Gaussian Processes
MATH 728

Will Consagra
Department of Statistics
University of South Carolina

Thanks to Dr. Ray Bai!

Outline

Section 1

Introduction to Gaussian Processes

What is a Gaussian Process?

Formal Definition

A stochastic process $\{f(x) : x \in \mathcal{X} \subset \mathbb{R}^d,\ d \geq 1\}$, where $\mathcal{X}$ is a continuous set, is called a *Gaussian process (GP)* if and only if for every finite set of points $x_1, \ldots, x_k$ in $\mathcal{X}$, the k-dimensional vector

$$\boldsymbol{f} := (f(x_1), \ldots, f(x_k))^\top$$

is a multivariate Gaussian random vector.

What is a Gaussian Process?

Formal Definition

A stochastic process $\{f(x) : x \in \mathcal{X} \subset \mathbb{R}^d,\ d \geq 1\}$, where $\mathcal{X}$ is a continuous set, is called a *Gaussian process (GP)* if and only if for every finite set of points $x_1, \ldots, x_k$ in $\mathcal{X}$, the k-dimensional vector

$$\boldsymbol{f} := (f(x_1), \ldots, f(x_k))^\top$$

is a multivariate Gaussian random vector.

Key Idea: GPs define distributions over functions.

- Fully specified by mean *function* $m(x)$ and covariance *function* $k(x, x')$
- Notation: $f(x) \sim \mathrm{GP}(m(x), k(x, x'))$

GP as a Distribution Over Functions

- Mean vector: $\boldsymbol{m}_X = (m(x_1), \ldots, m(x_k))^\top \in \mathbb{R}^K$
- Covariance matrix: $\mathbf{K}_{X,X} \in \mathbb{R}^{K \times K}$, with element-wise definition $\mathbf{K}_{X,X}(i,j) = k(x_i, x_j)$
- Joint distribution over our function discretization

$$\boldsymbol{f} \sim \mathcal{N}_k(\boldsymbol{m}_X, \mathbf{K}_{X,X})$$

We can get high-fidelity (high-resolution) approximation to the infinite-dimensional f samples by taking a finer grid X (a larger K).

Applications of Gaussian Processes

- **Modeling and simulation**: Brownian motion $f \sim GP(0, k)$, where $k(s, t) = \sigma^2 \min(s, t)$
  - Random motion of molecules
  - Random price movements in a financial market

Applications of Gaussian Processes

- **Modeling and simulation**: Brownian motion $f \sim GP(0, k)$, where $k(s, t) = \sigma^2 \min(s, t)$
  - Random motion of molecules
  - Random price movements in a financial market
- **Machine Learning:** Supervised regression/classification
  - Use a GP to approximate function mapping predictors to labels: $Y = f(X)$

Applications of Gaussian Processes

- **Modeling and simulation**: Brownian motion $f \sim GP(0, k)$, where $k(s, t) = \sigma^2 \min(s, t)$
  - Random motion of molecules
  - Random price movements in a financial market
- **Machine Learning:** Supervised regression/classification
  - Use a GP to approximate function mapping predictors to labels: $Y = f(X)$
- **"Uncertainty Quantification":**
  - Emulation: Approximate output of expensive computer code (PDE)
  - Inversion: Solving an inverse problem when your unknown is a function
  - Discrepancy: Non-parametric model for data-model mismatch

Section 2

Gaussian Process Regression

Nonparametric vs Parametric Regression

We observe data $\mathcal{D} := (x_1, y_1), ..., (x_n, y_n) \in \mathbb{R}^p \times \mathbb{R}$

Nonparametric vs Parametric Regression

We observe data $\mathcal{D} := (x_1, y_1), ..., (x_n, y_n) \in \mathbb{R}^p \times \mathbb{R}$

Parametric

- Data generating model: $y_i = x_i^\top \beta + \varepsilon_i$, $\mathbb{E}[\varepsilon_i] = 0$
- Estimate with OLS (frequentist)

$$\widehat{\beta} = \min_{\beta} \sum_{i=1}^{n} (y_i - x_i^\top \beta)^2$$

  or Bayesian $p(\beta|\mathcal{D}) \propto p(\mathcal{D}|\beta)p(\beta)$
- Confidence intervals/credible intervals can be formed using standard techniques

