## **Optimization in Machine Learning**

# Bayesian Optimization Noisy Bayesian Optimization

× 0 0 × × ×



#### Learning goals

- Noisy surrogate modeling
- Noisy acquisition functions
- Final best point

#### **NOISY EVALUATIONS**

In many real-life applications, we cannot access the true function values  $f(\mathbf{x})$  but only a **noisy** version thereof

 $f(\mathbf{x}) + \epsilon(\mathbf{x})$ 

For the sake of simplicity, we assume  $\epsilon(\mathbf{x}) \sim \mathcal{N}\left(0, \sigma_{\epsilon}^2\right)$  for now





#### **NOISY EVALUATIONS**

In many real-life applications, we cannot access the true function values  $f(\mathbf{x})$  but only a **noisy** version thereof

 $f(\mathbf{x}) + \epsilon(\mathbf{x})$ 

For the sake of simplicity, we assume  $\epsilon(\mathbf{x}) \sim \mathcal{N}\left(0, \sigma_{\epsilon}^2\right)$  for now

× 0 0 × 0 × ×

Examples:

- HPO (due to non-deterministic learning algorithm and/or resampling technique)
- Oil drilling optimization (an oil sample is only an estimate)
- Robot gait optimization (velocity of a run of a robot is an estimate of true velocity)

### NOISY EVALUATIONS

This raises the following problems:

• Surrogate modeling: So far we used an interpolating GP that is based on noise-free observations; as a consequence, the variance is modeled as 0

$$s^2(\mathbf{x}^{[i]}) = 0$$

for design points  $(\mathbf{x}^{[i]}, y^{[i]}) \in \mathcal{D}^{[t]}$ . This is problematic.

- Acquisition functions: Most acquisition functions are based on the best observed value *f*<sub>min</sub> so far. If evaluations are noisy, we do not know this value (it is a random variable).
- Final best point: The design point evaluated best is not necessarily the true best point in design (overestimation).

#### SURROGATE MODEL

In case of noisy evaluations, a nugget-effect GP (GP regression) should be used instead of an interpolating GP.

The posterior predictive distribution for a new test point  $\mathbf{x} \in S$  under a GP assuming homoscedastic noise  $(\sigma_{\epsilon}^2)$  is:



$$Y(\mathbf{x}) \mid \mathbf{x}, \mathcal{D}^{[t]} \sim \mathcal{N}\left(\hat{f}(\mathbf{x}), \hat{s}^2(\mathbf{x})\right)$$

with

$$\hat{f}(\mathbf{x}) = k(\mathbf{x})^{\top} (\mathbf{K} + \sigma_{\epsilon}^{2} \mathbf{I}_{t})^{-1} \mathbf{y} \hat{s}^{2}(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x})^{\top} (\mathbf{K} + \sigma_{\epsilon}^{2} \mathbf{I}_{t})^{-1} k(\mathbf{x})$$

#### NOISY ACQUISITION FUNCTIONS: AEI

Augmented Expected Improvement (Huang et al., 2006)

$$a_{\mathsf{AEI}}(\mathbf{x}) = a_{\mathsf{EI}_{f_{\mathsf{min}_*}}}(\mathbf{x}) \left(1 - \frac{\sigma_{\epsilon}}{\sqrt{\hat{\mathbf{s}}^2(\mathbf{x}) + \sigma_{\epsilon}^2}}\right).$$

× × ×

Here,  $a_{\text{El}_{f_{\min}*}}$  denotes the **Expected Improvement with Plugin**. It uses the **effective best solution** as a plugin for the (unknown) best observed value  $f_{\min}$ 

$$f_{\min_*} = \min_{\mathbf{x} \in \{\mathbf{x}^{[1]}, \dots, \mathbf{x}^{[t]}\}} \hat{f}(\mathbf{x}) + c\hat{s}(\mathbf{x}),$$

where c > 0 is a constant that controls the risk aversion.

 $\sigma_{\epsilon}^{\rm 2}$  is the nugget-effect as estimated by the GP regression.

#### NOISY ACQUISITION FUNCTIONS: AEI / 2

In addition, it takes into account the nugget-effect  $\sigma_{\epsilon}^2$  by a penalty term:

$$\left(1 - \frac{\sigma_{\epsilon}}{\sqrt{\hat{s}^2(\mathbf{x}) + \sigma_{\epsilon}^2}}\right)$$

The penalty is justified to "account for the diminishing return of additional replicates as the predictions become more accurate" (*Huang et al., 2006*)





- Designs with small predictive variance \$\hfrac{s}^2(x)\$ are penalized in favor of more exploration.
- If  $\sigma_{\epsilon}^2 = 0$  (noise-free), the AEI corresponds to the EI with plugin.

#### REINTERPOLATION

Clean noise from the model and then apply a general acquisition function (EI, PI, LCB,  $\dots$ )

The RP suggests to build **two models**: a nugget-effect GP (regression model; left) and then, on the predictions from the first model (grey), an interpolating GP (right)

× 0 0 × 0 × ×





#### REINTERPOLATION

#### Algorithm Reinterpolation Procedure

- 1: Build a nugget-effect GP model based on noisy evaluations
- 2: Compute predictions for all points in the design  $\hat{f}(\mathbf{x}^{[1]}), \ldots, \hat{f}(\mathbf{x}^{[t]})$
- 3: Train an interpolating GP on  $\left\{ \left( \mathbf{x}^{[1]}, \hat{f}(\mathbf{x}^{[1]}) \right), \dots, \left( \mathbf{x}^{[t]}, \hat{f}(\mathbf{x}^{[t]}) \right) \right\}$
- 4: Based on the interpolating model, obtain a new candidate using a noise-free acquisition function

× 0 0 × × ×





### **IDENTIFICATION OF FINAL BEST POINT**

Another problem is the identification of a final best point:

- Assume that all evaluations are noisy
- The probability is high that by chance
  - bad points get overrated
  - good points get overlooked



× 0 0 × 0 × ×

### **IDENTIFICATION OF FINAL BEST POINT / 2**

Possibilities to reduce the risk of falsely returning a bad point:

- Return the best predicted point:  $\arg \min_{\mathbf{x} \in {\mathbf{x}^{[1]},...,\mathbf{x}^{[t]}}} \hat{f}(\mathbf{x})$
- Repeated evaluations of the final point: infer guarantees about final point (however if final point is "bad" unclear how to find a better one)
- Repeated evaluations of all design points: reduce noise during optimization and risk of falsely returning a bad point
- More advanced replication strategies, e.g. incumbent strategies: also re-evaluate the "incumbent" in each iteration

0 0 X X 0 X X