Gaussian Processes

Oswin Krause, PML, 2022

UNIVERSITY OF COPENHAGEN

Reminder Random Fields

- Let Ω be an event space (e.g., $\mathbb{R}^N$)
- Let $\mathcal{X}$ be an index set (e.g. $\mathbb{N}$ or $\mathbb{R}^d$)
- A random field is a collection of random variables
  - $F_x \in \Omega, \forall x \in \mathcal{X}$ with realizations f_x
  - Intuitively: A function that assigns a random variable to each point $x \in \mathcal{X}$
- If $\mathcal{X} = \mathbb{R}^d$ it is also called a *random process*

Reminder Random Fields

- Let Ω be an event space (e.g., $\mathbb{R}^N$)
- Let $\mathcal{X}$ be an index set (e.g. $\mathbb{N}$ or $\mathbb{R}^d$)
- A random field is a collection of random variables
  - $F_x \in \Omega, \forall x \in \mathcal{X}$ with realizations f_x
  - Intuitively: A function that assigns a random variable to each point $x \in \mathcal{X}$
- If $\mathcal{X} = \mathbb{R}^d$ it is also called a *random process*
- Random Fields are defined by their Marginals:
  - Pick any finite subset $S_\ell = \{x_1, \dots, x_\ell\} \subseteq \mathcal{X}$
  - Marginal: $p(f_1, \dots, f_\ell | S_\ell) = p(f_{x_1}, \dots, f_{x_\ell})$

Gaussian Processes

Generalization?

We have seen two examples of random processes with:

- Marginal distributions of f conditioned on S are normal distributed
- The mean of f_i depends only on x_i
- the covariance matrix consists of entries of pairs of points

Can we generalize that?

Kernels

Definition

Let $\mathcal{X}$ be some set. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. If for all $S = \{x_1, \dots, x_\ell\} \subset \mathcal{X}$ and any $\ell \in \mathbb{N}$ it holds

$$K(S) = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_\ell) \\ \vdots & \ddots & \vdots \\ k(x_\ell, x_1) & \dots & k(x_\ell, x_\ell) \end{bmatrix} \text{ is symmetric positive semi-definite}$$

We call k a kernel.

Reminder: A matrix is positive semi-definite, if all its eigenvalues are ≥ 0

Generalization: Gaussian Processes

Definition

Let $\mathcal{X}$ be an index set

A random field $F_x \in \mathbb{R}$ whose marginals $p(f|S)$ are Multivariate Normal distributions, is called a Gaussian Process.

Moreover, there exists a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and a function $m : \mathcal{X} \to \mathbb{R}$ such, that

$$p(f|S) = \mathcal{N}(m(S), K(S)), \forall S = \{x_1, \dots, x_\ell\} \subset \mathcal{X}, \forall \ell \in \mathbb{N}$$

with $m(S) = (m(x_1), \dots, m(x_\ell))$ and $K(S)_{ij} = k(x_i, x_j)$. If m and k are known, we write

$$f \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$$

Generalization: Gaussian Processes

We have already seen examples for Gaussian Processes

- The Wiener process with kernel

$$k(x, x') = \min\{x, x'\}$$

- Bayesian Linear Regression with kernel

$$k(x, x') = \phi(x)^T \Sigma_\theta \phi(x')$$

Universal Kernels

- Problem: $K(S)$ might not be positive definite

$\rightarrow$ There is no pdf for the marginal. Bad for learning!

Example:

- GP using Bayesian Linear Regression Kernel $k(x, x') = \phi(x)^T \Sigma_\theta \phi(x')$
- $\phi(x) = (1, x, x^2, x^3)^T$
- $\rightarrow$ Sampled functions are third degree polynomials
- 4 observations are enough to uniquely define them
- $\ell > 4$: $\ell - 4$ observations have no randomness.

$\rightarrow$ $\text{rank}(K(S)) \leq 4$.

Universal Kernels I

Definition

If k is a kernel and for all $S = \{x_1, \dots, x_\ell\} \subset \mathcal{X}$ with $x_i \neq x_j, i \neq j$ additionally holds

$$K(S) \text{ is positive definite}$$

Then, we call k universal.

Reminder: A multivariate normal distribution only has a pdf if the covariance matrix is positive definite!