Nonparametric vs Parametric Regression

We observe data $\mathcal{D} := (x_1, y_1), ..., (x_n, y_n) \in \mathbb{R}^p \times \mathbb{R}$

Parametric

- Data generating model: $y_i = x_i^\top \beta + \varepsilon_i$, $\mathbb{E}[\varepsilon_i] = 0$
- Estimate with OLS (frequentist)

$$\widehat{\beta} = \min_{\beta} \sum_{i=1}^{n} (y_i - x_i^\top \beta)^2$$

  or Bayesian $p(\beta|\mathcal{D}) \propto p(\mathcal{D}|\beta)p(\beta)$
- Confidence intervals/credible intervals can be formed using standard techniques

Nonparametric:

- Data generating model: $y_i = f(x_i) + \varepsilon_i$, where f is infinite-dimensional (e.g. belongs to some function space).
- Frequentist: Kernel smoothing, basis expansion, neural networks, etc.
- Bayesian: Use a GP prior on f and calculate the posterior "$p(f|\mathcal{D})$"

Gaussian Process Regression Model

Bayesian Model

Prior $f|k \sim \mathrm{GP}(0, k(x, x'))$

Likelihood $y_i|f, \sigma^2 = N(f(x_i), \sigma^2) \quad i = 1, ..., n$

where $\sigma^2 = \mathrm{Var}[\varepsilon_i]$.

Gaussian Process Regression Model

Bayesian Model

$$\textbf{Prior} \qquad f|k \sim \mathrm{GP}(0, k(x, x'))$$
$$\textbf{Likelihood} \qquad y_i|f, \sigma^2 = N(f(x_i), \sigma^2) \quad i = 1, ..., n$$

where $\sigma^2 = \mathrm{Var}[\varepsilon_i]$.

Posterior

- We will define the posterior over some finite discretization
  - Can be points or basis function, we'll focus on points.
- Define a discretization of the input ("test points") $X^* := (x_1^*, ..., x_k^*)$, and the corresponding function values $\mathbf{f}^* := (f(x_1^*), ..., f(x_k^*))$.
- Our goal is perform Bayesian inference to obtain the posterior $p(\mathbf{f}^*|\mathcal{D})$ under the Bayesian model above.

Specifying the Functional Prior: Covariance Functions

- The choice of covariance function k (and its hyperparameters) in our GP prior reflect prior beliefs on the function smoothness, “wiggliness”, periodicity, etc.

Specifying the Functional Prior: Covariance Functions

- The choice of covariance function k (and its hyperparameters) in our GP prior reflect prior beliefs on the function smoothness, "wiggliness", periodicity, etc.

Common Kernels:

- Squared Exponential kerenl: $k(x, x') = \tau^2 \exp\left(-\frac{\|x-x'\|^2}{2\ell^2}\right)$

Specifying the Functional Prior: Covariance Functions

- The choice of covariance function k (and its hyperparameters) in our GP prior reflect prior beliefs on the function smoothness, "wiggliness", periodicity, etc.

Common Kernels:

- Squared Exponential kerenl: $k(x, x') = \tau^2 \exp\left(-\frac{\|x-x'\|^2}{2\ell^2}\right)$
  - **Assumption**: f is an infinitely differentiable (very smooth) function

Specifying the Functional Prior: Covariance Functions

- The choice of covariance function k (and its hyperparameters) in our GP prior reflect prior beliefs on the function smoothness, "wiggliness", periodicity, etc.

Common Kernels:

- Squared Exponential kerenl: $k(x, x') = \tau^2 \exp\left(-\frac{\|x-x'\|^2}{2\ell^2}\right)$
  - **Assumption**: f is an infinitely differentiable (very smooth) function
- Ornstein-Uhlenbeck kernel: $k(x, x') = \tau^2 \exp\left(-\frac{\|x-x'\|_2}{\ell}\right)$

Specifying the Functional Prior: Covariance Functions

- The choice of covariance function k (and its hyperparameters) in our GP prior reflect prior beliefs on the function smoothness, "wiggliness", periodicity, etc.

