Navigating the Unknown: Intelligent and Efficient Solution Discovery via Bayesian Optimization

Wang MA

24 Summer Seminar - Week 1

maw6@rpi.edu

July 22, 2024

Contents

Problem Setup

Bayesian Optimization is a class of machine-learning-based optimization methods focused on sloving the problem

$$\hat{x} = \max_{x \in A} f(x), \tag{1}$$

where the feasible set and objective function typically follow some assumptions/properties.

- The input x is in $\mathbb{R}^d$, where typically $d \leq 20$.
- The feasible set A is a simple set, e.g., box constraints (hyper-rectangle) or a simplex.
- f is continuous but lacks special structure, e.g., non-convex.
- f is derivative-free, evaluations do not give gradient info.
- f is expensive to evaluate.
- f may be noisy.

Scenarios Suitable for Bayesian Optimization

1. **Expensive function evaluations**: the objective function is costly to evaluate, such as requiring significant computational resources or time
2. Black-box functions: the objective function is a black-box with no explicit analytical form and no available gradient information.
3. Hyperparameter Optimization: hyperparameter tuning in machine learning models, such as optimizing learning rates, regularization parameters, etc..

Optimization of expensive functions ariese in

- fitting machine learning models
- tuning algorithms via backtesting
- optimizing physics-based models
- drug and materials discovery

Contents

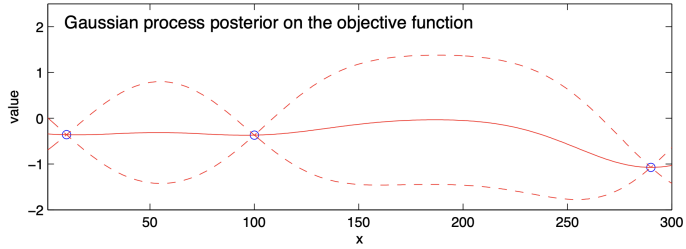
Bayesian Optimization

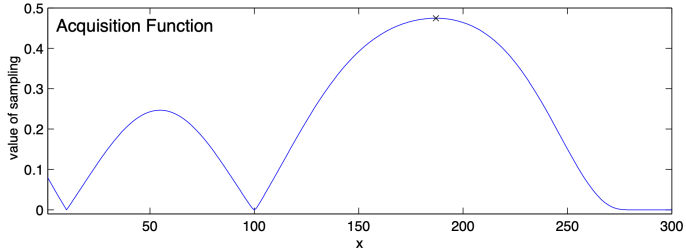
Algorithm 1 BayesOpt

Assume a Bayesian prior on f
(usually a Gaussian process prior, a **probabilistic model** of the function)

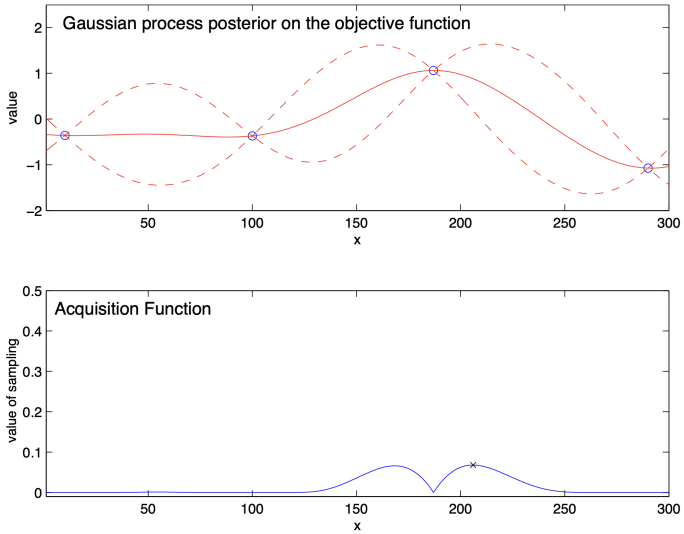
while *budget is not exhausted* **do**
- Find x that maximizes **acquisition function** $(x, \textit{posterior})$
- Sample x and observe $f(x)$
- Update the posterior distribution on f

