Bayesian Optimization in Machine Learning

José Jiménez

Master Thesis
Master's degree in Statistics and Operations Research

supervised by Josep Ginebra[1]
[1] Department of Statistics and Operations Research. ETSEIB.

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

UNIVERSITAT DE
BARCELONA

Goals of this thesis

This Master's thesis aims to be a multi-objective optimization task:

- Provide an introduction to both Gaussian Process regression and Bayesian optimization.
- Show that the Bayesian Optimization framework works in several real-world machine learning tasks.
- Write a complete software package (pyGPGO) for users to apply Bayesian Optimization in their research.

Organization of the work

Organized in 5 self-contained chapters.

- **Chapter 2** focuses on an introduction to regression problems using Gaussian Processes. These are surrogate models we will use for Bayesian Optimization.
- **Chapter 3** covers the main topic in this work, Bayesian Optimization.
- **Chapter 4** presents experiments using the Bayesian Optimization framework. Mostly mid-sized supervised-learning problems.
- **Chapter 5** provides technical explanations for pyGPGO, the software developed alongside this manual.

A brief introduction

Overall, Bayesian Optimization focuses on:

$$\max_{\mathbf{x} \in \mathcal{A}} f(x) \tag{1}$$

We make almost no assumptions about f:

- f may not have a closed-form expression.
- Evaluations of f may be noisy.
- Gradient information is optional.

These situations arise when optimizing the *loss* function of a machine-learning model, depending on its hyperparameters (e.g. log-loss):

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{n} \sum_i \left(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right) \tag{2}$$

Gaussian Process Regression: basic definitions

Definition 1.

A Gaussian Process is a collection of random variables, any finite number of which have a joint Gaussian distribution. This process is defined by two functions. Its *mean function*:

$$m(\mathbf{x}) = \mathbb{E}\left[f(\mathbf{x})\right] \tag{3}$$

and its *covariance function*:

$$k(\mathbf{x}, \mathbf{x'}) = \mathbb{E}\left[\left(f(\mathbf{x}) - m(\mathbf{x})\right)\left(f(\mathbf{x'}) - m(\mathbf{x'})\right)\right] \tag{4}$$

We say that f is a Gaussian Process with mean $m(\mathbf{x})$ and covariance function $k(x, x')$ and write:

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x'})\right) \tag{5}$$

Gaussian Process Regression: basic definitions

Define then a covariance function, such as the *squared exponential* kernel:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}|\mathbf{x} - \mathbf{x}'|^2\right) \tag{6}$$

where $|.|$ denotes the standard L_2 norm. Drawing samples from a Gaussian Process, assuming zero mean, for given finite inputs X_* simplifies to sampling from:

$$\mathbf{f}_* \sim \mathcal{N}(\mathbf{0}, K(X_*, X_*)) \tag{7}$$

Gaussian Process Regression: prediction

Assume training data $\mathcal{D} = \{(\mathbf{x_i}, y_i) \,|i = 1, \ldots, n\}$

Prediction using GP prior

Let $\mathbf{y}$ and $\mathbf{f}_*$ be jointly Gaussian:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{8}$$