Universal Kernels II

Examples

- Wiener Process kernel

$$k(x, y) = \min\{x, y\}$$

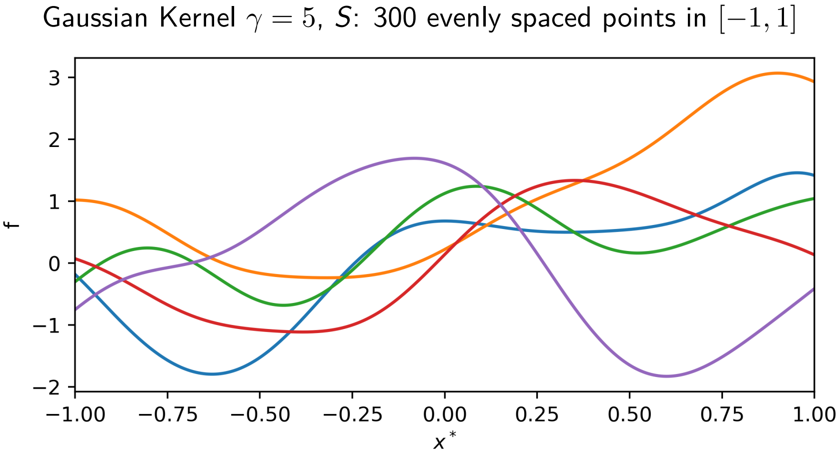
- Gaussian kernel

$$k(x, y) = \exp(-\gamma \|x - y\|^2)$$

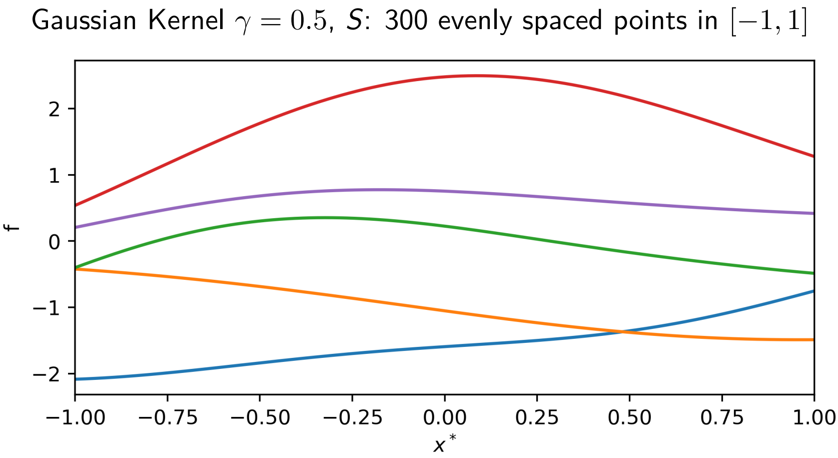
- Matern 3/2

$$k(x, y) = \left(1 + \frac{\sqrt{3}\|x - y\|}{\rho}\right) \exp\left(-\frac{\sqrt{5}\|x - y\|}{\rho}\right)$$

Universal Kernels: Example Draws from the Process

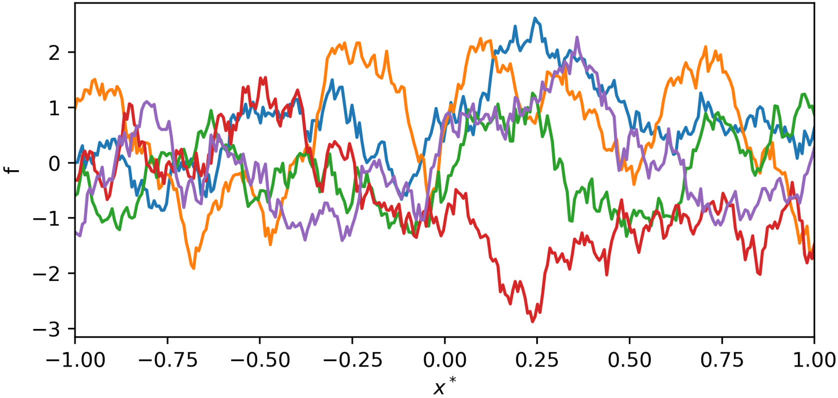


Universal Kernels: Example Draws from the Process



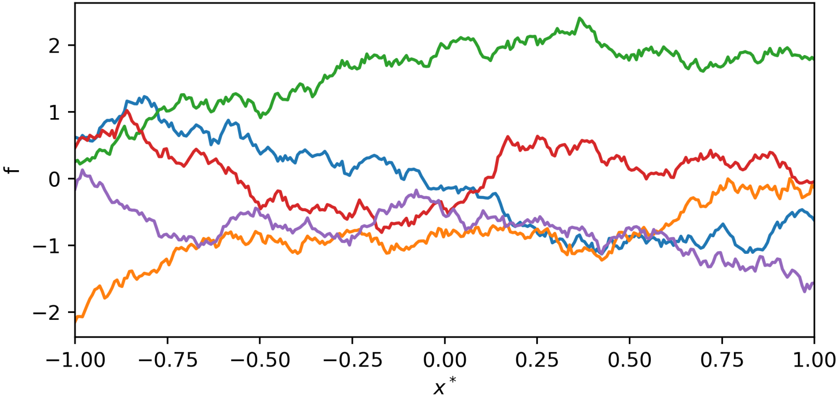
Universal Kernels: Example Draws from the Process

Matern 3/2 Kernel $\gamma = 5$, S: 300 evenly spaced points in $[-1, 1]$



Universal Kernels: Example Draws from the Process

Matern 3/2 Kernel $\gamma = 0.5$, S: 300 evenly spaced points in $[-1, 1]$



Reminder Random Fields

- The choice of Kernel and Parameters have a huge influence on the shape
  - Width of valleys
  - Ruggedness
  - Magnitude of function values
- We will see
  - The choice of kernel reflects what kind of function we expect to see
  - The choice of kernel has consequences on learning.

Mercer's Theorem (simplified)

There are two theorems connecting Gaussian Processes and Bayesian linear regression

Theorem

Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and bounded. Let $k : \mathcal{X} \times \mathcal{X}$ be a kernel. Then, there exists a sequence of features $\phi_1, \phi_2, \ldots$ such, that

$$k(x, x') = \sum_{i=1}^{\infty} \phi_i(x)\phi_i(x')$$

For universal kernels the sequence is infinite.

No proof.

