A Framework for Bayesian Optimization in Embedded Subspaces

Alexander Munteanu [1] [*] **Amin Nayebi** [2] [*] **Matthias Poloczek** [3] [2]

Abstract

We present a theoretically founded approach for high-dimensional Bayesian optimization based on low-dimensional subspace embeddings. We prove that the error in the Gaussian process model is bounded tightly when going from the original high-dimensional search domain to the low-dimensional embedding. This implies that the optimization process in the low-dimensional embedding proceeds essentially as if it were run directly on an unknown active subspace of low dimensionality. The argument applies to a large class of algorithms and GP models, including non-stationary kernels. Moreover, we provide an efficient implementation based on hashing and demonstrate empirically that this subspace embedding achieves considerably better results than the previously proposed methods for high-dimensional BO based on Gaussian matrix projections and structure-learning.

1. Introduction

Bayesian optimization (BO) has recently emerged as powerful technique for the global optimization of expensive-to-evaluate black-box functions (Brochu et al., 2010; Shahriari et al., 2016; Frazier, 2018). Here 'black-box' means that we may evaluate the objective at any point to observe its value, possibly with noise but without derivative information. The advantages of Bayesian optimization are sample-efficiency, convergence to a global optimum, and a low computational overhead. A critical limitation is the number of parameters that BO can optimize. The limit is usually seen around 15 parameters for the most common form of BO that uses Gaussian process (GP) regression as a surrogate model for the objective function. Thus, it is not surprising that expanding BO to higher-dimensional search spaces is widely acknowledged as one of the most important goals in the field. For example, Frazier (2018, p. 16) states that *"developing Bayesian optimization methods that work well in high dimensions is of great practical and theoretical interest"*.

This paper advances the field in the theory of high-dimensional Bayesian optimization and improves the practical performance. Specifically, the contributions are:

1. A theoretically founded framework for Bayesian optimization based on subspace embeddings. The core is a rigorous proof that any GP-based BO algorithm proceeds on the embedded space as it would if it was run directly on an unknown active subspace of low dimensionality.

2. An extension of this result to large classes of parametrized GP models. Note that the argument is agnostic to the acquisition criterion used by the BO algorithm and thus can be easily combined with many state-of-the-art methods, including batch acquisition, multifidelity modeling, network architecture search, and many more.

3. The Hashing-enhanced Subspace BO (HeSBO) method for high-dimensional problems has i) strong accuracy guarantees and ii) low computational overhead, while being iii) conceptually simple. In particular, it avoids complex corrections of projections that caused complications previously.

4. An experimental evaluation that demonstrates state-of-the-art performance when the proposed embedding is combined with a low-dimensional BO algorithm, e.g., Knowledge Gradient (KG) (Frazier et al., 2009) or BLOSSOM (BM) (McLeod et al., 2018).

Related Work. Bayesian optimization has recently received significant interest for the optimization of expensive-to-evaluate black-box functions. We refer to (Brochu et al., 2010; Shahriari et al., 2016; Frazier, 2018) for an overview. Aside of the general setting of optimizing an objective black-box function, several extensions have been studied, including constrained (Gardner et al., 2014; Hernández-Lobato et al., 2015; Picheny et al., 2016) and multifidelity optimization (Kennedy & O'Hagan, 2000; Huang et al., 2006; Swersky et al., 2013; Kandasamy et al., 2016; Poloczek et al., 2017), expensive functions with derivative information (Wu et al., 2017; Eriksson et al., 2018), and neural

[*] Equal contribution [1] Dortmund Data Science Center, Faculties of Statistics and Computer Science, TU Dortmund, Dortmund, Germany [2] Department of Systems and Industrial Engineering, University of Arizona, Tucson, AZ, USA [3] Uber AI Labs, San Francisco, CA, USA. Correspondence to: Matthias Poloczek <poloczek@uber.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

network architecture search (Klein et al., 2017; Kandasamy et al., 2018).

Kandasamy et al. (2015) extended Bayesian optimization to higher-dimensional search spaces composed of disjoint low-dimensional subspaces that can be optimized over separately if the latent additive structure is learned. The ADD algorithm of Gardner et al. (2017) and SKL of Wang et al. (2017) use sophisticated sampling procedures to learn the decomposition, and Rolland et al. (2018) and Mutny & Krause (2018) extended the idea to overlapping subspaces. Inferring the latent structure of the search space from data via extensive sampling introduces a considerable computational load in practice, which limits the applicability to objective functions with high evaluation cost (see Sect. 3). This bottleneck was recently addressed by Wang et al. (2018) whose ensemble BO method (EBO) uses an ensemble of additive GP models for scalability. We evaluated ADD, EBO and SKL from this line of research and found EBO indeed more scalable than SKL, but still considerably more expensive than the other approaches that we evaluated.

Chen et al. (2012); Wang et al. (2016b) bypassed scalability issues by leveraging the observation that for many applications only a small number of tunable parameters have a significant effect on the objective function. This phenomenon is known as “active subspace” and “effective dimensionality”. Wang et al. (2016b) used a projection matrix with standard Gaussian entries and showed that the active subspace is rank-preserved but strongly dilated. Thus, they expanded the low-dimensional search space to ensure that its projection to the high-dimensional space contains an optimal solution with reasonable probability. Their REMBO algorithm fits a GP model to the low-dimensional space and projects points back to the high-dimensional space using the inverse random projection. Points whose high-dimensional image is outside the search domain $\mathcal{X}$ are convex-projected to the boundary $\partial\mathcal{X}$, which may lead to over-exploration of the boundary regions. Binois et al. (2015) suggested a new warping kernel k_ψ to address this issue by an improved preservation of distances among such points. Recently, Binois et al. (2019) proposed a novel choice of the low-dimensional search space to handle the distortion of Gaussian projections. Djolonga et al. (2013) and Eriksson et al. (2018) propose learning the active subspace, in the latter paper from derivative information, and therefore avoid the random projection. We compare to the methods in (Wang et al., 2016b; Binois et al., 2015) in Sect. 3. We also compare to the BOCK algorithm of Oh et al. (2018) that is based on a cylindrical transformation. This provides scalability to high dimensional problems and an implicit prior that the optimizer is in the interior of the search space.

Outline. Sect. 2 introduces probabilistic subspace embeddings and the HeSBO method, followed by the theoretical foundation. The experimental evaluation is given in Sect. 3 and the discussion is in Sect. 4. Sections denoted by letters are found in the Supplement.

2. Probabilistic Subspace Embeddings of Gaussian processes

This section introduces probabilistic subspace embeddings and their use in BO. Sect. 2.2 motivates the approach. The theoretical foundation is given in Sect. 2.3 and generalized to large classes of popular kernels in Sect. 2.4.

