

Multi-criteria Optimization

Bayesian Optimization

Bernd Bischl Frank Hutter Lars Kotthoff
Marius Lindauer Joaquin Vanschoren

Recap: Bayesian Optimization I

Advantages of BO

- Sample efficient
- Can handle noise
- Native incorporation of priors
- Does not require gradients
- Theoretical guarantees

We will now extend BO to multiple cost functions.

Recap: Bayesian Optimization II

Bayesian optimization loop

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

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

- 1 Initialize data $\mathcal{D}^{(0)}$ with initial observations
 - 2 **for** $t = 1$ **to** T **do**
 - 3 Fit predictive model $\hat{c}^{(t)}$ on $\mathcal{D}^{(t-1)}$
 - 4 Select next query point: $\lambda^{(t)} \in \arg \max_{\lambda \in \Lambda} u(\lambda; \mathcal{D}^{(t-1)}, \hat{c}^{(t)})$
 - 5 Query $c(\lambda^{(t)})$
 - 6 Update data: $\mathcal{D}^{(t)} \leftarrow \mathcal{D}^{(t-1)} \cup \{\langle \lambda^{(t)}, c(\lambda^{(t)}) \rangle\}$
-

Multi-Criteria Bayesian Optimization

Goal: Extend Bayesian optimization to multiple cost functions

$$\min_{\boldsymbol{\lambda} \in \Lambda} c(\boldsymbol{\lambda}) \Leftrightarrow \min_{\boldsymbol{\lambda} \in \Lambda} (c_1(\boldsymbol{\lambda}), c_2(\boldsymbol{\lambda}), \dots, c_m(\boldsymbol{\lambda})) .$$

There are two basic approaches:

- 1 Simplify the problem by scalarizing the cost functions, or
- 2 define acquisition functions for multiple cost functions.

Scalarization

Idea: Aggregate all cost functions

$$\min_{\lambda \in \Lambda} \sum_{i=1}^m w_i c_i(\lambda) \quad \text{with } w_i \geq 0$$

- **Obvious problem:** How to choose w_1, \dots, w_m ?
 - ▶ Expert knowledge?
 - ▶ Systematic variation?
 - ▶ Random variation?
- If expert knowledge is not available a-priori, we need to ensure that different trade-offs between cost functions are explored.
- Simplifies multi-criteria optimization problem to single-objective
 - Bayesian optimization can be used without adaption of the general algorithm.

Scalarize the cost functions using the augmented Tchebycheff norm / achievement function

$$c = \max_{i=1,\dots,m} (w_i c_i(\boldsymbol{\lambda})) + \rho \sum_{i=1}^m w_i c_i(\boldsymbol{\lambda}),$$

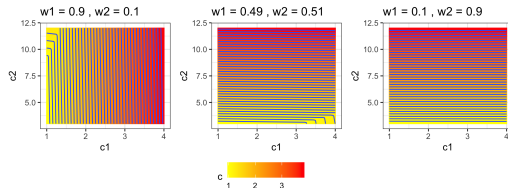
- The weights $w \in W$ are drawn from

$$W = \left\{ w = (w_1, \dots, w_m) \mid \sum_{i=1}^m w_i = 1, w_i = \frac{l}{s} \wedge, l \in 0, \dots, s \right\},$$

with $|W| = \binom{s+m-1}{k-1} 1$.

- New weights are drawn in every BO iteration.
- ρ is a small parameter suggested to be set to 0.05.
- s selects the number of different weights to draw from.

Why the Tchebycheff norm?



$$c = \max_{i=1,\dots,m} (w_i c_i(\lambda)) + \rho \sum_{i=1}^m w_i c_i(\lambda),$$

- The norm consists of two components:
 - ▶ $\max_{i=1,\dots,m} (w_i c_i(\lambda))$ takes only the maximum weighted cost into account.
 - ▶ $\sum_{i=1}^m w_i c_i(\lambda)$ is the weighted sum of all cost functions.
- ρ describes the trade-off between these components.
- By the randomized weights in each iteration and the usually small value of $\rho = 0.05$, this allows exploration of extreme points of single cost functions.
- One can prove: **Every solution of the scalarized problem is pareto-optimal!**

ParEGO Algorithm

ParEGO loop

Require: Search space Λ , cost function c , acquisition function u , predictive model \hat{c} , maximal number of function evaluations T , ρ , l , s

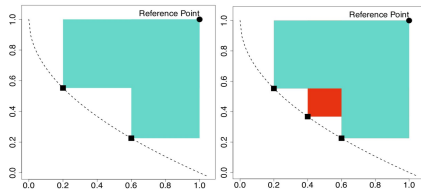
Result : Best configuration $\hat{\lambda}$ (according to \mathcal{D} or \hat{c})

