Advanced Machine Learning

Bayesian Optimization

Based on Slides by
Sourish Das and Madhavan Mukund

Chennai Mathematical Institute

2022

Pranabendu Misra

Introduction

Bayesian optimization (BayesOpt) is a class of ML optimization methods focused on solving the problem

$$\max_{\mathbf{x} \in A} f(\mathbf{x}),$$

where

- $\mathbf{x} \in \mathbb{R}^d$, typically $d \leq 20$
- Typically $A = \{\mathbf{x} \in \mathbb{R}^d | a_i \leq x_i \leq b_i\}$ is a hyper-rectangle
- f is expensive to evaluate.

Ex: f is a deep network model and with L many layers and q many nodes in each layer; $L = 2, 3, 4 \cdots$; $q = 2, 3, 4, \cdots$; so $\mathbf{x} = (L, q)$ and $f = RMSE$ in validation dataset

Nature of f

- f is expensive to evaluate – each evaluation may that maybe performed may take substantial amount of time and/or monetory cost (e.g., buying cloud computing power)
- f lacks known special structure like concavity or linearity
- When we evaluate f, we observe on $f(\mathbf{x})$ and no first and second order derivatives available
- so gradient descent type algorithms are not possible
- f is a '**black box**.'
- Goal: Find a global rather than local optimum.

Overview of BayesOpt

- BayesOpt is designed for black-box derivative free global optimization.
- BayesOpt consists of two main components:
  1. Bayesian statistical model for modeling the objective function f
  2. Acquisition function for deciding where to sample next.

Basic pseudo-code for Bayesian optimization

- Place a Gaussian process prior model on f
- Set $n = n_0$, observe f at n_0 different points, i.e., $f(\mathbf{x}_1), f(\mathbf{x}_2), \cdots, f(\mathbf{x}_{n_0})$
- `while` $n \leq N$ `do`
  1. Update the posterior probability distribution on f
  2. Let $\mathbf{x}_n$ be a maximizer of the acquisition α function over $\mathbf{x}$

  Note Acquisition function $\alpha(\mathbf{x})$ is computed using the current posterior distribution.

  3. Observe $y_n = f(\mathbf{x}_n)$
  4. $n = n + 1$
- `end while`
- Return a solution: the point evaluated with the ~~largest~~ smallest $f(\mathbf{x})$

- we have $(x_i, y_i = f(x_i))$ for $i \in \{1, 2, \dots n\}$
- Assume we have a multivariate gaussian dist-

$$[y_1, y_2 \dots y_n] \sim N(\mu, \Sigma)$$

$$\mu = [\mu_1, \dots \mu_n] \qquad \Sigma = \begin{bmatrix} \Sigma_{1,1} & \Sigma_{1,2} & \dots & \Sigma_{1,n} \\ \Sigma_{2,1} & \Sigma_{2,2} & \dots & \Sigma_{2,n} \\ \vdots & & & \\ \Sigma_{n,1} & \Sigma_{n,2} & \dots & \Sigma_{n,n} \end{bmatrix}$$

$\Sigma_{i,j}$ is the Covariance of y_i and y_j

mean

Co-variances

- We assume that f can be approximated by a multivariate gaussian dist.

Gaussian Process $\equiv n \to \infty$

- We choose some suitable initial value for μ_i

mean function $\mu(x_i) = \mu_i$ Typically $\mu_i = 0$

- The choice of the co-variances $\Sigma_{i,j}$ is more important

  - covariance function or Kernel $\underbrace{\Sigma_{i,j} = \Sigma(x_i, x_j)}$

directly compute covariances without evaluating $y_i = f(x_i)$

- Gaussian Kernel

$$\Sigma(x_i, x_j) = a\, e^{-\left(\frac{\|x_i - x_j\|^2}{l^2}\right)}$$

A popular choice

- we have $(x_i, y_i = f(x_i))$ for $i \in \{1, 2, .. n\}$
- Assume we have a multivariate gaussian dist-

$$[y_1, y_2, \dots y_n] \sim N(\mu, \Sigma)$$

→ if we had $y_1 \ y_2 \dots y_{n-1}$ then we get a Prob dist on possible values of y_n

$$y_n \mid [y_1 \cdots y_{n-1}] \sim N(\hat{\mu}_n, \hat{\sigma}^2_n)$$

$$\hat{\mu}_n = [\Sigma_{n,1}, \dots \Sigma_{n,n-1}] \Sigma^{-1} [y_i - \mu_i]_{i=1 \dots n-1} + \mu_n$$

$$\hat{\sigma}^2_n = \Sigma_{n,n} - [\Sigma_{n,1}, \dots, \Sigma_{n,n-1}] \Sigma^{-1} [\Sigma_{1,n}, \dots \Sigma_{n-1,n}]$$