Common Kernels:

- Squared Exponential kerenl: $k(x, x') = \tau^2 \exp\left(-\frac{\|x-x'\|^2}{2\ell^2}\right)$
  - **Assumption**: f is an infinitely differentiable (very smooth) function
- Ornstein-Uhlenbeck kernel: $k(x, x') = \tau^2 \exp\left(-\frac{\|x-x'\|_2}{\ell}\right)$
  - **Assumption**: f is a continuous but not differentiable (not very smooth) function

Specifying the Functional Prior: Covariance Functions

- The choice of covariance function k (and its hyperparameters) in our GP prior reflect prior beliefs on the function smoothness, "wiggliness", periodicity, etc.

Common Kernels:

- Squared Exponential kerenl: $k(x, x') = \tau^2 \exp\left(-\frac{\|x-x'\|^2}{2\ell^2}\right)$
  - **Assumption**: f is an infinitely differentiable (very smooth) function
- Ornstein-Uhlenbeck kernel: $k(x, x') = \tau^2 \exp\left(-\frac{\|x-x'\|_2}{\ell}\right)$
  - **Assumption**: f is a continuous but not differentiable (not very smooth) function
- Periodic kernel: $k(x, x') = \tau^2 \exp\left(-\frac{2\sin^2(\pi\|x-x'\|/p)}{\ell^2}\right)$

Specifying the Functional Prior: Covariance Functions

- The choice of covariance function k (and its hyperparameters) in our GP prior reflect prior beliefs on the function smoothness, "wiggliness", periodicity, etc.

Common Kernels:

- Squared Exponential kerenl: $k(x, x') = \tau^2 \exp\left(-\frac{\|x-x'\|^2}{2\ell^2}\right)$
  - **Assumption**: f is an infinitely differentiable (very smooth) function
- Ornstein-Uhlenbeck kernel: $k(x, x') = \tau^2 \exp\left(-\frac{\|x-x'\|_2}{\ell}\right)$
  - **Assumption**: f is a continuous but not differentiable (not very smooth) function
- Periodic kernel: $k(x, x') = \tau^2 \exp\left(-\frac{2\sin^2(\pi\|x-x'\|/p)}{\ell^2}\right)$
  - **Assumption**: f is a periodic over some period p

Prior Draws from Different Covariance Functions

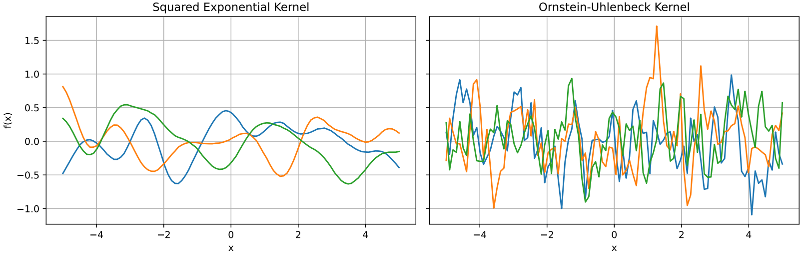


Figure: Samples from mean-zero GP priors with different kernels to encode different assumptions on the function smoothness.

Measurement Model

- Recall that we have the data generating model for $i = 1, ..., n$:

$$y_i = f(x_i) + \varepsilon_i, \varepsilon_i \sim N(0, \sigma^2)$$

- Denote $\boldsymbol{y} = (y_1, ..., y_n)^\intercal$, $\boldsymbol{f} = (f(x_1), ..., f(x_n))^\intercal$, $\boldsymbol{\varepsilon} = (\varepsilon_1, ..., \varepsilon_n)$.
- The vectorized model is given by

$$\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\epsilon}$$

Measurement Model

- Recall that we have the data generating model for $i = 1, ..., n$:

$$y_i = f(x_i) + \varepsilon_i, \varepsilon_i \sim N(0, \sigma^2)$$

- Denote $\boldsymbol{y} = (y_1, ..., y_n)^\intercal$, $\boldsymbol{f} = (f(x_1), ..., f(x_n))^\intercal$, $\boldsymbol{\varepsilon} = (\varepsilon_1, ..., \varepsilon_n)$.
- The vectorized model is given by