- 1 Initialize data $\mathcal{D}^{(0)}$ with initial observations
 - 2 **for** $t = 1$ **to** T **do**
 - 3 Sample w from $\{w = (w_1, \dots, w_m) \mid \sum_{i=1}^m w_i = 1, w_i = \frac{l}{s} \wedge, l \in 0, \dots, s\}$;
 - 4 Compute scalarization $c^{(t)} = \max_{i=1, \dots, m} (w_i c_i(\lambda)) + \rho \sum_{i=1}^m w_i c_i(\lambda)$;
 - 5 Fit predictive model $\hat{c}^{(t)}$ on $\mathcal{D}^{(t-1)}$
 - 6 Select next query point: $\lambda^{(t)} \in \arg \max_{\lambda \in \Lambda} u(\lambda; \mathcal{D}^{(t-1)}, \hat{c}^{(t)})$
 - 7 Query $c(\lambda^{(t)})$
 - 8 Update data: $\mathcal{D}^{(t)} \leftarrow \mathcal{D}^{(t-1)} \cup \{\langle \lambda^{(t)}, c(\lambda^{(t)}) \rangle\}$
-

Hypervolume based Acquisition Functions

Idea: Define acquisition function that directly models contribution to dominated HV.

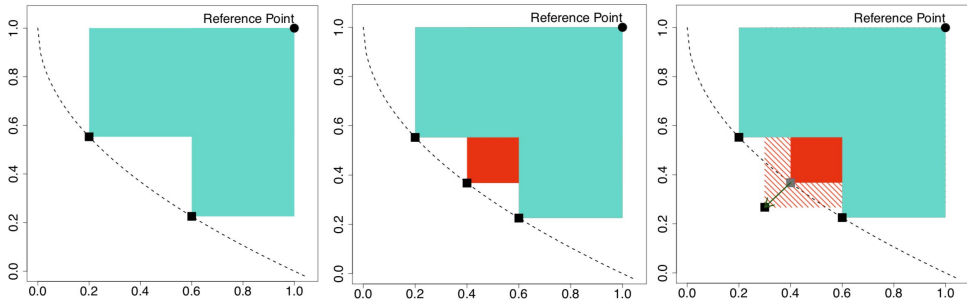
$$\max(0, S(\mathcal{P} \cup \boldsymbol{\lambda}, R) - S(\mathcal{P}, R))$$



- Fit m single-objective surrogate models $\hat{c}_1, \dots, \hat{c}_m$
- Acquisition function takes all surrogate models into account.
- Single-criteria optimization of acquisition function.

S-Metric Selection-based EGO I

Using the Lower Confidence bound $u_{\text{LCB},1}(\lambda), \dots, u_{\text{LCB},m}(\lambda)$, an optimistic estimate of hypervolume contribution can be calculated.



S-Metric Selection-based EGO II

Problem: Based on the way the hypervolume contribution is measured large plateaus of zero improvement are present.

- These make optimization much harder.
- An adaptive penalty is added to regions in which the lower confidence bound is dominated.

This method is referred to as SMS-EGO [Ponweiser et al. 2008].

Further Hypervolume based Acquisition Functions

Expected Hypervolume Improvement (EHI) [Yang et al. 2019]

$$u_{EI, \mathcal{H}}(\boldsymbol{\lambda}) = \int_{-\infty}^{\infty} p(c \mid \boldsymbol{\lambda}) \times \mathcal{H}(\boldsymbol{\lambda}) \, dc,$$

with $\mathcal{H}(\boldsymbol{\lambda}) = S(\mathcal{P} \cup \boldsymbol{\lambda}, R) - S(\mathcal{P}, R)$.

- Direct extension of u_{EI} to the hypervolume.
- $p(c \mid \boldsymbol{\lambda})$ is the joint density of the surrogate model predictions at $\boldsymbol{\lambda}$.
- As the surrogates are GPs and modeled independently of each other, this is just an integral over m univariate normal distributions.
- Efficient computations for $m \leq 3$ exist, beyond that expensive simulation-based computation is required.

Further hypervolume based acquisition functions:

- **Stepwise Uncertainty Reduction** (SUR) based on the probability of improvement.
- **Expected Maximin Improvement** (EMI) based on the ϵ -indicator.

Hypervolume based BO Algorithm

Hypervolume based Bayesian optimization loop

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

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

- 1 Initialize data $\mathcal{D}^{(0)}$ with initial observations
 - 2 **for** $t = 1$ **to** T **do**
 - 3 Fit predictive models $\hat{c}_1^{(t)}, \dots, \hat{c}_m^{(t)}$ on $\mathcal{D}^{(t-1)}$
 - 4 Select next query point: $\lambda^{(t)} \in \arg \max_{\lambda \in \Lambda} u(\lambda; \mathcal{D}^{(t-1)}, \hat{c}_1^{(t)}, \dots, \hat{c}_m^{(t)})$
 - 5 Query $c(\lambda^{(t)})$
 - 6 Update data: $\mathcal{D}^{(t)} \leftarrow \mathcal{D}^{(t-1)} \cup \{\langle \lambda^{(t)}, c(\lambda^{(t)}) \rangle\}$
-