Gaussian Processes for Regression

Outline

- Review of Bayesian estimation
- Review of Bayesian linear regression
- Gaussian process regression
- Bayesian optimization

Bayesian Estimation

- Maximum likelihood assumes that $\boldsymbol{\theta}$ is nonrandom
- In Bayesian estimation, $\boldsymbol{\theta}$ is viewed as random
- Our uncertainty about $\boldsymbol{\theta}$ depends on whether we have observed data or not
- Let $\underline{\boldsymbol{Z}} = (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_n)$ denote all observed quantities
- Let $p(\boldsymbol{\theta})$ denote the distribution of $\boldsymbol{\theta}$ *before* observing a realization $\underline{\boldsymbol{z}}$ of $\underline{\boldsymbol{Z}}$.
- Let $p(\boldsymbol{\theta} \mid \underline{\boldsymbol{z}})$ denote the distribution of $\boldsymbol{\theta}$ *after* observing a realization $\underline{\boldsymbol{z}}$ of $\underline{\boldsymbol{Z}}$.
- These are related by Bayes rule:

$$p(\theta \mid \underline{z}) = \frac{p(\underline{z} \mid \theta)\, p(\theta)}{p(\underline{z})}$$

$$p(\underline{z}) = \int p(\underline{z} \mid \theta)\, p(\theta)\, d\theta$$

user-specified

$(\underline{Z}, \theta)$ jointly distributed

prior distribution

posterior distribution

Bayesian Estimation

- A parameter estimate can be obtained from the posterior in multiple ways:
- Posterior mean: $\hat{\theta}(\underline{z}) = \mathbb{E}[\theta \mid \underline{Z} = \underline{z}] = \int \theta \, p(\theta \mid \underline{z}) \, d\theta$

- Maximum a posteriori (MAP) estimation:

$$\hat{\theta}(\underline{z}) = \arg\max_{\theta} \log p(\theta \mid \underline{z})$$

$$= \arg\max_{\theta} \log p(\underline{z} \mid \theta) + \log p(\theta)$$

- In addition to parameter estimates, the main advantage of Bayesian methods over frequentist methods is they also provide natural confidence intervals.

Bayesian Linear Regression

- Training data: $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$. View $\boldsymbol{x}_i$ as fixed.
- No offset, $\boldsymbol{\theta} = \boldsymbol{w}$
- Gaussian likelihood:

$$y_i = \boldsymbol{w}^T \boldsymbol{x}_i + \epsilon_i, \quad \Longleftrightarrow \quad y_i \sim N(w^T x_i, \sigma_e^2)$$

  where $\epsilon_i \sim \mathcal{N}(0, \sigma_e^2)$

- Prior:

$$w \sim N(0, \sigma_w^2 \mathbf{I}) \Leftrightarrow p(w) = (2\pi\sigma_w^2)^{-\frac{d}{2}} \exp\left\{-\frac{\|w\|^2}{2\sigma_w^2}\right\}$$

$$\Longleftrightarrow \quad w = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}, \quad w_i \overset{iid}{\sim} N(0, \sigma_w^2)$$

- σ_e^2, σ_w^2 are hyperparameters

Bayesian Linear Regression

- Posterior:

$$w|y \sim \mathcal{N}(\mu(y), \Sigma(y))$$

where

$$\hat{w} = \boldsymbol{\mu}(\boldsymbol{y}) = \left(\boldsymbol{X}\boldsymbol{X}^T + \frac{\sigma_e^2}{\sigma_w^2}\boldsymbol{I}\right)^{-1}\boldsymbol{X}\boldsymbol{y}$$

$$\boldsymbol{\Sigma}(\boldsymbol{y}) = \left(\frac{1}{\sigma_e^2}\boldsymbol{X}\boldsymbol{X}^T + \frac{1}{\sigma_w^2}\boldsymbol{I}\right)^{-1}$$

and

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_n \end{bmatrix} \in \mathbb{R}^{d \times n}$$

Bayesian Linear Regression

- Linear transformation of a MVG is another MVG
- In particular, if $\boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and $\boldsymbol{A}$ is a matrix and $\boldsymbol{b}$ a column vector of appropriate dimensions, then

$$\boldsymbol{AZ} + \boldsymbol{b} \sim \mathcal{N}(\boldsymbol{A\mu} + \boldsymbol{b}, \boldsymbol{A\Sigma A}^T)$$