$$\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\epsilon}$$

Using standard results from multivariate statistics

- $\mathbb{E}(\mathbf{y}) = \mathbb{E}[\mathbb{E}(\mathbf{y} \mid \mathbf{f})] = \mathbb{E}(\mathbf{f}) = \mathbf{0}_n$
- $\mathrm{Cov}(\mathbf{y}) = \mathrm{Cov}(\mathbf{f} + \varepsilon) = \mathrm{Cov}(\mathbf{f}) + \mathrm{Cov}(\varepsilon) = \mathbf{K}_{X,X} + \sigma^2 \mathbf{I}_n$
- $\mathrm{Cov}(\mathbf{y}, \mathbf{f}_\star) = \mathrm{Cov}(\mathbf{f} + \varepsilon, \mathbf{f}_\star) = \mathrm{Cov}(\mathbf{f}, \mathbf{f}_\star) = \mathbf{K}_{X,X_\star}$
- $\mathrm{Cov}(\mathbf{f}_\star, \mathbf{y}) = [\mathrm{Cov}(\mathbf{y}, \mathbf{f}_\star)]^\top = \mathbf{K}_{X_\star,X}$

Deriving the Posterior Predictive Distribution

Approach I: Properties of Conditional Multivariate Normals

- **Joint Distribution:**

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_\star \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{X,X} + \sigma^2 \mathbf{I} & \mathbf{K}_{X,X_\star} \\ \mathbf{K}_{X_\star,X} & \mathbf{K}_{X_\star,X_\star} \end{bmatrix} \right),$$

where this comes from the means and covariances derived on the previous side.

Deriving the Posterior Predictive Distribution

Approach I: Properties of Conditional Multivariate Normals

- **Joint Distribution:**

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_\star \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{X,X} + \sigma^2 \mathbf{I} & \mathbf{K}_{X,X_\star} \\ \mathbf{K}_{X_\star,X} & \mathbf{K}_{X_\star,X_\star} \end{bmatrix} \right),$$

where this comes from the means and covariances derived on the previous side.

- **Posterior:**

$$p(\mathbf{f}_\star|\mathbf{y}) \sim \mathcal{N}(\boldsymbol{\mu}_\star, \boldsymbol{\Sigma}_\star)$$

where

$$\boldsymbol{\mu}_\star = \mathbf{K}_{X_\star,X}[\mathbf{K}_{X,X} + \sigma^2 \mathbf{I}_n]^{-1}\mathbf{y}$$

$$\boldsymbol{\Sigma}_\star = \mathbf{K}_{X_\star,X_\star} - \mathbf{K}_{X_\star,X}[\mathbf{K}_{X,X} + \sigma^2 \mathbf{I}_n]^{-1}\mathbf{K}_{X,X_\star},$$

and this follows from standard conditioning properties of multivariate Gaussian.

Deriving the Posterior Predictive Distribution

Approach II: Bayes Rule

$$p(\mathbf{f}, \mathbf{f}_\star \mid \mathbf{y}) \propto \underbrace{\mathcal{N}(\mathbf{y} \mid \mathbf{f}, \sigma^2 \mathbf{I}_n)}_{\text{Likelihood}} \times \underbrace{\mathcal{N}\left(\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_\star \end{bmatrix} \middle| \mathbf{0}, \mathbf{K} \right)}_{\text{Prior}} \tag{12}$$

- Use the *completing the square* trick to derive the joint normal posterior $p(\mathbf{f}, \mathbf{f}_\star \mid \mathbf{y})$.
- Then apply the **marginalization property** of Gaussians to get:
  - the marginal posterior $p(\mathbf{f} \mid \mathbf{y})$
  - the predictive posterior $p(\mathbf{f}_\star \mid \mathbf{y})$

This is a standard derivation that you can find online.

Estimation and Uncertainty Quantification

Using the posterior predictive distribution we can compute:

- **Point estimates:** For each test point $x_{\star i}$, the predicted value is the posterior mean

$$\hat{f}(x_{\star i}) = \mu_{\star i}, \quad i = 1, \ldots, m,$$

  which is also the MAP since we are dealing with Gaussians.

- **95% Prediction intervals:** The interval for $f(x_{\star i})$ is based on the 2.5th and 97.5th percentiles of $\mathcal{N}(\mu_{\star i}, \Sigma_{\star ii})$:

$$\left[\mu_{\star i} - z_{0.975}\sqrt{\Sigma_{\star ii}},\ \mu_{\star i} + z_{0.975}\sqrt{\Sigma_{\star ii}}\right], \quad i = 1, \ldots, m$$

  GPs have a very fast, closed form expression for uncertainty quantification that avoids costly MCMC-type sampling!

Note: $z_{0.975} \approx 1.96$ for a standard normal distribution.