Karhunen-Lowe-Theorem (simplified)

There are two theorems connecting Gaussian Processes and Bayesian linear regression

Theorem

Let $f \sim \mathcal{GP}(0, k(\cdot,\cdot))$, ϕ_i *given by Mercer's theorem and* $x \in \mathcal{X}$, *then the random variable*

$$\tilde{f}_N = \sum_{i=1}^{N} \theta_i \phi_i(x),\ \theta_i \sim \mathcal{N}(0,1)$$

converges to f_x *as* $N \rightarrow \infty$, *where convergence is measured in squared norm.*

No proof.

Full circle

We have seen:

- Bayesian linear regression $\to$ Gaussian Process
  - Any choice of ϕ and Σ_θ leads to a kernel
- Gaussian process $\to$ Bayesian Linear regression
  - Mercer: Any kernel leads to a feature map ϕ_i, $i = 1, \dots$
  - Karhunen-Loewe: Prior $\theta_i \sim \mathcal{N}(0, 1)$, $i = 1, \dots$

This means that feature-maps and kernels are two sides of the same coin. But a kernel can be much cheaper to compute than the feature-map.

Learning with $\mathcal{GP}$-Priors

Predicting using a $\mathcal{GP}$

- We have seen the connection of Gaussian Processes to Bayesian Linear Regression
- Priors on parameters can be turned to priors on observations
- Can we learn likely models given observations?

Regression using a $\mathcal{GP}$

Given noisy observations $(x_i, y_i = g(x) + \epsilon_i)$, $\epsilon_i \sim \mathcal{N}(0, \sigma_y^2)$ $i = 1, \ldots, \ell$, what is the distribution of f^* at new point x^*, when $f \sim \mathcal{GP}(0, k(\cdot, \cdot))$?

Idea Constrain the prior on likely candidate functions by the noisy observations:

1. Compute the normal distribution of the GP prior on the set $S \cup x^*$
2. Add the noise variance of the measurement noise at observed locations
3. Condition the normal distribution on the noisy measurements at the observed points

$\mathcal{GP}$ Regression: Likelihood

Given noisy observations $(x_i, y_i = y_{\text{true}} + \epsilon_i)$, $\epsilon_i \sim \mathcal{N}(0, \sigma_y^2)$ $i = 1, \dots, \ell$, what is the distribution of f^* at new point x^*, when $f \sim \mathcal{GP}(0, k(\cdot, \cdot))$?

- GP-Prior: $p(f^*, f_S | S \cup x^*) = p(f_{x^*}, \underbrace{f_{x_1}, \dots, f_{x_\ell}}_{f_S})$

We will compute this using the generative model, no manipulation of pdfs!

$\mathcal{GP}$ Regression: Likelihood

Given noisy observations $(x_i, y_i = y_{\text{true}} + \epsilon_i)$, $\epsilon_i \sim \mathcal{N}(0, \sigma_y^2)$ $i = 1, \ldots, \ell$, what is the distribution of f^* at new point x^*, when $f \sim \mathcal{GP}(0, k(\cdot, \cdot))$?

- GP-Prior: $p(f^*, f_S | S \cup x^*) = p(f_{x^*}, \underbrace{f_{x_1}, \ldots, f_{x_\ell}}_{f_S})$
- Noisy observations: $p(y|f_S) = \mathcal{N}(y; f_S, \sigma_y^2 I_\ell)$

We will compute this using the generative model, no manipulation of pdfs!

$\mathcal{GP}$ Regression: Likelihood

Given noisy observations $(x_i, y_i = y_{\text{true}} + \epsilon_i)$, $\epsilon_i \sim \mathcal{N}(0, \sigma_y^2)$ $i = 1, \ldots, \ell$, what is the distribution of f^* at new point x^*, when $f \sim \mathcal{GP}(0, k(\cdot, \cdot))$?

- GP-Prior: $p(f^*, f_S | S \cup x^*) = p(f_{x^*}, \underbrace{f_{x_1}, \ldots, f_{x_\ell}}_{f_S})$
- Noisy observations: $p(y|f_S) = \mathcal{N}(y; f_S, \sigma_y^2 I_\ell)$
- Marginalize: $p(f^*, y|S \cup x^*) = \int p(y|f_S) p(f^*, f_S|S \cup x^*) df_S$

We will compute this using the generative model, no manipulation of pdfs!

$\mathcal{GP}$ Regression: Likelihood

Given noisy observations $(x_i, y_i = y_{\text{true}} + \epsilon_i)$, $\epsilon_i \sim \mathcal{N}(0, \sigma_y^2)$ $i = 1, \ldots, \ell$, what is the distribution of f^* at new point x^*, when $f \sim \mathcal{GP}(0, k(\cdot, \cdot))$?

- GP-Prior: $p(f^*, f_S | S \cup x^*) = p(f_{x^*}, \underbrace{f_{x_1}, \ldots, f_{x_\ell}}_{f_S})$
- Noisy observations: $p(y|f_S) = \mathcal{N}(y; f_S, \sigma_y^2 I_\ell)$
- Marginalize: $p(f^*, y|S \cup x^*) = \int p(y|f_S) p(f^*, f_S|S \cup x^*) df_S$
- Condition: $p(f^*|y, S \cup x^*) = \frac{\int p(y|f_S) p(f^*, f_S|S \cup x^*) df_s}{p(y|S \cup x^*)}$

We will compute this using the generative model, no manipulation of pdfs!