- If $\boldsymbol{x}$ is a test point, then the predicted output

$$\widehat{f}(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{w}$$

- Thus, the predicted output is a random variable with distribution

$$\widehat{f}(\boldsymbol{x}) \mid \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}^T\boldsymbol{\mu}(y), \boldsymbol{x}^T\boldsymbol{\Sigma}(y)\boldsymbol{x})$$

$$\sim \mathcal{N}\left(\underbrace{\boldsymbol{x}^T\left(\boldsymbol{X}\boldsymbol{X}^T + \frac{\sigma_\epsilon^2}{\sigma_w^2}\boldsymbol{I}\right)^{-1}\boldsymbol{X}\boldsymbol{y}}_{\mu_x(y)}, \underbrace{\boldsymbol{x}^T\left(\boldsymbol{X}\boldsymbol{X}^T + \frac{\sigma_\epsilon^2}{\sigma_w^2}\boldsymbol{I}\right)^{-1}\boldsymbol{x}}_{\sigma_x^2(y)}\right)$$

Bayesian Linear Regression

- Using matrix identities (e.g., the matrix inversion lemma), it can be shown that

$$\widehat{f}(\boldsymbol{x}) \mid \boldsymbol{y} \sim \mathcal{N}(\mu_{\boldsymbol{x}}(\boldsymbol{y}), \sigma^2_{\boldsymbol{x}}(\boldsymbol{y}))$$

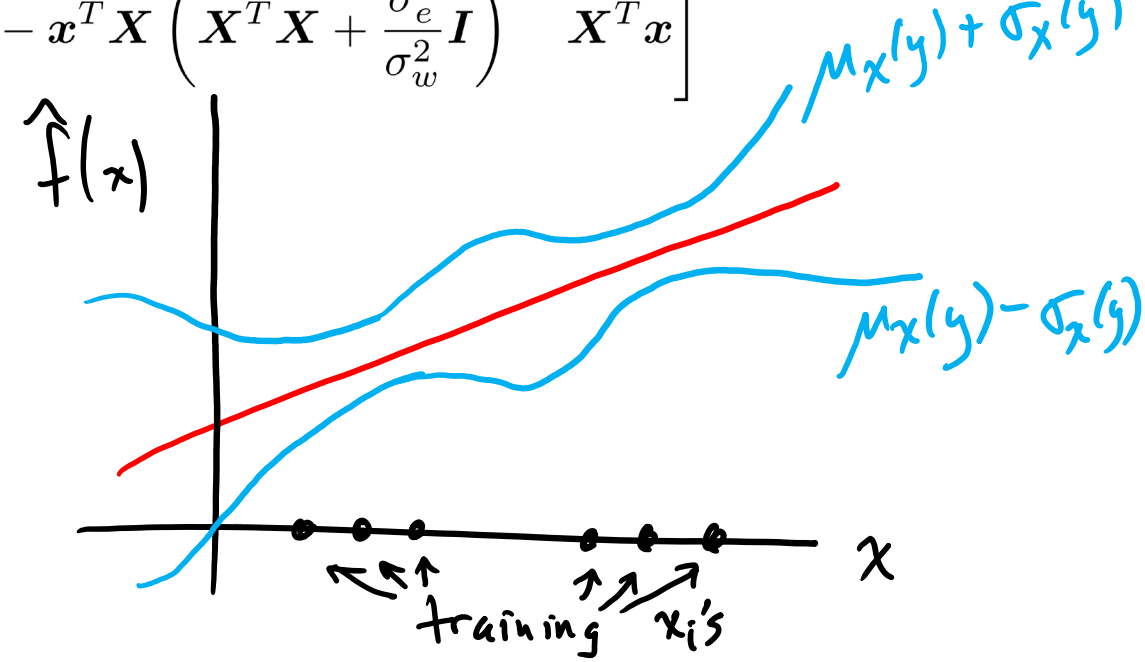
where

$$\mu_{\boldsymbol{x}}(\boldsymbol{y}) = \sigma_w^2 \boldsymbol{x}^T \boldsymbol{X} \left(\boldsymbol{X}^T \boldsymbol{X} + \frac{\sigma_e^2}{\sigma_w^2} \boldsymbol{I} \right)^{-1} \boldsymbol{y}$$

$$\sigma^2_{\boldsymbol{x}}(\boldsymbol{y}) = \sigma_w^2 \left[\boldsymbol{x}^T \boldsymbol{x} - \boldsymbol{x}^T \boldsymbol{X} \left(\boldsymbol{X}^T \boldsymbol{X} + \frac{\sigma_e^2}{\sigma_w^2} \boldsymbol{I} \right)^{-1} \boldsymbol{X}^T \boldsymbol{x} \right]$$

- Conclusion:

You can kernelize this method



Kernel Bayesian Linear Regression

- Let k be a symmetric, positive definite kernel
- Then Bayesian linear regression can be kernelized, leading to the prediction

$$\widehat{f}(\boldsymbol{x}) \mid \boldsymbol{y} \sim \mathcal{N}(\mu_{\boldsymbol{x}}(\boldsymbol{y}), \sigma^2_{\boldsymbol{x}}(\boldsymbol{y}))$$

where

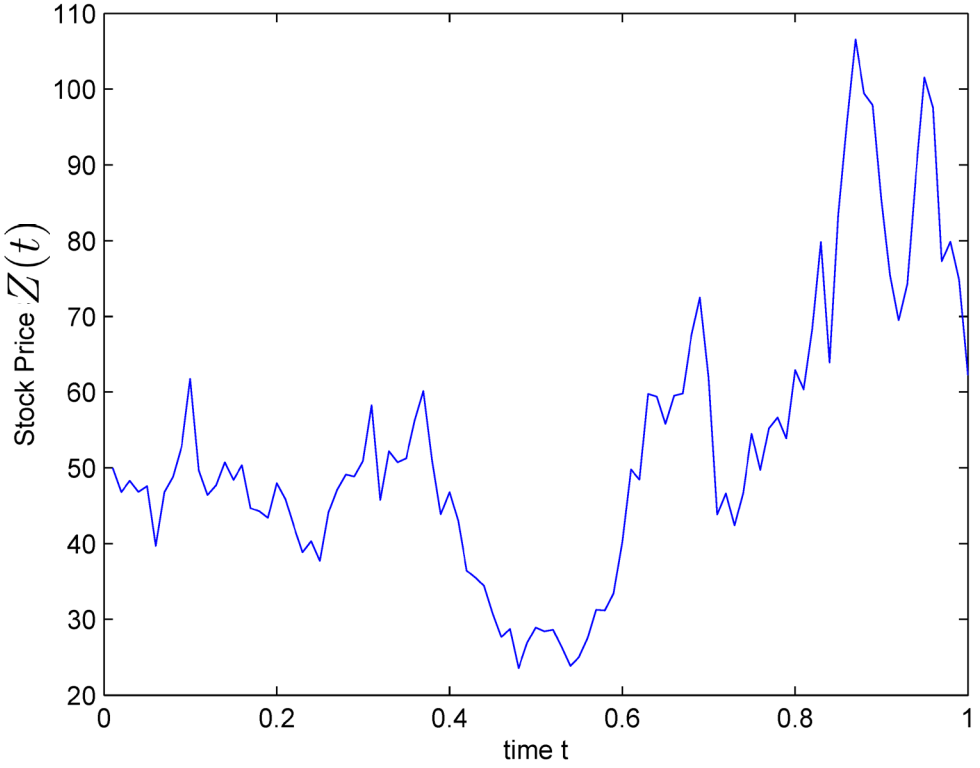
$$\mu_{\boldsymbol{x}}(\boldsymbol{y}) = \sigma_w^2 \boldsymbol{k}(\boldsymbol{x})^T \left(\boldsymbol{K} + \frac{\sigma_e^2}{\sigma_w^2} \boldsymbol{I} \right)^{-1} \boldsymbol{y}$$

$$\sigma^2_{\boldsymbol{x}}(\boldsymbol{y}) = \sigma_w^2 \left[k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x})^T \left(\boldsymbol{K} + \frac{\sigma_e^2}{\sigma_w^2} \boldsymbol{I} \right)^{-1} \boldsymbol{k}(\boldsymbol{x}) \right]$$

$$k(x) = \begin{bmatrix} k(x, x_1) \\ \vdots \\ k(x, x_n) \end{bmatrix}, \quad K = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & & \vdots \\ k(x_n, x_1) & & k(x_n, x_n) \end{bmatrix}$$

Random Processes

- A *random process* is a family of (scalar) random variables $\{Z(t)\}$, $t \in T$, where T is an index set (e.g., $T = \mathbb{R}$)
- A random process is *zero-mean* if $\mathbb{E}[Z(t)] = 0$ for all t.



https://www.probabilitycourse.com/chapter10/10_1_0_basic_concepts.php

Gaussian Processes

- Switch notation: $t \in T$ to $\boldsymbol{x} \in \mathbb{R}^d$

$\{Z(x)\}_{x \in \mathbb{R}^d} \leftrightarrow Z(x) = \text{random function on } \mathbb{R}^d$

- A *Gaussian process* is a random process, indexed by $\mathbb{R}^d$, such that for any finite set of indices $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$, the values $Z(\boldsymbol{x}_1), \ldots, Z(\boldsymbol{x}_N) \in \mathbb{R}$ are jointly Gaussian.