2.1. The HeSBO Algorithm

Given an objective function f defined on the high-dimensional domain $\mathcal{X} = [-1, +1]^D$, we suppose that there exists an *unknown* d_e-dimensional active subspace $\mathcal{Z}$. Informally, the objective f is invariant to coordinate changes outside $\mathcal{Z}$. Note that we do not suppose that $\mathcal{Z}$ is axis-aligned with $\mathcal{X}$. Our approach is to choose a d-dimensional subspace $\mathcal{Y}$ of $\mathcal{X}$ randomly such that the optimization process proceeds on $\mathcal{Y}$ essentially as it would on $\mathcal{Z}$ with good probability. This enables to run a BO method on $\mathcal{Y}$ in order to find an $x^* \in \arg\max_{x\in\mathcal{X}} f(x)$. The proposed embedding can easily be combined with many GP-based BO methods: Algorithm 1 shows the generic BO algorithm (Shahriari et al., 2016; Frazier, 2018) with the embedding incorporated via the *highlighted steps*. We call this *Hashing-enhanced Subspace Bayesian Optimization* (HeSBO).

The embedding is constructed in Algorithm 2 that initializes two hash functions which implicitly represent the embedding matrix S' (see Sect. 2.2 for details): The hash function h chooses one single non-zero entry for each of the D dimensions, and the function σ determines the sign of the corresponding non-zero entry. Algorithm 3 maps the low-dimensional vector $y \in \mathcal{Y}$ to the high-dimensional domain $\mathcal{X}$. The algorithm iterates over the high-dimensional coordinates: for each entry it invokes the two above hash functions to determine the associated coordinate in the low-dimensional vector y and the sign that the corresponding value of y is multiplied with. Note that by construction the columns of the matrix are orthogonal. Moreover, for any fixed coordinate (dimension) d_i of the low-dimensional space we have that the number of coordinates in the high dimensional space that map to d_i is concentrated at D/d, i.e., this number has mean D/d and a variance that drops (quickly) as D increases. The reason is that h picks the target of each dimension uniformly at random. The dilation is thus roughly $\sqrt{D/d}$, and the search domain $\mathcal{Y} = [-1, 1]^d$ is mapped to $S'\mathcal{Y} = \{S'y \mid y \in \mathcal{Y}\} \subset \mathcal{X} = [-1, 1]^D$ with low distortion. We contrast this with REMBO’s search space that was chosen as $[-\sqrt{d}, +\sqrt{d}]^d$ to cope with dilations; see Sect. B. It is critical to note that the operations of copying the coordinates and multiplying them by $\{-1, 1\}$ assert that no point is projected outside $\mathcal{X}$. Thus, HeSBO avoids complex corrections in contrast to the REMBO variants. The analysis in Sect. C in the supplement shows that REMBO applies these corrections frequently.

Algorithm 1 The Generic BO Algorithm with the probabilistic subspace embedding

```
Input: Objective $f : \mathcal{X} \to \mathbb{R}$; acquisition criterion $\alpha$ (e.g., EI); target dimension $d \in \mathbb{N}$.
Output: An optimizer $x^* \in \arg\max_{x \in \mathcal{X}} f(x)$.
Construct embedding $S' : \mathcal{Y} \to \mathcal{X}$ with Algorithm 2.
Sample initial points $Y_0 \in \mathcal{Y}$ using a space filling design and let $D_0 = (Y_0, \{\text{observation for } f(S'y) \mid y \in Y_0\})$.
Estimate the hyperparameters $\theta_0$ for the GP prior on $\mathcal{Y}$ given $D_0$. Then calculate the posterior conditioned on $\theta_0$ and $D_0$.
for $n = 1$ to $N$ do
    Compute $y^{n+1} \in \arg\max_{y \in \mathcal{Y}} \alpha(y, D_n)$.
    Evaluate $f(S'y^{n+1})$, for the projection of $y^{n+1}$ to $\mathcal{X}$ via Algorithm 3. Let $z^{n+1}$ be the (noisy) observation.
    Update the posterior distribution with the new observation $D_{n+1} = D_n \cup (y^{n+1}, z^{n+1})$.
end for
Return $x^* \in \arg\max_{x \in \mathcal{X}} f(x)$ if observations are noise-free, or a point with maximum expected value under the posterior given $D_N$ otherwise.
```

Algorithm 2 Construction of an inverse subspace embedding S' implicitly given by (h, σ), see details in the text.

```
Input: Input dimension $D$, target dimension $d$.
Output: Implicit representation of a linear map $S' : \mathbb{R}^d \to \mathbb{R}^D$, by two hash functions $h$ and $\sigma$.
Initialize a pairwise independent and uniform hash function $h : [D] \to [d]$.
Initialize a 4-wise independent and uniform hash function $\sigma : [D] \to \{-1, 1\}$.
Return $(h, \sigma)$, which implicitly defines $S'$, see detailed description in Sect. 2.2.
```

Algorithm 3 Computes $S'y \in \mathcal{X}$ via the inverse subspace embedding S' implicitly given by (h, σ)

```
Input: Low-dimensional vector $y \in \mathcal{Y} \subset \mathbb{R}^d$.
Output: High-dimensional vector $x = S'y \in \mathcal{X} \subset \mathbb{R}^D$.
for $i = 1$ to $D$ do
    $x_i = \sigma(i) \cdot y_{h(i)}$
end for
Return $x$.
```

2.2. Motivation for the Embedding

Next we provide the motivation and background for the proposed embedding, before we prove its theoretical properties in Sect. 2.3. *Random projections* have often been used to reduce the dependence on the dimension of an algorithm's running time, memory, or communication cost. The seminal result of Johnson & Lindenstrauss (1984), JL for short, states that any set of n points in a D-dimensional space can be embedded into $d \in O(\log n / \varepsilon^2)$ dimensions such that all pairwise ℓ_2 distances are preserved up to a $(1 \pm \varepsilon)$-factor. This can be achieved probabilistically by a linear map where each entry of the embedding $G \in \mathbb{R}^{d \times D}$ is an i.i.d. rescaled standard Gaussian. Much effort was put into making the construction computationally more efficient and sparse (Achlioptas, 2003; Ailon & Chazelle, 2009; Ailon & Liberty, 2009; Kane & Nelson, 2014). Sarlós (2006) was the first who generalized such embeddings to entire linear subspaces with target dimension roughly $O(d_e/\varepsilon^2)$ which is optimal, see (Nelson & Nguyên, 2014). Charikar et al. (2004) introduced an alternative projection, called *count-sketch*, based on hashing to identify *heavy hitters* in a large vector, often a data stream, based on a summary stored in a low-dimensional vector. The count-sketch can be formalized as a sparse matrix with only one non-zero entry per column that is drawn randomly in $\{-1, 1\}$. We give an example of such a map reducing from five to three dimensions:

$$\begin{pmatrix} 0 & 0 & 1 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} x_3 - x_4 \\ -x_1 \\ x_2 + x_5 \end{pmatrix}.$$

This is equivalent to uniformly hashing each entry x_i to a random coordinate $j = h(i) \in [d]$ and multiplying it by a random sign $\sigma_i \in \{-1, 1\}$ for all $i \in [D]$. The low-dimensional summary $y = Sx$ is given by $\forall j \in [d] \colon y_j = \sum_{i \colon h(i) = j} \sigma(i) x_i$. Charikar et al. (2004) showed that $\tilde{x}_i = \sigma(i) y_{h(i)}$ is a good estimate for x_i. We recover this idea for the projection from the low-dimensional to the high-dimensional domain, $x \approx S'y$, in Algorithms 2 and 3. This type of embedding is not capable of preserving pairwise distances for arbitrary sets of points, as JL does. However, Clarkson & Woodruff (2013) showed that it yields a subspace embedding for an entire linear subspace with constant probability. Informally, they used the fact that a d_e-dimensional (active) subspace can have only $O(d_e)$ coordinates that are large (have high leverage) and thus important. These are exactly the *heavy hitters* which can be estimated reasonably well. The other low values mostly cancel in the sum with random signs (up to a small error). Nelson & Nguyen (2013) improved and simplified their analysis and complemented with a tight lower bound, settling the target dimension of subspace embeddings via the *count-sketch* to $\Theta(d_e^2/(\varepsilon^2\delta))$, where δ is the failure probability. We define

ε-subspace embeddings more formally.