Regression using a $\mathcal{GP}$

Given noisy observations $(x_i, y_i = y_{\text{true}} + \epsilon_i)$, $\epsilon_i \sim \mathcal{N}(0, \sigma_y^2)$ $i = 1, \ldots, k$, what is the distribution of f^* at new point x^*, when $f \sim \mathcal{GP}(0, k(\cdot,\cdot))$?

Compute the normal distribution of the GP prior on the set $S \cup x^*$: $p(f, f^* | S \cup x^*)$

$$\left[\begin{array}{c} f \\ \hline f^* \end{array}\right] \sim \mathcal{N}\left(0, \left[\begin{array}{c|c} K(S) & k(S, x^*) \\ \hline k(S, x^*)^T & k(x^*, x^*) \end{array}\right]\right)$$

Here, $k(S, x^*) = (k(x_1, x^*), \ldots, k(x_\ell, x^*))^T$

Marginal Likelihood of $\mathcal{GP}$-Regression

$$p(f^*, y|S \cup x^*) = \int p(y|f_S)p(f^*, f_S|S \cup x^*)df_S$$

Easy way to compute: look how y, f^* is generated:

$$\left[\begin{array}{c} y \\ \hline f^* \end{array}\right] = \underbrace{\begin{bmatrix} f \\ f^* \end{bmatrix}}_{\text{MVN}} + \underbrace{\begin{bmatrix} \epsilon \\ 0 \end{bmatrix}}_{\text{MVN}}, \quad \epsilon \sim \mathcal{N}(0, \sigma_y^2 I_\ell)$$

Marginal Likelihood of $\mathcal{GP}$-Regression

$$p(f^*, y|S \cup x^*) = \int p(y|f_S)p(f^*, f_S|S \cup x^*)df_S$$

Easy way to compute: look how y, f^* is generated:

$$\left[\frac{y}{f^*}\right] = \underbrace{\begin{bmatrix} f \\ f^* \end{bmatrix}}_{\text{MVN}} + \underbrace{\begin{bmatrix} \epsilon \\ 0 \end{bmatrix}}_{\text{MVN}}, \quad \epsilon \sim \mathcal{N}(0, \sigma_y^2 I_\ell)$$

Using sum of multivariate normal distributed variables rule:

$$\left[\frac{y}{f^*}\right] \sim \mathcal{N}\left(0, \left[\begin{array}{c|c} K(S) + \sigma_y^2 I_\ell & k(S, x^*) \\ \hline k(S, x^*)^T & k(x^*, x^*) \end{array}\right]\right)$$

Predicting using the process

Last step: condition on y:

Predicting using the process

Last step: condition on y:

$$
\begin{aligned}
f^*|y &\sim \mathcal{N}(\mu^*, \sigma^*) \\
\mu^* &= k(S, x^*)^T((S) + \sigma_y^2 I_\ell)^{-1} y \\
(\sigma^*)^2 &= k(x^*, x^*) - k(S, x^*)^T((S) + \sigma_y^2 I_\ell)^{-1} k(S, x^*)
\end{aligned}
$$

$\mathcal{GP}$-Regression: Algorithm (Simple)

Training:

- Pick kernel $k(\cdot,\cdot)$ and noise variance $\sigma_y > 0$
- Get data $(x_1, y_1), \dots, (x_1, y_1)$, $S = \{x_1, \dots x_\ell\}$
- Compute $G = (\sigma_y^2 I_N + K(S))^{-1}$ and $\alpha = Gy$

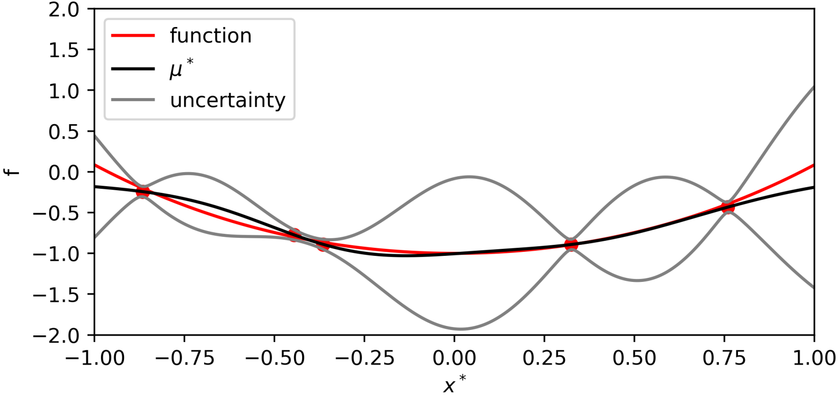
For a new point x^* to predict:

- Compute $\mu^* = k(S, x^*)^T \alpha$
- Compute $(\sigma^*)^2 = k(x^*, x^*) - k(S, x^*)^T G k(S, x^*)$

