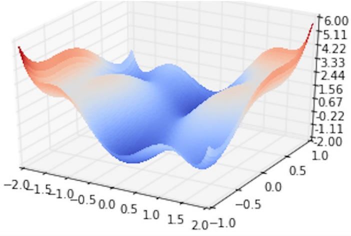
Pareto-efficient acquisition functions for Cost-Aware Bayesian-Optimization

Gauthier Guinet,
Valerio Perrone, Cédric Archambeau

Amazon Web Services

Goal and Challenges

- **Goal**: find $x_\star = \mathrm{argmin}_{x \in \mathcal{X}} f(x)$, where f is an expensive black-box function



Goal and Challenges

- **Goal**: find $x_\star = \operatorname{argmin}_{x \in \mathcal{X}} f(x)$, where f is an expensive **black-box** function.
  - No analytical form or gradient
  - Evaluations may be noisy
  - Grey-Box Setting is sometimes more realistic and useful in practice

Goal and Challenges

- **Goal**: find $x_\star = \operatorname{argmin}_{x \in \mathcal{X}} f(x)$, where f is an **expensive** black-box function.
  - Expensive is a relative notion
  - Real meaning is that we target Sample Efficiency or in other words, we are in limited budget scenario

Bayesian Optimization

Keys Ideas:

- Sequential Optimization
- Surrogate Model: Learn a probabilistic model $\mathcal{M}$ of f , which is cheap to evaluate
- Acquisition Function: Query f by balancing **exploitation against exploration**

Bayesian Optimization

Acquisition function:

- $EI(\mathrm{x}) = \mathrm{E}[\max(0, f(\mathrm{x}_{min}) - f(\mathrm{x}))]$ (Simple, efficient, closed form results)
- But also many others (Improvement-Based, Entropy-Based or Portfolio-Based...)

Surrogate Model:

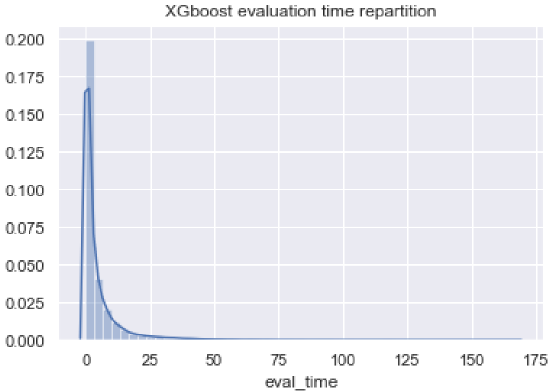
- Gaussian Process (Simple, closed form results)
- But also many others (Random Forest, Bayesian Neural Networks...)

Motivation

- **The Cost Assumption**: The cost of evaluation f is huge yet **homogeneous**.
- **Why? :**
  - Marginal contribution
  - Iteration framework
  - BO has a greedy way of working

Motivation

- **The Cost Assumption**: The cost of evaluation f is huge yet **homogeneous**.
- **Limits 1:** In practice, this is often not true, and by several orders of magnitude



Motivation

- **The Cost Assumption**: The cost of evaluation f is huge yet **homogeneous**.
- **Limits 2:** No control on cost for user aside from the opaque notion of iteration.

Existing Solutions

- **Maximum gain per cost**: $EIpu(x) = \frac{EI(x)}{c(x)}$ (Current de-facto standard)

Existing Solutions

- **Maximum gain per cost:** $EIpu(x) = \frac{EI(x)}{c(x)}$ (Current de-facto standard)
- **Early and cheap, late and expensive:** $EI - cool(x) = \frac{EI(x)}{c(x)^{\alpha}}$, where α is the percentage of remaining budget (Latest Paper on the topic)

