mapping $\mathbf{g}(\cdot)$.

2.1 Manifold Gaussian processes for response surface learning in feature space

In our optimization problem, we expect the response surface to predict the value of the black-box objective function f_X with calibrated uncertainty associated with each prediction. Gaussian processes (GPs) [42] are probabilistic models that allow for an analytic computation of posterior predictive function values within a Bayesian framework, and they are the standard model in BO for modeling the response surface.

A GP is a distribution over functions $f_Z \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$ and is fully specified by a *mean function* $m \colon \mathcal{Z} \to \mathbb{R}$, and a *covariance function/kernel* $k \colon \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$. The kernel computes the covariance between pairs of function values as a function of the corresponding inputs, i.e. $\mathrm{Cov}(f_Z(\mathbf{z}), f_Z(\mathbf{z}')) = k(\mathbf{z}, \mathbf{z}')$, and thereby encodes regularity assumptions about f_Z, such as smoothness or periodicity. Common kernel choices in the BO literature include the *squared exponential* and *Matérn* kernels [13].

In our feature space optimization, we address lines 5–6 of Algorithm 1 as a single learning problem. Therefore, we need a GP that learns useful representations $\mathbf{z}$ of inputs $\mathbf{x}$ for the regression task together with f_Z. A manifold Gaussian process (mGP) [36, 8, 56] addresses this issue by composing two mappings: The deterministic feature map $\mathbf{h}$ with parameters $\boldsymbol{\theta}_h$ and a GP $f_Z \sim \mathcal{GP}(m, k)$ with kernel hyper-parameters $\boldsymbol{\theta}_k$. The GP models the relationship between features $\mathbf{z}$ and function values y in observation space. The resulting composite model $f_X = f_Z \circ \mathbf{h}$ is a GP so that $f_X \sim \mathcal{GP}(m_m, k_m)$ with mean and covariance functions given by

$$m_m(\mathbf{x}) = m(\mathbf{h}(\mathbf{x})), \tag{2}$$

$$k_m(\mathbf{x}, \mathbf{x}') = k(\mathbf{h}(\mathbf{x}), \mathbf{h}(\mathbf{x}')), \tag{3}$$

respectively. Given high-dimensional training inputs $\mathbf{X}$ and corresponding observations $\mathbf{y}$ of the objective function, we train the model, which is parameterized by $\{\boldsymbol{\theta}_h, \boldsymbol{\theta}_k\}$, by maximizing the log-marginal likelihood (evidence) [42]

$$\{\boldsymbol{\theta}_h^*,\, \boldsymbol{\theta}_k^*\} \in \operatorname*{arg\,max}_{\boldsymbol{\theta}_h, \boldsymbol{\theta}_k} \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}_h, \boldsymbol{\theta}_k). \tag{4}$$

This objective allows us to learn a low-dimensional feature embedding as a by-product of the supervised GP regression framework. Unsupervised dimensionality reduction usually solves an orthogonal task to that of learning a response surface. Algorithms, such as PCA [40, 24] or variational auto-encoders [43, 27, 37], achieve compact data representations by optimizing objectives that are not necessarily useful in a supervised setting [53]. The mGP, instead, leads to low-dimensional representations that are optimal (locally) for the regression task at hand.

We use a multi-layer feedforward neural network with sigmoid activation functions as a feature map (encoder) $\mathbf{h}$, resulting in a feature space $\mathcal{Z} = [0,1]^d$. Neural networks as an explicit feature map within an mGP have already been applied successfully for modeling non-smooth responses in bipedal robot locomotion [8]. Deep network architectures have also proven useful for orientation extraction from high-dimensional images [56].

With a Gaussian likelihood, the mGP posterior predictive distribution at a test point $\mathbf{x}_\star \in \mathcal{X}$ is Gaussian distributed with mean and variance given by

$$\begin{aligned}\mathbb{E}[f_X(\mathbf{x}_\star)] &= m_m(\mathbf{x}_\star) + k_m(\mathbf{x}_\star, \mathbf{X})\mathbf{K}_{my}^{-1}(\mathbf{y} - m_m(\mathbf{X})) \\ &= m(\mathbf{z}_\star) + k(\mathbf{z}_\star, \mathbf{Z})\mathbf{K}_y^{-1}(\mathbf{y} - m(\mathbf{Z}))\end{aligned} \tag{5}$$