- A zero-mean Gaussian process is completely characterized by its *covariance function*

$$k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}\left[Z(x) Z(x')\right]$$

- Thus, for any $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$,

$$\begin{bmatrix} Z(x_1) \\ \vdots \\ Z(x_n) \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} k(x_1,x_1) & \cdots & k(x_1,x_n) \\ \vdots & \ddots & \vdots \\ k(x_n,x_1) & \cdots & k(x_n,x_n) \end{bmatrix} \right)$$

Covariance Function

- To be a valid covariance function, k must satisfy
  - $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}', \boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{x}'$
  - For all $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$, the matrix

$$\begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}$$

    must be positive semi-definite.

- In other words, k must be a symmetric, positive definite kernel
- Examples:

Gaussian: $k(x, x') = \sigma_k^2 \exp\left(-\gamma \|x - x'\|^2\right)$

Laplacian: $k(x, x') = \sigma_k^2 \exp\left(-\gamma \|x - x'\|\right)$

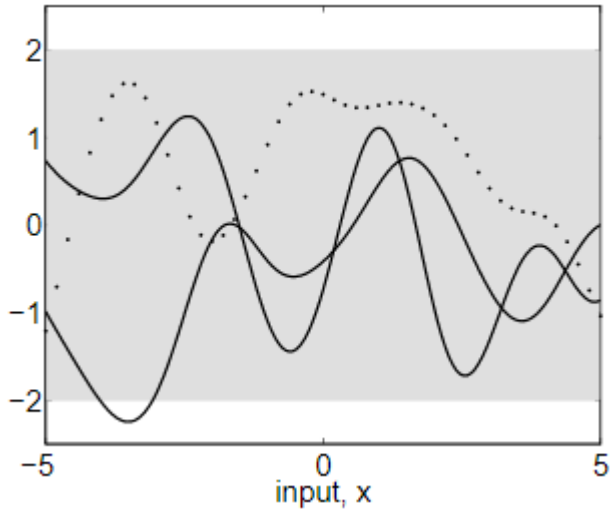
Smooth Realizations

- A realization of a Gaussian process, i.e., a random function $Z(\boldsymbol{x})$ defined on a domain $\mathcal{X} \subseteq \mathbb{R}^d$, is a continuous function provided

$\mathcal{X}$ is a compact set (e.g., a closed rectangle)

$k(x, x')$ is Lipschitz continuous (e.g., k is differentiable w/ bounded derivative)

- To plot a realization, choose a grid of points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ at which to plot the function, and simulate a realization of



$$\begin{bmatrix} Z(x_1) \\ \vdots \\ Z(x_n) \end{bmatrix} = N\left(0, \begin{bmatrix} k(x_1, x_1) & \cdots \\ \vdots & \end{bmatrix}\right)$$

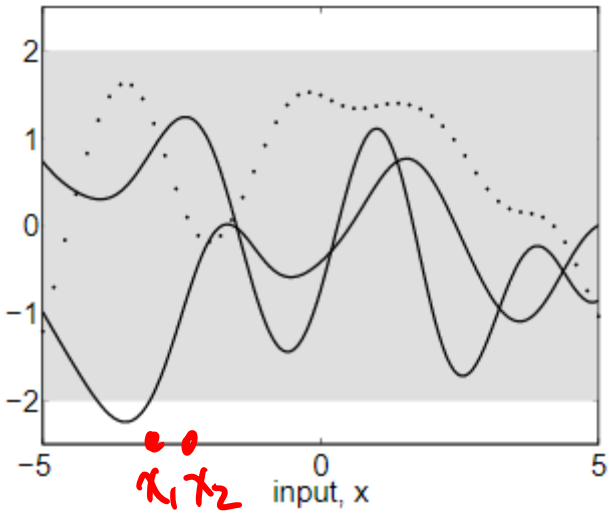
Poll

$\text{cov}(Z(x), Z(x'))$

Consider the Gaussian covariance $k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma\|\boldsymbol{x} - \boldsymbol{x}'\|^2)$. As γ increases, realizations of the associated GP become

(A) Smoother (less wiggly)

(B) Less smooth (more wiggly) ✓



$$\begin{bmatrix} Z(x_1) \\ Z(x_2) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} 1 & e^{-\gamma\|x_1 - x_2\|^2} \\ e^{-\gamma\|x_1 - x_2\|^2} & 1 \end{bmatrix}\right)$$

GPs for Regression

- Function to be estimated: $f(\boldsymbol{x})$
- Assumption: f is a realization of a Gaussian process (can be viewed as a prior distribution on f)
- This is a *nonparametric* Bayesian setup
- Training data: $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$
- Gaussian likelihood: $y_i = f(x_i) + \varepsilon_i, \quad \varepsilon_i \overset{iid}{\sim} N(0, \sigma_e^2)$
- Fixed test point: $x_{m+1}, \ldots, x_{m+n}$
- Predicted values at test points: Given $y_1, \ldots, y_n$, estimate $f(x_{m+i}), \; i = 1, \ldots, n.$
- Goal: Determine posterior distribution of

$$\mathbf{f} = \begin{bmatrix} f(x_{m+1}) \\ \vdots \\ f(x_{m+n}) \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

