Bayesian Optimization in High Dimensions via Random Embeddings

Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, Nando de Freitas

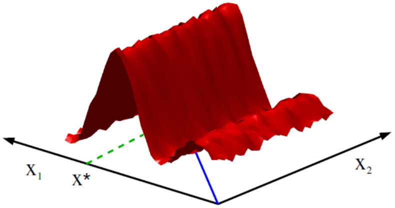
Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Aaron Klein

Automated Parameter Tuning and Algorithm Configuration
June 28 2013

Low Effective Dimensionality



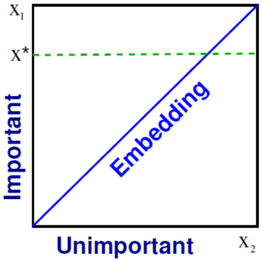


Table of Contents

Bayesian Optimization

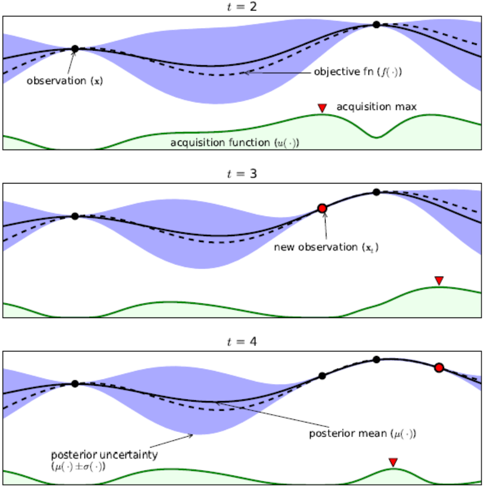


Table of Contents

Low Effective Dimensionality

UNI FREIBURG

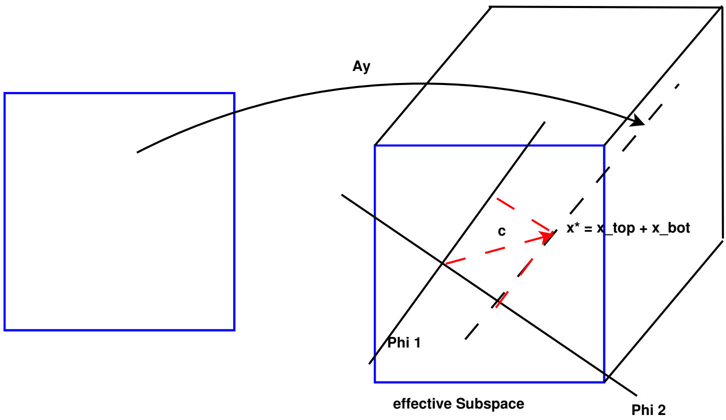
Definition

A function $f : \mathbb{R}^D \to \mathbb{R}$ has **effective dimensionality** d_e if:

- $\exists$ a linear subspace $\mathscr{T}$ of dimension d_e s.t. for all $x_\top \in \mathscr{T} \subset \mathbb{R}^D$ and $x_\perp \in \mathscr{T}^\perp \subset \mathbb{R}^D$ (orthogonal complement)
- $\Rightarrow f(x) = f(x_\top + x_\perp) = f(x_\top)$
- $\mathscr{T}$ is called the **effective subspace** of f and $\mathscr{T}^\perp$ the **constant subspace**

Theorem

If f has effective dimensionality d_e and a random Gaussian matrix $\mathbf{A} \in \mathbb{R}^{D \times d}$ with $d \geqslant d_e$:
$\Rightarrow$ with probability 1, for any $x \in \mathbb{R}^D$, $\exists$ a $y \in \mathbb{R}^d$ s.t. $f(x) = f(\mathbf{A}y)$.


Ay
c
x* = x_top + x_bot
Phi 1
effective Subspace
Phi 2

Proof Sketch.

Hence, $f(x) = f(x_\top + x_\perp) = f(x_\top)$

Proof Sketch.

Hence, $f(x) = f(x_\top + x_\perp) = f(x_\top)$

- it suffices to show that $\exists y \in \mathbb{R}^d$ s.t $f(x_\top) = f(\mathbf{A}y)$.