$$\begin{aligned}\mathbb{V}[f_X(\mathbf{x}_*)] &= k_m(\mathbf{x}_\star, \mathbf{x}_\star) - k_m(\mathbf{x}_\star, \mathbf{X})\mathbf{K}_{my}^{-1}k_m(\mathbf{X}, \mathbf{x}_\star) \\ &= k(\mathbf{z}_\star, \mathbf{z}_\star) - k(\mathbf{z}_\star, \mathbf{Z})\mathbf{K}_y^{-1}k(\mathbf{Z}, \mathbf{z}_\star),\end{aligned} \tag{6}$$

respectively. Here, $k_m(\mathbf{x}_\star, \mathbf{X}) = k(\mathbf{z}_\star, \mathbf{Z}) = [k(\mathbf{z}_\star, \mathbf{z}_i)]_{i=1}^N$, $\mathbf{K}_{my} := k_m(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I}$, $\mathbf{K}_y := k(\mathbf{Z}, \mathbf{Z}) + \sigma_n^2\mathbf{I}$, $k_m(\mathbf{X}, \mathbf{X}) = k(\mathbf{Z}, \mathbf{Z})$ and $m_m(\mathbf{X}) = m(\mathbf{Z}) = [m(\mathbf{z}_i)]_{i=1}^N$ computes the prior mean function evaluated at the embedded training inputs $\mathbf{Z} = \mathbf{h}(\mathbf{X})$. Note that posterior predictions can be computed using both the feature and data space.

The mGP defines a GP on $\mathcal{X}$, but allows us to learn a response surface in the lower-dimensional feature space $\mathcal{Z}$. This is key for optimizing the acquisition function in a low-dimensional space $\mathcal{Z}$ instead of the original data/parameter space $\mathcal{X}$. Thus far we have detailed the feature BO procedure up to line 8 in Algorithm 1. Once we have found an optimizer $\mathbf{z}_*$ of the acquisition function, we need to project it back into the original data space $\mathcal{X}$ in order to evaluate the true objective f_X. This can be done by means of a reconstruction mapping (decoder), which we detail in the following.

2.2 Input reconstruction with manifold multi-output Gaussian processes

Here, we present the reconstruction part (decoder) of our feature space optimization model described in Figure 1. We are interested in modeling the functional relationship between the feature space $\mathcal{Z}$ and the data space $\mathcal{X}$ for step 9 in Algorithm 1, which requires us to evaluate f_X. We therefore consider a vector-valued function $\mathbf{g} = \{g_i\}_{i=1}^D$, where each component function $g_i\colon \mathcal{Z} \to \mathcal{X}_i$ maps vectors in feature space to the i-th coordinate of high-dimensional data, i.e. $g_i(\mathbf{z}) = \tilde{x}^{(i)} \in \mathcal{X}$. Multi-output Gaussian processes (MOGPs) [4, 3, 58, 57, 2, 39, 47, 6] define a prior over vector-valued functions and explicitly allow for output correlations. An MOGP $\mathcal{GP}(\mathbf{m}, \mathbf{K})$ is fully specified by a mean vector function $\mathbf{m}\colon \mathcal{Z} \to \mathbb{R}^D$ and a positive, semi-definite matrix-valued covariance function $\mathbf{K}\colon \mathcal{Z} \to \mathbb{R}^{D\times D}$, which expresses the correlation between observations in the same output coordinate and cross-correlation between the D different outputs. Various formulations of the matrix-valued kernel correspond to specific generative model assumptions for the multiple outputs g_i.

In our work, we consider the *intrinsic coregionalization model* (ICM) [19, 52], which structures the covariance matrix as a Kronecker product and allows for efficient training and predictions. This model is particularly suitable for trading off number of model parameters and expressiveness of the vector valued function. In particular, the ICM facilitates information sharing across different tasks by adopting the same covariance function and has successfully been adopted in robotics for learning inverse dynamics [55]. Hence, this model requires fewer parameters than the *linear model of coregionalization* [4], and allows for exploiting properties of the Kronecker product for efficient training and posterior computation.

Intrinsic coregionalization model The ICM [19, 52] applies a linear mapping to a set of latent functions. In particular, we consider a set of P latent functions $u_i\colon \mathcal{Z} \to \mathbb{R}$, that are assumed to be *sample paths*, i.e. sample functions independently drawn from the same GP prior $\mathcal{GP}(m_c, k_c)$. The ICM model expresses the vector-valued function as a linear combination of these sample functions

$$\mathbf{g}(\mathbf{z}) = \mathbf{A}\mathbf{u}(\mathbf{z}), \tag{7}$$

where $\mathbf{u}(\mathbf{z}) \in \mathbb{R}^P$ is the collection of the P sample paths' evaluations at $\mathbf{z}$, and $\mathbf{A} \in \mathbb{R}^{D\times P}$ is the linear mapping that couples the independent vector and parameterizes the ICM model. As a result,