A 1-dim Example to Bayesian Optimization

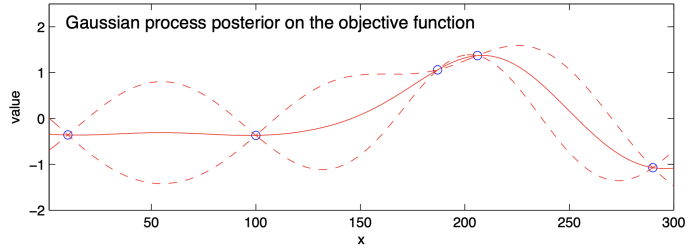


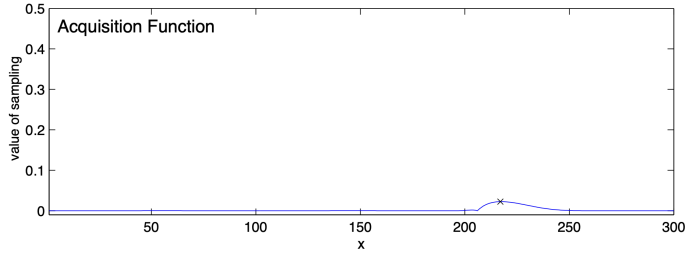


A 1-dim Example to Bayesian Optimization

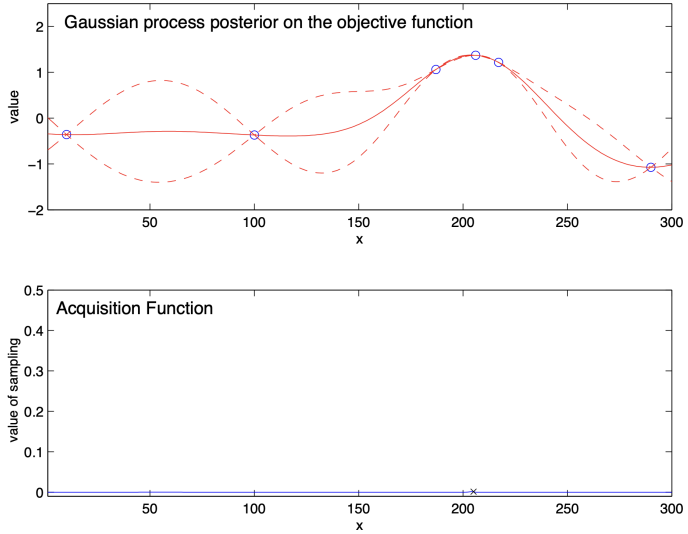


A 1-dim Example to Bayesian Optimization





A 1-dim Example to Bayesian Optimization



Contents

Gaussian Process: Definition

A GP is fully specified by its mean function $m(x)$ and covariance function $k(x, x')$. When we model our function as $f(x) \sim GP(m(x), k(x, x')$, we are saying that

- Mean function: $m(x) = \mathbb{E}[f(x)]$
- Covariance function: $k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$

The mean function $m(x)$ often assumed to be constant or zero for simplicity, say, $m(x) = 0$. Covariance functions (kernels) decreases with $|x - x'||$, commonly used $k(x, x')$:

- Squared Exponential (RBF): $k(x, x') = \sigma_f^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)$
- Matern: $k(x, x') = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}|x-x'|}{\ell}\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}|x-x'|}{\ell}\right)$
- Rational Quadratic: $k(x, x') = \sigma_f^2 \left(1 + \frac{(x-x')^2}{2\alpha\ell^2}\right)^{-\alpha}$

Gaussian Process

Gaussian Process

A Gaussian Process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

As a concrete example, let's choose:

$$m(x) = 0 \tag{2}$$

$$k(x, x') = \exp(-\frac{1}{2}(x - x')^T(x - x')). \tag{3}$$

Given observations $\mathbf{f} = [f(x_1), f(x_2), ..., f(x_t)]$, we would like to make a **prediction** at a new point x^*. According to the GP prior, $f(x^*)$ is jointly normally distributed with $\mathbf{f}$ so that

$$\Pr\left(\begin{bmatrix} \boldsymbol{f} \\ f^* \end{bmatrix}\right) = \text{Norm}\left(\mathbf{0}, \begin{bmatrix} \boldsymbol{K}[\boldsymbol{X}, \boldsymbol{X}] & \boldsymbol{K}[\boldsymbol{X}, \boldsymbol{x}^*] \\ \boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{X}] & \boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{x}^*] \end{bmatrix}\right). \tag{4}$$

Gaussian Process: Predicting

Given the jointly normal property, we have that

$$\Pr(\boldsymbol{f}^*|\boldsymbol{f}) = \text{Norm}(\mu[\boldsymbol{x}^*], \sigma^2[\boldsymbol{x}^*]), \tag{5}$$