Definition 1 (ε-subspace embedding, cf. Sarlós (2006); Woodruff (2014))**.** *Given a matrix $V \in \mathbb{R}^{D\times d_e}$ with orthonormal columns, an integer $d \leq D$ and an approximation parameter $\varepsilon \in (0, \frac{1}{2})$, an ε-subspace embedding for V is a map $S\colon \mathbb{R}^D \to \mathbb{R}^d$ such that $\forall x \in \mathbb{R}^d$:*

$$(1-\varepsilon)\left\|Vx\right\|_2^2 \leq \left\|SVx\right\|_2^2 \leq (1+\varepsilon)\left\|Vx\right\|_2^2, \quad (1)$$

or, equivalently (cf. Paul et al. (2014); Cohen et al. (2016)) $\left\|V'S'SV - I_d\right\|_2 \leq \varepsilon.$ (2)

Note that (1) is closer to the original JL property preserving Euclidean norms and thus distances, but generalized to the entire subspace spanned by V, while (2) is often analytically more convenient and intuitively states that the isometry property $V'V = I_d$ is approximately preserved under the random projection. Note that the definition directly implies that the singular values of an embedded subspace are preserved up to $(1 \pm \varepsilon)$ multiplicative distortion, which in particular means that its rank is preserved.

2.3. Theoretical Foundation for the Embedding

Assuming the existence of a low-dimensional active subspace, Wang et al. (2016b) showed that preserving the rank and controlling the dilation ensures that an optimal point is contained in a low-dimensional projection of this subspace. We improve on their analysis by leveraging the above definition. Our main theoretical contribution is to show that any ε-subspace embedding preserves the mean and variance functions of a Gaussian process with linear kernel, i.e., the standard inner product in Euclidean space. In Sect. 2.4 we further extend this result to other important classes of kernel functions like polynomial kernels, (squared) exponential and Matérn kernels. Note that the guarantee is independent of how the subspace embedding is achieved algorithmically.

Priors based on Gaussian process regression are popular in BO. Let the error distribution be normal with mean 0 and positive semidefinite kernel $K(x, y)$. Then the predictive distribution, given data matrix $X = (x_1, \ldots, x_l)$ and the vector of their function evaluations $f = (f(x_1), \ldots, f(x_l))$, is also normal with mean and variance functions

$$\mu(x) = k'(x)K^- f, \quad (2)$$
$$\sigma^2(x) = K(x, x) - k'(x)K^- k(x), \quad (3)$$

where $x \in \mathcal{X}$ and $k(x)$ denotes the vector given by $(K(x, x_1) \ldots K(x, x_l))'$, and the kernel matrix is defined by $K_{i,j} = K(x_i, x_j), i, j \in [l]$. By K^- we denote the Moore-Penrose pseudoinverse which coincides with the standard inverse but generalizes to non-invertible cases. For the sake of illustration, assume for the moment that we have the simple linear kernel $K(x, y) = x'y$, i.e., the standard inner product. Then Equations (2) and (3) simplify to

$$\mu(x) = x'X(X'X)^- f, \quad (4)$$
$$\sigma^2(x) = x'x - x'X(X'X)^- X'x. \quad (5)$$

Now, to bound the effect of an ε-subspace embedding S under the assumption of low effective dimensionality, consider V, an orthonormal basis for the active subspace, and $V^\perp$, an orthonormal basis for its nullspace. We can replace any point $x = Vy + V^\perp z$ by $x^\top = Vy$, its projection to the subspace, since $f(x) = f(Vy + V^\perp z) = f(Vy) = f(x^\top)$. We stress that V and $V^\perp$ can be arbitrary bases for those subspaces. While Definition 1 and our results do not depend on the representative, it will be convenient to choose V derived from a singular value decomposition (SVD). Our main result states that the Gaussian process in this subspace is simulated very accurately conditioned on the (probabilistic) event that we have an ε-subspace embedding.

Theorem 2. *Consider a Gaussian process that acts directly in the unknown active subspace of dimension d_e with mean and variance functions $\mu(\cdot), \sigma^2(\cdot)$. Let $\tilde{\mu}(\cdot), \tilde{\sigma}^2(\cdot)$ be their approximations using an ε-subspace embedding for the active subspace. Then we have for every $x \in \mathcal{X}$*
1. $|\mu(x) - \tilde{\mu}(x)| \leq 5\varepsilon \left\|x\right\| \left\|X^- f\right\|$
2. $\left|\sigma^2(x) - \tilde{\sigma}^2(x)\right| \leq 12\varepsilon \left\|x\right\|^2$.

Proof. We can express $x = Vy$ and by the SVD we have $X = V\Sigma U'$. Then for the mean function and its approximation we have by the triangle inequality

$$\begin{aligned}
&|\mu(x) - \tilde{\mu}(x)| \\
&= \left|x'X(X'X)^- f - x'S'SX(X'S'SX)^- f\right| \\
&\leq \left|x'X(X'X)^- f - x'S'SX(X'X)^- f\right| \quad (6)\\
&+ \left|x'S'SX(X'X)^- f - x'S'SX(X'S'SX)^- f\right|. \quad (7)
\end{aligned}$$

We bound both terms separately

$$\begin{aligned}
(6) &\leq \Big|y'V'V\Sigma U'(U\Sigma V'V\Sigma U')^- f \\
&\qquad - y'V'S'SV\Sigma U'(U\Sigma V'V\Sigma U')^- f\Big| \\
&\leq \left\|y\right\| \left\|V'S'SV - I\right\| \left\|\Sigma U'(U\Sigma V'V\Sigma U')^- f\right\| \\
&\leq \varepsilon \left\|y\right\| \left\|V\Sigma U'U(\Sigma^2)^- U'f\right\| = \varepsilon \left\|y\right\| \left\|V\Sigma^- U'f\right\| \\
&= \varepsilon \left\|y\right\| \left\|X^- f\right\|
\end{aligned}$$

and for the second summand (7),

$$\begin{aligned}
(7) &\leq \Big|y'V'S'SV\Sigma U'(U\Sigma V'V\Sigma U')^- f \\
&\qquad - y'V'S'SV\Sigma U'(U\Sigma V'S'SV\Sigma U')^- f\Big| \\
&\leq \Big|y'V'S'SV\Sigma U'U\Sigma^- V'V\Sigma^- U'f \\
&\qquad - y'V'S'SV\Sigma U'U\Sigma^-(V'S'SV)^-\Sigma^- U'f\Big| \\
&\leq \left\|y'V'S'SV\right\| \left\|(V'S'SV)^- - V'V\right\| \left\|\Sigma^- U'f\right\| \\
&\leq (1+\varepsilon)2\varepsilon \left\|y\right\| \left\|X^- f\right\|
\end{aligned}$$