$\mathbf{g}$ is a MOGP $\mathcal{GP}(\mathbf{m}, \mathbf{K})$ with mean function $\mathbf{m} = \mathbf{A}\mathbf{m}_c$, where $\mathbf{m}_c = [m_c]_{i=1}^{P}$ is obtained by repeating the single-valued mean function m_c in a P-vector. The covariance function is expressed as $\mathbf{K}(\mathbf{z}, \mathbf{z}') = \mathbf{A}\mathbf{A}^T \otimes k_c(\mathbf{z}, \mathbf{z}')$, where k_c is the covariance function for the GP prior and $\otimes$ denotes the Kronecker product.

Reconstruction Model For the reconstruction task in line 9 of Algorithm 1, we introduce the manifold MOGP with intrinsic coregionalization model (mMOGP), which shares the feature map $\mathbf{h}$ with the manifold GP used for learning the response surface in Section 2.1. In our work, we assume without loss of generality a zero-mean vector function for the mMOGP $\mathcal{GP}(\mathbf{0}, \mathbf{B} \otimes k_M)$, where $k_M(\mathbf{x}, \mathbf{x}') = k_c(\mathbf{h}(\mathbf{x}), \mathbf{h}(\mathbf{x}'))$ and the matrix $\mathbf{B} = \mathbf{A}\mathbf{A}^T$. We model the composition $\mathbf{g} \circ \mathbf{h}$, which describes the relationship between the data space $\mathcal{X}$ and feature space $\mathcal{Z}$, jointly with the MOGP mapping from feature space back to the data space $\mathcal{X}$. Albeit sharing the same set of parameters $\boldsymbol{\theta}_h$ for the feature mapping, the mMOGP uses a kernel $k_c \neq k$ that differs from the one used for modeling the response surface (see Section 2.1).

2.3 Joint training

The joint training of the mGP, which models the response surface, and the mMOGP, which is used for the reconstruction (see also Figure 1) is performed via log marginal likelihood maximization.

$$\mathcal{L} \propto -\mathbf{y}^T\mathbf{K}_y^{-1}\mathbf{y} - \log|\mathbf{K}_y| - \mathbf{x}_V^T\mathbf{K}_V^{-1}\mathbf{x}_V - \log|\mathbf{K}_V| + \text{const} \tag{8}$$

Here, $\mathcal{L}$ comprises terms from both the mGP and mMOGP models, where $\mathbf{K}_y$ is defined in (5), and the covariance matrix of the mMOGP $\mathbf{K}_V = \bar{\mathbf{K}} + \sigma_n^2\mathbf{I}$ is obtained by evaluating the Kronecker product $\bar{\mathbf{K}} = \mathbf{B} \otimes k_c(\mathbf{Z}, \mathbf{Z})$ with the mMOGP kernel k_c. The vector $\mathbf{x}_V$ is a concatenation of the columns of the data $\mathbf{X}$. The maximizers $[\boldsymbol{\theta}_h^*, \boldsymbol{\theta}_k^*, \boldsymbol{\theta}_c^*]$ of the log-marginal likelihood are the parameters of the feature map $\mathbf{h}$ (which is shared between the mGP and the mMOGP) and the hyper-parameters of the two kernels k and k_c for the mGP and mMOGP, respectively. Optimization of (8) is performed via gradient-based methods [34, 59].

Modeling the black-box objective function f_X is orthogonal to the reconstruction problem. However, when training these tasks jointly, they have a regularization effect on the optimization of the parameters $\boldsymbol{\theta}_h$ of the feature embedding in the sense that the mapping $\mathbf{h}$ will not overfit to a single regression task: the parameters $\boldsymbol{\theta}_h$ will give rise to a feature space embedding that is useful for both the modeling of the objective and the reconstruction of the original inputs.

The major computational bottleneck for evaluating the marginal likelihood comes from the term $\mathbf{x}_V^T\mathbf{K}_V^{-1}\mathbf{x}_V$, which requires inverting an $ND \times ND$ covariance. We reduce the computational complexity of this operation to $\mathcal{O}(N^3) + \mathcal{O}(D^3)$ by exploiting the properties of the Kronecker product, tensor algebra [44] and structured GPs [15, 45]. Details can be found in the Appendix.

