A Tutorial on Bayesian Optimization

Darian Nwankwo

Introduction

Problem motivation

We're focused on solving problems of the form

$$x^* = \underset{x \in \mathcal{X}}{\text{argmax}}\, f(x)$$

where f is a black-box.

Problem motivation

We're focused on solving problems of the form

$$x^* = \underset{x \in \mathcal{X}}{\operatorname{argmax}}\, f(x)$$

where f is a black-box. Subject to:

- evaluation is expensive

Problem motivation

We're focused on solving problems of the form

$$x^* = \underset{x \in \mathcal{X}}{\operatorname{argmax}}\, f(x)$$

where f is a black-box. Subject to:

- evaluation is expensive
- observations may be noisy

Problem motivation

We're focused on solving problems of the form

$$x^* = \underset{x \in \mathcal{X}}{\operatorname{argmax}}\, f(x)$$

where f is a black-box. Subject to:

- evaluation is expensive
- observations may be noisy
- may only observe the function value (no gradients)

Real-world examples

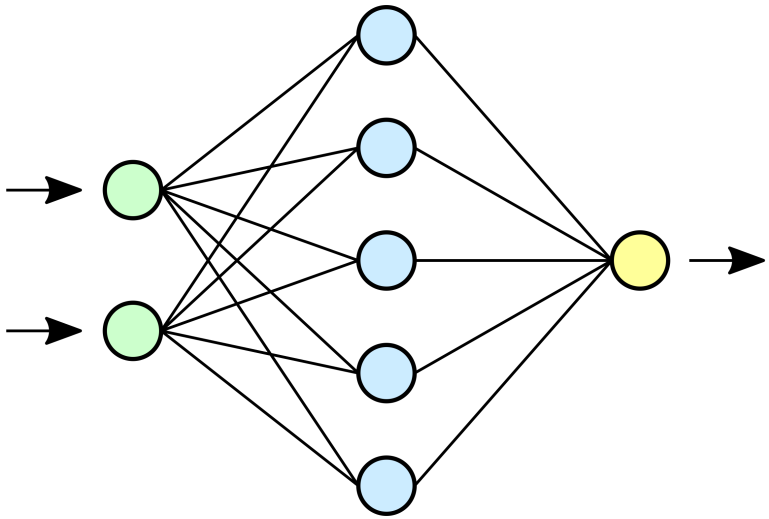


Figure: Neural Network

Hyperparameter configuration for deep neural networks

Real-world examples

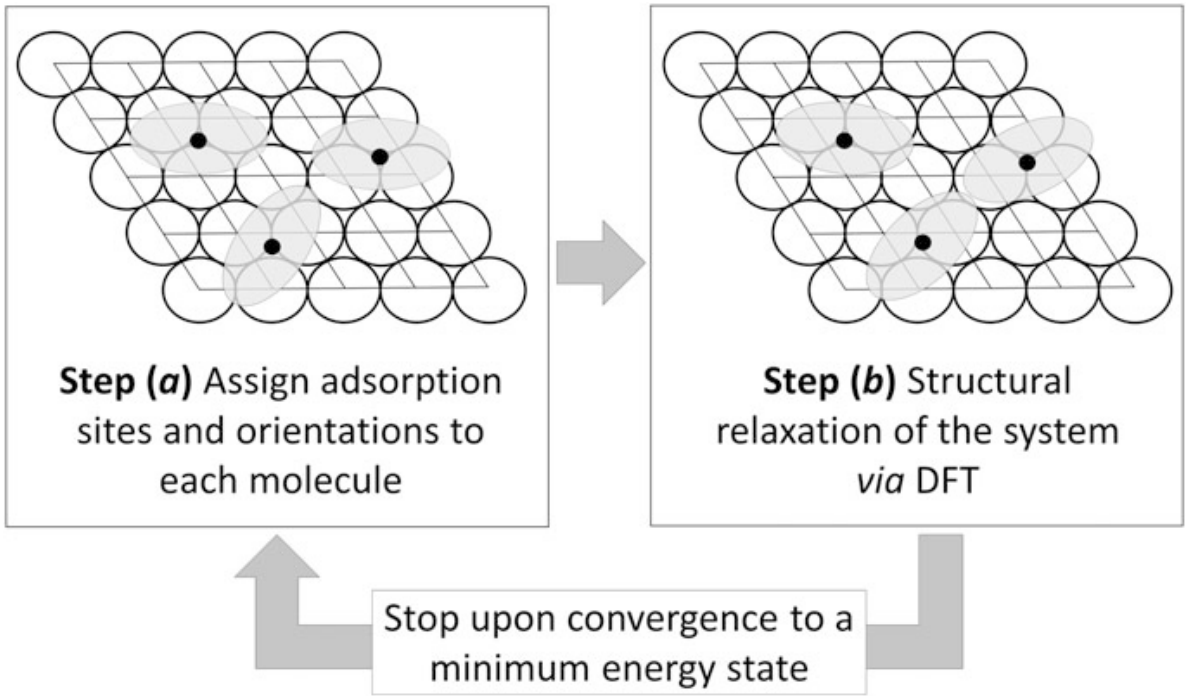


Figure: Conformation of molecules absorbed to a surface

Optimization of absorption sites and orientations for molecules in material science

Real-world examples

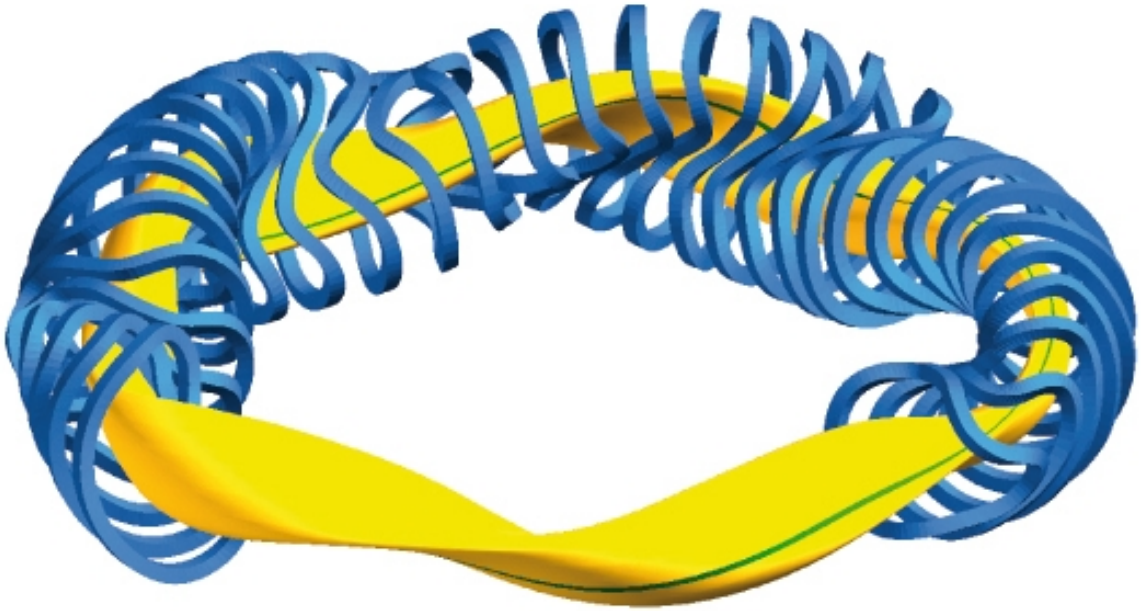
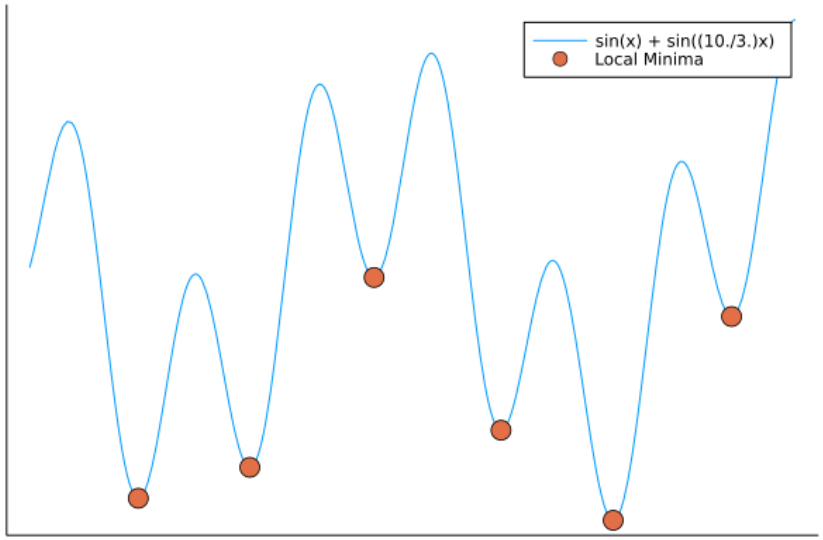