Proof Sketch.

Hence, $f(x) = f(x_\top + x_\perp) = f(x_\top)$

- it suffices to show that $\exists y \in \mathbb{R}^d$ s.t $f(x_\top) = f(\mathbf{A}y)$.
- $\Phi \in \mathbb{R}^{D \times d_e}$ whose columns form an orthonormal basis for $\mathscr{T}$. For each $x_\top \in \mathscr{T}$ it exists a $c \in \mathbb{R}^{d_e}$ s.t. $x_\top = \Phi c$.

REMBO

Proof Sketch.

Hence, $f(x) = f(x_\top + x_\perp) = f(x_\top)$

- it suffices to show that $\exists y \in \mathbb{R}^d$ s.t $f(x_\top) = f(\mathbf{A}y)$.
- $\Phi \in \mathbb{R}^{D \times d_e}$ whose columns form an orthonormal basis for $\mathcal{T}$. For each $x_\top \in \mathcal{T}$ it exists a $c \in \mathbb{R}^{d_e}$ s.t. $x_\top = \Phi c$.
- $rank(\Phi^\top A) = d_e$ and therefore we have $(\Phi^\top A)y = c$ for some y

REMBO

Proof Sketch.

Hence, $f(x) = f(x_\top + x_\perp) = f(x_\top)$

- it suffices to show that $\exists y \in \mathbb{R}^d$ s.t $f(x_\top) = f(\mathbf{A}y)$.
- $\Phi \in \mathbb{R}^{D \times d_e}$ whose columns form an orthonormal basis for $\mathscr{T}$. For each $x_\top \in \mathscr{T}$ it exists a $c \in \mathbb{R}^{d_e}$ s.t. $x_\top = \Phi c$.
- $rank(\Phi^\top A) = d_e$ and therefore we have $(\Phi^\top A)y = c$ for some y
- orthogonal projection Ay onto $\mathscr{T}$ is given by $\Phi \Phi^T A y = \Phi c = x_\top$.

Proof Sketch.

Hence, $f(x) = f(x_\top + x_\perp) = f(x_\top)$

- it suffices to show that $\exists y \in \mathbb{R}^d$ s.t $f(x_\top) = f(\mathbf{A}y)$.
- $\Phi \in \mathbb{R}^{D \times d_e}$ whose columns form an orthonormal basis for $\mathscr{T}$. For each $x_\top \in \mathscr{T}$ it exists a $c \in \mathbb{R}^{d_e}$ s.t. $x_\top = \Phi c$.
- $rank(\Phi^\top A) = d_e$ and therefore we have $(\Phi^\top A)y = c$ for some y
- orthogonal projection Ay onto $\mathscr{T}$ is given by $\Phi\Phi^T Ay = \Phi c = x_\top$.
- $Ay = x_\top + x'$ with $x' \in \mathscr{T}^\perp$ and $f(Ay) = f(x_\top + x') = f(x_\top)$

$\square$

Table of Contents

Random Embeddings for Bayesian Optimization

The basic idea of REMBO:

- For any $x \in \mathbb{R}^D$ and a random matrix $A \in \mathbb{R}^{D \times d}$, there is a point $y \in \mathbb{R}^d$ such that $f(x) = f(Ay)$
- Thus, for any optimizer $x^\star \in \mathbb{R}^D$ it exists a $y^\star \in \mathbb{R}^d$ such that $f(x^\star) = f(\mathbf{A}y^\star)$
- Instead of optimizing in the high dimensional space, we optimize $g(\mathbf{y}) = f(\mathbf{Ay})$