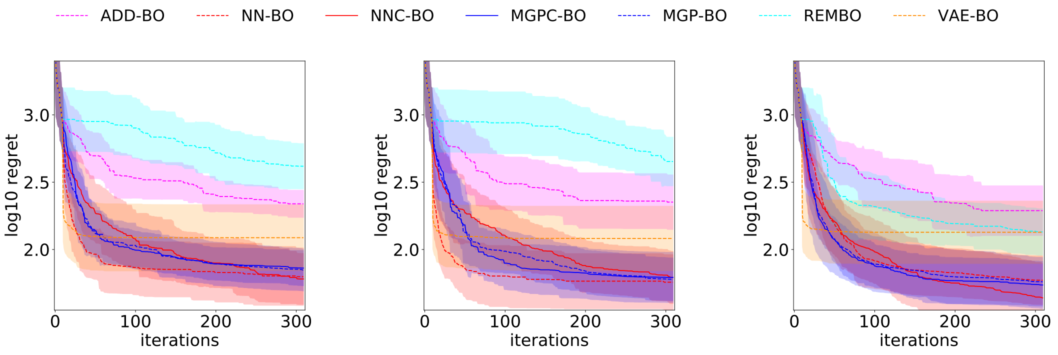
3 Constrained acquisition

We defined a joint probabilistic model for the response surface learning and the input reconstruction tasks, summarized in lines 4–6 and 9 of Algorithm 1, respectively. We are now concerned with the maximization of the acquisition function in feature space as described in line 8 of Algorithm 1. We aim at maximizing the acquisition function in a low-dimensional feature space of the original data/ parameter space $\mathcal{X}$. However, one problem that arises with the mMOGP decoding is that locations in feature space that are too far away from data will be mapped back to the mMOGP prior. Since the acquisition function is a key driver of exploration in BO, this is a problem. We address this limitation by introducing a constraint based on the Lipschitz continuity of the mMOGP posterior. This will ensure that candidates $\mathbf{z}_* \in \mathcal{Z}$ selected in feature space will not collapse to the origin $\mathbf{0} \in \mathbb{R}^D$ if the reconstruction is defined as $\tilde{\mathbf{x}}_* = \boldsymbol{\mu}(\mathbf{z}_*)$, where $\boldsymbol{\mu}$ is the posterior mean of the mMOGP.

We want to leverage information from observed data for the multi-output mapping and exploit it when optimizing the acquisition function in feature space. This can be achieved by introducing an upper bound to the Euclidean distance

$$\text{dist}(\mathbf{z}, \mathbf{Z}_t) = \min_{1\leq i\leq N_t} \|\mathbf{z}_i - \mathbf{z}\|_2 \tag{9}$$

in feature space between the optimization variable $\mathbf{z}$ and the embedded training data $\mathbf{Z}_t = [\mathbf{z}_1, ..., \mathbf{z}_{N_t}]$. Here, N_t is the number of data points available at BO iteration t. The desired upper bound is



(a) EI

(b) UCB

(c) PI

Figure 2: Results of BO in feature space with linear embedding. Baselines MGPC-BO and NNC-BO (solid) apply nonlinearly constrained acquisition maximization and recover same regret as the unconstrained versions MGP-BO and NN-BO.

obtained by exploiting the Lipschitz continuity property of the multi-output posterior mean for which $\|\boldsymbol{\mu}(\mathbf{z}) - \boldsymbol{\mu}(\mathbf{z}')\| \leq L\|\mathbf{z} - \mathbf{z}'\|$. Here, L denotes the Lipschitz constant of the posterior mean $\boldsymbol{\mu}$ of the mMOGP. For common kernels, such as Matérn$_{52}$ and squared exponential, the posterior mean is (at least) twice differentiable, and therefore Lipschitz continuous. The upper bound

$$\text{dist}(\mathbf{z}, \mathbf{Z}_t) \leq \mu_{\max}(\mathbf{z}^*)/L \tag{10}$$

allows us to specify how far from the data we can move in feature space without falling back to the prior on all coordinates of the reconstruction. Here $\mathbf{z}^*$ minimizes the Euclidean distance in (9), while the numerator on the right-hand side is the component-wise maximum of $\boldsymbol{\mu}(\mathbf{z}^*)$. We estimate the Lipschitz constant as the maximum norm of the Jacobian of the posterior mean of the mMOGP [18]

$$L = \arg\max_{\mathbf{z} \in \mathcal{Z}} \|\nabla_{\mathbf{z}} \boldsymbol{\mu}(\mathbf{z})\|. \tag{11}$$