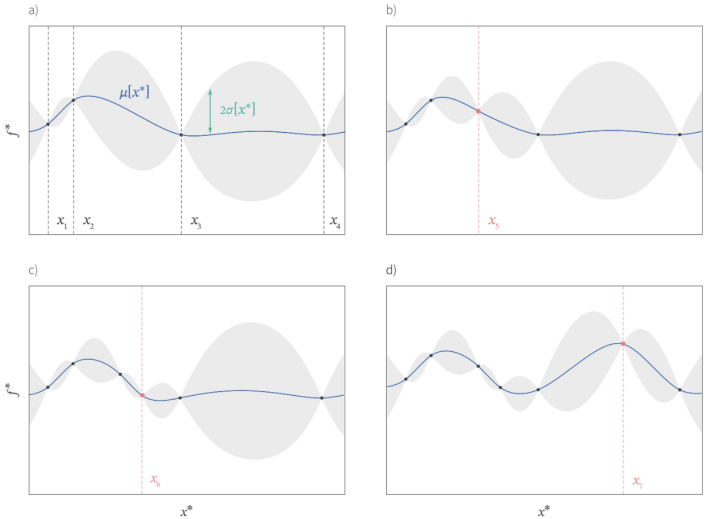
where

$$\mu[\boldsymbol{x}^*] = \boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{X}]\boldsymbol{K}[\boldsymbol{X}, \boldsymbol{X}]^{-1}\boldsymbol{f} \tag{6}$$

$$\sigma^2[\boldsymbol{x}^*] = \boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{x}^*] - \boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{X}]\boldsymbol{K}[\boldsymbol{X}, \boldsymbol{X}]^{-1}\boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{x}^*]. \tag{7}$$

Using above formula, we can estimate the distribution at any new point $\boldsymbol{x}^*$. The best estimate of the funciton value is given by the mean $\mu[x]$, and the uncertainty is given by the variance $\sigma^2[x]$.

Gaussian Process Model



Contents

Acquisition Functions

- Acquisition functions guide **the selection of the next point** to evaluate by balancing exploration and exploitation.

- Common acquisition functions include:
  - Probability of Improvement (PI)
  - Expected Improvement (EI)
  - Upper Confidence Bound (UCB)

Upper Confidence Bound (UCB)

- UCB selects points based on an upper confidence bound on the surrogate model's predictions.
- Mathematically defined as:

$$\text{UCB}[x^*] = \mu[x^*] + \kappa\sigma[x^*], \tag{8}$$

  where κ is a parameter that controls the trade-off between exploration and exploitation.
- This favors either (i) regions where is $\mu[x^*]$ large (for exploitation) or (ii) regions where $\sigma[x^*]$ is large (for exploration). The positive parameter κ trades off these two tendencies.

Probability of Improvement (PI)

- PI aims to maximize the probability that the next sample will improve over the current maximum.
- Mathematically defined as:

$$\mathrm{PI}[x^*] = \int_{f[\hat{x}]}^{\infty} \mathrm{Norm}_{f[x^*]}[\mu[x^*], \sigma[x^*]] df[x^*], \tag{9}$$

  where $f[\hat{x}]$ is the current maximum.

This acquisition function computes the likelihood that the function at x^* will return a result higher than current maximum. For each point x^*, we integrate the part of the associated normal distribution that is above the current maximum.

Expected Improvement (EI)

- EI balances exploration and exploitation by considering both the magnitude and the probability of improvement.
- Mathematically defined as:

$$\mathrm{EI}[x^*] = \mathrm{E}_n[(f[x^*] - f[\hat{x}])^+] \tag{10}$$

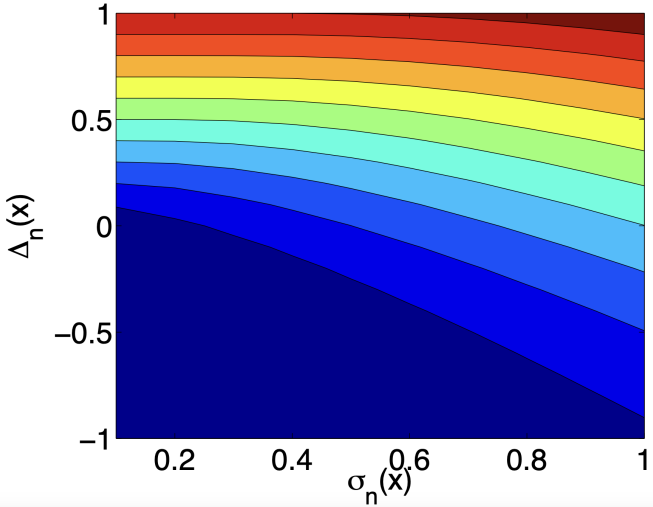
$$= \int_{f[\hat{x}]}^{\infty} (f[x^*] - f[\hat{x}])\mathrm{Norm}_{f[x^*]}[\mu[x^*], \sigma[x^*]]df[x^*]. \tag{11}$$

Expected Improvement (EI) takes into account how much the improvement will be, so that we can find the favorable larger improvement. A closed form of EI:

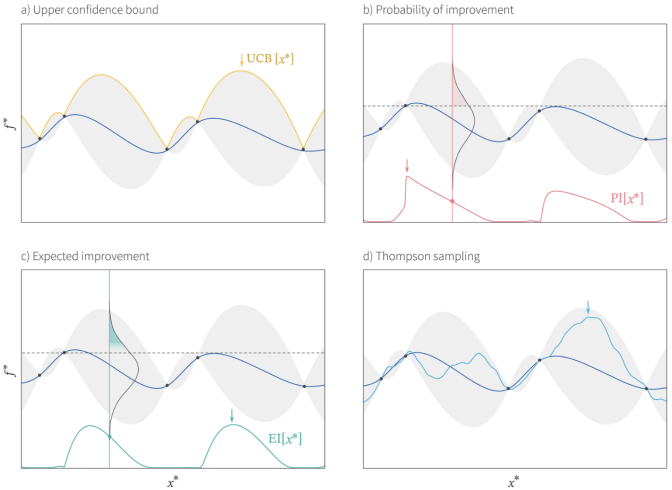
$$\mathrm{EI}[x^*] = [\Delta(x^*)]^+ + \sigma(x^*)\varphi\left(\frac{\Delta(x^*)}{\sigma(x^*)}\right) - |\Delta(x^*)|\Phi\left(-\frac{|\Delta(x^*)|}{\sigma(x^*)}\right), \tag{12}$$

where $\Delta(x^*) = \mu(x^*) - f[\hat{x}]$.

Expected Improvement: Exploration ($\sigma[x^*]$) vs. Exploitation ($\Delta[x^*]$)



Acqusition Functions: Where should we sample next?



Parallel Expected Improvement: Setting

We can parallelize EI:

$$\mathrm{EI}[x_{1:q}] = \mathbb{E}_n[(\max(f[x_1], ..., f[x_q]) - f[\hat{x}])^+] \tag{13}$$

How to maximize the parallel expected improvement?

1. Construct an unbiased estimator of the gradient of $\mathrm{EI}[x_{1:q}]$.
2. Use **multistart stochastic gradient ascent** to appoximately maximize $\mathrm{EI}[x_{1:q}]$.

PEI: Esitimate ∇EI

Here's how we estimate ∇EI:

- $Y = [f[x_1], ..., f[x_q]]$ is multivariate normal (GP prior)
- Let $m = \mathbf{E}[Y]$ and $C = \text{Chol}(\text{Cov}[Y])$
- then $Y = m + CZ$, where Z is a vector of independent standard normals
- $\text{EI}[x_{1:q}] = \mathbf{E}[h(Y)]$, where $h(Y) = (\max[Y] - f[\hat{x}])^+$
- Assume the problem is well-behaved, then we can switch derivative and expectation:

$$\nabla \text{EI}[x_{1:q}] = \mathbb{E}[\nabla h(m + CZ)]$$

PEI: Esitimate ∇EI

Here's how we estimate ∇EI:

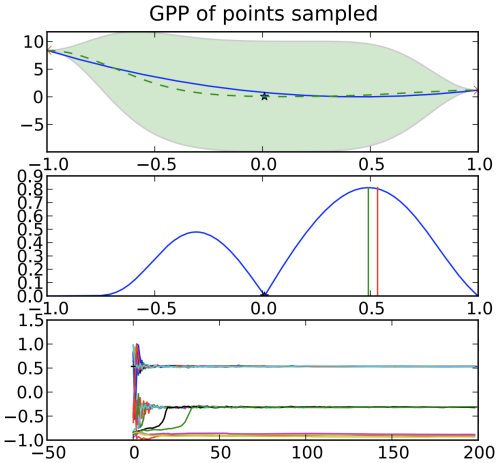
- Simulate a vector Z of independent standard normals
- Calculate $m = \mathbb{E}[Y]$ and $C = \text{Chol}(\text{Cov}[Y])$
- Then the estimator of $\nabla\text{EI}[x_1, ..., x_q]$ is $\nabla h(m + CZ)$, where $h(Y) = (\max[Y] - f[\hat{x}])^+$

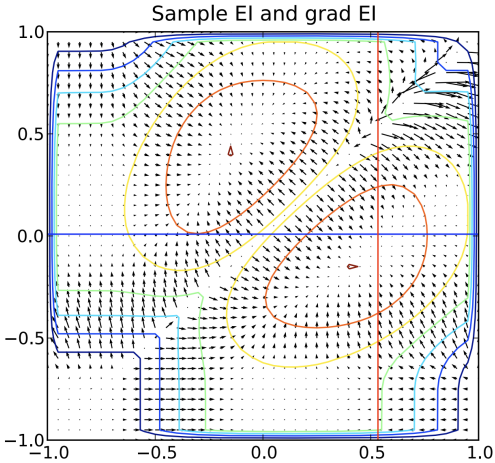
Approximately Maximize $\mathrm{EI}[x_{1:q}]$

We use the previous estimator of $\nabla \mathrm{EI}$ in multistart stochastic gradient ascent:

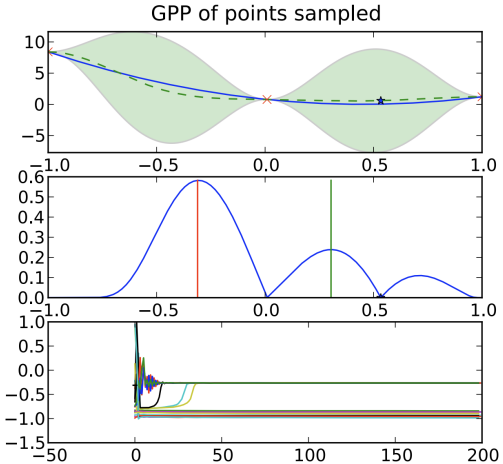
1. Select several starting points, uniformly at random
2. From each starting point, iterate using the stochastic gradient method until convregence.
$$(\vec{x}_1, ..., \vec{x}_q) \longleftarrow (\vec{x}_1, ..., \vec{x}_q) + \alpha_n g(\vec{x}_1, ..., \vec{x}_q, \omega),$$
where (α_n) is a stepsize sequence,
3. For each starting point, average the iterates to get an estimated stationary point. (Polyak-Ruppert averaging)
4. Select the estimated stationary point with the best estimated value as the solution.

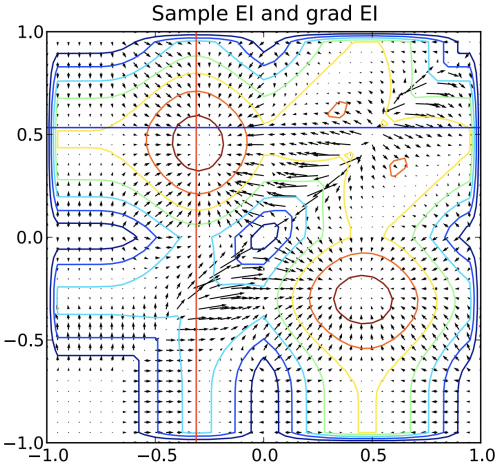
Parallel Expected Improvement: Illustration



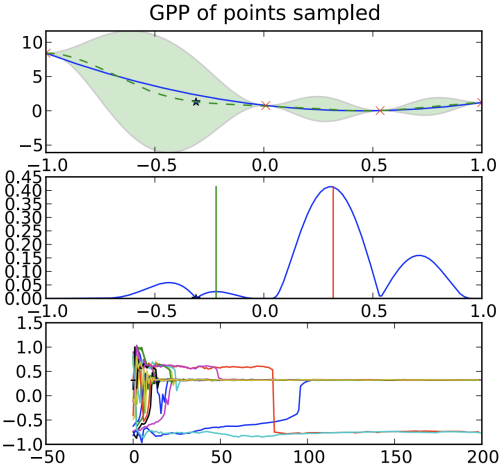


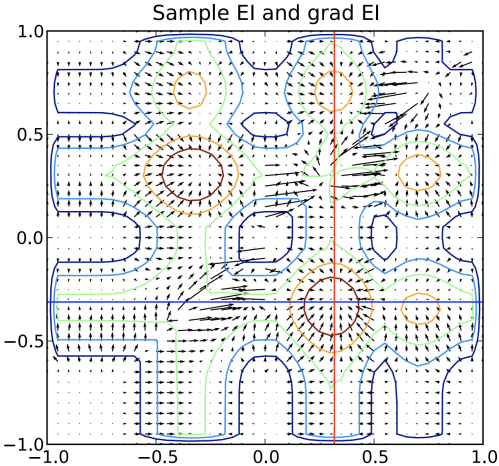
Parallel Expected Improvement: Illustration





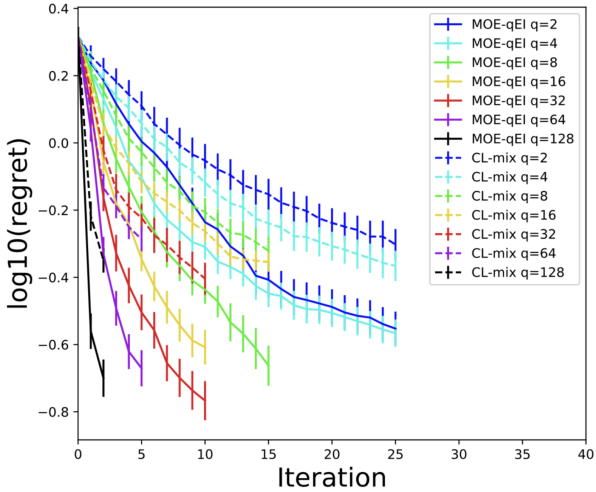
Parallel Expected Improvement: Illustration





Parallel Expected Improvement: Illustration

Here's a demonstration on a 6-dimensional Bayesian Optimization problem with up to 128 parallel evaluations.