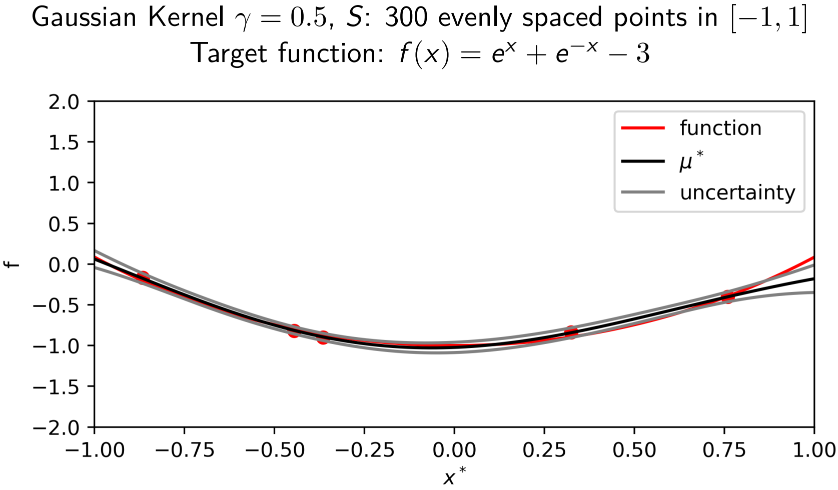
μ^*: maximum likelihood estimate for f^*
$\mu^* \pm 1.96(\sigma^*)^2$: 95% confidence interval for location of f^*

Visualization $p(f^*|y, S \cup x^*)$

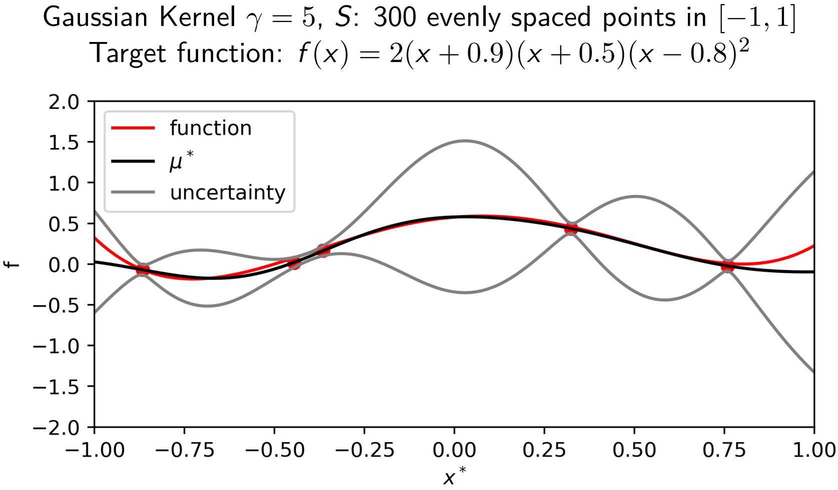
Gaussian Kernel $\gamma = 5$, S: 300 evenly spaced points in $[-1, 1]$
Target function: $f(x) = e^x + e^{-x} - 3$



Visualization $p(f^*|y, S \cup x^*)$

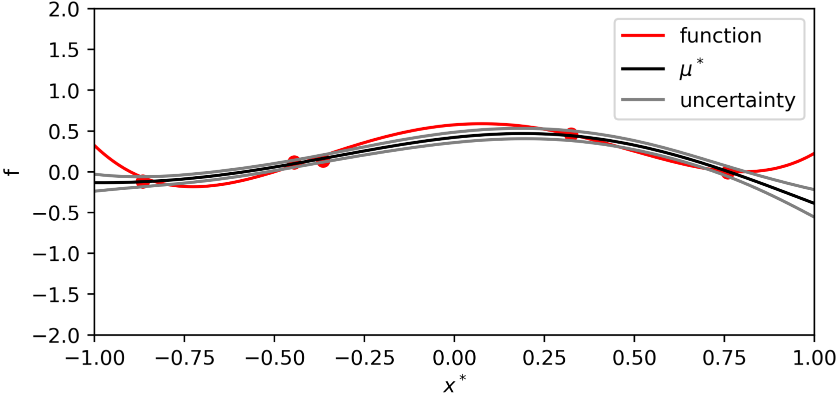


Visualization $p(f^*|y, S \cup x^*)$



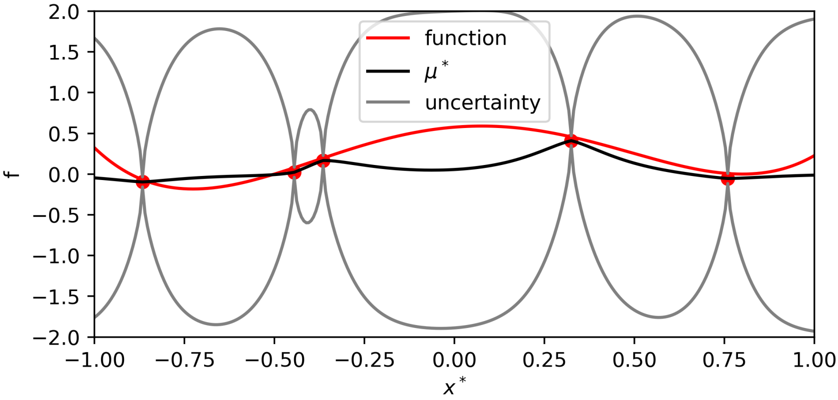
Visualization $p(f^*|y, S \cup x^*)$

Gaussian Kernel $\gamma = 0.5$, S: 300 evenly spaced points in $[-1, 1]$
Target function: $f(x) = 2(x + 0.9)(x + 0.5)(x - 0.8)^2$