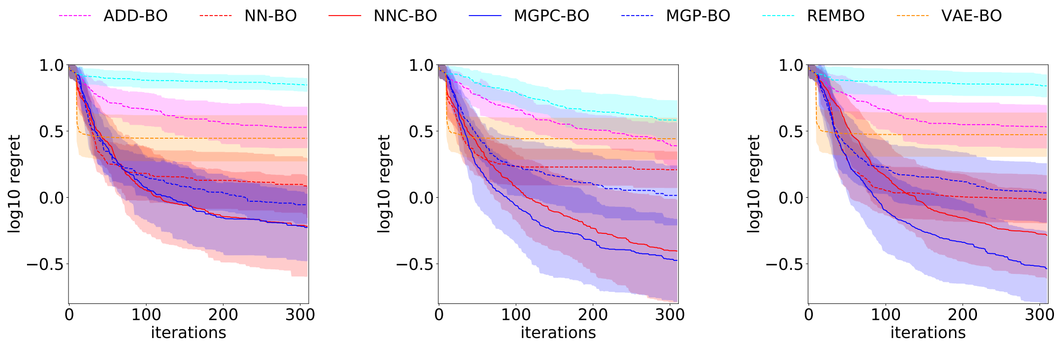
This maximization returns a valid Lipschitz constant [18] for the multi-output mapping for any choice of norm in (11). The Jacobian of the posterior mean is represented by a $D \times d$ matrix and we adopt the max norm $\|\nabla_{\mathbf{z}} \boldsymbol{\mu}(\mathbf{z})\|_\infty = \max |\mu^{'}_{i,j}|$ for $i = 1, ..., D$ and $j = 1, ..., d$. Lower values of valid Lipschitz constants L allow for exploration in larger regions of the feature space that still satisfy the nonlinear constraint in (10).

4 Results

We report results on a set of high-dimensional benchmark functions that possess an intrinsic low dimensionality. In particular, we (i) assess the benefits of adopting a model structure as presented in Figure 1; (ii) analyze the benefits of the constrained optimization of the acquisition function. Our purpose is to compare empirical performances across (a) different characterizations of the feature spaces, e.g. linear/nonlinear subspaces; (b) different properties of the objective function, e.g. additivity/non additivity.

We compare our approach (MGPC-BO) with the random embeddings optimization (REMBO) [54], which performs BO on a random linear subspace of the inputs. Additional baselines include additive models (ADD-BO) [26], which assumes an additive structure (across dimensions) of the objective f_X, and one recently proposed VAE-based model (VAE-BO) [16] that learns a feature space with deep networks offline. We also include a version of our model presented in Figure 1 (NNC-BO) that uses a hierarchical ICM for the input reconstruction mapping $\mathbf{g}$. The hierarchical ICM partitions the data space into low-dimensional disjoint subsets, i.e. $\mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_Q$, $\mathcal{X}_i \subset \mathbb{R}^3$, and assumes independence between reconstructions of different subsets, i.e. $\tilde{\mathbf{x}}^{(i)} \perp \tilde{\mathbf{x}}^{(j)}$, where $\tilde{\mathbf{x}}^{(i)} \in \mathcal{X}_i$, $\tilde{\mathbf{x}}^{(j)} \in \mathcal{X}_j$ for $i \neq j$. Moreover, the baselines MGP-BO and NN-BO correspond to same modeling as in MGPC-BO and NNC-BO, respectively, but without applying the nonlinear constraint in (10).

We evaluate the performances of all baselines across a set of common choices of acquisition functions: expected improvement (EI) [38], upper confidence bound (UCB) [49] and probability of improvement



(a) EI

(b) UCB

(c) PI

Figure 3: Optimization progression on product of sines with EI 3(a), UCB 3(b) and PI 3(c). Both MGPC-BO and NNC-BO learn low-dimensional representations of the objective that are useful for optimization.

(PI) [28]. The maximization of the acquisition function is identical for all baselines: We first perform a random search step with $5,000$ samples drawn uniformly at random, then we select the best 100 locations and apply gradient-based optimization from these starting locations. For box-constrained acquisition optimization we use L-BFGS-B [34, 59]. For constrained acquisition optimization with nonlinear constraints we use a trust-region interior point method [7].