for $t = 1, 2, ..$ **do**

Find $x_{t+1} \in \mathbb{R}^D$ by optimizing the acquisition function

$u : x_{t+1} = \arg\max_{x \in \mathscr{X}} u(\mathbf{x}|D_t)$

Augment the data $D_{t+1} = \{D_t, (x_{t+1}, f(\mathbf{x_{t+1}}))\}$

end

Algorithmus 1: Bayesian Optimization

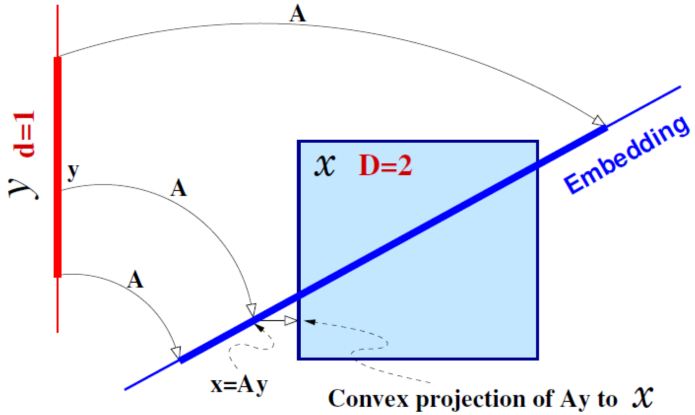
Random Embeddings for Bayesian Optimization

Generate a random matrix A;
for $t = 1, 2, ..$ **do**
  Find $\mathbf{y_{t+1}} \in \mathbb{R}^d$ by optimizing the acquisition function $u : \mathbf{y_{t+1}} = \arg\max_{\mathbf{y} \in \mathscr{Y}} u(\mathbf{y}|D_t)$
  Augment the data $D_{t+1} = \{D_t, (\mathbf{y_{t+1}}, f(\mathbf{A}\mathbf{y_{t+1}})\}$
end

Algorithmus 2: REMBO

Table of Contents

Bounded Regions



Bounded Regions

How do we choose the bounded region $\mathscr{Y} \subset \mathbb{R}^d$ where REMBO performs the Bayesian Optimization?

- Is $\mathscr{Y}$ too small, it is more likely that it does not include the global optimizer
- Is $\mathscr{Y}$ too big, it is more time consuming

Bounded Regions

Theorem

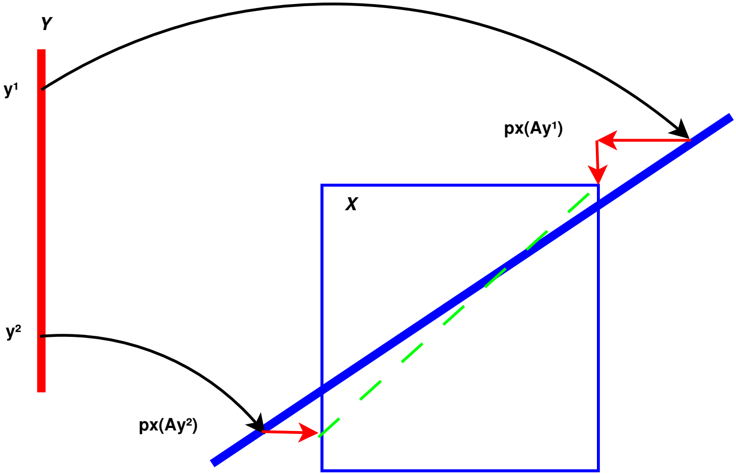
If $\mathbf{A}$ is a $D \times d$ random Gaussian matrix, there exists an optimizer $y^\star \in \mathbb{R}^d$ such that $f(\mathbf{A}y^\star) = f(x_\top^\star)$ and $\|y^\star\|_2 \leqslant \frac{\sqrt{d_e}}{\varepsilon}\|x_\top^\star\|_2$ with probability at least $1 - \varepsilon$.

Random Embeddings for Bayesian Optimization

- The Theorem says that with probability $1-\varepsilon$ the optimizer is in $\mathscr{Y}$ and that with probability $\delta \leqslant \varepsilon$ the optimizer lies outside of $\mathscr{Y}$
- To increase the success rate:
  - Run REMBO k times with different independently drawn random embeddings. This decreases δ to δ^k
  - Increase the dimensionality d. If $d > d_e$ there are $\binom{d}{d_e}$ different embeddings and it is more likely that the optimizer is included.