Posterior Prob Distribution

Modeling objective function with GP Regression

- Consider the following

$$\mathbf{y} = f(\mathbf{x}) + \boldsymbol{\epsilon}$$

- Represents $f(\mathbf{x})$ as

$$f(\mathbf{x}) = \sum_{j=1}^{K} \phi_j(\mathbf{x})\beta_j = \boldsymbol{\phi}\boldsymbol{\beta},$$

  we say ϕ is a basis system for $f(\mathbf{x})$, where $\phi_j(\mathbf{x})$ is completely known.

- Problem is $\boldsymbol{\beta}$ is unknown - hence we estimate $\boldsymbol{\beta}$.

Modeling objective function with GP Regression

can be infinite

- We are writing the function with its basis expansion

$$\mathbf{y} = \phi\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

- The basis ϕ is fully known, such as
  - $\phi = \{1, \sin(\omega\mathbf{x}), \cos(\omega\mathbf{x}), \sin(2\omega\mathbf{x}), \cos(2\omega\mathbf{x}) \cdots\}$, ω is known
  - $\phi = \{1, \exp(-\lambda_1(\mathbf{x} - c_1)^2), \exp(-\lambda_2(\mathbf{x} - c_2)^2) \cdots\}$

- Problem is $\boldsymbol{\beta}$ is unknown - hence we estimate $\boldsymbol{\beta}$.

Bayesian method

- Model:

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

$$\epsilon \sim \mathbf{N}(0, \sigma^2\mathbf{I}) \implies \mathbf{y} \sim \mathbf{N}(f(\mathbf{x}), \sigma^2\mathbf{I}),$$

$$f(\mathbf{x}) \quad = \quad \phi\boldsymbol{\beta} = \sum_{k=1}^{K} \phi_k(\mathbf{x})\beta_k + \sum_{k=K+1}^{\infty} \phi_k(\mathbf{x})\beta_k,$$

  where $|\sum_{k=K+1}^{\infty} \phi_k(\mathbf{x})\beta_k| < \epsilon;\ \epsilon \geq 0$

- $\boldsymbol{\beta}$ is unknown and we want to estimate

  Assuming $\boldsymbol{\beta}$'s are uncorrelated random variable and $\phi_k(\mathbf{x})$ are known deterministic real-valued functions.

- Then due to **Kosambi-Karhunen-Loeve** theorem, we can say that $f(\mathbf{x})$ is a random realisation from a stochastic process.

Gaussian Process Prior

- As $f(\mathbf{x})$ is a stochastic process, if we assume $\boldsymbol{\beta} \sim \mathbf{N}(0, \sigma^2 \mathbf{I})$ then $f(\mathbf{x}) = \phi\boldsymbol{\beta}$ follow Gaussian process.
- Since $f(\mathbf{x})$ is unknown function; therefore induced process on $f(\mathbf{x})$ is known as '**Gaussian Process Prior**'.

  Prior on $\boldsymbol{\beta}$:

$$p(\boldsymbol{\beta}) \propto \exp\left(-\frac{1}{2\sigma^2}\boldsymbol{\beta}^T\boldsymbol{\beta}\right)$$

  Induced Prior on $f = \phi\boldsymbol{\beta}$:

$$p(f) \propto \exp\left(-\frac{1}{2\sigma^2}\boldsymbol{\beta}^T\boldsymbol{\phi}^T\mathbf{K}^{-1}\phi\boldsymbol{\beta}\right)$$

Gaussian Process Prior

- The prior mean and covariance of $f(\mathbf{x})$ are given by

$$\mathbf{E}[f(\mathbf{x})] = \phi(\mathbf{x})E[\beta] = \phi\beta_0$$

$$\begin{aligned}\mathbf{cov}[f(\mathbf{x})] &= \mathbf{E}[f(\mathbf{x}).f(\mathbf{x}')^T] = \phi(\mathbf{x}).\mathbf{E}[\beta.\beta^{\mathbf{T}}]\phi(t')^T \\ &= \sigma^2\phi(\mathbf{x}).\phi(\mathbf{x}')^T = \mathbf{K}(\mathbf{x},\mathbf{x}')\end{aligned}$$

$$f(\mathbf{x}) \sim \mathcal{N}_n(\phi(\mathbf{x})\beta_0, \mathbf{K}(\mathbf{x},\mathbf{x}')), \quad \epsilon \sim \mathcal{N}_n(0, \sigma^2\mathbf{I})$$

$$y(\mathbf{x}) \sim \mathcal{N}_n\Big(\phi(\mathbf{x})\beta_0, \mathbf{K}(\mathbf{x},\mathbf{x}') + \sigma^2\mathbf{I}\Big)$$