Toy Example

True function: $f(x) = x \sin(x)$
Measurement error variance $\sigma^2 = 0.3^2$

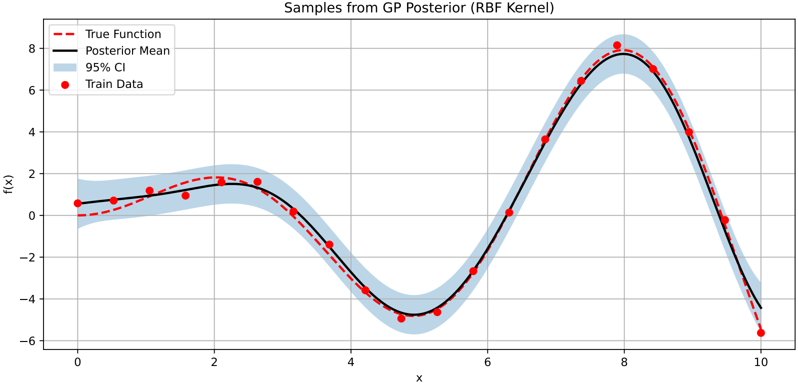


Figure: Posterior GP inference on dense grid.

Computational

Exact GP Inferenence:

- Requires inverting $n \times n$ matrix: $[\mathbf{K}_{X,X} + \sigma^2 \mathbf{I}_n]^{-1}$
- Time complexity $\mathcal{O}(n^3)$, space complexity $\mathcal{O}(n^2)$
- Intractable for large n (e.g., $n > 10{,}000$)

Computational

Exact GP Inferenence:

- Requires inverting $n \times n$ matrix: $[\mathbf{K}_{X,X} + \sigma^2 \mathbf{I}_n]^{-1}$
- Time complexity $\mathcal{O}(n^3)$, space complexity $\mathcal{O}(n^2)$
- Intractable for large n (e.g., $n > 10{,}000$)

In many modern applications, $n > 1{,}000{,}000$, e.g. 3D brain imaging!

What can we do?

Computational

Exact GP Inferenence:

- Requires inverting $n \times n$ matrix: $[\mathbf{K}_{X,X} + \sigma^2 \mathbf{I}_n]^{-1}$
- Time complexity $\mathcal{O}(n^3)$, space complexity $\mathcal{O}(n^2)$
- Intractable for large n (e.g., $n > 10{,}000$)

In many modern applications, $n > 1{,}000{,}000$, e.g. 3D brain imaging!

What can we do?
Possible Solutions: Approximate the distribution Sparse GP Approximations (using $q << n$ inducing inputs)

For more details, see:
Approximation Methods for Gaussian Process Regression,
Quiñonero-Candela, Rasmussen, and Williams.

Section 3

Hyperparameter Selection

Hyperparameter Selection

Marginal Maximum Likelihood

- Previous analysis assumes kernel parameters (e.g. ℓ, τ) and noise σ^2 are known
  - This is often not the case!

Hyperparameter Selection

Marginal Maximum Likelihood

- Previous analysis assumes kernel parameters (e.g. ℓ, τ) and noise σ^2 are known
  - This is often not the case!
- Proper hyperparameter selection can have an enormous effect on the quality of the inference, both in terms of estimation accuracy and well-calibrated uncertainty.

Hyperparameter Selection

Marginal Maximum Likelihood

- Previous analysis assumes kernel parameters (e.g. ℓ, τ) and noise σ^2 are known
  - This is often not the case!
- Proper hyperparameter selection can have an enormous effect on the quality of the inference, both in terms of estimation accuracy and well-calibrated uncertainty.
- We will use the *marginal likelihood*

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f})d\boldsymbol{f} = \mathcal{N}_n(\mathbf{0}, \boldsymbol{K}_{X,X} + \sigma^2 \boldsymbol{I}_n)$$

  - "The probability of the observed data $\boldsymbol{y}$ given the model ($y = f(x) + \varepsilon$) over the prior ($p(f)$)".