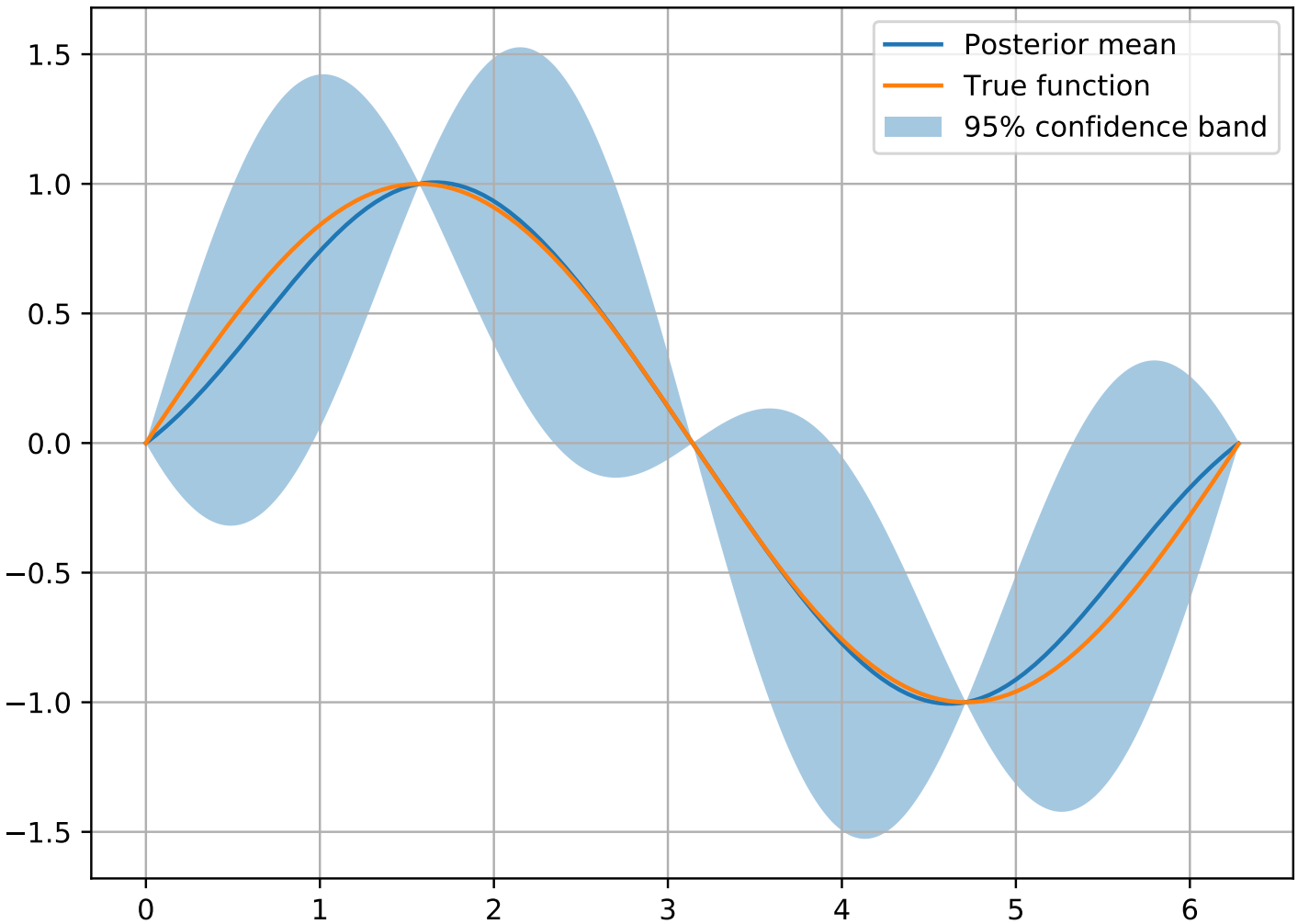
We want to condition $\mathbf{f}_*$ over $\mathbf{y}$.

$$\mathbf{f}_* | \mathbf{y} \sim \mathcal{N}(\overline{\mathbf{f}_*}, Cov(\mathbf{f}_*)) \tag{9}$$

where:

$$\begin{aligned} \overline{\mathbf{f}_*} &= K(X_*,X)\left(K(X,X) + \sigma_n^2 I\right)^{-1} \mathbf{y} \\ Cov(\mathbf{f}_*) &= K(X_*,X_*) - K(X_*,X)\left(K(X,X) + \sigma_n^2 I\right)^{-1} K(X,X_*) \end{aligned} \tag{10}$$

Gaussian Process Regression: an example



Gaussian Process Regression: on covariance functions

Some common covariance function choices

$$k_{SE}(r) = \exp\left(-\frac{r^2}{2l^2}\right) \qquad k_{\text{Matèrn}}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}r}{l}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}r}{l}\right)$$

$$k_{\text{GE}}(r) = \exp\left(-\left(\frac{r}{l}\right)^{\gamma}\right) \qquad k_{RQ}(r) = \left(1 + \frac{r^2}{2\alpha l^2}\right)^{-\alpha} \tag{11}$$

where $r = |\mathbf{x} - \mathbf{x}'|$. Observations are noisy:

$$k^y(\mathbf{x}_p, \mathbf{x}_q) = \sigma_f^2 k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq} \tag{12}$$

Gaussian Process Regression: Type II Maximum-Likelihood

An empirical Bayes approach to choosing hyperparameters. Noticing that $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, K + \sigma_n^2 I)$

Marginal log-likelihood

$$\log p(\mathbf{y}|X) = -\frac{1}{2}\mathbf{y}^T(K + \sigma_n^2 I)^{-1}\mathbf{y} - \frac{1}{2}\log|K + \sigma_n^2 I| - \frac{n}{2}\log 2\pi \quad (13)$$

Derivative w.r.t hyperpameters

$$\frac{\partial}{\partial \theta_j}\log p(\mathbf{y}|X, \boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^T K^{-1}\frac{\partial K}{\partial \theta_j}K^{-1}\mathbf{y} - \frac{1}{2}\mathrm{tr}\left(K^{-1}\frac{\partial K}{\partial \theta_j}\right) \quad (14)$$

Gaussian Process Regression: incorporating diff. priors

We have assumed $m(\mathbf{x}) = 0$ for simplicity. If we dispose of prior knowledge, we can use it

Different prior mean

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}^*)\right) \tag{15}$$

Posterior mean becomes:

$$\mathbf{f}_* = \mathbf{m}(X_*) + k(X_*, X)K^{-1}\left(\mathbf{y} - \mathbf{m}(X)\right) \tag{16}$$

Posterior variance remains unchanged.