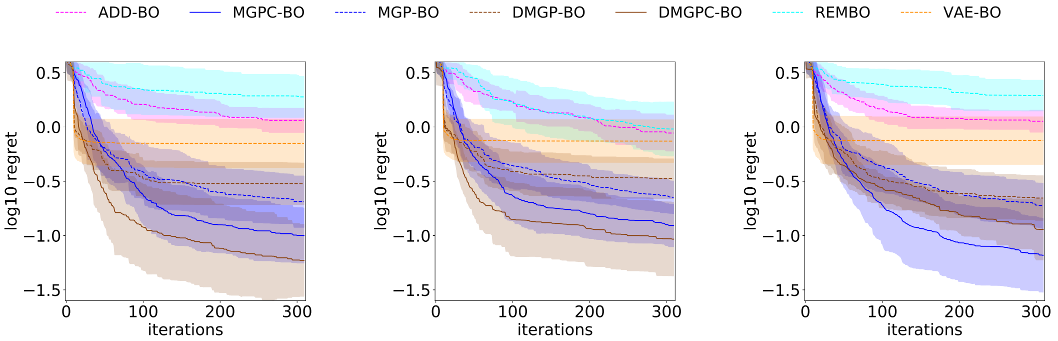
Each BO progression curve shows the mean and standard error of the immediate logarithmic regret $\log_{10} |f(\mathbf{x}_{\text{best}}(t)) - f_{\text{min}}|$, where f_{min} is the true minimum of f_X and $\mathbf{x}_{\text{best}}(t) \in \arg\min_{i=1:t} f_X(\mathbf{x}_i)$. Mean and standard error are computed over 20 experiments with different random initializations. All optimization experiments start with a budget of 10 data points and perform a total of 300 iterations.

4.1 Linear feature space

We consider benchmark functions that are defined in a $d = 10$-dimensional space. We map their input space to a $D = 60$-dimensional space using an orthogonal matrix $\mathbf{R}^{d\times D}$ so that the overall objective is $f_X(\mathbf{x}) = f(\mathbf{z}) = f(\mathbf{R}\mathbf{x})$.

Additive objective We minimize the *Rosenbrock* benchmark function $f(\mathbf{z}) = \sum_{i=1}^{d-1}[100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2]$ in a 10-dimensional feature space. Figure 2 shows that MGPC-BO and NNC-BO baselines descend quickly to relatively low regret in the early stages of optimization and recover the same regret at termination as the unconstrained baselines MGP-BO and NN-BO. This highlights the fast learning of feature-space representations that are effective for optimization when only few data samples are available. The VAE-BO baseline also improves quickly but lacks exploration due to an insufficiently expressive reconstruction mapping from feature space to data space. We highlight that the VAE-BO model was trained on a budget of 500 inputs-observations pairs prior to starting the BO experiments. This additional budget, however, still does not allow the VAE-BO to compare well with baselines that learn a feature mapping during optimization. REMBO shows a slower descent due to a limited exploration in at most d-directions of the data space, while the ADD-BO baseline suffers from the coupling effects of the linear dimensionality reduction $\mathbf{R}$.

Non-additive objective Here, we optimize the *Product of Sines* function $f(\mathbf{z}) = 10\sin(z_1)\prod_{i=1}^{10}\sin(z_i)$ and compare results when the additivity assumption is not satisfied. Figure 3 shows the regret curves obtained optimizing the objective on a 10-dimensional feature space. Solid lines describe the Lipschitz-regularized baselines MGPC-BO and NNC-BO (with nonlinear constraint), while dashed lines are baselines that apply box-constrained maximization of the acquisition in feature space. The NN-BO and MGP-BO regrets flatten early. The reason for this is that the acquisition function highlights locations in feature space that are too far away from the training data. In this setting, the decoder $\mathbf{g}$ returns the same high-dimensional reconstruction, which prevents BO from exploring. The constrained maximization of the acquisition is beneficial for both models. We also note that the REMBO baseline conforms to the intrinsic linear low-dimensionality assumption described Section 4.1. However, the linear reconstruction mapping applied by REMBO also suffers



(a) EI

(b) UCB

(c) PI

Figure 4: BO performances expressed as log regret of the product of sines function in a nonlinear embedding. Results are shown for EI 4(a), UCB 4(b) and PI 4(c). This limits the exploration in the high-dimensional space.

from non-injectivity, and this slows exploration in the high-dimensional space. The linear projection deteriorates performances of the additive model. ADD-BO assumes independence between axis-aligned projections of the high-dimensional space, while the linear mapping $\mathbf{R}$ couples all subsets of dimensions. This renders the optimization of independent additive components not effective. The VAE-BO approach requires much larger amounts of data to learn a meaningful reconstruction mapping than what is allowed in our BO experiment. As a result most locations in feature space are mapped to similar reconstructions. This explains the flat curve observed on all VAE-BO progressions with different acquisitions.

