Bayesian Optimization for Hyperparameter Optimization

High-Dimensional Bayesian Optimization

Bernd Bischl   Frank Hutter   Lars Kotthoff
Marius Lindauer   Joaquin Vanschoren

High-Dimensional Bayesian Optimization: Motivation

- Issue: Standard BO works best on problems of moderate dimensions $d \leq 20$
  - Standard Gaussian processes do not tend to fit well in high dimensions
  - Maximizing the acquisition function is also computationally challenging

High-Dimensional Bayesian Optimization: Motivation

- Issue: Standard BO works best on problems of moderate dimensions $d \leq 20$
  - Standard Gaussian processes do not tend to fit well in high dimensions
  - Maximizing the acquisition function is also computationally challenging
- Possible solutions we will discuss:
  - Different models, in particular random forests [Hutter et al. 2011]
  - Embedding into a low-dimensional space (REMBO) [Wang et al. 2016]
  - Additive models [Kandasamy et al. 2015]

Low Effective Dimensionality

- Many optimization problems in practice have low effective dimensionality
  - E.g., HPO for deep neural networks [Bergstra et al. 2012]
  - E.g., algorithm configuration for combinatorial optimization solvers [Hutter et al. 2014]

Low Effective Dimensionality

- Many optimization problems in practice have low effective dimensionality
  - E.g., HPO for deep neural networks [Bergstra et al. 2012]
  - E.g., algorithm configuration for combinatorial optimization solvers [Hutter et al. 2014]

- Idea: Exploit low effective dimensionality to cover a lower-dimensional space well

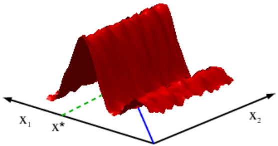
Using Random Forests for High Dimensions / Low Effective Dimensionality

- Random forests are automatic feature detectors
  - They automatically select the important (axis-aligned) inputs
- Random forests have indeed be used effectively on spaces of more than 700 hyperparameters
  - In terms of computational efficiency, they do not pose a bottleneck
  - In terms of statistical efficiency, they scale more gracefully to high dimensions than GPs

Random Embeddings for Exploiting Low Effective Dimensionality: Overview

Given a $D = 2$ dimensional black-box function $c(x_1, x_2)$:

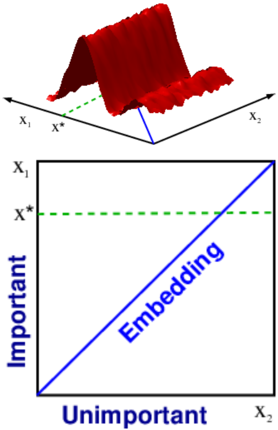
- Assume we know c has only $d = 1$ important dimensions, but we don't know which one it is.



Random Embeddings for Exploiting Low Effective Dimensionality: Overview

Given a $D = 2$ dimensional black-box function $c(x_1, x_2)$:

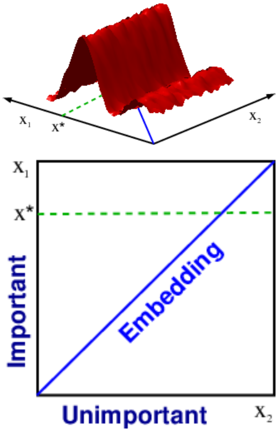
- Assume we know c has only $d = 1$ important dimensions, but we don't know which one it is.
- Subspace $x_1 = x_2$ is guaranteed to include the optimum.



Random Embeddings for Exploiting Low Effective Dimensionality: Overview

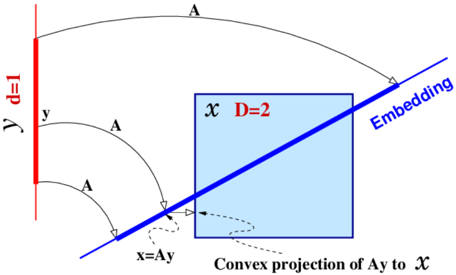
Given a $D = 2$ dimensional black-box function $c(x_1, x_2)$:

- Assume we know c has only $d = 1$ important dimensions, but we don't know which one it is.
- Subspace $x_1 = x_2$ is guaranteed to include the optimum.
- This idea applies to any d-dimensional linear subspace; allows scaling to arbitrary D (e.g., $D = 1$ billion)



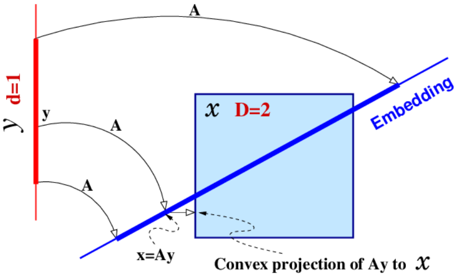
Random Embedding Bayesian Optimization (REMBO)

- Generate a random matrix $A \in \mathbb{R}^{D \times d}$
- Use BO to optimize $g(\boldsymbol{\lambda}) = c(\boldsymbol{Ay})$ instead of high dimensional $c(\boldsymbol{\lambda})$



Random Embedding Bayesian Optimization (REMBO)

- Generate a random matrix $A \in \mathbb{R}^{D \times d}$
- Use BO to optimize $g(\boldsymbol{\lambda}) = c(\boldsymbol{A}\boldsymbol{y})$ instead of high dimensional $c(\boldsymbol{\lambda})$



Theorem

If the effective dimensionality of c is at most d, then with probability 1, for any $\boldsymbol{\lambda} \in \mathbb{R}^D$, there exists a $\boldsymbol{y} \in \mathbb{R}^d$ such that $c(\boldsymbol{\lambda}) = c(\boldsymbol{A}\boldsymbol{y})$.