Hyperparameter Selection

Marginal Maximum Likelihood

- Previous analysis assumes kernel parameters (e.g. ℓ, τ) and noise σ^2 are known
  - This is often not the case!
- Proper hyperparameter selection can have an enormous effect on the quality of the inference, both in terms of estimation accuracy and well-calibrated uncertainty.
- We will use the *marginal likelihood*

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f})d\boldsymbol{f} = \mathcal{N}_n(\mathbf{0}, \boldsymbol{K}_{X,X} + \sigma^2 \boldsymbol{I}_n)$$

  - "The probability of the observed data $\boldsymbol{y}$ given the model $(y = f(x) + \varepsilon)$ over the prior $(p(f))$".
- Select the parameters

$$(\widehat{\ell}, \widehat{\tau}, \widehat{\sigma}^2) = \max_{(\ell,\tau,\sigma^2)} p(\boldsymbol{y}),$$

  using some numerical optimizer.

Hyperparameter Selection

Hyperpriors

- A "fully Bayesian" approach places priors on hyperparameters:

$$\sigma^2 \sim p(\sigma^2), \quad (\ell, \tau) \sim p(\ell)p(\tau)$$

Hyperparameter Selection

Hyperpriors

- A "fully Bayesian" approach places priors on hyperparameters:

$$\sigma^2 \sim p(\sigma^2), \quad (\ell, \tau) \sim p(\ell)p(\tau)$$

- These are called *hyperpriors*, and model our uncertainty in hyperparameter values.

Hyperparameter Selection

Hyperpriors

- A "fully Bayesian" approach places priors on hyperparameters:

$$\sigma^2 \sim p(\sigma^2), \quad (\ell, \tau) \sim p(\ell)p(\tau)$$

- These are called *hyperpriors*, and model our uncertainty in hyperparameter values.
- But marginalizing over unknown hyperparameters involves an integral:

$$p(\mathbf{f}_\star \mid \mathbf{y}) = \int p(\mathbf{f}_\star, \ell, \tau, \sigma^2 \mid \mathbf{y})\, d\sigma^2\, d\ell d\tau \tag{1}$$

  and hence $p(\mathbf{f}_\star \mid \mathbf{y})$ is no longer multivariate normal, but some more complicated distribution.

Hyperparameter Selection

Hyperpriors

- A "fully Bayesian" approach places priors on hyperparameters:

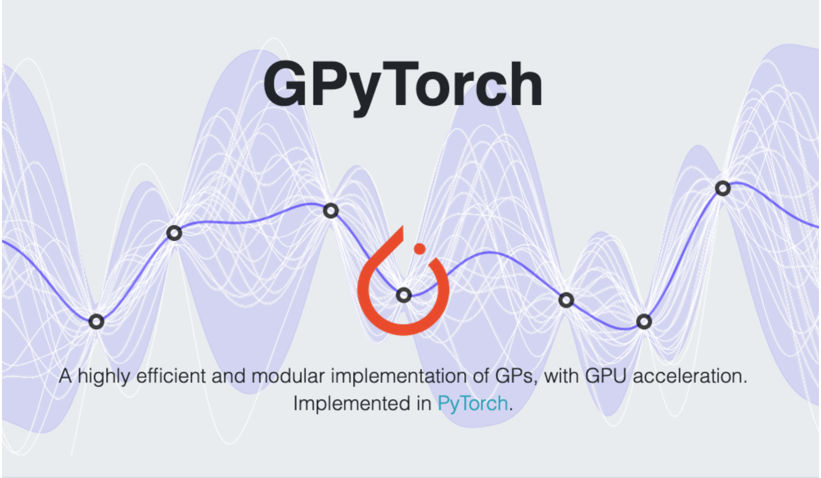
$$\sigma^2 \sim p(\sigma^2), \quad (\ell, \tau) \sim p(\ell)p(\tau)$$

- These are called *hyperpriors*, and model our uncertainty in hyperparameter values.
- But marginalizing over unknown hyperparameters involves an integral:

$$p(\mathbf{f}_\star \mid \mathbf{y}) = \int p(\mathbf{f}_\star, \ell, \tau, \sigma^2 \mid \mathbf{y}) \, d\sigma^2 \, d\ell d\tau \tag{1}$$

  and hence $p(\mathbf{f}_\star \mid \mathbf{y})$ is no longer multivariate normal, but some more complicated distribution.
- To approximate the posterior (1) we will have to use a more complicated inference algorithm:
  - Some Markov chain Monte Carlo (MCMC) variant or variational inference

Software


GPyTorch
A highly efficient and modular implementation of GPs, with GPU acceleration.
Implemented in PyTorch.