Figure: Stellarator Configuration

Optimal geometry of stellarators in plasma physics for fusion energy

Local and Global Optimization

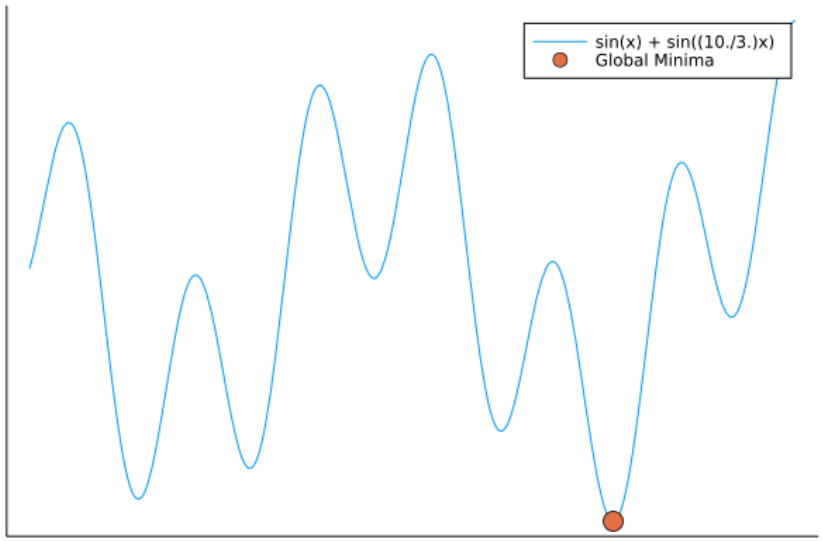
Local optimization



The point x^* is a strong local minimum of F if there exists $\delta > 0$ such that

- $F(x)$ is defined on $N(x^*, \delta)$
- $F(x^*) < F(y)$ for all $y \in N(x^*, \delta)$, $y \neq x^*$

Global optimization



The point x^* is a global minimum of F if

- $F(x^*) < F(y)$ for all $y \in D$, $y \neq x^*$

Recap and takeaways

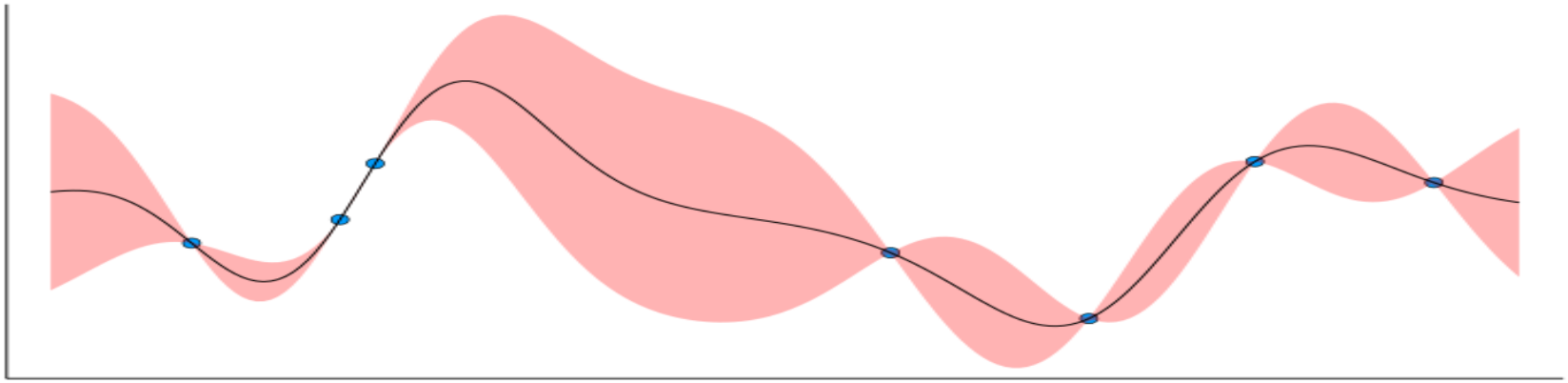
- Characterizing local optima is easy

Recap and takeaways

- Characterizing local optima is easy
- Characterizing global optima is harder

Gaussian Processes

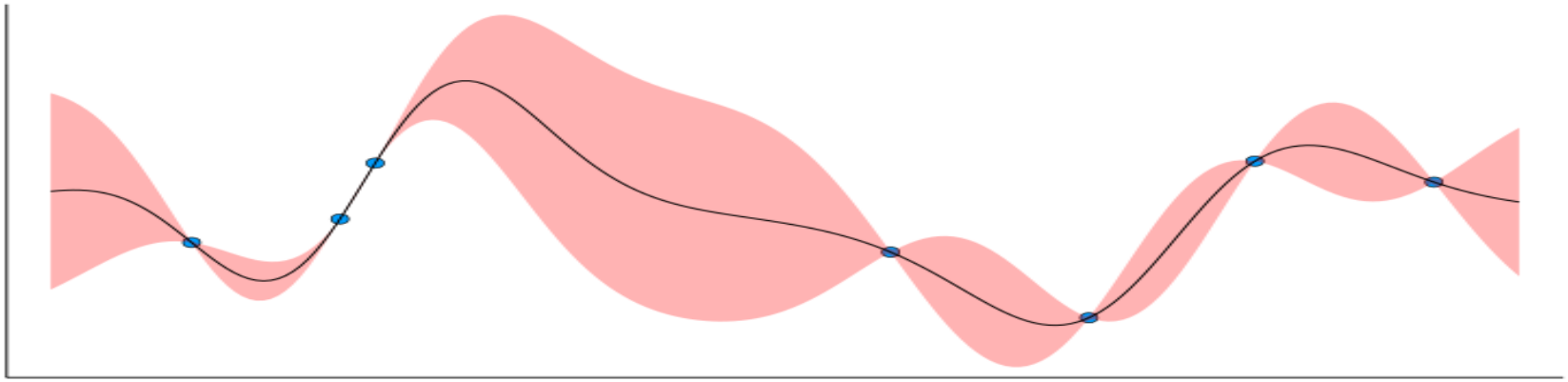
Definition of a Gaussian process



A Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution.