GPs for Regression

- The posterior is just the conditional distribution of $\boldsymbol{f}$ given $\boldsymbol{y}$.
- We will use the Gaussian conditioning property:

  If $\boldsymbol{A} \in \mathbb{R}^p$ and $\boldsymbol{B} \in \mathbb{R}^q$ are such that

  $$\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu_A} \\ \boldsymbol{\mu_B} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma_{AA}} & \boldsymbol{\Sigma_{AB}} \\ \boldsymbol{\Sigma_{BA}} & \boldsymbol{\Sigma_{BB}} \end{bmatrix} \right),$$

  then

  $$\boldsymbol{A} \mid \boldsymbol{B} = \boldsymbol{b} \sim \mathcal{N}\left(\boldsymbol{\mu_A} + \boldsymbol{\Sigma_{AB}}\boldsymbol{\Sigma_{BB}^{-1}}(\boldsymbol{b} - \boldsymbol{\mu_B}), \boldsymbol{\Sigma_{AA}} - \boldsymbol{\Sigma_{AB}}\boldsymbol{\Sigma_{BB}^{-1}}\boldsymbol{\Sigma_{BA}} \right).$$

$y_j = f(x_j) + \varepsilon_i$

GPs for Regression

(i,j) entry → $k(x_i, x_j)$, $k(x_i, x_{m+j})$, $k(x_{m+i}, x_{m+j})$

- Introduce the notation

$$\boldsymbol{K} = \text{train-train kernel matrix } (m \times m)$$
$$\boldsymbol{K}_* = \text{train-test kernel matrix } (m \times n)$$
$$\boldsymbol{K}_{**} = \text{test-test kernel matrix } (n \times n)$$

- Covariances
  - $\mathrm{Cov}(f(\boldsymbol{x}_{m+i}), f(\boldsymbol{x}_{m+j})) = k(x_{m+i}, x_{m+j})$
  - $\mathrm{Cov}(f(\boldsymbol{x}_{m+i}), y_j) = \mathrm{Cov}(f(x_{m+i}), f(x_j) + \varepsilon_j) = \mathrm{Cov}(f(x_{m+i}), f(x_j)) = k(x_{m+i}, x_j)$
  - $\mathrm{Cov}(y_i, y_j) = \mathrm{Cov}(f(x_i) + \varepsilon_i, f(x_j) + \varepsilon_j) = k(x_i, x_j) + \sigma_e^2 \mathbb{1}_{\{i=j\}}$

- The joint distribution of $\boldsymbol{f}$ and $\boldsymbol{y}$ is

$$\begin{bmatrix} f \\ y \end{bmatrix} \sim N\left(0, \begin{bmatrix} K_{**} & K_*^T \\ K_* & K + \sigma_e^2 I \end{bmatrix}\right)$$

GPs for Regression

- Therefore the posterior distribution of the predicted test outputs is

$$\boldsymbol{f} \mid \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{K}_*^T(\boldsymbol{K} + \sigma_e^2\boldsymbol{I})^{-1}\boldsymbol{y}, \boldsymbol{K}_{**} - \boldsymbol{K}_*^T(\boldsymbol{K} + \sigma_e^2\boldsymbol{I})^{-1}\boldsymbol{K}_*).$$