The last inequality follows from the following two observations. First, the inverse singular values are approximated to within a factor $(1 \pm 2\varepsilon)$ by Observation 1 in (Geppert et al., 2017). Thus $\|(V'S'SV)^{-} - V'V\| \leq 2\varepsilon$. Second, we have

$$\begin{aligned} \|y'V'S'SV\| &\leq \|y'V'S'SV - y'V'V\| + \|y'\| \\ &\leq \|y'\| \, \|V'S'SV - I\| + \|y'\| \leq (1+\varepsilon)\,\|y\| \end{aligned}$$

Summing up, the triangle inequality implies

$$|\mu(x) - \tilde{\mu}(x)| \leq (6) + (7) \leq 5\varepsilon \,\|y\| \, \left\|X^{-} f\right\|.$$

Similarly, we can show for the variance function that

$$\begin{aligned} &\left|\sigma^2(x) - \tilde{\sigma}^2(x)\right| \\ &= \left|x'x - x'X(X'X)^{-}X'x \right. \\ &\qquad\qquad \left. - x'S'Sx + x'S'SX(X'S'SX)^{-}X'S'Sx\right| \\ &\leq |x'x - x'S'Sx| \\ &+ \left|x'X(X'X)^{-}X'x \right. \\ &\qquad\qquad \left. - x'S'SX(X'S'SX)^{-}X'S'Sx\right| \\ &\leq |x'x - x'S'Sx| && (8) \\ &+ \left|x'X(X'X)^{-}X'x - x'S'SX(X'X)^{-}X'x\right| && (9) \\ &+ \left|x'S'SX(X'X)^{-}X'x \right. \\ &\qquad\qquad \left. - x'S'SX(X'X)^{-}X'S'Sx\right| && (10) \\ &+ \left|x'S'SX(X'X)^{-}X'S'Sx \right. \\ &\qquad\qquad \left. - x'S'SX(X'S'SX)^{-}X'S'Sx\right| && (11) \end{aligned}$$

Again we bound items (8)-(11) separately. We have

$$\begin{aligned} (8) &\leq |y'V'Vy - y'V'S'SVy| \\ &\leq \|y\|^2 \, \|V'S'SV - I\| \leq \varepsilon \,\|y\|^2, \end{aligned}$$

and

$$\begin{aligned} (9) &\leq \left|y'V'V\Sigma U'(X'X)^{-}X'x \right. \\ &\qquad\qquad \left. - y'V'S'SV\Sigma U'(X'X)^{-}X'x\right| \\ &\leq \|y\| \, \|V'S'SV - I\| \, \left\|\Sigma U'(X'X)^{-}X'x\right\| \\ &\leq \varepsilon \,\|y\| \, \left\|\Sigma U'U\Sigma^{-}V'V\Sigma^{-}U'U\Sigma V'Vy\right\| \\ &\leq \varepsilon \,\|y\| \, \underbrace{\left\|\Sigma U'U\Sigma^{-}V'V\Sigma^{-}U'U\Sigma\right\|}_{=1} \|y\| \\ &= \varepsilon \,\|y\|^2, \end{aligned}$$

and

$$\begin{aligned} (10) &\leq \left|x'S'SX(X'X)^{-}U\Sigma V'Vy \right. \\ &\qquad\qquad \left. - x'S'SX(X'X)^{-}U\Sigma V'S'SVy\right| \\ &\leq \left\|x'S'SX(X'X)^{-}U\Sigma\right\| \, \|V'S'SV - I\| \, \|y\| \\ &\leq \varepsilon \,\|y\| \, \left\|y'V'S'SV\Sigma U'(X'X)^{-}U\Sigma\right\| \\ &\leq \varepsilon \,\|y\| \left(\left\|y'(V'S'SV - I)\Sigma U'(X'X)^{-}U\Sigma\right\| \right. \\ &\qquad\qquad \left. + \left\|y'V'V\Sigma U'(X'X)^{-}U\Sigma\right\|\right) \\ &\leq \varepsilon \,\|y\| \left(\|y\| \, \|V'S'SV - I\| \, \left\|\Sigma U'(X'X)^{-}U\Sigma\right\| \right. \\ &\qquad\qquad \left. + \|y\| \, \left\|\Sigma U'(X'X)^{-}U\Sigma\right\|\right) \\ &\leq \varepsilon(1+\varepsilon)\,\|y\|^2 \underbrace{\left\|\Sigma U'U\Sigma^{-}V'V\Sigma^{-}U'U\Sigma\right\|}_{=1} \\ &\leq 2\varepsilon \,\|y\|^2. \end{aligned}$$

For the last summand, observe

$$\begin{aligned} (11) &\leq \left|x'S'SXU\Sigma^{-}V'V\Sigma^{-}U'X'S'Sx \right. \\ &\qquad\qquad \left. - x'S'SXU\Sigma^{-}(V'S'SV)^{-}\Sigma^{-}U'X'S'Sx\right| \\ &\leq \left\|(V'S'SV)^{-} - I\right\| \, \left\|x'S'SXU\Sigma^{-}\right\|^2 \\ &\leq 2\varepsilon \left\|y'V'S'SV\Sigma U'U\Sigma^{-}\right\|^2 \\ &\leq 2\varepsilon \left(\left\|y'(V'S'SV - I)\Sigma U'U\Sigma^{-}\right\| \right. \\ &\qquad\qquad \left. + \left\|y'V'V\Sigma U'U\Sigma^{-}\right\|\right)^2 \\ &\leq 2\varepsilon \left(\|y\| \, \|V'S'SV - I\| \, \left\|\Sigma U'U\Sigma^{-}\right\| \right. \\ &\qquad\qquad \left. + \|y\| \, \left\|\Sigma U'U\Sigma^{-}\right\|\right)^2 \\ &\leq 2\varepsilon(1+\varepsilon)^2 \,\|y\|^2 \underbrace{\left\|\Sigma U'U\Sigma^{-}\right\|^2}_{=1} \\ &\leq 8\varepsilon \,\|y\|^2 \end{aligned}$$

The second part of Theorem 2 follows by triangle inequality

$$\begin{aligned} \left|\sigma^2(x) - \tilde{\sigma}^2(x)\right| &\leq (8) + (9) + (10) + (11) \\ &\leq 12\varepsilon \,\|y\|^2. \end{aligned}$$

□

2.4. Generalization to other Popular Kernels

Sarlós (2006) original argument to construct a subspace embedding relies on an additive approximation for inner products. The analysis is conducted on unit norm vectors and extends to the entire space by linearity. The assumption is thus not restrictive but simplifies the presentation. We show small error approximation guarantees for several popular classes of kernels. For the linear kernel $K(x,y) = x'y$, this is an immediate consequence of Definition 1 via the parallelogram rule (Arriaga & Vempala, 2006):

Corollary 3. *If S is an ε-subspace embedding for V then for all $x, y \in \{v \in \mathbb{R}^D \mid v = Vu, u \in R^d\}$ we have*

$$x'y - \varepsilon \,\|x\| \, \|y\| \leq x'S'Sy \leq x'y + \varepsilon \,\|x\| \, \|y\|.$$

We begin with the polynomial kernel $K(x,y) = (x'y + c)^p$.