Table of Contents

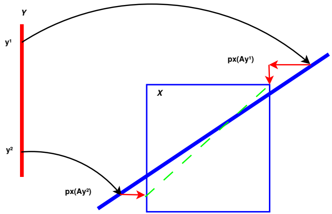
Kernels



Kernels

Low dimensional kernel:

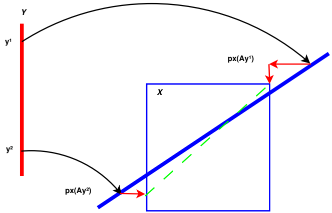
- $k_l^d(y^{(1)}, y^{(2)}) = \exp\left(-\frac{\|y^{(1)} - y^{(2)}\|^2}{2l^2}\right)$
- Squared exponential kernel in low dimensions
- It constructs a GP in the d-dimensional space
- Effective in continuous space



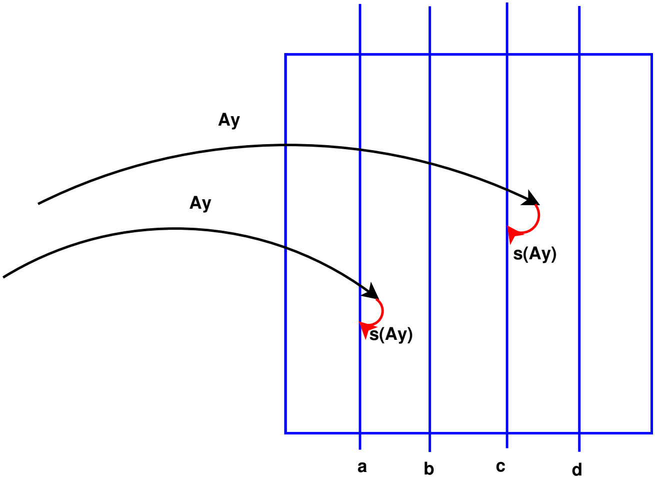
Kernels

High dimensional Kernel:

- $k_l^D(y^{(1)}, y^{(2)}) = \exp\left(-\frac{\|p_X(Ay^{(1)}) - p_X(Ay^{(2)})\|^2}{2l^2}\right)$ with $p_X : \mathbb{R}^D \to \mathbb{R}^D$ as the standard projection operator
- It constructs a GP in the D-dimensional space
- Search space is not $\mathscr{X}$ any more, it is now the smaller subspace $\{p_x(Ay) : y \in \mathscr{Y}\}$



Categorical Kernel:



Kernels

Categorical Kernel:

- $k_\lambda^D(y^{(1)}, y^{(1)}) = \exp\left(-\frac{\lambda}{2} g(s(Ay^{(1)}), s(Ay^{(2)}))^2\right)$
  with $g(x^{(1)}, x^{(2)}) = |\{i : x_i^{(1)} \neq x_i^{(2)}\}|$ as the Hamming distance and the function s that maps continuous vectors to discrete vectors
- Measures the distance between two low dimensional points as the distance between their high dimensional mappings

Table of Contents

Bayesian Optimization in a Billion Dimensions

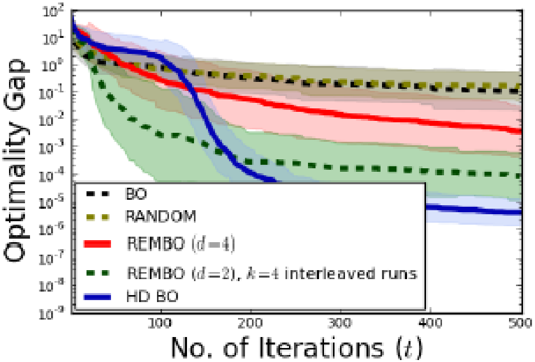
- Bayesian Optimization of the $d_e = 2$-dimensional Branin function embedded in a D-dimensional space with $D - d_e$ unimportant dimensions.
- The optimality gap as the performance measurement, i.e. the difference of the best function value and the optimal function value
- 500 function evaluations and $\frac{500}{k}$ for each run
- Tested with $D = 25$ and $D = 10^9$ and different internal dimensions d