4.2 Nonlinear feature space with non-additive objective

We consider the product of sines functions and apply a nonlinear dimensionality reduction. We define a single-layer neural network mapping to elevate the dimensionality of the objective to $D = 60$, i.e. $f_X(\mathbf{x}) = f(\sigma(\mathbf{R}\mathbf{x}))$. Here σ is the sigmoid activation function. We also compare with a different parametrization of the covariance function of the decoder $\mathbf{g}$. The baseline DMGP-BO and DMGPC-BO define a single kernel k_c for the reconstruction task while NN-BO and NNC-BO define different kernels $\{k_c^i\}_{i=1}^{Q}$, one for each subset of the partitioning. Figure 4 shows the progression of the regret over 300 BO iterations. We can observe consistent improvements of MGPC-BO and DMGPC-BO with respect to VAE-BO which also assumes a nonlinear embedding for the objective. The Performance of MGPC-BO and DMGPC-BO also retain better regret at termination with box-constrained acquisition maximization, namely MGP-BO and DMGP-BO.

Overall, we observe that the constrained maximization of the acquisition function is beneficial for the proposed model and variants of its reconstruction mapping, i.e. NN-BO, DMGP-BO. The advantages are more more evident with the product of sines objective while with the Rosenbrock we retain no worse regret. We also highlight that our performances improve as we move to problems which are characterized by intrinsic low-dimensionality characterized by a nonlinear dimensionality reduction.

5 Conclusion

We proposed a framework for efficient Bayesian optimization of intrinsically low-dimensional black-box functions based on nonlinear embeddings. In our model, the manifold GP learns useful low-dimensional feature representations of high-dimensional data by jointly learning the response surface and the reconstruction mapping. As a reconstruction mapping we use a manifold MOGP. Our approach allows for optimizing acquisition functions in a low-dimensional feature space. However, since exploration in feature space (driven by the acquisition function) does not necessarily mean exploration in the high-dimensional parameter space, we introduce a nonlinear constraint based on Lipschitz continuity of predicitons of the mMOGP, which encourages exploration in the vicinity of the training data and eliminates un-identifiability issues in data space, which would otherwise hinder optimization.

References

[1] G. Abbati, A. Tosi, M. A. Osborne, and S. Flaxman. Adageo: adaptive geometric learning for optimization and sampling. *International Conference on Artificial Intelligence and Statistics*, 2018.

[2] M. A. Alvarez and N. D. Lawrence. Sparse convolved Gaussian processes for multi-output regression. *Advances in Neural Information Processing Systems*, 2009.

[3] M. A. Alvarez and N. D. Lawrence. Computationally efficient convolved multiple output Gaussian processes. *Journal of Machine Learning Research*, 12:1459—1500, 2011.

[4] M. A. Alvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: a review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2011.

[5] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kegl. Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, 2011.

[6] P. Boyle and M. Frean. Dependent Gaussian processes. *Advances in Neural Information Processing Systems*, 2005.

[7] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[8] R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth. Manifold Gaussian processes for regression. *International Joint Conference on Neural Networks*, 2016.

[9] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76:5–23, 2016.

[10] Z. Dai, A. Damianou, J. González, and N. D. Lawrence. Variational auto-encoded deep Gaussian processes. *International Conference on Learning Representations*, 2016.

[11] A. Damianou. *Deep Gaussian processes and variational propagation of uncertainty*. PhD dissertation, 2015.

[12] A. Damianou and N. D. Lawrence. Deep Gaussian processes. *Artificial Intelligence and Statistics*, 2013.

[13] P. I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[14] R. Garnett, M. A. Osborne, and P. Hennig. Active learning of linear embeddings for Gaussian processes. *Conference on Uncertainty in Artificial Intelligence*, 2014.

[15] E. Gilboa, Y. Saatçi, and J. P. Cunningham. Scaling multidimensional inference for structured Gaussian processes. *Institute of Electrical and Electronics Engineers*, 37(2):424–436, 2015.

[16] R. Gomez-Bombarelli, N. W. Jennifer, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4:268–276, 2018.

[17] J. Gonzalez, J. Longworth, D. C. James, and N. D. Lawrence. Bayesian optimization for synthetic gene design. *arXiv preprint arXiv:1505.01627*, 2015.

[18] J. González, Z. Dai, P. Hennig, and N. Lawrence. Batch Bayesian optimization via local penalization. *International Conference on Artificial Intelligence and Statistics*, 2016.

[19] P. Goovaerts. *Geostatistics for natural resources evaluation*. Oxford University Press, 1997.