Lemma 4. *Let $K(x,y) = (x'y + c)^p, p \in \mathbb{N}$ be the polynomial kernel with $c \geq 2$. Let S be an ε-subspace embedding for V. Then for all $x, y \in \{v \in \mathbb{R}^D \mid v = Vu, u \in R^d\}$ with $\|x\| = \|y\| = 1$ we have $|K(x,y) - K(Sx, Sy)| \leq \varepsilon p K(x,y)$.*

Proof. By Corollary 3 we have

$$\tilde{K}(x,y) = (x'S'Sy + c)^p$$

$$
\begin{aligned}
&\leq (x'y + \varepsilon \|x\| \|y\| + c)^p \\
&= (x'y + c)^p \left(1 + \varepsilon \frac{\|x\| \|y\|}{x'y + c}\right)^p
\end{aligned}
$$

and

$$
\begin{aligned}
\tilde{K}(x, y) &= (x'S'Sy + c)^p \\
&\geq (x'y - \varepsilon \|x\| \|y\| + c)^p \\
&= (x'y + c)^p \left(1 - \varepsilon \frac{\|x\| \|y\|}{x'y + c}\right)^p .
\end{aligned}
$$

Thus by Bernoulli's inequality

$$
\begin{aligned}
&\left|K(x, y) - \tilde{K}(x, y)\right| = |(x'y + c)^p - (x'S'Sy + c)^p| \\
&\leq \left|(x'y + c)^p \left(1 - \left(1 \pm \varepsilon \frac{\|x\| \|y\|}{x'y + c}\right)^p\right)\right| \\
&\leq \left|(x'y + c)^p \left(1 - 1 + \varepsilon p \frac{\|x\| \|y\|}{x'y + c}\right)\right| \\
&\leq (x'y + c)^{p-1} \varepsilon p \|x\| \|y\| \leq \varepsilon p K(x, y). \qquad \square
\end{aligned}
$$

Next we bound the approximation guarantee under ε-subspace embeddings for the (squared) exponential kernel $K(x, y) = \exp\left(-\frac{\|x-y\|^p}{l^p}\right)$ for $p \in \{1, 2\}$. To this end, we begin with a claim proven in the Sect..

Claim 5. *Let $z \in [0, 1]$. Fix $\delta \in (0, \frac{1}{2})$. Then we have $0 \leq z - z^{1+\delta} \leq \delta$ and $0 \leq z^{1-\delta} - z \leq \delta$.*

Lemma 6. *Let $K(x, y) = \exp\left(-\|x - y\|^p / l^p\right), p \in \{1, 2\}$ be the (squared) exponential kernel. Let S be an ε-subspace embedding for V. Then for all $x, y \in \{v \in \mathbb{R}^D \mid v = Vu, u \in R^d\}$ we have $|K(x, y) - K(Sx, Sy)| \leq \varepsilon$.*

Proof. S is an ε-subspace embedding. Thus for any x, y in the embedded subspace there exists some $\delta \in (-\frac{1}{2}, \frac{1}{2})$ with $|\delta| \leq \varepsilon$ such that

$$
\begin{aligned}
&\left|K(x, y) - \tilde{K}(x, y)\right| \\
&= \left|\exp\left(-\frac{\|x - y\|^2}{l^2}\right) - \exp\left(-\frac{\|Sx - Sy\|^2}{l^2}\right)\right| \\
&= \left|\exp\left(-\frac{\|x - y\|^2}{l^2}\right) - \exp\left(-(1 + \delta)\frac{\|x - y\|^2}{l^2}\right)\right| \\
&= \left|K(x, y) - K(x, y)^{1+\delta}\right| \leq |\delta| \leq \varepsilon,
\end{aligned}
$$

where the last line holds by Claim 5 since $K(x, y) = \exp(-\|x - y\|^2 / l^2) \in [0, 1]$.

Note that the approximation guarantee of Definition 1 implies that the (non-squared) Euclidean norms are approximated to within $\sqrt{1 + \varepsilon} \leq 1 + \varepsilon$ and $\sqrt{1 - \varepsilon} \geq 1 - \varepsilon$ respectively. The proof thus applies unchanged to the simple exponential kernel $K(x, y) = \exp(-\|x - y\| / l)$. $\square$

The Matérn kernel is defined as

$$
K(x, y) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{l} \|x - y\|\right)^\nu B_\nu \left(\frac{\sqrt{2\nu}}{l} \|x - y\|\right),
$$

where $\nu > 0$ is a parameter and B_ν is a second kind Bessel function. For parameters of the form $\nu = p + \frac{1}{2}, p \in \mathbb{N}$ the above characterization simplifies and can in particular be expressed as the product of a polynomial of degree p and an exponential function. To this end for x, y and length scale parameter l let $\rho = \sqrt{2\nu} \|x - y\| / l$. Then

$$
K(x, y) = K(\rho) = (1 + \operatorname{poly}(\rho)) \exp(-\rho),
$$

where $\operatorname{poly}(\rho) = \sum_{i=1}^p c_i \rho^i$ for coefficients $c_i > 0, i \in [p]$, depending on ν and p. The limiting cases are the simple exponential kernel for $p = 0, \nu = \frac{1}{2}$ and the squared exponential kernel for $\nu \to \infty$. The Matérn kernel converges quickly for larger values $p \geq 3, \nu \geq \frac{7}{2}$ and is nearly indistinguishable from the limiting case for finite and noisy data. We thus give two examples which are most interesting in machine learning (Rasmussen & Williams, 2006):

$$
\begin{aligned}
&\text{for } p = 1\text{: } K_{3/2}(\rho) = (1 + \rho) \exp(-\rho), \\
&\text{for } p = 2\text{: } K_{5/2}(\rho) = \left(1 + \rho + \rho^2/3\right) \exp(-\rho).
\end{aligned}
$$

Lemma 7. *Let $K(x, y)$ be the Matérn kernel with parameter $\nu = p + \frac{1}{2}, p \in \mathbb{N}$. Let S be an ε-subspace embedding for V. Then for all $x, y \in \{v \in \mathbb{R}^D \mid v = Vu, u \in R^d\}$ we have $|K(x, y) - K(Sx, Sy)| \leq 2\varepsilon$.*