Gaussian Process Regression: marginalizing over hyperparameters

The full Bayesian approach does not optimize the marginal likelihood, but integrates the uncertainty of hyperparameters θ into the model, either using MCMC or Variational Inference techniques.

Full Bayesian approach

$$\theta \sim p_h(\theta) \tag{17}$$

$$\mathbf{f} \sim \mathcal{N}(0, \Sigma_\theta) \tag{18}$$

$$\mathcal{L}(\mathbf{f}) = p(\text{data}|\mathbf{f}) = p(\mathbf{y}|\mathbf{f}) \tag{19}$$

We wish to sample from the joint posterior under unknowns:

$$p(\mathbf{f}, \theta|\,\text{data}) \propto \mathcal{L}(\mathbf{f})p(\mathbf{f})p_h(\theta) \tag{20}$$

Several strategies for sampling from latent Gaussian models proposed in main text.

Bayesian Optimization: introduction

Again, assume:

$$\max_{\mathbf{x} \in \mathcal{A}} f(\mathbf{x}) \tag{21}$$

Assume that we have sampled our function f to optimize a small number of times n. Providing us with training data:

$$\mathcal{D}_n = \{\mathbf{x}_i, y_i, i = 1, \ldots, n\}\,. \tag{22}$$

Our steps here are:

1. Fit a Gaussian Process regression model on $\mathcal{D}_n$.
2. Choose the next point to sample, according to an *acquisition function*, depending on GP.
3. Evaluate said point. Augment training data. Repeat.

Bayesian optimization: algorithm

Algorithm 1 Bayesian optimization framework

1: Sample a small number of points $\mathbf{x} \in \mathcal{A}$. Evaluate $f(\mathbf{x})$ to get $\mathcal{D}_n$
2: **for** $n = 1, 2, \ldots$ **do**
3: Fit a GP regression model on $\mathcal{D}_n$
4: $\mathbf{x}_{n+1} \leftarrow \arg\max_{\mathbf{x}} \alpha(\mathbf{x}, \mathcal{D}_n)$
5: Evaluate $f(\mathbf{x}_{n+1}) = y_{n+1}$
6: Augment data $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (\mathbf{x}_{n+1}, y_{n+1})\}$
7: **end for**

Bayesian optimization: on acquisition functions

Three popular categories: improvement-based, optimistic, and information-based policies.

Improvement-based

τ denotes our best evaluation so far.
Probability of improvement:

$$\alpha_{\mathrm{PI}}(\mathbf{x}, \mathcal{D}_n) = P(\nu > \tau) = \Phi\left(\frac{\mu_n(\mathbf{x}) - \tau}{\sigma_n(\mathbf{x})}\right) \tag{23}$$

Expected improvement:

$$\alpha_{\mathrm{EI}}(\mathbf{x}, \mathcal{D}_n) = (\mu_n(\mathbf{x}) - \tau)\Phi\left(\frac{\mu_n(\mathbf{x}) - \tau}{\sigma_n(\mathbf{x})}\right) + \sigma_n(\mathbf{x})\phi\left(\frac{\mu_n(\mathbf{x}) - \tau}{\sigma_n(\mathbf{x})}\right) \tag{24}$$

Bayesian optimization: on acquisition functions

Optimistic acquisitions

Upper confidence bound:

$$\alpha_{\mathrm{UCB}}(\mathbf{x}, \mathcal{D}_n) = \mu_n(\mathbf{x}) + \beta_n \sigma_n(\mathbf{x}) \tag{25}$$

Information-based

Entropy-based:

$$\alpha_{\mathrm{ES}}(\mathbf{x}|\mathcal{D}_n) = H(\mathbf{x}^*|\mathcal{D}_n) - \mathbb{E}_{y|\mathcal{D}_n,\mathbf{x}}\left[H(\mathbf{x}^*|\mathcal{D}_n \cup \{(\mathbf{x}, y)\})\right] \tag{26}$$

Predictive entropy search:

$$\alpha_{\mathrm{PES}}(\mathbf{x}, \mathcal{D}_n) = H(y|\mathcal{D}_n, \mathbf{x}) - \mathbb{E}_{\mathbf{x}^*|\mathcal{D}_n}\left[H(y|\mathcal{D}_n, \mathbf{x}, \mathbf{x}^*)\right] \tag{27}$$