One intuition, two problems

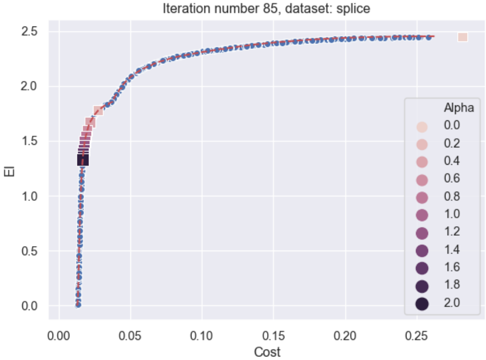
- **Optimal Time Allocation Problem**: Allocate a **maximum time budget** and try to maximize accuracy with no more constraints on maximum number of iteration.
- **Bi-Optimization Problem:** Allocate a **maximum iteration budget** and look for the **best trade-off gain in time vs loss in accuracy** at the end of iteration count.

A Pareto Front intuition - Introduction

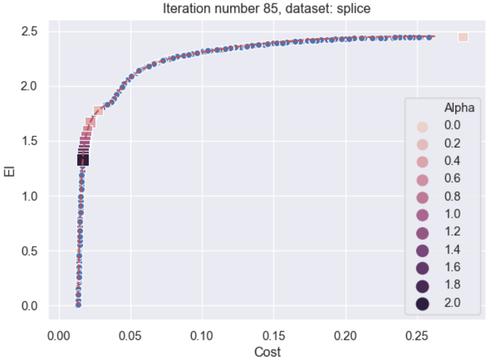
How to better understand cost impact when considered with EI?

- Each $x \in \mathcal{X}$ leads to a given cost and EI value, at time step t
- Some of these values are **Pareto-optimal**.

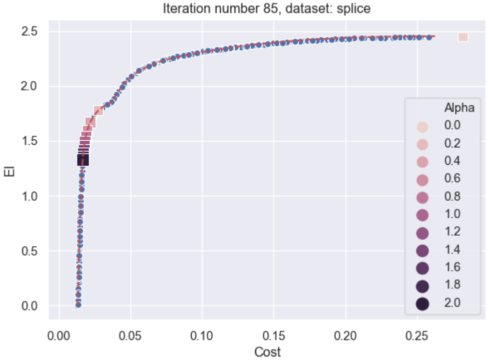
A Pareto Front intuition



A Pareto Front intuition



A Pareto Front intuition



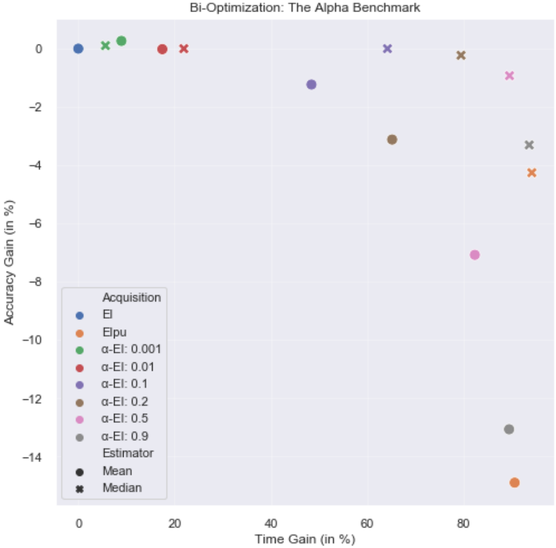
A Pareto Front intuition - Quick summary of results

- **Strong and general functional form**
- **Quite unpredictable evolution**
- **Lack of optimality persistence**