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

Definition of a Gaussian process



A Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution.

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

where $m(\cdot)$ is the mean function and $k(\cdot, \cdot)$ is the covariance function.

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}\left[f(\mathbf{x})\right] \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}\left[\left(f(\mathbf{x}) - m(\mathbf{x})\right)\left(f(\mathbf{x}') - m(\mathbf{x}')\right)\right] \end{aligned}$$

Kernels as a measure of covariance

Kernel	Formula
Squared Exponential (SE)	$s^2 \exp(-\frac{r^2}{2\ell^2})$
Matern 3/2	$s^2(1+\frac{\sqrt{3}r}{\ell})\exp(-\frac{\sqrt{3}r}{\ell})$
Linear	$x^T x'$

Figure: A few popular RBF kernels.

Kernels as a measure of covariance

Kernel	Formula
Squared Exponential (SE)	$s^2 \exp(-\frac{r^2}{2\ell^2})$
Matern 3/2	$s^2(1+\frac{\sqrt{3}r}{\ell})\exp(-\frac{\sqrt{3}r}{\ell})$
Linear	$x^T x'$

Figure: A few popular RBF kernels.

- In this setting, our covariance function $k(\mathbf{x}, \mathbf{x}')$ is a kernel function that captures the correlation between two points.

Kernels as a measure of covariance

Kernel	Formula
Squared Exponential (SE)	$s^2 \exp(-\frac{r^2}{2\ell^2})$
Matern 3/2	$s^2(1+\frac{\sqrt{3}r}{\ell})\exp(-\frac{\sqrt{3}r}{\ell})$
Linear	$x^T x'$

Figure: A few popular RBF kernels.

- In this setting, our covariance function $k(\mathbf{x}, \mathbf{x}')$ is a kernel function that captures the correlation between two points.
- A kernel dependent on $r = \|x - x'\|_2$ i.e. $k(x, x') = \phi(r)$ is a radial basis function (RBF) kernel.

Kernels as a measure of covariance

Kernel	Formula
Squared Exponential (SE)	$s^2 \exp(-\frac{r^2}{2\ell^2})$
Matern 3/2	$s^2(1+\frac{\sqrt{3}r}{\ell})\exp(-\frac{\sqrt{3}r}{\ell})$
Linear	$x^T x'$

Figure: A few popular RBF kernels.

- In this setting, our covariance function $k(\mathbf{x}, \mathbf{x}')$ is a kernel function that captures the correlation between two points.
- A kernel dependent on $r = \|x - x'\|_2$ i.e. $k(x, x') = \phi(r)$ is a radial basis function (RBF) kernel.
- Each kernel has hyperparameters θ

Learning optimal kernel hyperparameters

- To learn hyperparameters in kernel, we need values and gradients of:

$$\mathcal{L}(y_X|\theta) = \mathcal{L}_y + \mathcal{L}_{|K|} - \frac{n}{2}\log(2\pi).$$

Learning optimal kernel hyperparameters

- To learn hyperparameters in kernel, we need values and gradients of:

$$\mathcal{L}(y_X|\theta) = \mathcal{L}_y + \mathcal{L}_{|K|} - \frac{n}{2}\log(2\pi).$$

- Model fit term:

$$\mathcal{L}_y = -\frac{1}{2}(y_X - \mu_X)^T \left[\widetilde{K}_{XX}\right]^{-1}(y_X - \mu_X).$$

Learning optimal kernel hyperparameters

- To learn hyperparameters in kernel, we need values and gradients of:

$$\mathcal{L}(y_X|\theta) = \mathcal{L}_y + \mathcal{L}_{|K|} - \frac{n}{2}\log(2\pi).$$

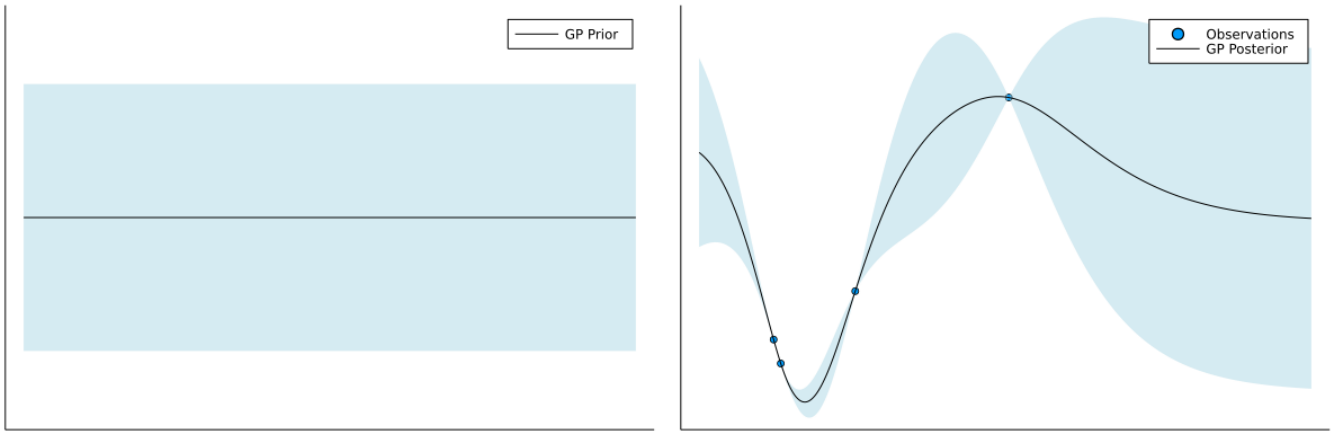
- Model fit term:

$$\mathcal{L}_y = -\frac{1}{2}(y_X - \mu_X)^T \left[\widetilde{K}_{XX}\right]^{-1}(y_X - \mu_X).$$

- Complexity penalty:

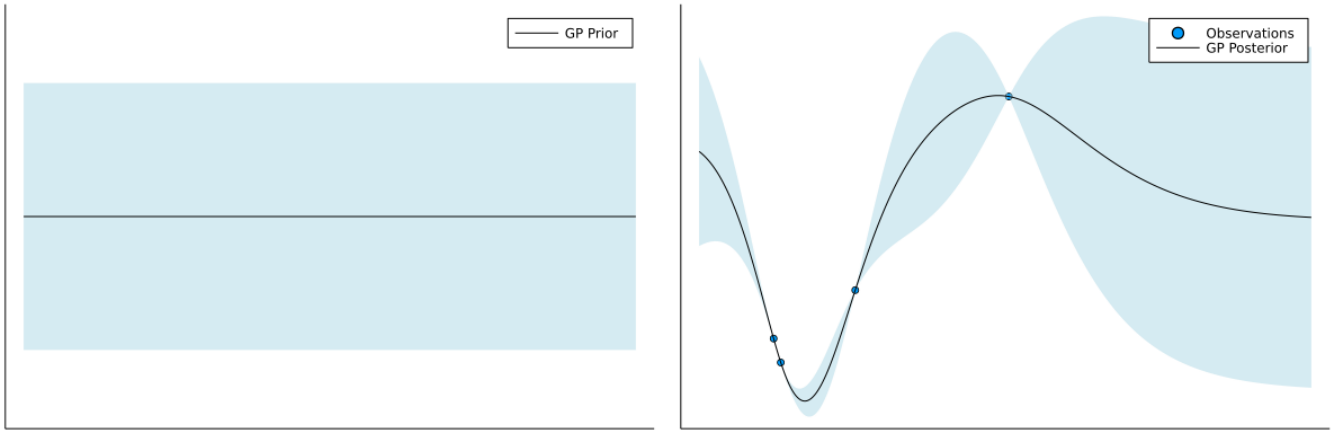
$$\mathcal{L}_{|K|} = -\frac{1}{2}\log|\widetilde{K}_{XX}|.$$