High Dimensional Bayesian Optimization: REMBO Pseudocode

REMBO: Bayesian Optimization with Random Embedding

Require: Search space $\boldsymbol{\Lambda}$, cost function c, acquisition function u, predictive model $\hat{c}$, maximal number of function evaluations T

Result : Best observed configuration $\hat{\boldsymbol{\lambda}}$ according to $\mathcal{D}^{(T)}$ or $\hat{c}$

```
Generate a random matrix $\boldsymbol{A} \in \mathbb{R}^{D \times d}$
$\mathcal{D}^{(0)} \leftarrow \varnothing$
for $t = 1$ to $T$ do
    $\hat{c}^{(t)} \leftarrow$ fit predictive model on $\mathcal{D}^{(t-1)}$
    $\boldsymbol{y} \leftarrow \boldsymbol{y} \in \arg\max_{\boldsymbol{y} \in \mathcal{Y}} u(\boldsymbol{y}|\mathcal{D}^{(t-1)}, \hat{c}^{(t)})$
    Query $c(\boldsymbol{Ay})$;
    $\mathcal{D}^{(t)} \leftarrow \mathcal{D}^{(t-1)} \cup \{\langle \boldsymbol{Ay}, c(\boldsymbol{Ay}) \rangle\}$
```

Advantages

- Exploits low effective dimensionality
- Allows scaling to arbitrarily high extrinsic dimensions
- Applies to both continuous and categorical variables
- Trivial modification of BO algorithm
- Coordinate independent (invariant under rotations)

Advantages

- Exploits low effective dimensionality
- Allows scaling to arbitrarily high extrinsic dimensions
- Applies to both continuous and categorical variables
- Trivial modification of BO algorithm
- Coordinate independent (invariant under rotations)

Disadvantages

- Sensitive to the definition of the bounded low dimensional constrained space $\mathcal{Y}$
- Assumes truly unimportant dimensions

High Dimensional Bayesian Optimization via Additive Models

- Recall:
  - Standard GPs do not tend to fit well in high dimensions
  - Maximizing the acquisition function is also computationally challenging

High Dimensional Bayesian Optimization via Additive Models

- Recall:
  - Standard GPs do not tend to fit well in high dimensions
  - Maximizing the acquisition function is also computationally challenging
- Idea:
  - Assume additive structure of the objective function [Kandasamy et al. 2015]:

$$f(\boldsymbol{\lambda}) = f^{(1)}(\boldsymbol{\lambda}^{(1)}) + f^{(2)}(\boldsymbol{\lambda}^{(2)}) + ... + f^{(M)}(\boldsymbol{\lambda}^{(M)})$$

  - Model each $f^{(i)}$ by an individual GP

High Dimensional Bayesian Optimization via Additive Models

- Recall:
  - Standard GPs do not tend to fit well in high dimensions
  - Maximizing the acquisition function is also computationally challenging
- Idea:
  - Assume additive structure of the objective function [Kandasamy et al. 2015]:

$$f(\boldsymbol{\lambda}) = f^{(1)}(\boldsymbol{\lambda}^{(1)}) + f^{(2)}(\boldsymbol{\lambda}^{(2)}) + \ldots + f^{(M)}(\boldsymbol{\lambda}^{(M)})$$

  - Model each $f^{(i)}$ by an individual GP
  - If the decomposition does not overlap:
    can maximize acquisition function separately for each of the $f^{(i)}$

High Dimensional Bayesian Optimization via Additive Models

- Recall:
  - Standard GPs do not tend to fit well in high dimensions
  - Maximizing the acquisition function is also computationally challenging
- Idea:
  - Assume additive structure of the objective function [Kandasamy et al. 2015]:

$$f(\boldsymbol{\lambda}) = f^{(1)}(\boldsymbol{\lambda}^{(1)}) + f^{(2)}(\boldsymbol{\lambda}^{(2)}) + \ldots + f^{(M)}(\boldsymbol{\lambda}^{(M)})$$

  - Model each $f^{(i)}$ by an individual GP
  - If the decomposition does not overlap:
    can maximize acquisition function separately for each of the $f^{(i)}$
  - Best results for known decomposition, but also possible to learn decomposition from the data

High Dimensional Bayesian Optimization via Additive Models

Advantages

- Exploits low effective dimensionality
- Scales GPs to high-dimensional parameter spaces
- Regret is linearly dependent on the dimension D when c is additive

High Dimensional Bayesian Optimization via Additive Models

Advantages

- Exploits low effective dimensionality
- Scales GPs to high-dimensional parameter spaces
- Regret is linearly dependent on the dimension D when c is additive

Disadvantages

- Sensitive to the number of additive components
- Restricted to an axis-aligned representation
- Relies on assumption of additivity

Questions to Answer for Yourself / Discuss with Friends

- Repetition. What is the main assumption behind REMBO?
- Repetition. What is the main assumption behind additive modelling?
- Discussion. Are these assumptions likely satisfied for tuning deep neural networks?
- Discussion. How do random forests help deal with high dimensions and low effective dimensionality? Can they also model additive structure?