- Special case of $n = 1$ test point: x

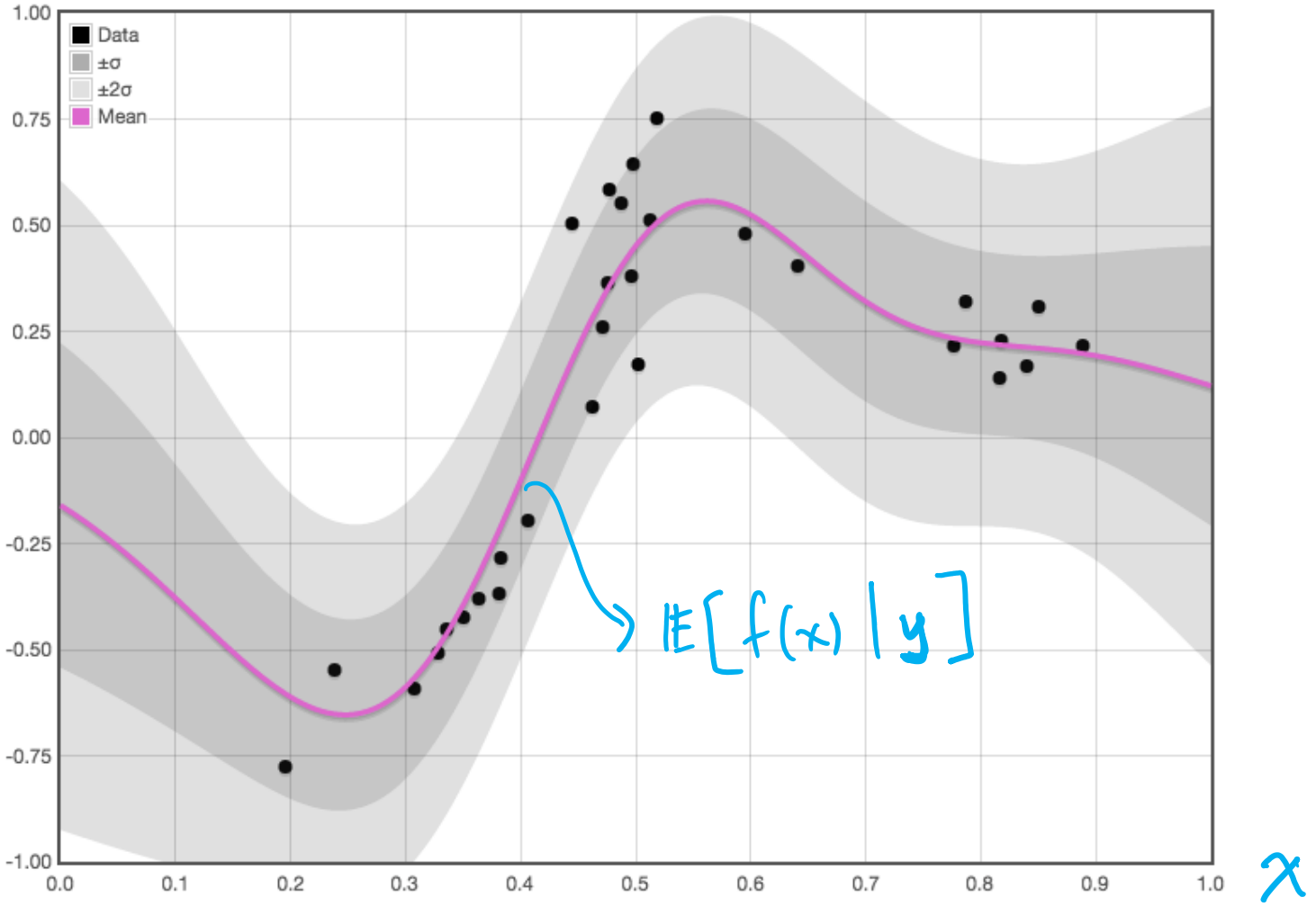
$$f(x) \mid \boldsymbol{y} \sim \mathcal{N}\left(k(x)^T(K + \sigma_e^2 I)^{-1} y,\ k(x,x) - k(x)^T(K + \sigma_e^2 I)^{-1} k(x)\right)$$

where

$$k(x) = \begin{bmatrix} k(x, x_1) \\ \vdots \\ k(x, x_m) \end{bmatrix}$$

- This agrees with our kernelized version of Bayesian linear regression
- While the estimator (posterior mean/mode) is a linear function of $\boldsymbol{y}$, it is a nonlinear function of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{m+n}$.