Gaussian process regression



- Suppose we have computed n potentially noisy samples $y_j = f(x^j) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$
- $X = \begin{bmatrix} x^1 & x^2 & \dots & x^n \end{bmatrix}$ and values $y = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}^T$

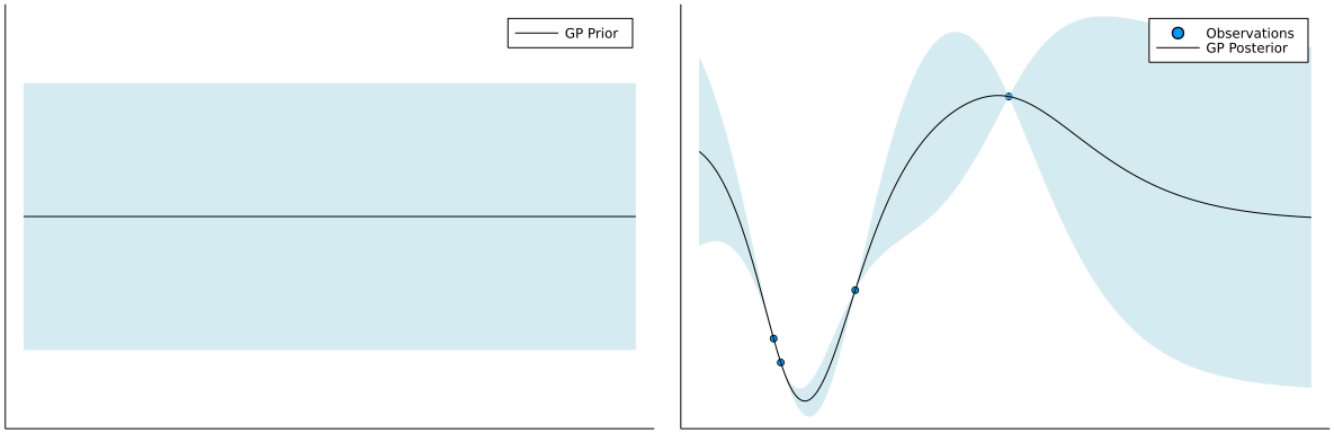
Gaussian process regression



- Suppose we have computed n potentially noisy samples $y_j = f(x^j) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$
- $X = \begin{bmatrix} x^1 & x^2 & \dots & x^n \end{bmatrix}$ and values $y = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}^T$
- Posterior prediction at $\bar{x}$ is $\hat{f}(\bar{x}) \sim \mathcal{N}(\mu_{\bar{x}}, \hat{K}_{\bar{x}\bar{x}})$:

$$\mu_{\bar{x}} = K_{X\bar{x}}^T c \ , \ \hat{K}_{\bar{x}\bar{x}} = K_{\bar{x}\bar{x}} - K_{X\bar{x}}^T \widetilde{K}_{XX}^{-1} K_{X\bar{x}}$$

Gaussian process regression



- Suppose we have computed n potentially noisy samples $y_j = f(x^j) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$
- $X = \begin{bmatrix} x^1 & x^2 & \dots & x^n \end{bmatrix}$ and values $y = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}^T$
- Posterior prediction at $\bar{x}$ is $\hat{f}(\bar{x}) \sim \mathcal{N}(\mu_{\bar{x}}, \hat{K}_{\bar{x}\bar{x}})$:

$$\mu_{\bar{x}} = K_{X\bar{x}}^T c \ , \ \hat{K}_{\bar{x}\bar{x}} = K_{\bar{x}\bar{x}} - K_{X\bar{x}}^T \widetilde{K}_{XX}^{-1} K_{X\bar{x}}$$

- Where the coefficients c are the solution to this system of equations:

$$\widetilde{K}_{XX} c = y, \widetilde{K}_{XX} = K_{XX} + \sigma_n^2 \mathcal{I}$$

Recap and takeaways

- Gaussian processes (GP) define a distribution over functions

Recap and takeaways

- Gaussian processes (GP) define a distribution over functions
- Our choice of kernel defines the function space

Recap and takeaways

- Gaussian processes (GP) define a distribution over functions
- Our choice of kernel defines the function space
- GPs allow us to perform inference while quantifying our uncertainty

Bayesian Optimization

Fundamental steps for doing Bayesian optimization

1. Gather initial samples

Fundamental steps for doing Bayesian optimization

1. Gather initial samples
2. Initialize the surrogate model

Fundamental steps for doing Bayesian optimization

1. Gather initial samples
2. Initialize the surrogate model
3. Construct the acquisition function $\alpha(x)$

Fundamental steps for doing Bayesian optimization

1. Gather initial samples
2. Initialize the surrogate model
3. Construct the acquisition function $\alpha(x)$
4. Optimize the acquisition function $x^* = \text{argmin}\, \alpha(x)$

Fundamental steps for doing Bayesian optimization

1. Gather initial samples
2. Initialize the surrogate model
3. Construct the acquisition function $\alpha(x)$
4. Optimize the acquisition function $x^* = \text{argmin}\, \alpha(x)$
5. Sample new data at x^* and update surrogate model

Fundamental steps for doing Bayesian optimization

1. Gather initial samples
2. Initialize the surrogate model
3. Construct the acquisition function $\alpha(x)$
4. Optimize the acquisition function $x^* = \text{argmin}\, \alpha(x)$
5. Sample new data at x^* and update surrogate model
6. Repeat until the budget is exhausted

Fundamental steps for doing Bayesian optimization

1. Gather initial samples
2. Initialize the surrogate model
3. Construct the acquisition function $\alpha(x)$
4. Optimize the acquisition function $x^* = \operatorname{argmin} \alpha(x)$
5. Sample new data at x^* and update surrogate model
6. Repeat until the budget is exhausted
7. Make final recommendation

BO: 1. Gather Initial Samples (if any)

Gathering our initial samples is trivial and we denote the collection of covariates and their respective observations as such:

$$X = \begin{bmatrix} x^1 & x^2 & \dots & x^n \end{bmatrix}$$
$$\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}^T$$

where $x^j \in \mathbb{R}^d$, $\forall \quad 1 \leq j \leq n$

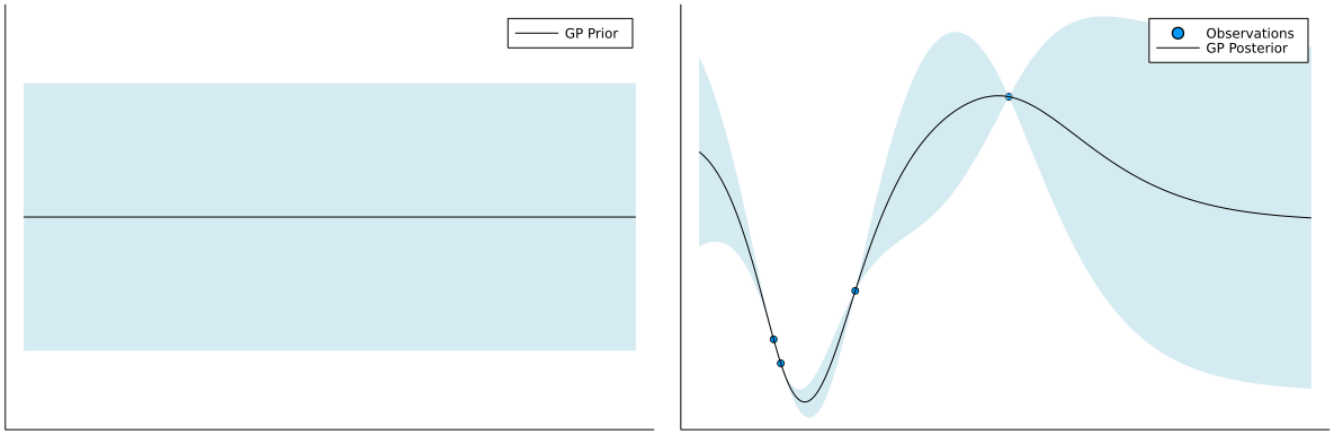
BO: 2. Initialize the Surrogate Model

- Our surrogate of choice is a Gaussian process and initializing our model is akin to conditioning our model on our observations.