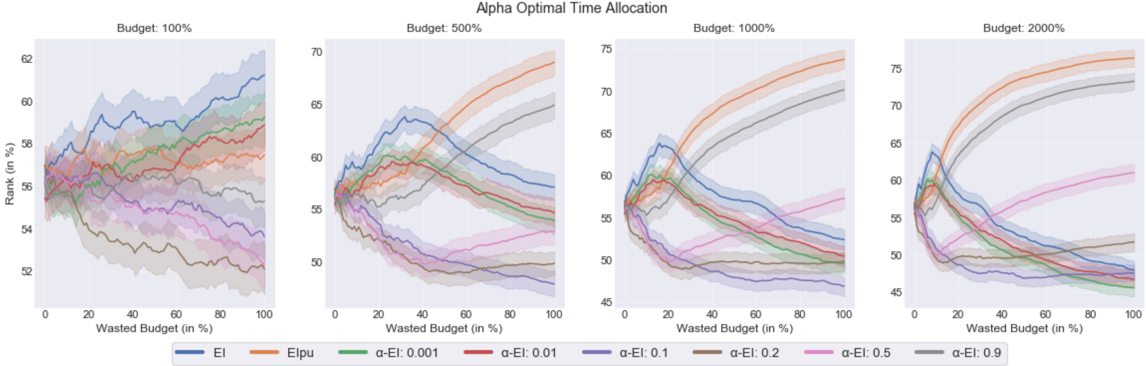
A Pareto Front parametric study

- $\alpha - EI(x) = \frac{EI(x)}{c(x)^{\alpha}}\ ,\ \alpha \in \mathbb{R}^{+}$
- 161 production type datasets, XGboost for Regression and Classification Tasks
- Low-Variance Cost-Model

A Pareto Front parametric study - Bi-optimization



A Pareto Front parametric study - Optimal Time Allocation



A Contextual Approach

- **Idea:** Identify best Alpha in function of current present context.

Towards Pareto-efficient solutions

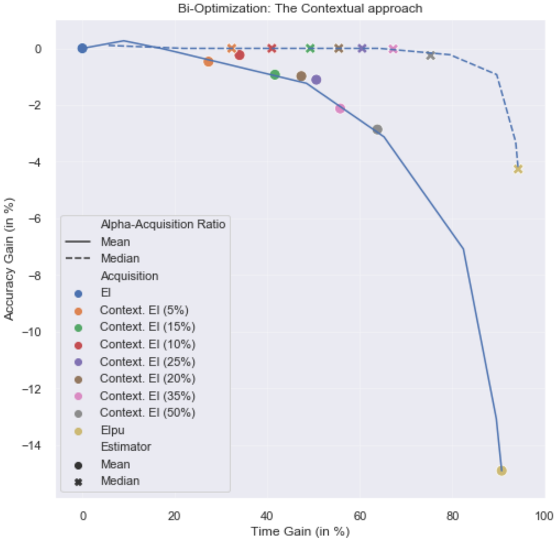
- **Goal:** Dynamic Alpha Allocation
- **Information to leverage:**
  - Past: Performances of Alpha-Acquistion Functions in previous iterations
  - Present: State of the Pareto Front and other type of information (budget)
  - Future: Lookahead, going further than simple greedy allocation (sampling)
  - Offline: Performance on other optimization tasks

A Contextual Approach

- **Idea:** Identify best Alpha in function of current present context.
- **Implementation**:

$$Contextual - EI(x) = \begin{cases} cost(x) & \text{if } EI(x) \geq (1-\lambda) * \max_{x \in \mathcal{X}}(EI(x)) \\ +\infty & \text{sinon.} \end{cases} \quad , \ \lambda \in [0,1] \tag{1}$$

A Contextual Approach - Results



Cost Modeling - Goals

Goals:

- Online: Better online cost-modeling
- Offline: Forward-simulate wall-clock time
- Offline: Budget Forecasting Problem

Online Cost Modeling - Models

- **GP:** Cost $c(\mathrm{x})$ is modeled with a warped GP that fits the log cost $\gamma(\mathrm{x})$. It is then predicted by $c(\mathrm{x}) = \exp(\gamma(\mathrm{x}))$
- **Low-Variance Models (Grey Box setting):** A linear model with low number of features is trained is trained instead of the GP

Conclusion

- Need for clear benchmark and customer use cases
- Context is useful in BO but it's a big challenge to isolate it effect.
- Lot can be done with cost modeling

Thank you!