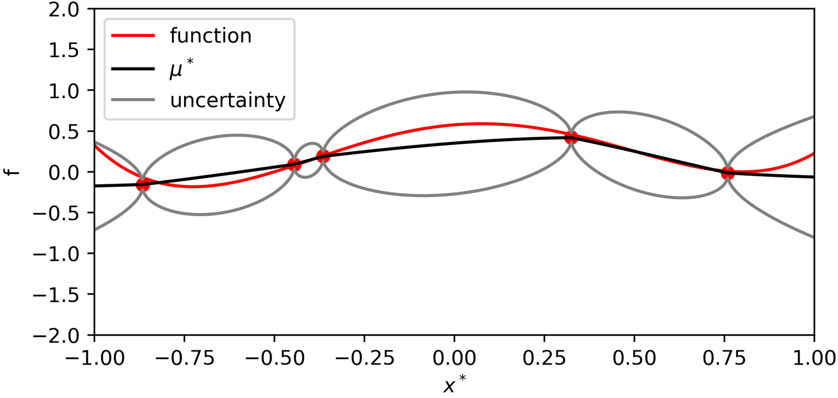
Visualization $p(f^*|y, S \cup x^*)$

Matern Kernel $\gamma = 5$, S: 300 evenly spaced points in $[-1, 1]$
Target function: $f(x) = 2(x + 0.9)(x + 0.5)(x - 0.8)^2$



Visualization $p(f^*|y, S \cup x^*)$

Matern Kernel $\gamma = 0.5$, S: 300 evenly spaced points in $[-1, 1]$
Target function: $f(x) = 2(x + 0.9)(x + 0.5)(x - 0.8)^2$

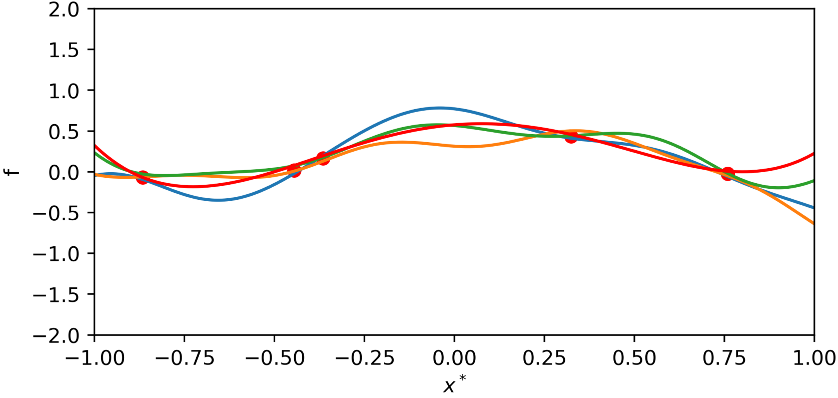


Sampling multiple points

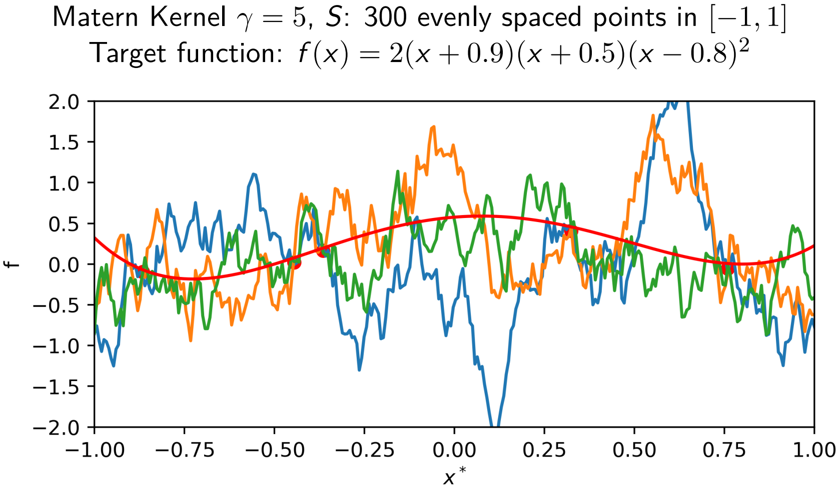
- So far we computed $p(f^*|y, S \cup x^*)$ for a single new point x^*
- We can redo the derivation for a set of points $S^* = \{x_1^*, \dots, x_M^*\}$
- This introduces additional dependencies on $f_{S^*} = (f_1^*, \dots, f_M^*)$.
- Can help us understand how real function samples between observations might look like.
- Derivation skipped for brevity

Visualization $p(f^*|y, S \cup x^*)$

Gaussian Kernel $\gamma = 5$, S: 300 evenly spaced points in $[-1, 1]$
Target function: $f(x) = 2(x + 0.9)(x + 0.5)(x - 0.8)^2$



Visualization $p(f^*|y, S \cup x^*)$



Conclusions from Visualizations

- Choice of kernel und its parameters radically impacts predictions
- Sampled posterior functions closely reflect kernel prior function shapes
- Uncertainties can be misleading
- Bayesian: Uncertainties are a belief, no verifiable fact.

Next: can we optimize the kernel?

Kernel optimization

Data Likelihood (Evidence)

How can we optimize the choice of kernel and parameterS?

- Given: k_η: kernel with parameter vector η
- Idea: pick the kernel parameters and noise σ_y^2 that make y most likely
- We call η and σ_y^2 hyperparameters.