BO: 2. Initialize the Surrogate Model

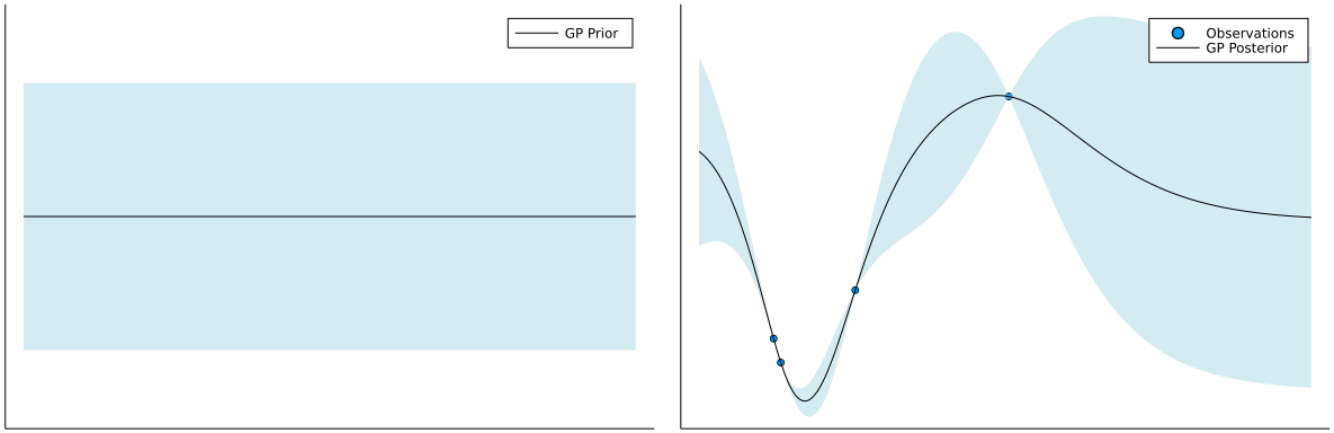
- Our surrogate of choice is a Gaussian process and initializing our model is akin to conditioning our model on our observations.
- Or, in the nomenclature of Bayesian statistics, computing the posterior probability distribution.

BO: 2. Initialize the Surrogate Model



Therefore, predictions (f_*) at test locations (X_*) is computed as follows:

BO: 2. Initialize the Surrogate Model



Therefore, predictions (f_*) at test locations (X_*) is computed as follows:

$$f_*|X, \mathbf{y}, X_* \sim \mathcal{N}(\bar{f}_*, cov(f_*)), \quad \textit{where}$$

$$\bar{f}_* \triangleq \mathbb{E}\left[f_*|X, \mathbf{y}, X_*\right]$$

$$= K(X_*, X)\left[K(X, X) + \sigma_n^2 \mathcal{I}\right]^{-1} \mathbf{y}$$

$$cov(f_*) = K(X_*, X_*) - K(X_*, X)\left[K(X, X) + \sigma_n^2 \mathcal{I}\right]^{-1} K(X, X_*)$$

BO: 3. Construct Acquisition Function $\alpha(x)$

- The acquisition function drives our sampling process, i.e. where should we look next to update our beliefs so as to gain as much "information" as possible, while leveraging well known regions as well.

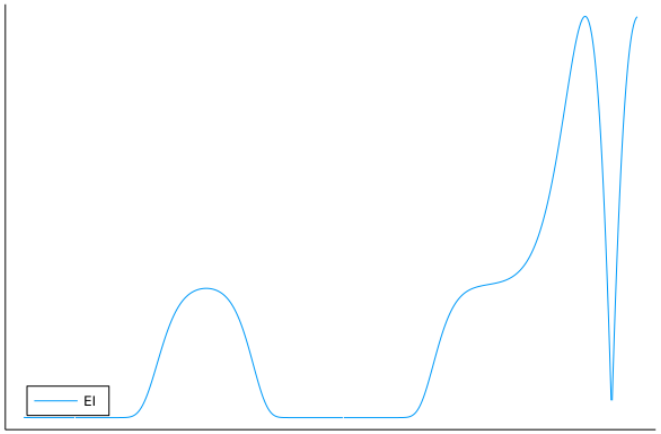
BO: 3. Construct Acquisition Function $\alpha(x)$

- The acquisition function drives our sampling process, i.e. where should we look next to update our beliefs so as to gain as much "information" as possible, while leveraging well known regions as well.
- Our choice of where to evaluate is based on a tradeoff between high expected performance and high uncertainty.

BO: 3. Construct Acquisition Function $\alpha(x)$

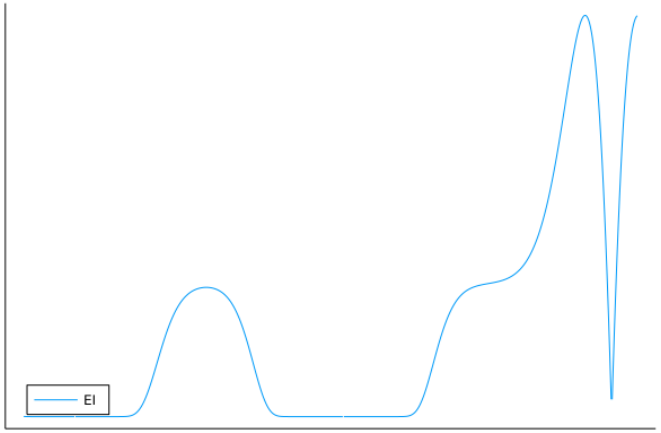
- The acquisition function drives our sampling process, i.e. where should we look next to update our beliefs so as to gain as much "information" as possible, while leveraging well known regions as well.
- Our choice of where to evaluate is based on a tradeoff between high expected performance and high uncertainty.
  - We call this the "exploration-exploitation tradeoff".

BO: 3. Construct Acquisition Function $\alpha(x)$



Perhaps the three most popular acquisition functions are Expected Improvement (EI), Probability of Improvement (POI) and Upper Confidence Bound (UCB):

BO: 3. Construct Acquisition Function $\alpha(x)$



Perhaps the three most popular acquisition functions are Expected Improvement (EI), Probability of Improvement (POI) and Upper Confidence Bound (UCB):

$$EI(x) = z\sigma\Phi(z) + \sigma\phi(z)$$
$$POI(x) = \Phi(z)$$
$$UCB(x) = \mu + \kappa\sigma$$

where $z = \left(\mu - f^{+} - \xi\right)/\sigma$, μ, σ are functions of x, Φ is standard normal CDF and ϕ is standard normal PDF.