Proof. $K(x, y)$ is normalized since it is bounded by the limiting squared exponential kernel $K(\rho) \leq \exp(-C\rho^2) \leq 1$ for some absolute constant $C > 0$, cf. (Rasmussen & Williams, 2006). S is an ε-subspace embedding. Thus for any x, y in the embedded subspace there exists some $\delta \in (-\frac{1}{2}, \frac{1}{2})$ with $|\delta| \leq \varepsilon$ such that $\|x - y\| = (1 + \delta) \|Sx - Sy\|$. Thus we have

$$
\begin{aligned}
&\left|K(\rho) - \tilde{K}(\rho)\right| \\
&= \left|\frac{1 + \operatorname{poly}(\rho)}{\exp(\rho)} - \frac{1 + \operatorname{poly}((1 + \delta)\rho)}{\exp((1 + \delta)\rho)}\right| \\
&\leq \left|\frac{1 + \operatorname{poly}(\rho)}{\exp(\rho)} - \left(\frac{1 + \operatorname{poly}(\rho)}{\exp(\rho)}\right)^{1+\delta}\right| \qquad (12) \\
&+ \left|\left(\frac{1 + \operatorname{poly}(\rho)}{\exp(\rho)}\right)^{1+\delta} - \frac{1 + \operatorname{poly}((1 + \delta)\rho)}{\exp((1 + \delta)\rho)}\right|. \qquad (13)
\end{aligned}
$$

Since $K(\rho) \in [0, 1]$ it follows again from Claim 5 that (12) $= \left|K(\rho) - K(\rho)^{1+\delta}\right| \leq |\delta| \leq \varepsilon$. In the second term the denominators are equal, since $(\exp(\rho))^{1+\delta} = \exp(\rho(1 + \delta))$, so we continue only with the enumerators of (13). The first term is always greater than the second for positive δ. Moreover note that $\operatorname{poly}((1 + \delta)\rho) \geq (1 + \delta) \operatorname{poly}(\rho)$ since in each term we have $(1+\delta)^i \geq (1+\delta)$. For negative δ this is reversed but then also $(1+\delta)^i \leq (1+\delta)$ is reversed. It can be shown similarly to Claim 5 that

$$\left|(1+\mathrm{poly}(\rho))^{1+\delta} - (1+(1+\delta)\,\mathrm{poly}(\rho))\right|$$
$$\leq |\delta| \exp((1+\delta)\rho) \leq \varepsilon \exp((1+\delta)\rho).$$

So (13) $\leq \varepsilon$ and thus $\left|K(\rho) - \tilde{K}(\rho)\right| \leq 2\varepsilon$. □

When generalizing Theorem 2 for the kernel functions above, we face two problems. First, since the implicit feature space may have infinite dimension it is unclear whether the singular value decomposition (SVD) exists. To see that it does, note that the rank of the kernel matrix might grow as large as the number of points. But it remains bounded and it was shown in (Bell, 2014) that any bounded rank linear operator has an SVD. Second, we have to argue that we have a subspace embedding restricted to the space spanned by the feature vectors. Recall that in the original proof of Sarlós (2006), it is sufficient to have an additive error guarantee for the inner product of unit vectors as shown above.

Corollary 8. *The claims of Theorem 2 hold when using the normalized kernel functions with ε replaced by the additive error bounds given in Lemmas 4 - 7.*

3. Numerical Results

We demonstrate the performance of the proposed embedding technique on a variety of functions, combining it with `KG` of `Cornell-MOE` (Wu & Frazier, 2016; Wu et al., 2017), `BM` (McLeod et al., 2018), and `EI` that was also used for `REMBO`, referred to as `HeSBO-KG`, `HeSBO-BM`, and `HeSBO-EI` respectively. The evaluation compares `HeSBO` to the state-of-the-art for high-dimensional BO (see Sect. 1): `REMBO`-$k_{\mathcal{Y}}$ that fits a GP model to $\mathcal{Y}$ using the distances in $\mathcal{Y}$, and `REMBO`-$k_{\mathcal{X}}$ that uses distances after projection to $\mathcal{X}$. This addresses the problem that `REMBO`-$k_{\mathcal{Y}}$ tends to over-explore the boundary of $\mathcal{X}$ since those points may appear distant when selected in $\mathcal{Y}$ (Wang et al., 2016b, cf. p. 371). `REMBO`-k_{ψ} (Binois et al., 2015) uses a more sophisticated warping to handle points mapped to the boundary of $\mathcal{X}$ better. All `REMBO` variants use the `EI` criterion. `ADD` (Gardner et al., 2017), `SKL` (Wang et al., 2017), and `EBO` (Wang et al., 2018) represent methods that learn the structure of the search space. They were omitted for 1000-dimensional test functions due to high computational cost. `BOCK` (Oh et al., 2018) uses a cylindrical transformation of the space to achieve an excellent performance on problems of higher dimension. We implemented `HeSBO-EI` and all `REMBO` variants in `Python 3` using `GPy`. The code for the embedding is available at github.com/aminnayebi/HesBO. For all other algorithms we used the authors' reference implementations (see the bibliography).

Experimental Setup. We report the the function values of the recommended solutions as a function of (1) the number of steps performed by the algorithm, or (2) of the wall-clock time. For the latter it is important to note that all experiments were performed on dedicated machines with identical resources. Error bars are shown at the median ± two standard errors of the median, averaged over at least 100 replications. See Sect. D for more details. When algorithms show large discrepancies in the performance, we only plot the best for clarity. See Sect. E for additional plots that show all algorithms.

Performance on Test Functions. We compared the algorithms on the following test functions: (1) Branin, (2) Hartmann-6, (3) Rosenbrock, and (4) Styblinski-Tang (StybTang) with input dimension $D \in \{25, 100, 1000\}$. The first three have a low-dimensional active subspace: 2 dimensions for Branin and Rosenbrock, 6 for Hartmann-6. StybTang is defined on all D dimensions. The proposed subspace embedding in combination with `KG` and `BM`, two state-of-the-art methods for low-dimensional BO, outperforms the state-of-the-art on all benchmarks, including the high-dimensional StybTang, except for Hartmann-6. Looking closer, we found that `EI`, `KG` and `BM`—without the embedding— did not converge on the 6-d Hartmann function within this budget, hence we believe that this is not caused by the embedding but likely due to the algorithms' default settings. Fig. 1 shows the results for $D = 100$; see Sect. E for $D \in \{25, 1000\}$. `ADD`'s performance is comparable to `HeSBO-EI` but has considerably higher computational cost. We will revisit to this aspect below.

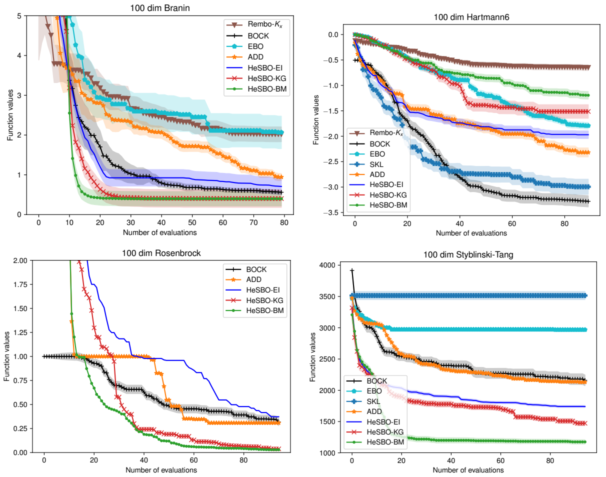

Figure 1. Performances on Branin (ul) with target dimension $d = 4$, on Hartmann-6 (ur) with $d = 6$, on Rosenbrock (bl) with $d = 4$, and on StybTang (br) with $d = 12$. The input dimension is $D = 100$. We see that the proposed embedding in combination with `KG` and `BM` achieves considerably better solutions at the same number of function evaluations than the state-of-the-art. An exception is Hartmann, where `BOCK`, `SKL`, and `ADD` perform best.