Knowledge Gradient

- KG measures the expected increase in the maximum value of the objective function after a new observation.
- We will not select the previously evaluated point as the final solution: the exploration can be ”unsuccessful”
- Formally, for a candidate point x, KG is defined as:

$$KG(x) = \mathbb{E}[\mu^*_{n+1} - \mu^*_n | \text{observing} x_{n+1}]$$

- Here, $\mu^*_n = \max_x \mu_n[x] = \max_x \mathbb{E}_n[f(x)]$ is the best expected value of our ovjective under the posterior at tiem step n.

Knowledge Gradient: Formula

- Let $\mu(x)$ and $\sigma(x)$ be the posterior mean and standard deviation of the objective function at x.

- The KG acquisition function can be expressed as:

$$KG(x) = (\mu(x) - \mu_n^*)\Phi\left(\frac{\Delta(x)}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{\Delta(x)}{\sigma(x)}\right)$$

  Where:
  - μ_n^* is the current best observed value.
  - Φ is the cumulative distribution function of the standard normal distribution.
  - ϕ is the probability density function of the standard normal distribution.

Our approach for optimizing parallel EI also works for optimizing KG

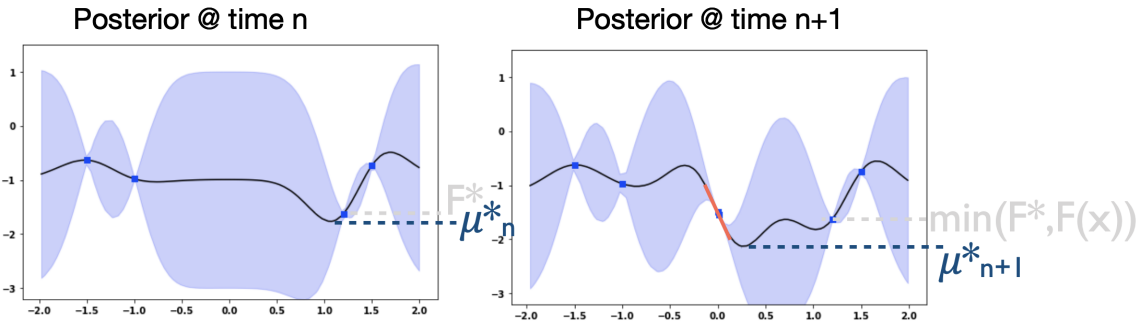
1. Estimate $\nabla \mathrm{KG}(\mathbf{x}_{1:q})$ **using infinitesimal perturbation analysis (IPA)** & the envelope theorem:

$$\nabla \mathrm{KG} = \nabla \mu_{n+1}(x^*; m(x_{1:q}) + CZ, x_{1:q}),$$

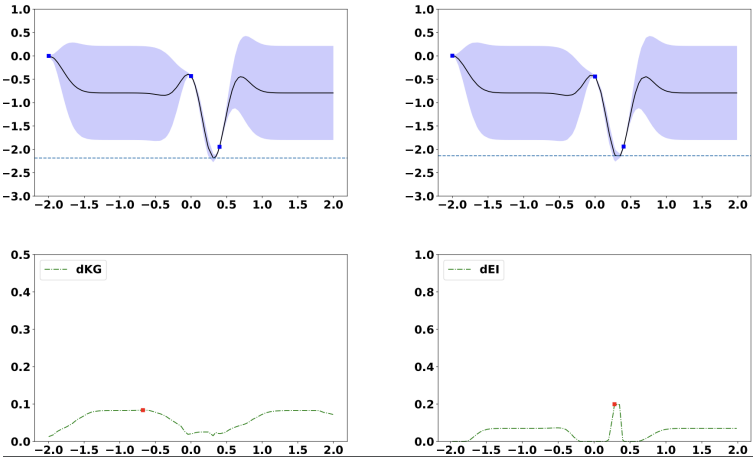
   calculating the gradient holding x^* fixed.