BO: 3. Construct Acquisition Function $\alpha(x)$

Regardless of the acquisition function, we denote it as:

$$\alpha(x)$$

which can be thought of as a scoring function on the viability of candidate location x.

BO: 4. Optimize Acquisition Function

Optimizing our acquisition functions informs us of where we should sample next:

$$x_{opt} = \underset{x \in A}{\text{argmax}}\, \alpha(x) \implies \nabla \alpha(x_{opt}) = \mathbf{0}$$

BO: 5. Update Surrogate w/New Data

Condition our model on our new observation:

$$X_{update} = \begin{bmatrix} x^1 & x^2 & \dots & x^n & x_{opt} \end{bmatrix}$$
$$\mathbf{y}_{update} = \begin{bmatrix} y_1 & y_2 & \dots & y_n & y_{opt} \end{bmatrix}^T$$

where $x^j \in \mathbb{R}^d$, $\forall \quad 1 \leq j \leq n+1$

BO: 6. Repeat Until Budget Is Exhausted

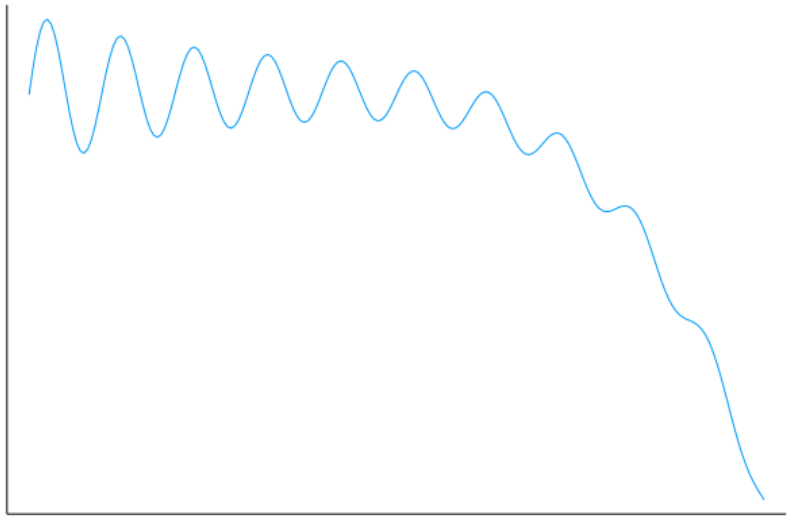
AND REPEAT

BO: 7. Make Final Recommendation

Once our budget is exhausted, we propose the best point we've seen thus far as our final recommendation.

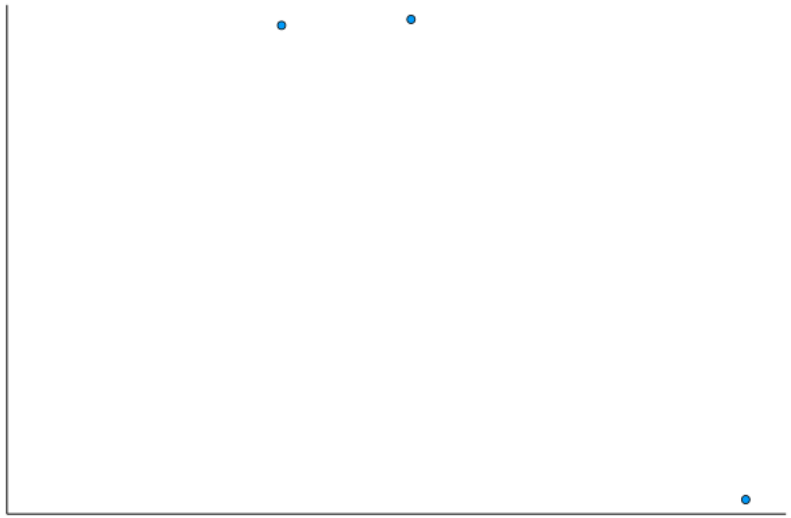
Bayesian Optimization in 1D

7 Steps to Bayesian Optimization



Bayesian optimization steps:

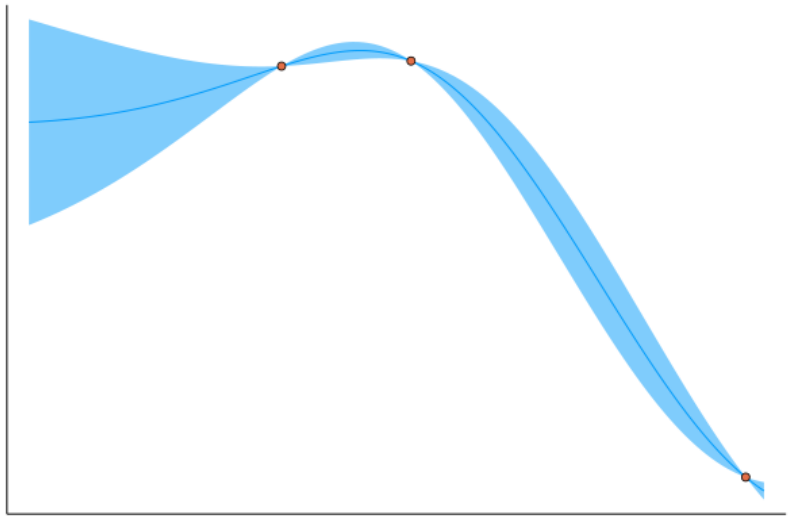
7 Steps to Bayesian Optimization



Bayesian optimization steps:

1. Initial samples

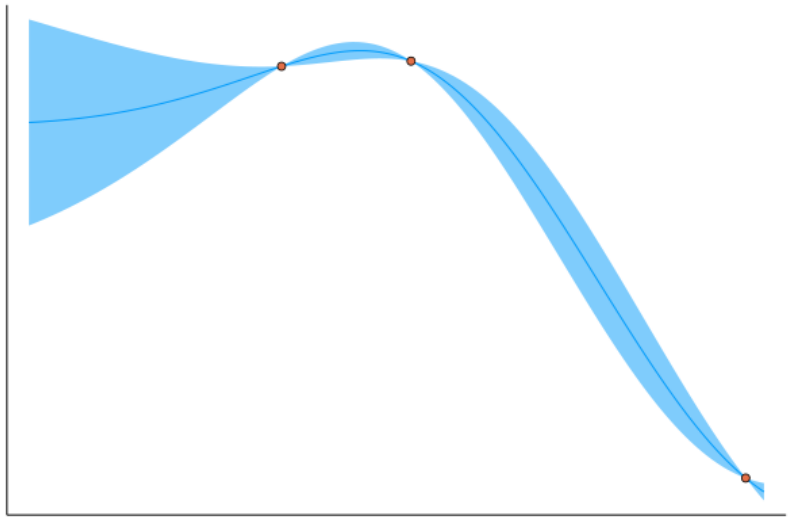
7 Steps to Bayesian Optimization

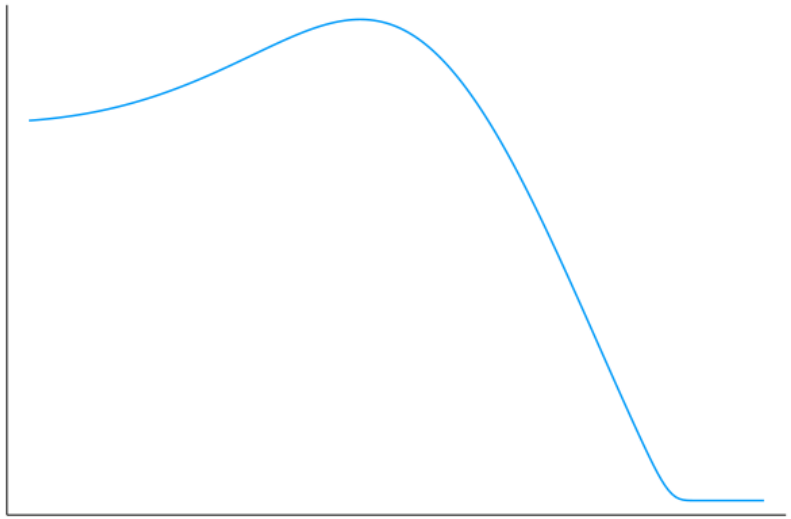


Bayesian optimization steps:

1. Initial samples
2. Initialize surrogate model

7 Steps to Bayesian Optimization

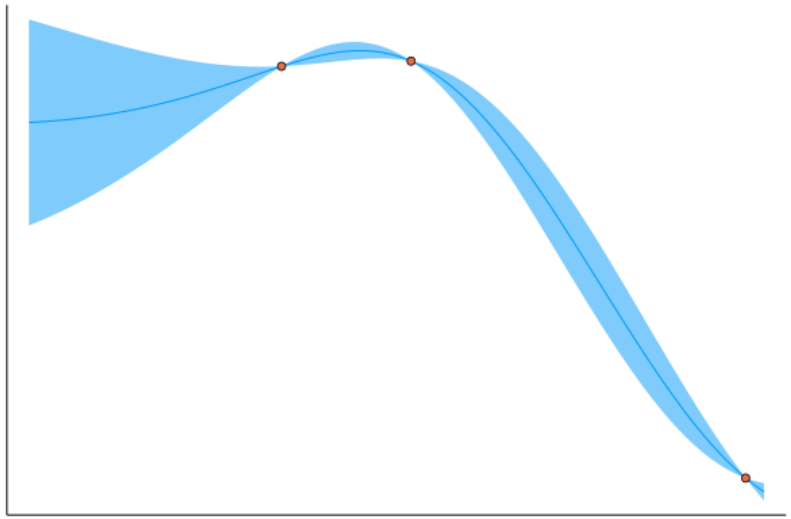


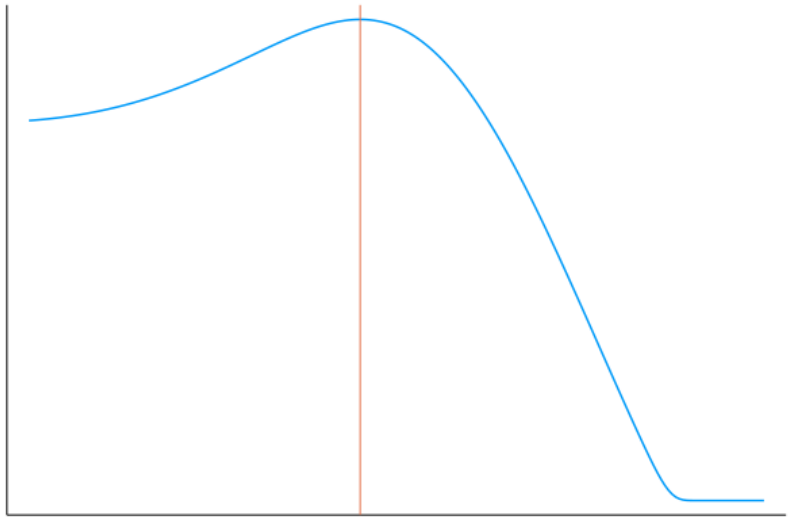


Bayesian optimization steps:

1. Initial samples
2. Initialize surrogate model
3. Get acquisition function $\alpha(x)$

7 Steps to Bayesian Optimization

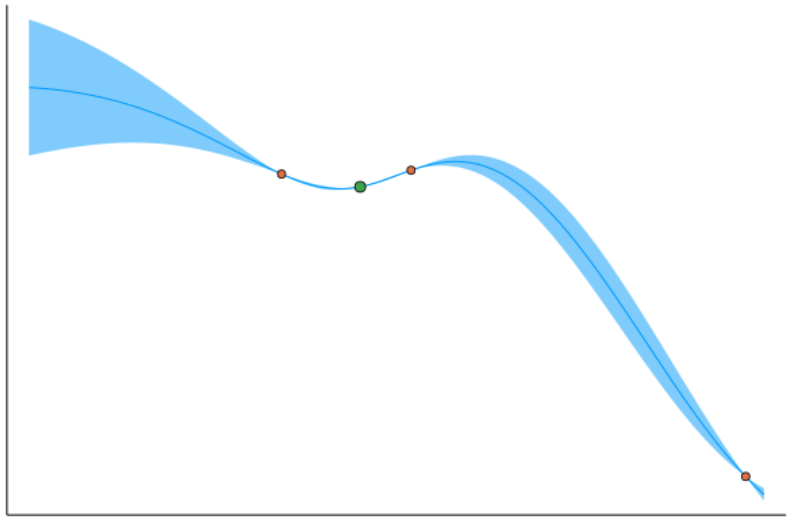


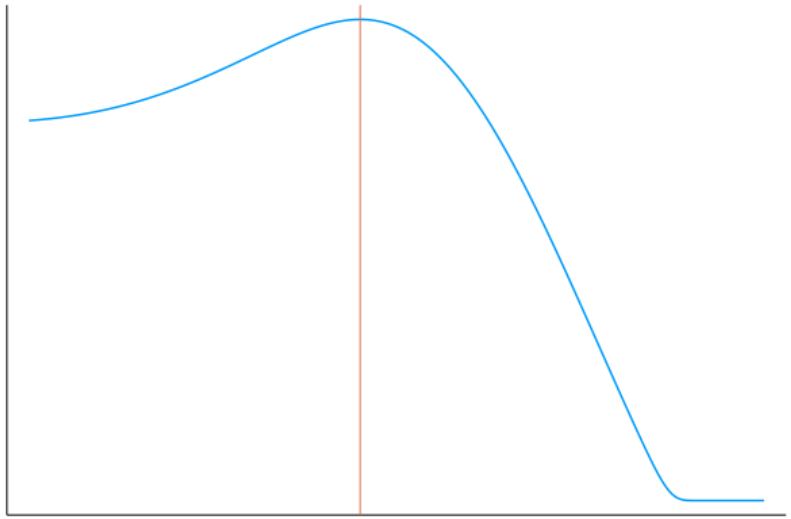


Bayesian optimization steps:

1. Initial samples
2. Initialize surrogate model
3. Get acquisition function $\alpha(x)$
4. Optimize $\alpha(x)$:
   $x_{next} = \text{argmax}\, \alpha(x)$