Robustness to Target Dimension d and Input Dimension D. We study the robustness of the embedding based algorithms with respect to the target dimension d chosen for the embedding and the input dimension D. Fig 2 shows the performances on Hartmann-6 with input dimensions $D \in \{25, 1000\}$. We see that HeSBO-EI achieves essentially the same performance for all target dimensions d across the different input dimensions D. Note that in particular HeSBO-EI's performance does not degrade when the target dimension d is chosen *smaller* than six, the dimensionality of Hartmann-6. We also tested the robustness on Hartmann-6 with input dimension $D = 100$ and for Branin, and observed a similar robustness in both cases. HeSBO-EI and the three REMBO variants use the same GP model and the same acquisition function EI, thus we attribute the considerably better robustness to the hashing based subspace embedding proposed in Sect. 2.

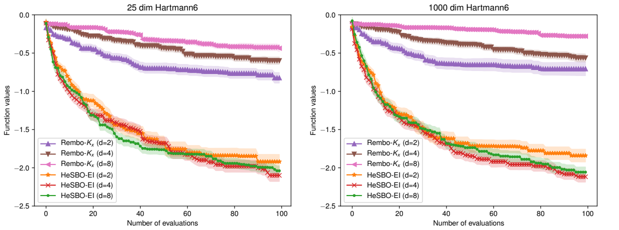

Figure 2. Robustness regarding the target dimension d on Hartmann-6 with input dimensions $D = 25$ (l) and $D = 1000$ (r). The performance of the new subspace embedding HeSBO-EI is robust for the different target dimensions and outperforms the best REMBO variant k_X for all choices of d.

Analysis of the Scalability. We study the scalability of the algorithms in Fig. 3. Here we examine the performance as a function of the wall-clock time that essentially equals the computational overhead of the respective methods. We observe that the embedding-based approaches HeSBO-EI and HeSBO-BM obtain considerably better solutions than the other algorithms at the same cost.

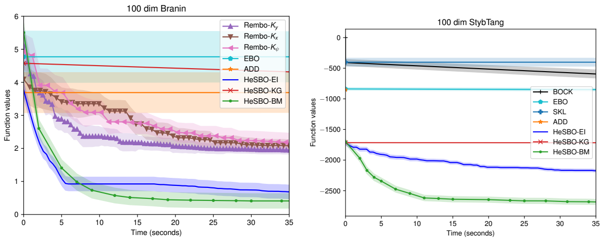

Figure 3. Comparison of the running times on Branin with target dimensions $d = 4$ (l) and on 100d-StybTang with $d = 12$ (r). The input dimension is $D = 100$. We see that the proposed subspace embedding achieves considerably better solutions at the same computational cost compared to the other baselines, in particular REMBO. HeSBO-EI and HeSBO-BM perform best here.

Neural Network Parameter Search. We evaluate the algorithms on a 100-dimensional neural network (NN) optimization task proposed by Oh et al. (2018). Here we are given a NN with one hidden layer of size ten. The goal is to choose the weights between the hidden layer and the outputs in order to minimize the loss on the MNIST data set (LeCun et al., 2017). The other weights and biases are optimized by Adam (Kingma & Ba, 2014). We refer to their paper for details. Fig. 4 summarizes the observed performances on this benchmark for target dimension $d = 12$. We note in passing that $d = 6$ and $d = 24$ gave similar results. HeSBO-KG and BOCK obtain the best performance, followed by HeSBO-EI. Particularly remarkable for this benchmark is that the simple, embedding-based HeSBO-EI finds better solutions than EBO, although the latter operates directly on the high-dimensional search space. This suggests that this high-dimensional problem possesses a low-dimensional approximation of sufficient accuracy.

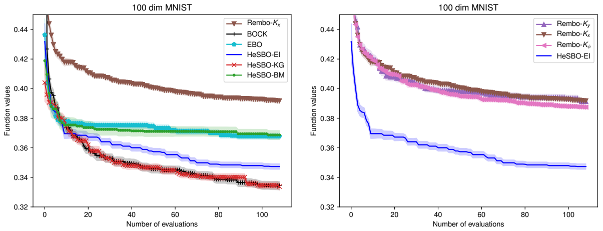

Figure 4. The NN benchmark with target dimension $d = 12$. We observe that HeSBO-KG and BOCK achieve the best performance. Note that HeSBO-EI outperforms EBO that operates on the high-dimensional domain.

4. Conclusion

Enabling Bayesian optimization to high dimensional problems is seen of great practical and theoretical interest, e.g., see (Frazier, 2018). This paper advances both directions. We have proven for a large class of GP-based BO methods that the optimization process would proceed essentially as it would if invoked directly on the active subspace. The experimental evaluation demonstrated the practical value of the proposed embedding technique on a variety of benchmarks. In particular, HeSBO performs better than the state-of-the-art for high-dimensional BO based on random embeddings and structure learning. It is important to note that the subspace embedding has a negligible computational overhead and is agnostic to the way data is acquired. It is thus straightforward to combine it with existing methods, e.g., for batch acquisition (Chevalier & Ginsbourger, 2013; Marmin et al., 2015; Shah & Ghahramani, 2015; Wu & Frazier, 2016; Wang et al., 2016a; Wu et al., 2017). We look forward to "exotic" applications, e.g., in multifidelity optimization.

Acknowledgements

The authors would like to thank Peter Frazier, Jian Wu, Pu Yang, Yunxiang Duke Zhang, Jake Gardner, and Mark McLeod for their help with their respective algorithms. Moreover, they are grateful to the anonymous reviewers for their valuable comments. Alexander Munteanu was partly supported by the German Science Foundation (DFG), Collaborative Research Center SFB 876, project C4 and by the Dortmund Data Science Center (DoDSc).

References

Achlioptas, D. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, **66**(4):671–687, 2003.

Ailon, N. and Chazelle, B. The fast Johnson–Lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39 (1):302–322, 2009.

Ailon, N. and Liberty, E. Fast dimension reduction using Rademacher series on dual BCH codes. *Discrete & Computational Geometry*, **42**(4):615–630, 2009.

Arriaga, R. I. and Vempala, S. An algorithmic theory of learning: Robust concepts and random projection. *Machine Learning*, 63 (2):161–182, 2006.

Bell, J. The singular value decomposition of compact operators on Hilbert spaces. unpublished, 2014.

Binois, M., Ginsbourger, D., and Roustant, O. A warped kernel improving robustness in Bayesian optimization via random embeddings. In *International Conference on Learning and Intelligent Optimization*, pp. 281–286. Springer, 2015.

Binois, M., Ginsbourger, D., and Roustant, O. On the choice of the low-dimensional domain for global optimization via random embeddings. *Journal of Global Optimization*, 2019. Accepted for Publication.

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

Charikar, M., Chen, K. C., and Farach-Colton, M. Finding frequent items in data streams. *Theoretical Computer Science*, **312**(1): 3–15, 2004.

Chen, B., Castro, R. M., and Krause, A. Joint optimization and variable selection of high-dimensional Gaussian processes. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1379–1386. Omnipress, 2012.