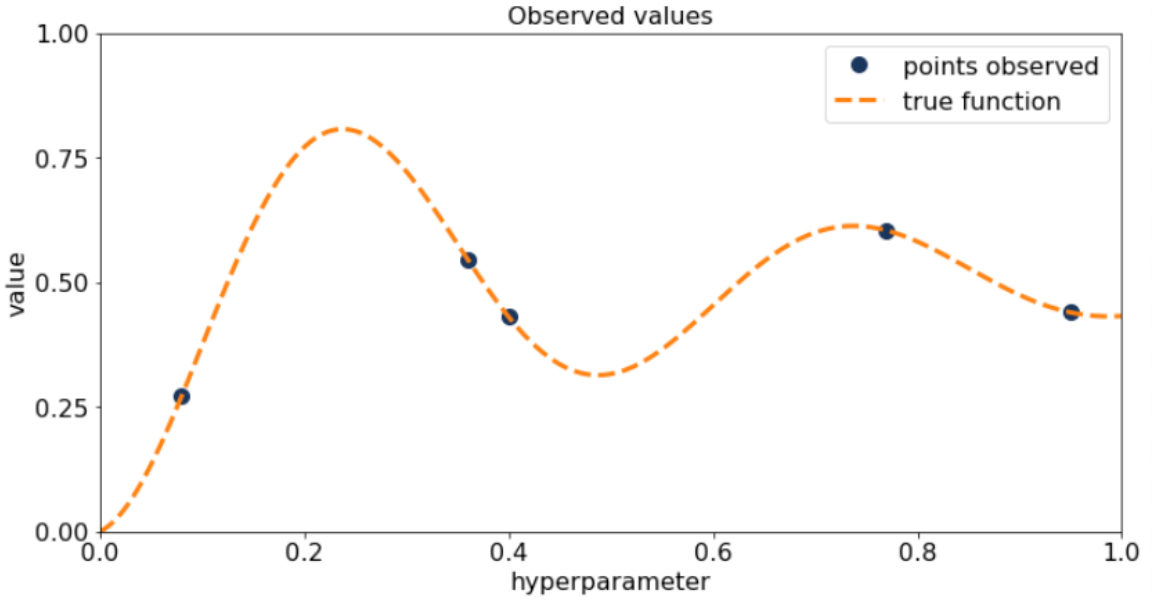
GPs for Regression



https://danmackinlay.name/notebook/gp_regression.html

Bayesian Optimization

- Method for *black box optimization*, when objective function is hard to evaluate and can't calculate derivatives
- Need to determine next point to sample
- Application: model selection in machine learning (e.g., objective function is cross-validation accuracy estimate of model performance)