2. Use multistart stochastic gradient ascent to maximize $\mathrm{KG}(\mathbf{x}_{1:q})$

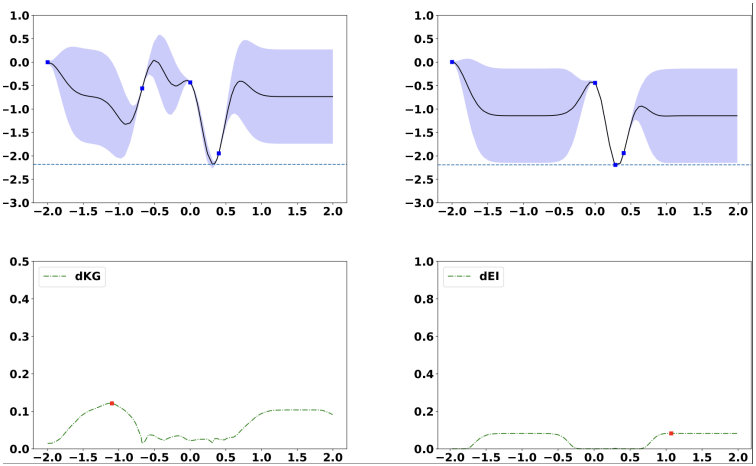
EI May Make Poor Decisions



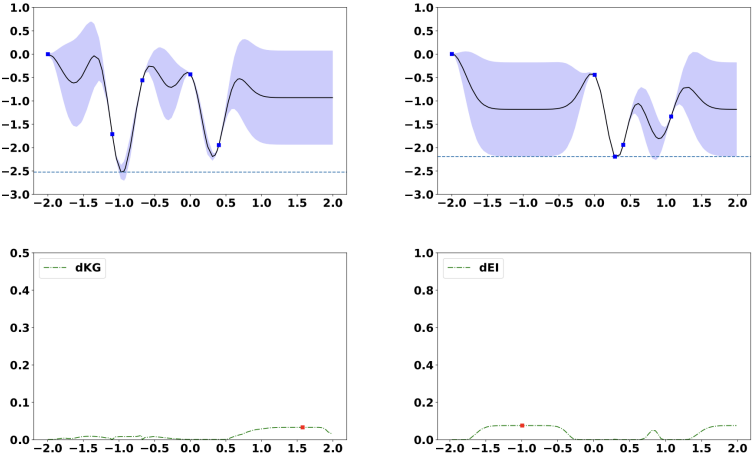
Knowledge Gradient is Efficient



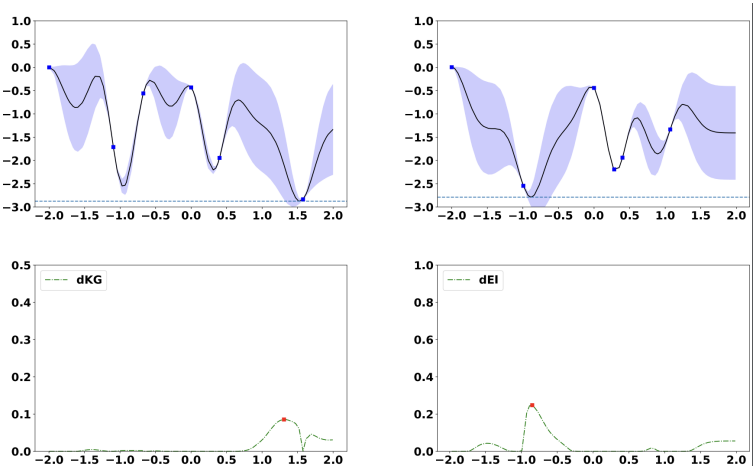
Knowledge Gradient is Efficient



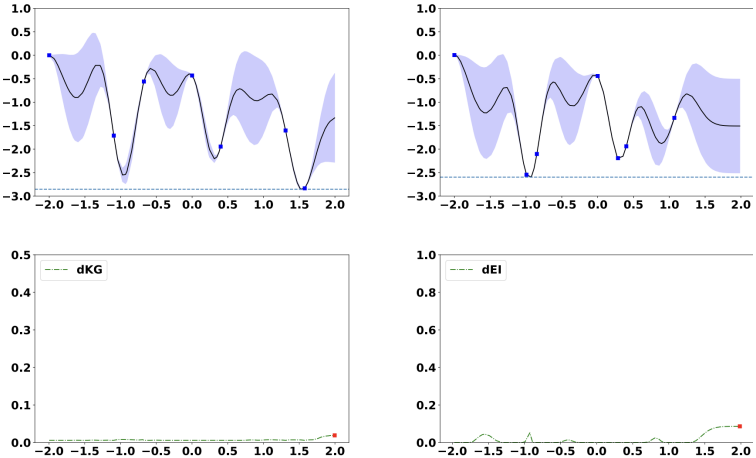
Knowledge Gradient is Efficient



Knowledge Gradient is Efficient



Knowledge Gradient is Efficient



Incorporating Noisy Measurements

We add an extra noise term to the expression for the Gaussian process covariance

$$\mathbb{E}[(y(x) - m(x))(y(x') - m(x'))] = k(x, x') + \sigma_n^2, \tag{14}$$

where $y(x) = f(x) + \epsilon$ is a noisy observation. And the joint normal becomes

$$\Pr\left(\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}^* \end{bmatrix}\right) = \text{Norm}\left(\mathbf{0}, \begin{bmatrix} \boldsymbol{K}[\boldsymbol{X}, \boldsymbol{X}] + \sigma_n^2 I & \boldsymbol{K}[\boldsymbol{X}, \boldsymbol{x}^*] \\ \boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{X}] & \boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{x}^*] \end{bmatrix}\right). \tag{15}$$

Then We have the conditional distribution with noise:

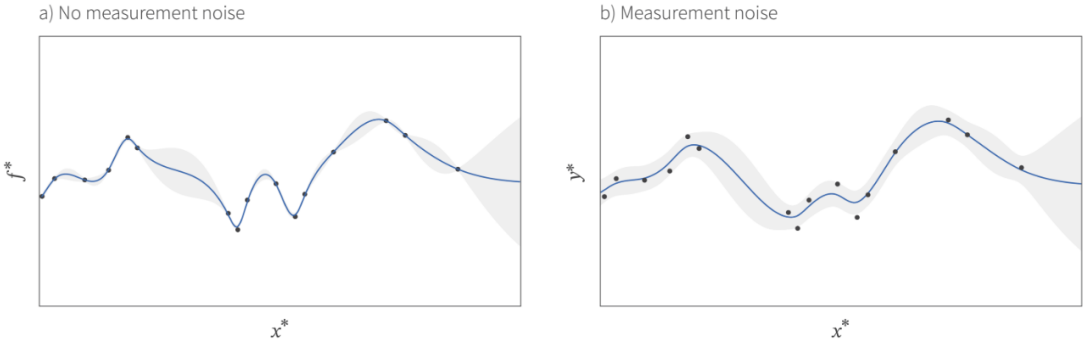
$$\Pr(\boldsymbol{f}^* | \boldsymbol{f}) = \text{Norm}(\mu[\boldsymbol{x}^*], \sigma^2[\boldsymbol{x}^*]), \tag{16}$$

where

$$\mu[\boldsymbol{x}^*] = \boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{X}] \left[\boldsymbol{K}[\boldsymbol{X}, \boldsymbol{X}] + \sigma_n^2 I\right]^{-1} \boldsymbol{f} \tag{17}$$

$$\sigma^2[\boldsymbol{x}^*] = \boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{x}^*] - \boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{X}] \left[\boldsymbol{K}[\boldsymbol{X}, \boldsymbol{X}] + \sigma_n^2 I\right]^{-1} \boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{x}^*]. \tag{18}$$

Noisy Measurements Illustration



Further Reading

- Learning GP parameters: MLE, full Bayesian Approach
- Different Probabilistic Models: Random Forest (SMAC)
- Discrete Variables: Beta-Bernoulli Bandit
- Kernel Choices
- Tips, tricks, and limitations
  - Inducing points
  - Decomposing the kernel
  - Using random projections

Contents

Conclusion: Bayesian Optimization

Algorithm 2 BayesOpt

Assume a Bayesian prior on f
(usually a Gaussian process prior, a **probabilistic model** of the function)

while *budget is not exhausted* **do**
- Find x that maximizes **acquisition function** $(x, \textit{posterior})$
- Sample x and observe $f(x)$
- Update the posterior distribution on f

Conclusion: Gaussian Process Regression

$$\Pr(\boldsymbol{f}^*|\boldsymbol{f}) = \text{Norm}(\mu[\boldsymbol{x}^*], \sigma^2[\boldsymbol{x}^*]), \tag{19}$$

where

$$\mu[\boldsymbol{x}^*] = \boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{X}]\boldsymbol{K}[\boldsymbol{X}, \boldsymbol{X}]^{-1}\boldsymbol{f} \tag{20}$$

$$\sigma^2[\boldsymbol{x}^*] = \boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{x}^*] - \boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{X}]\boldsymbol{K}[\boldsymbol{X}, \boldsymbol{X}]^{-1}\boldsymbol{K}[\boldsymbol{x}^*, \boldsymbol{x}^*]. \tag{21}$$

Conclusion: Acquisition Functions

- Acquisition functions guide **the selection of the next point** to evaluate by balancing exploration and exploitation.

- Common acquisition functions include:
  - Probability of Improvement (PI)

$$\mathrm{PI}[x^*] = \int_{f[\hat{x}]}^{\infty} \mathrm{Norm}_{f[x^*]}[\mu[x^*], \sigma[x^*]] df[x^*], \tag{22}$$

  - Expected Improvement (EI)

$$\mathrm{EI}[x^*] = \mathrm{E}_n[(f[x^*] - f[\hat{x}])^+] \tag{23}$$

$$= \int_{f[\hat{x}]}^{\infty} (f[x^*] - f[\hat{x}]) \mathrm{Norm}_{f[x^*]}[\mu[x^*], \sigma[x^*]] df[x^*]. \tag{24}$$

  - Upper Confidence Bound (UCB)

$$\mathrm{UCB}[x^*] = \mu[x^*] + \kappa\sigma[x^*], \tag{25}$$

Review

1. Introduction
2. An Overview of Bayesian Optimization
3. A model of the function: Gaussian Process
4. Acquisition Functions
5. Conclusion

Bayesian Optimization: END

Thank you!

Questions and Opinions are Welcome!