Bayesian Optimization in a Billion Dimensions

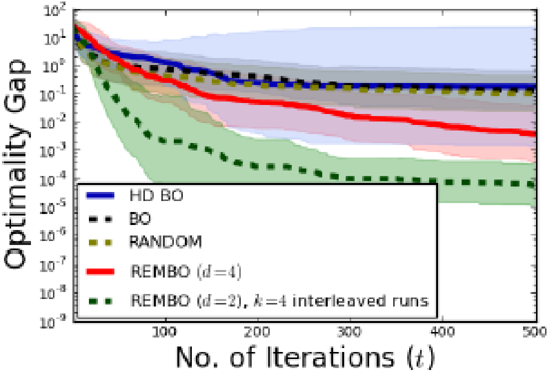
k	$d = 2$	$d = 4$	$d = 6$
10	0.0022 ± 0.0035	0.1553 ± 0.1601	0.4865 ± 0.4796
5	0.0004 ± 0.0011	0.0908 ± 0.1252	0.2586 ± 0.3702
4	0.0001 ± 0.0003	0.0654 ± 0.0877	0.3379 ± 0.3170
2	0.1514 ± 0.9154	0.0309 ± 0.0687	0.1643 ± 0.1877
1	0.7406 ± 1.8996	0.0143 ± 0.0406	0.1137 ± 0.1202

BO of the Branin function embedded in $D = 25$ dimensions

Bayesian Optimization in a Billion Dimensions



Bayesian Optimization in a Billion Dimensions



Bayesian Optimization in a Billion Dimensions

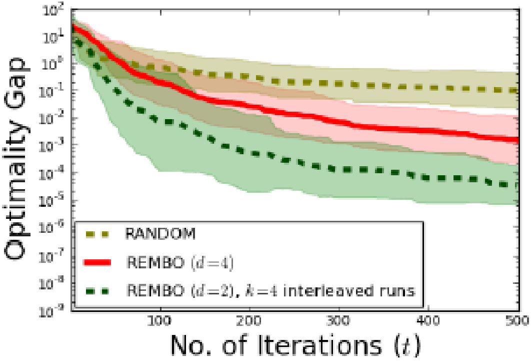


Table of Contents

Optimizing LPSOLVE

LPSOLVE

- Popular mixed integer linear programming solver

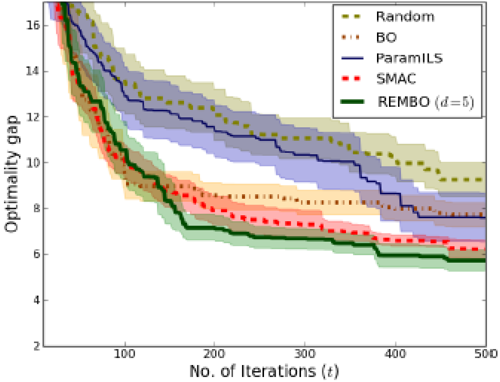
Problem instance

- Deterministic blackbox optimization problem
- The configuration problem includes 40 binary and 7 categorical parameters of LPSOLVE

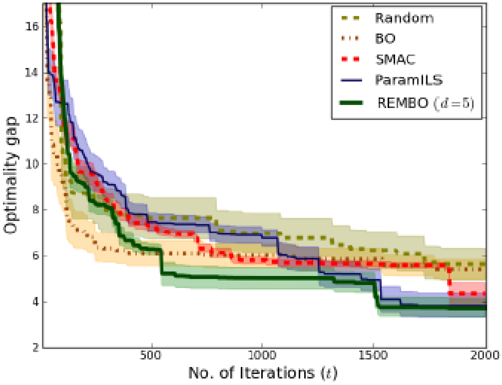
Experiment

- Comparison of BO, REMBO, ParamILS, SMAC and Random Search
- Due to the categorical parameters, the high-dimensional kernel is used for REMBO

Optimizing LPSOLVE



Optimizing LPSOLVE



Discussion

- REMBO performs Bayesian Optimization in a random embedded subspace in the high dimensional space of the objective function. The assumption is that the objective function has a low effective dimensionality.
- Normally no parameters are totally unimportant
- How do we choose the internal dimension d?
- It is not quite clear yet, how many practical problems fall into this class