Gaussian Process Prior

- If $\beta_0 = 0$ then

$$\mathbf{E}[f(\mathbf{x})] = \phi(\mathbf{x})E[\beta] = \phi\beta_0 = 0$$

$$f(\mathbf{x}) \sim \mathcal{N}_n(\mathbf{0}, \mathbf{K}(\mathbf{x}, \mathbf{x}')), \quad \epsilon \sim \mathcal{N}_n(0, \sigma^2\mathbf{I})$$

$$y(\mathbf{x}) \sim \mathcal{N}_n\left(\mathbf{0}, \mathbf{K}(\mathbf{x}, \mathbf{x}') + \sigma^2\mathbf{I}\right)$$

Gaussian Process Regression

- The estimated value of $\mathbf{y}$ for a given $\mathbf{x}_*$ is the mean (expected) value of the functions sampled from from the posterior at that value of $\mathbf{x}_*$.

- Suppose $\mu(\mathbf{x}) = \phi(\mathbf{x})\beta_0 = 0$, then expected value of the estimate at a given $\mathbf{x}_*$ is given by

$$\begin{aligned} \hat{f}(\mathbf{x}_*) &= \mathbf{E}(f(\mathbf{x}_*)|\mathbf{x}, \mathbf{y}) \\ &= \mathbf{K}(\mathbf{x}_*, \mathbf{x}).\underbrace{[\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma^2.\mathbf{I}]^{-1}}_{\text{Matrix of order } n}.\mathbf{y} \end{aligned}$$

- The time complexity of the matrix inversion is $\mathcal{O}(n^3)$

- mean function $\mu(x_i) = \mu_i$
- covariance function or Kernel $\Sigma_{i,j} = \Sigma(x_i, x_j)$
  - Gaussian Kernel

$$\Sigma(x_i, x_j) = a\, e^{-\left(\frac{\|x_i - x_j\|^2}{\ell^2}\right)}$$

A popular choice

- How can we choose the parameters μ, a, ℓ ?

Likelihood Method: Gaussian Process Prior Model

- Data model:

$$\mathbf{y}(\mathbf{x}) \;\sim\; \mathcal{N}_n\Big(\mathbf{0}, \mathbf{K}_{\alpha,\rho}(\mathbf{x},\mathbf{x}') + \sigma^2\mathbf{I}\Big)$$

- Static or Hyperparameters: $\boldsymbol{\theta} = \{\alpha, \rho, \sigma^2\}$
- Likelihood function:

$$f(\boldsymbol{\beta}|\mathbf{y},\phi,\sigma^2) \propto (\sigma^2)^{-p/2}\exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}-f)^T[\mathbf{K}+\sigma^2\mathbf{I}]^{-1}(\mathbf{y}-f)\right)$$

- Negative Log-likelihood function:

$$l(\boldsymbol{\beta}) \propto \frac{1}{2\sigma^2}\mathbf{y}^T[\mathbf{K}+\sigma^2\mathbf{I}]^{-1}\mathbf{y}$$

Gaussian Process Prior Model

- Negative log-posterior:

$$p(\boldsymbol{\beta}) \propto \frac{1}{2\sigma^2}\left(\mathbf{y}^T[\mathbf{K}+\sigma^2\mathbf{I}]^{-1}\mathbf{y}+\boldsymbol{\beta}^T\phi^T\mathbf{K}^{-1}\phi\boldsymbol{\beta}\right)$$

- Hence the induced penalty matrix in the Gaussian process prior is identity matrix

- Still hyperparameters: $\boldsymbol{\theta}=\{\alpha,\rho,\sigma^2\}$ are unknown.

- One can use **optimization** routine to estimate the MLE/MAP.