Data Likelihood (also called Evidence):

$$p(y|S) = \int \underbrace{p(y|f_S)}_{\text{measurement noise}} \underbrace{p(f_S|S)}_{\text{GP prior}} df_S$$

Data Likelihood (Evidence)

How can we optimize the choice of kernel and parameterS?

- Given: k_η: kernel with parameter vector η
- Idea: pick the kernel parameters and noise σ_y^2 that make y most likely
- We call η and σ_y^2 hyperparameters.

Data Likelihood (explicit parameters):

$$p(y|S,\eta,\sigma_y^2) = \int p(y|f_S,\sigma_y^2)p(f_S|S,\eta)df_s$$

Data Likelihood (Evidence)

How can we optimize the choice of kernel and parameterS?

- Given: k_η: kernel with parameter vector η
- Idea: pick the kernel parameters and noise σ_y^2 that make y most likely
- We call η and σ_y^2 hyperparameters.

Data Likelihood (explicit parameters):

$$p(y|S, \eta, \sigma_y^2) = \mathcal{N}\left(y; 0, \sigma_y^2 I_\ell + K(S|\eta)\right)$$

$K(S|\eta)_{ij} = k_\eta(x_i, x_j)$

Data Likelihood

Data Likelihood:

$$p(y|S,\eta,\sigma_y^2) = \mathcal{N}\left(y; 0, \sigma_y^2 I_\ell + K(S|\eta)\right)$$

- Idea: find hyperparameters that maximize the log-likelihood
- Problem 1: This function is multi-modal, gradient-descent gets stuck

$\rightarrow$ Standard optimization: grid-search, random-search, gradient-descent with restart

- Problem 2: The likelihood is numerically unstable
  - Eigenvalues of $K(S) + \sigma_y^2 I_\ell$ are lower bounded by σ_y^2
  - Non-universal kernel and $\sigma_y^2 = 0 \rightarrow$ pdf might not exist

  $\rightarrow$ Pick numerical safe lower-bound for σ_y^2, e.g., 10^{-4}

Data Likelihood

Data Log-Likelihood:

$$\log p(y|S,\eta,\sigma_y^2) = -\frac{1}{2}y^T(\sigma_y^2 I_\ell + K_\eta(S))^{-1}y - \frac{1}{2}\log\det(\sigma_y^2 I_\ell + K_\eta(S)) - \frac{\ell}{2}\log\sqrt{2\pi}$$

- Idea: find hyperparameters that maximize the log-likelihood
- Problem 1: This function is multi-modal, gradient-descent gets stuck

$\rightarrow$ Standard optimization: grid-search, random-search, gradient-descent with restart

- Problem 2: The likelihood is numerically unstable
  - Eigenvalues of $K(S) + \sigma_y^2 I_\ell$ are lower bounded by σ_y^2
  - Non-universal kernel and $\sigma_y^2 = 0 \rightarrow$ pdf might not exist

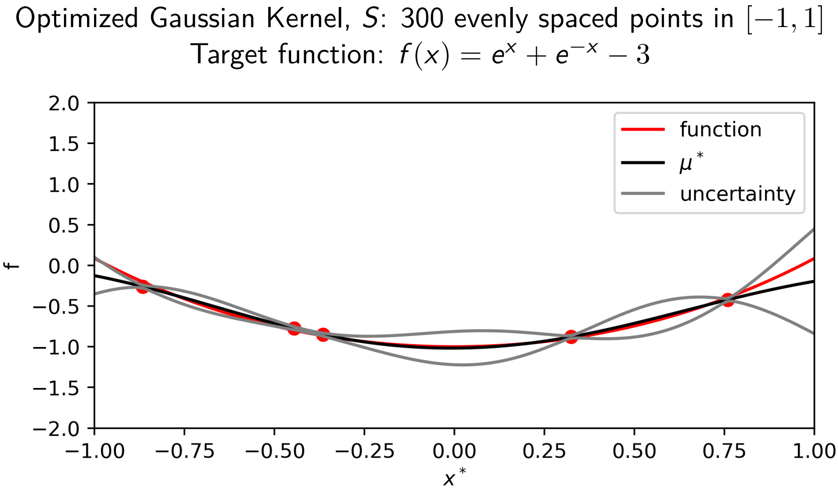
  $\rightarrow$ Pick numerical safe lower-bound for σ_y^2, e.g., 10^{-4}

Gridsearch, python

```
import scipy.optimize as opt #For Grid Search

def negLogLikelihood(params):
    noise_y = params[0]
    eta = params[1]
    ...
#noise_y and eta are bounded between 1.e-4 and 5
ranges = ((1.e-4,5), (1.e-4,5))
gridElements = 20
#run grid search, this algorithm does minimization
opt_params =opt.brute(negLogLikelihood, ranges,
    Ns=gridElements, finish=None).x
```