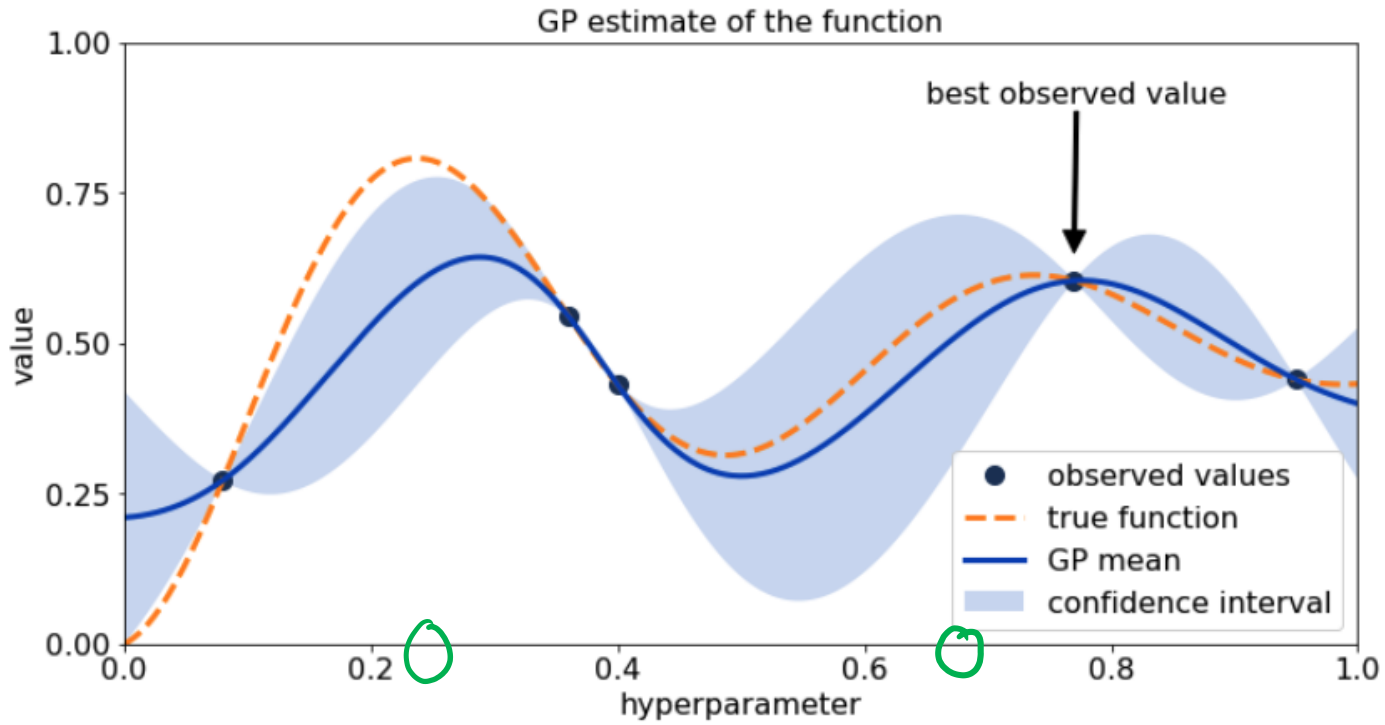


https://sigopt.com/blog/bayesian-optimization-101/

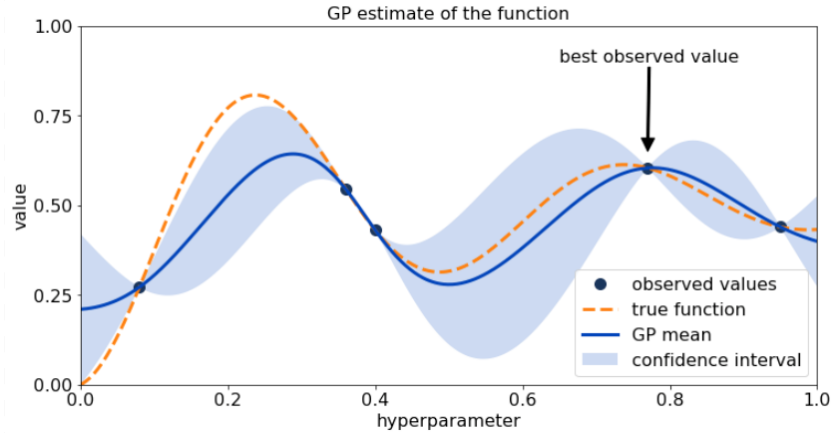
Bayesian Optimization

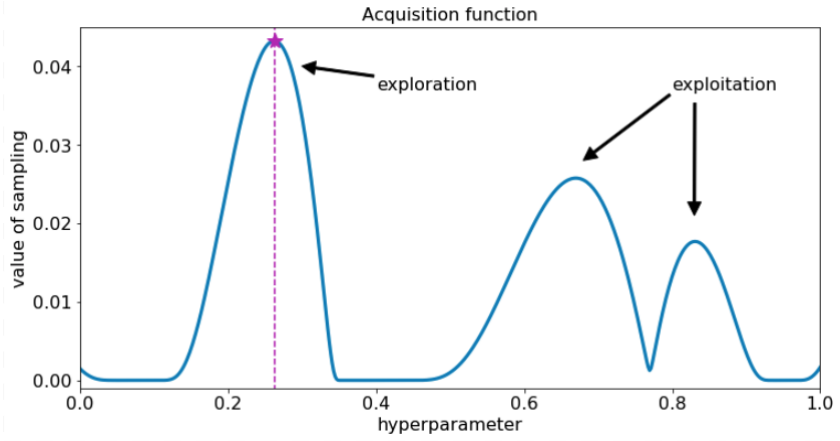
- Assume a GP prior on the unknown function
- Use posterior to characterize uncertainty in the unknown function, and determine next point to evaluate

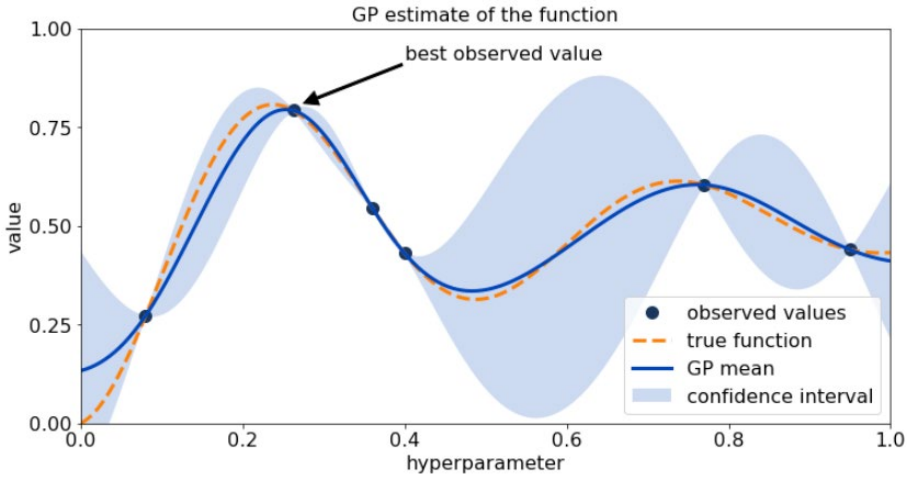
$\sigma_e^2 = 0$



Bayesian Optimization







Final Thoughts on GPs

- Free parameters can be tuned by maximum likelihood estimation – much faster than cross-validation (although possibly not as accurate)
- In regression, we used a GP prior and a Gaussian likelihood. The math works out nicely because the Gaussian prior is *conjugate* to the Gaussian likelihood.
- GPs can be used with the logistic likelihood as a nonlinear model for classification.
- This gives a nonparametric Bayesian version of kernel logistic regression
- Math is more challenging because Gaussian prior is not conjugate to the logistic likelihood
- Bayesian optimization: black-box optimization without grid search