Pick θ that maximizes the probability of the seen data

Experiment with GP Regression

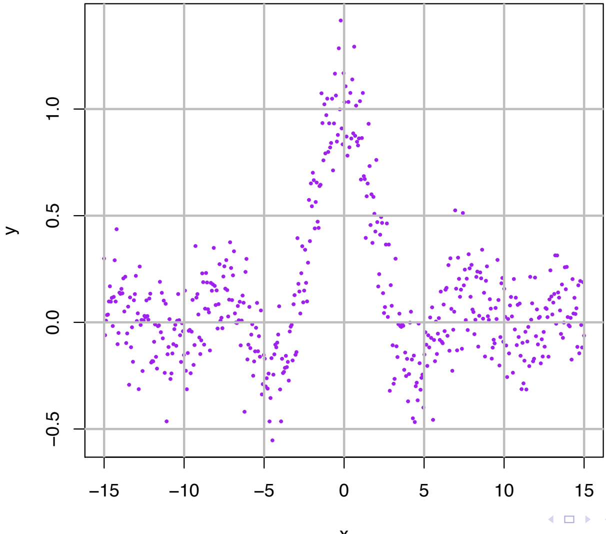
- Model:

$$y = \frac{\sin(x)}{x} + \epsilon,$$

  where $\epsilon \sim N(0, \tau)$.
- Simulate data from the above model and pretend we don't know the true function.
- **Objective** is to estimate/learn the function.

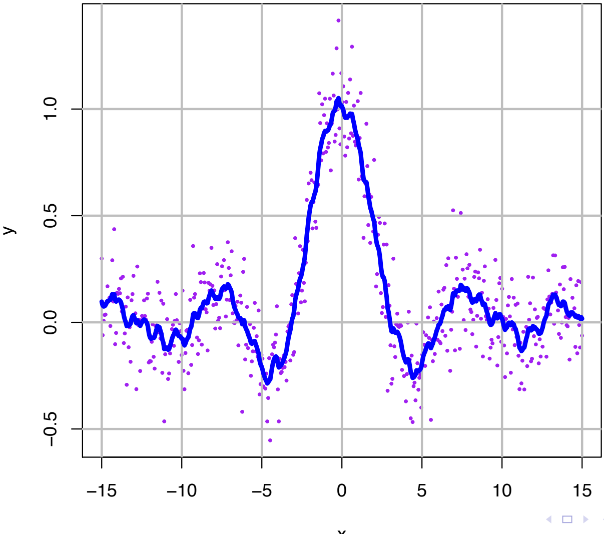
Experiment with GP Regression

Objective is to estimate/learn the function.



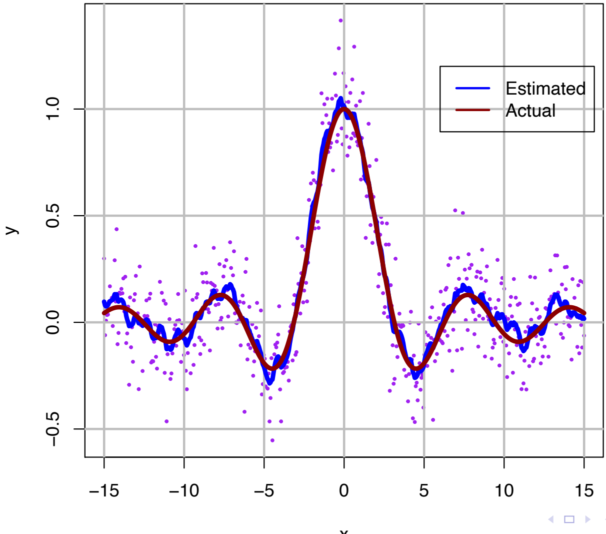
Experiment with GP Regression

Objective is to estimate/learn the function.



Experiment with GP Regression

Objective is to estimate/learn the function.



Back to BayesOpt

Obs: Only need to model near the optimum, not everywhere

- As we modeled the objective function f by $\hat{f}$
- With f we try to predict the performance of the deep network model for a possible choices of hyper-parameter $\mathbf{x}$.
- Next we model the acusition function which recomend where will be the next point of hyper-parameter will be