Chevalier, C. and Ginsbourger, D. Fast computation of the multi-points expected improvement with applications in batch selection. In *Learning and Intelligent Optimization*, pp. 59–69. Springer, 2013.

Clarkson, K. L. and Woodruff, D. P. Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 81–90, 2013.

Cohen, M. B., Nelson, J., and Woodruff, D. P. Optimal approximate matrix product in terms of stable rank. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, pp. 11:1–11:14, 2016.

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

Eriksson, D., Dong, K., Lee, E., Bindel, D., and Wilson, A. G. Scaling Gaussian process regression with derivatives. In *Advances in Neural Information Processing Systems*, pp. 6868–6878, 2018.

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

Frazier, P. I., Powell, W. B., and Dayanik, S. The Knowledge Gradient Policy for Correlated Normal Beliefs. *INFORMS Journal on Computing*, 21(4):599–613, 2009.

Frean, M. and Boyle, P. Using Gaussian processes to optimize expensive functions. In *Australasian Joint Conference on Artificial Intelligence*, pp. 258–267. Springer, 2008.

Gardner, J., Guo, C., Weinberger, K., Garnett, R., and Grosse, R. Discovering and exploiting additive structure for Bayesian optimization. In *Artificial Intelligence and Statistics*, pp. 1311–1319, 2017.

Gardner, J. R., Kusner, M. J., Xu, Z. E., Weinberger, K. Q., and Cunningham, J. P. Bayesian optimization with inequality constraints. In *Proceedings of the 31th International Conference on Machine Learning*, pp. 937–945, 2014.

Geppert, L. N., Ickstadt, K., Munteanu, A., Quedenfeld, J., and Sohler, C. Random projections for Bayesian regression. *Statistics and Computing*, **27**(1):79–101, 2017.

Hernández-Lobato, J. M., Gelbart, M. A., Hoffman, M. W., Adams, R. P., and Ghahramani, Z. Predictive entropy search for Bayesian optimization with unknown constraints. In *ICML*, 2015.

Huang, D., Allen, T., Notz, W., and Miller, R. Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32(5):369–382, 2006.

Johnson, W. B. and Lindenstrauss, J. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, **26** (1):189–206, 1984.

Kandasamy, K., Schneider, J., and Póczos, B. High dimensional Bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning*, pp. 295–304, 2015.

Kandasamy, K., Dasarathy, G., Oliva, J. B., Schneider, J., and Poczos, B. Gaussian process bandit optimisation with multi-fidelity evaluations. In *Advances in Neural Information Processing Systems*, 2016. The code is available at `https://github.com/kirthevasank/mf-gp-ucb`. Last Accessed on 05/01/2019.

Kandasamy, K., Neiswanger, W., Schneider, J., Poczos, B., and Xing, E. Neural architecture search with Bayesian optimisation and optimal transport. *arXiv preprint arXiv:1802.07191*, 2018.

Kane, D. M. and Nelson, J. Sparser Johnson-Lindenstrauss transforms. *J. ACM*, 61(1):4:1–4:23, 2014.

Kennedy, M. C. and O'Hagan, A. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Klein, A., Falkner, S., Bartels, S., Hennig, P., and Hutter, F. Fast Bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial Intelligence and Statistics*, pp. 528–536, 2017.

LeCun, Y., Cortes, C., and Burges, C. J. The MNIST database of handwritten digits, 2017. http://yann.lecun.com/exdb/mnist/. Last Accessed on 05/13/2019.

Marmin, S., Chevalier, C., and Ginsbourger, D. Differentiating the multipoint expected improvement for optimal batch design. *hal-01133220v2*, 2015.

McLeod, M., Roberts, S., and Osborne, M. A. Optimization, fast and slow: Optimally switching between local and Bayesian optimization. In *International Conference on Machine Learning*, pp. 3440–3449, 2018.

Mutny, M. and Krause, A. Efficient high dimensional bayesian optimization with additivity and quadrature fourier features. In *Advances in Neural Information Processing Systems*, pp. 9005–9016, 2018.

Nelson, J. and Nguyen, H. L. Sparsity lower bounds for dimensionality reducing maps. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 101–110, 2013.

Nelson, J. and Nguyên, H. L. Lower bounds for oblivious subspace embeddings. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, pp. 883–894, 2014.

Oh, C., Gavves, E., and Welling, M. BOCK: Bayesian optimization with cylindrical kernels. In *International Conference on Machine Learning*, 2018. The implementation is available at https://github.com/ChangYong-Oh/HyperSphere. Last Accessed on 05/10/2019.

Paul, S., Boutsidis, C., Magdon-Ismail, M., and Drineas, P. Random projections for linear support vector machines. *ACM Transactions on Knowledge Discovery from Data*, **8**(4):22:1–22:25, 2014.

Picheny, V., Gramacy, R. B., Wild, S., and Le Digabel, S. Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian. In *Advances in Neural Information Processing Systems*, pp. 1435–1443, 2016.

Poloczek, M., Wang, J., and Frazier, P. I. Multi-information source optimization. In *Advances in Neural Information Processing Systems*, pp. 4288–4298, 2017.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006.

Rolland, P., Scarlett, J., Bogunovic, I., and Cevher, V. High-dimensional Bayesian optimization via additive models with overlapping groups. In *International Conference on Artificial Intelligence and Statistics*, pp. 298–307, 2018.

Sarlós, T. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 143–152, 2006.

Shah, A. and Ghahramani, Z. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Advances in Neural Information Processing Systems*, pp. 3330–3338, 2015.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Swersky, K., Snoek, J., and Adams, R. P. Multi-task Bayesian optimization. In *Advances in Neural Information Processing Systems*, pp. 2004–2012, 2013.

Wang, J., Clark, S. C., Liu, E., and Frazier, P. I. Parallel Bayesian global optimization of expensive functions. *arXiv preprint arXiv:1602.05149*, 2016a.

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and de Freitas, N. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55: 361–387, 2016b.

Wang, Z., Li, C., Jegelka, S., and Kohli, P. Batched high-dimensional Bayesian optimization via structural kernel learning. In *International Conference on Machine Learning*, pp. 3656–3664, 2017. The implementation is available at https://github.com/zi-w/Structural-Kernel-Learning-for-HDBBO. Last Accessed on 05/10/2019.

Wang, Z., Gehring, C., Kohli, P., and Jegelka, S. Batched large-scale Bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, pp. 745–754, 2018. The implementation is available at https://github.com/zi-w/Ensemble-Bayesian-Optimization. Last Accessed on 05/10/2019.

Woodruff, D. P. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, **10** (1-2):1–157, 2014.

Wu, J. and Frazier, P. The parallel knowledge gradient method for batch Bayesian optimization. In *Advances in Neural Information Processing Systems*, pp. 3126–3134, 2016. Cornell-MOE is available at https://github.com/wujian16/Cornell-MOE. Last Accessed on 05/10/2019.

Wu, J., Poloczek, M., Wilson, A. G., and Frazier, P. I. Bayesian optimization with gradients. In *Advances in Neural Information Processing Systems*, 2017. Cornell-MOE is available at https://github.com/wujian16/Cornell-MOE. Last Accessed on 05/10/2019.