[20] R. R. Griffiths and J. M. Hernández-Lobato. Constrained Bayesian optimization for automatic chemical design. *arXiv preprint arXiv:1709.05501*, 2017.

[21] A. Hebbal, L. Brevault, M. Balesdent, E.G. Talbi, and N. Melab. Bayesian optimization using deep Gaussian processes. *arXiv preprint arXiv:1905.03350*, 2019.

[22] J. Hensman and N. D. Lawrence. Nested variational compression in deep Gaussian processes. *arXiv preprint arXiv:1412.1370*, 2014.

[23] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. *International conference on Learning and Intelligent Optimization*, 2011.

[24] I. Jolliffe. *Principal component analysis*. Springer, 2011.

[25] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[26] K. Kandasamy, J. Schneider, and B. Poczos. High dimensional Bayesian optimisation and bandits via additive models. *International Conference on Machine Learning*, 2015.

[27] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations*, 2014.

[28] H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

[29] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato. Grammar variational autoencoder. *International Conference on Machine Learning*, 2017.

[30] N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in Neural Information Processing Systems*, 2004.

[31] N. D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of machine learning research*, 2005.

[32] N. D. Lawrence and J. Quiñonero-Candela. Local distance preservation in the gp-lvm through back constraints. *International Conference on Machine Learning*, 2006.

[33] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 2015.

[34] P. Lu, J. Nocedal, C. Zhu, and R. H. Byrd. A limited-memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1994.

[35] X. Lu, J. Gonzalez, Z. Dai, and N. Lawrence. Structured variationally auto-encoded optimization. *International Conference on Machine Learning*, 2018.

[36] D. J. C. MacKay. Introduction to Gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, pages 133–166, 1998.

[37] A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.

[38] J. Močkus. On Bayesian methods for seeking the extremum. *Optimization Techniques IFIP Technical Conference*, 1975.

[39] M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. *International Conference on Information Processing in Sensor Networks*, 2008.

[40] K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[41] S. Rana, C. Li, S. Gupta, V. Nguyen, and S. Venkatesh. High dimensional Bayesian optimization with elastic Gaussian process. *International Conference on Machine Learning*, 2017.

[42] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. The MIT Press, 2006.

[43] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and variational inference in deep latent Gaussian models. *International Conference on Machine Learning*, 2014.

[44] K. F. Riley, M. P. Hobson, and S. J. Bence. *Mathematical methods for physics and engineering*. Cambridge University Press, 1999.

[45] Y. Saatçi. *Scalable inference for structured Gaussian process models*. PhD dissertation, 2012.

[46] H. Salimbeni and M. P. Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. *Advances in Neural Information Processing Systems*, 2017.

[47] M. Seeger, Y. W. Teh, and M. Jordan. Semiparametric latent factor models. *Technical Report*, 2005.

[48] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: a review of Bayesian optimization. *Institute of Electrical and Electronics Engineers*, 104:148–175, 2016.

[49] N. Srinivas, A. Krause, S.M. Kakade, and M. W. Seeger. Gaussian process bandits without regret: an experimental design approach. *International Conference on Machine Learning*, 2010.

[50] Y. Sui, Al. Gotovos, J. Burdick, and A. Krause. Safe exploration for optimization with Gaussian processes. *International Conference on Machine Learning*, 2015.

[51] M. Titsias and N. D. Lawrence. Bayesian Gaussian process latent variable model. *International Conference on Artificial Intelligence and Statistics*, 2010.

[52] H. Wackernagel. *Multivariate geostatistics: an introduction with applications*. Springer, 2013.

[53] N. Wahlström, T. B. Schön, and M. P. Deisenroth. From pixels to torques: Policy learning with deep dynamical models. *arXiv preprint arXiv:1502.02251*, 2015.

[54] Z. Wang, M. Zoghi, F. Hutter, D. Matheson, and N. De Freitas. Bayesian optimization in high dimensions via random embeddings. *International Joint Conference on Artificial Intelligence*, 2013.

[55] C. Williams, S. Klanke, S. Vijayakumar, and K. M. Chai. Multi-task Gaussian process learning of robot inverse dynamics. *Advances in Neural Information Processing Systems*, 2009.

[56] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. *International Conference on Artificial Intelligence and Statistics*, 2016.

[57] A. G. Wilson, D. A. Knowles, and Z. Ghahramani. Gaussian process regression networks. *International Conference on Machine Learning*, 2012.

[58] B. M. Yu, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani. Gaussian process factor analysis for low-dimensional single-trial analysis of neural population activity. *Advances in Neural Information Processing Systems*, 2009.

[59] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, 1997.