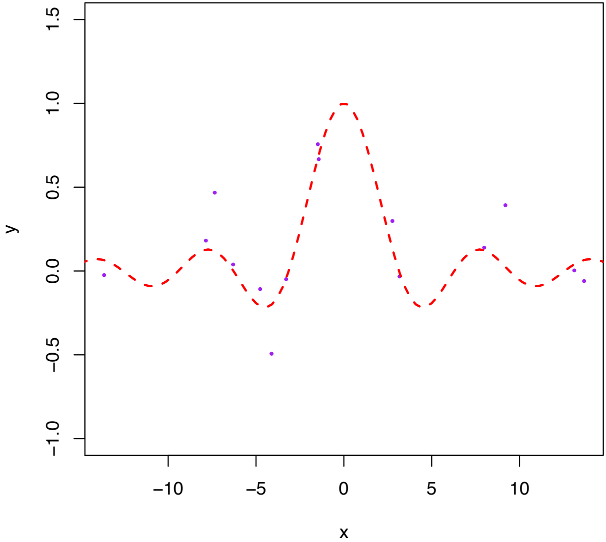
where to sample $f(x)$ next

- One can use the $\hat{f}$ directly as acquisition function or one can sample the acquisition function $\alpha(\mathbf{x})$ from the posterion distribution of f, i.e.,

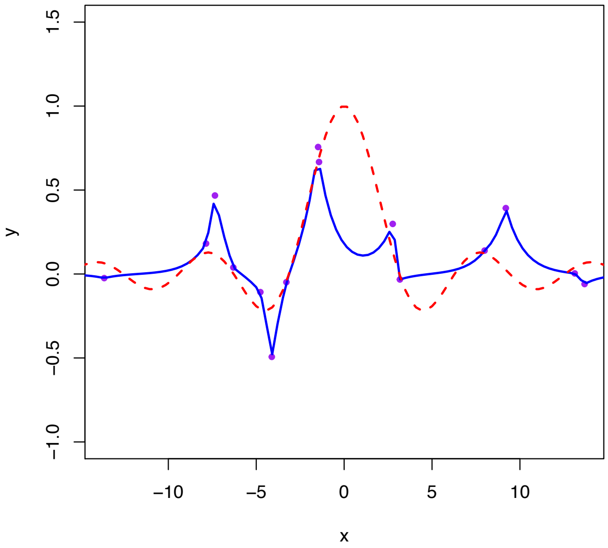
$$\alpha(\mathbf{x}) \sim \mathcal{N}(\hat{f}, cov(\hat{f}))$$

Bayesian Optimization: First Iteration (maximization)

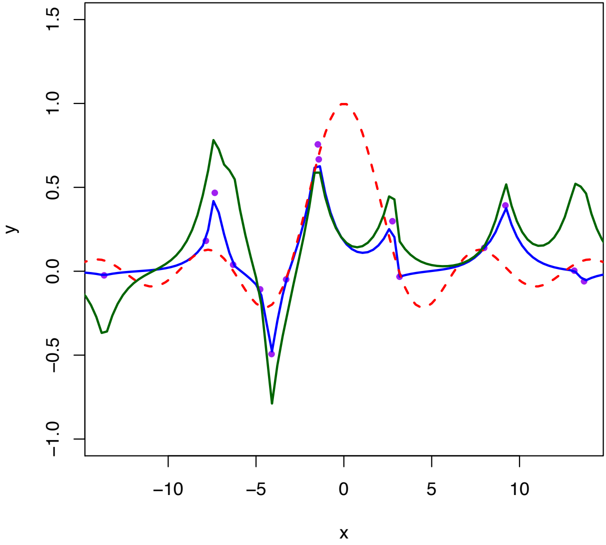
Noisy data



Bayesian Optimization: First Iteration

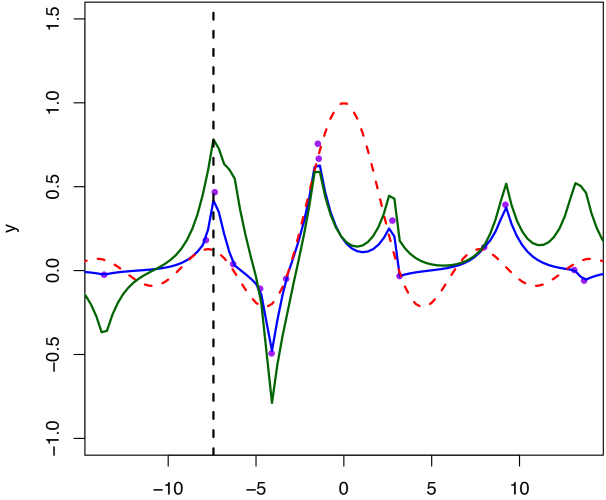


Bayesian Optimization: First Iteration



Bayesian Optimization: First Iteration

```
[1] -7.424242
```



Bayesian Optimization: Iteration = 50

```
[1] 0.2705411
```