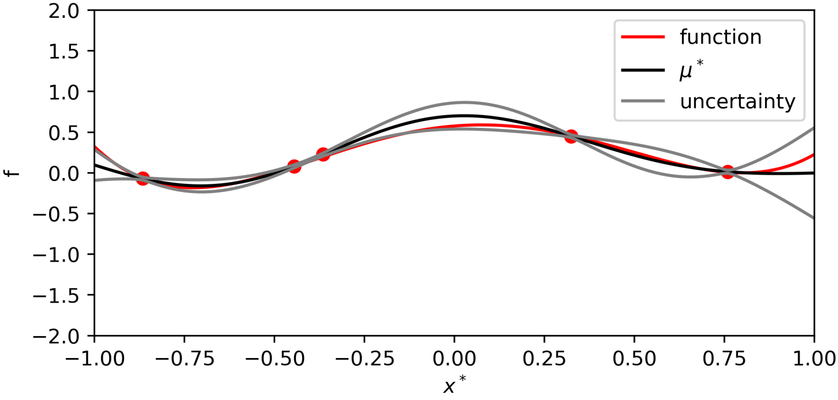
Optimized GP



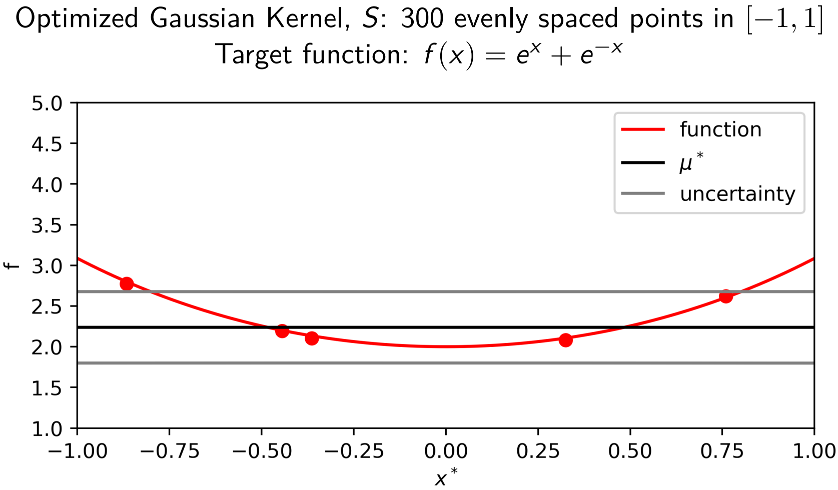
Optimizated GP

Optimized Gaussian Kernel, S: 300 evenly spaced points in $[-1, 1]$

Target function: $f(x) = 2(x + 0.9)(x + 0.5)(x - 0.8)^2$



Optimized GP



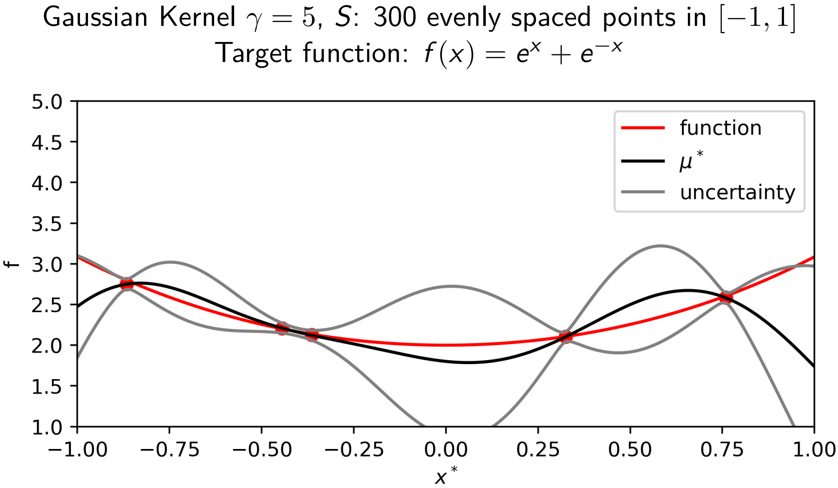
Observations

- First two functions: decent fit
- Third function: failure.
  - The third function is just an offseted version of the first
  - Parameters found: $\gamma = 10^{-4}$, $\sigma_y^2 = 0.26$
  - Optimized prior: approximately constant functions with lots of observation noise
  - Why?!?

Observations

- First two functions: decent fit
- Third function: failure.
  - The third function is just an offseted version of the first
  - Parameters found: $\gamma = 10^{-4}$, $\sigma_y^2 = 0.26$
  - Optimized prior: approximately constant functions with lots of observation noise
  - Why?!?

Visualization $p(f^*|y, S \cup x^*)$, third function



Towards An explanation

- Mean seems to be drawn towards 0 between observations
- Remember Mean function:

$$\mu^* = k(S, x^*)^T \alpha = \sum_{i=1}^{\ell} \alpha_i k(x_i, x^*)$$

- The Gaussian kernel is just a Gaussian hat!

$\rightarrow$ each $k(x_i, x^*)$ eventually goes to 0

$\rightarrow$ The Gaussian kernel assumes functions that fluctuate around 0.

How can we fix this?

- Solution 1: normalization
  - Gaussian kernel assumes functions with mena 0 and variance 1
  - Just normalize y before fitting the GP.
  - When predicting: undo normalization on the predicted value
- Solution 2: Adapt kernel
  - Add a "constant feature" to the kernel
  - Add a scaling parameter
  - How can we do that?

Kernel combinations

Let k_1, k_2 be kernels, $a > 0$, $b \in \mathbb{R}$. Kernel combination rules

- $k(x, x') = \sigma^2 k_1(x, x')$ is a kernel
- $k(x, x') = k_1(x, x') + k_2(x, x')$ is a kernel
- $k(x, x') = k_1(x, x') + a$ is a kernel

Interpretation:

- Scales the kernel variance
- Adds functions from the priors of both kernels
- Adds a constant function with unknown constant

Combining kernels is an art. There are more rules