7 Steps to Bayesian Optimization

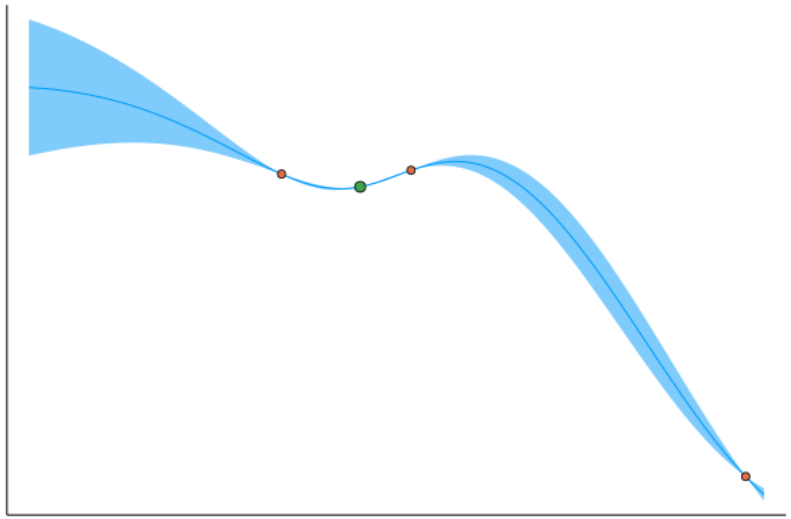


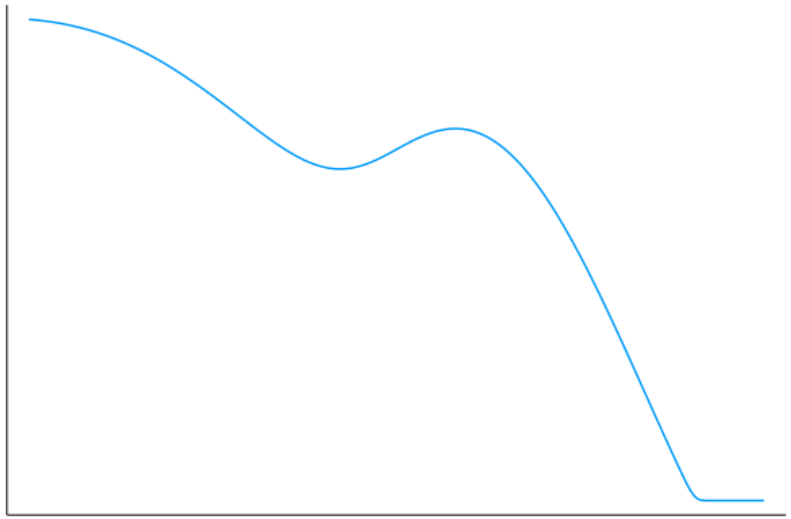


Bayesian optimization steps:

1. Initial samples
2. Initialize surrogate model
3. Get acquisition function $\alpha(x)$
4. Optimize $\alpha(x)$:
   $x_{next} = \text{argmax}\, \alpha(x)$
5. Sample new data and update surrogate

7 Steps to Bayesian Optimization

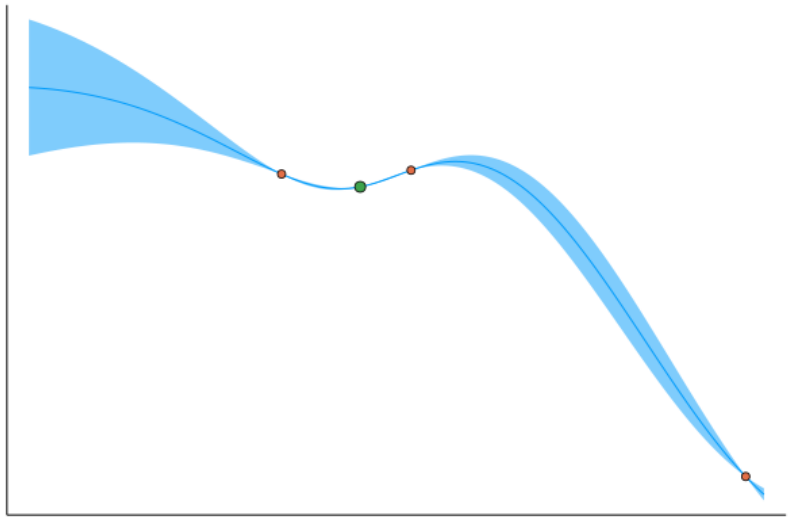


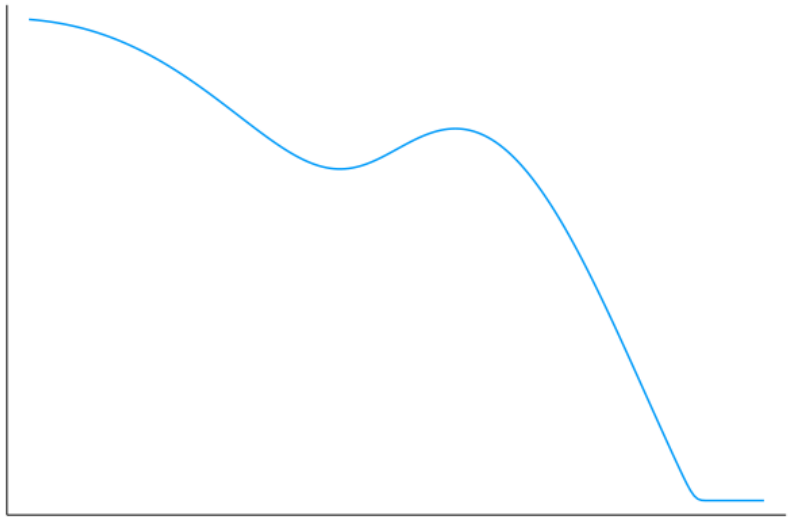


Bayesian optimization steps:

1. Initial samples
2. Initialize surrogate model
3. Get acquisition function $\alpha(x)$
4. Optimize $\alpha(x)$:
   $x_{next} = \text{argmax}\, \alpha(x)$
5. Sample new data and update surrogate
6. Go to step 3 and repeat

7 Steps to Bayesian Optimization





Bayesian optimization steps:

1. Initial samples
2. Initialize surrogate model
3. Get acquisition function $\alpha(x)$
4. Optimize $\alpha(x)$:
   $x_{next} = \text{argmax}\, \alpha(x)$
5. Sample new data and update surrogate
6. Go to step 3 and repeat
7. Make final recommendation

Fundamental problem in Bayesian optimization

Given the sequential nature of how we seek to globally optimize f, we have the appropriately named issue: exploration-exploitation trade-off.

Fundamental problem in Bayesian optimization

Given the sequential nature of how we seek to globally optimize f, we have the appropriately named issue: exploration-exploitation trade-off.

- When should I consider regions of high uncertainty at the possibility of receiving a better reward?

Fundamental problem in Bayesian optimization

Given the sequential nature of how we seek to globally optimize f, we have the appropriately named issue: exploration-exploitation trade-off.

- When should I consider regions of high uncertainty at the possibility of receiving a better reward?
- When should I ignore regions of high uncertainty and exploit regions of low uncertainty but with a "reasonably" good reward?

What do we mean by myopic Bayesian optimization?

- Our strategy doesn't consider the impact our current decision has on subsequent decisions

What do we mean by myopic Bayesian optimization?

- Our strategy doesn't consider the impact our current decision has on subsequent decisions
- This is by design, due to how we construct the acquisition function

Recap and takeaways

- BO is a sequential design strategy used to find the global optimum of an expensive-to-evaluate function

Recap and takeaways

- BO is a sequential design strategy used to find the global optimum of an expensive-to-evaluate function
- How we construct the acquisition function directly influences what locations we try next

Recap and takeaways

- BO is a sequential design strategy used to find the global optimum of an expensive-to-evaluate function
- How we construct the acquisition function directly influences what locations we try next
- Myopia is the tendency to focus on the short-term rather than the long-term implications of our actions

Additional Resources

1. (Paper) A Tutorial on Bayesian Optimization by Peter Frazier
2. (Book) Bayesian Optimization by Roman Garnett
3. (Book) Gaussian Processes for Machine Learning by Carl Rasmussen