Chapter 6, Part 2: Fitting Gaussian Processes to Data

Advanced Topics in Statistical Machine Learning

Tom Rainforth

Hilary 2024

rainforth@stats.ox.ac.uk

Gaussian Process Regression

By conditioning on observations, we can use Gaussian processes (GPs) to perform a Bayesian **nonparametric** regression that provides **uncertainty estimates** rather than just the best fit

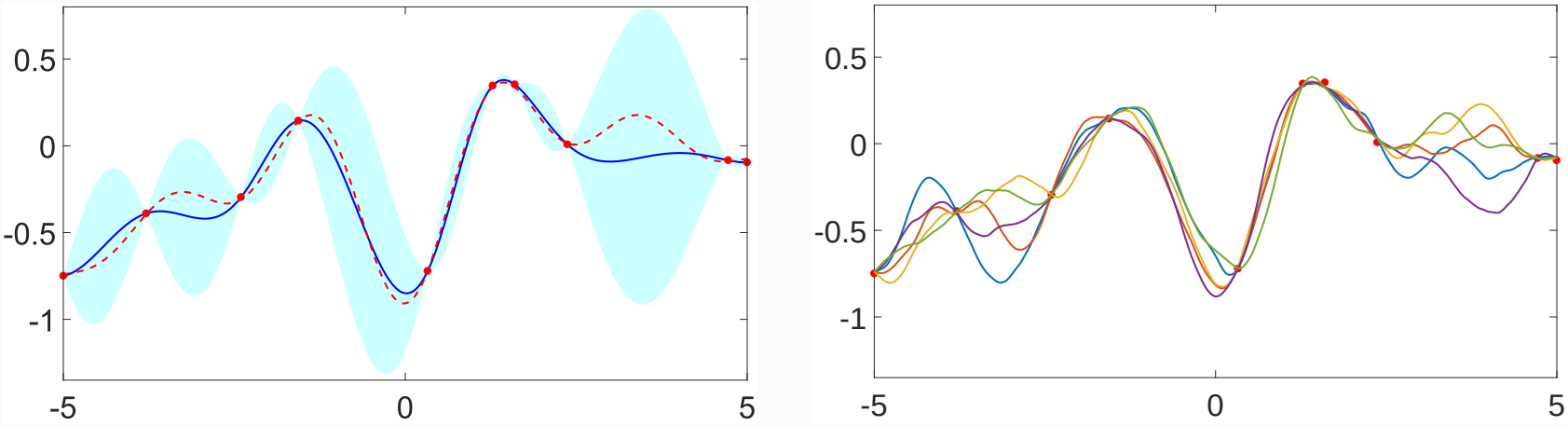


Figure 1: Example GP regression for points shown in red dots. [Left] GP posterior where the solid line is the mean prediction, the shading is the mean $\pm$ 2 standard deviations, and the dashed line is the ground truth function. [Right] five example functions drawn from this posterior.

Basic Results of Gaussians

There are a number of important results for multivariate Gaussians that we can exploit for GPs:[1]

- All affine transformations of Gaussian random vectors are also Gaussian distributed (by extension, sums of jointly Gaussian random vectors are also Gaussian)
- If X is Gaussian and $Y|X \sim \mathcal{N}(AX + \mu, \Sigma)$ for some A, μ, and Σ that do not depend on X, then $\begin{bmatrix} X \\ Y \end{bmatrix}$ is jointly Gaussian (note this includes the case where X and Y are independent)
- A Gaussian prior $p(\theta) = \mathcal{N}(\theta; \mu, \Sigma)$ is **conjugate** to Gaussian likelihoods of the form $p(\mathcal{D}|\theta) = \prod_{i=1}^{n} \mathcal{N}(y_i; \theta^T x_i, \sigma^2)$, which means that they also produce a Gaussian posterior

[1] Chapter 8 of `https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf` is very helpful reference here

Gaussian Marginals and Conditionals

Another important rule is that all marginal and conditional distributions of jointly Gaussian random variables are also Gaussian with the following forms

Gaussian Marginals and Conditionals

Let $\mathbf{z} \sim \mathcal{N}(\mu, \Sigma)$ and consider splitting its dimensions as

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}, \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}.$$

Note that $\Sigma_{21} = \Sigma_{12}^\top$ due to symmetry. The marginal and conditional distributions are given as follows

$$p(\mathbf{z}_1) = \mathcal{N}(\mathbf{z}_1; \mu_1, \Sigma_{11})$$
$$p(\mathbf{z}_2|\mathbf{z}_1) = \mathcal{N}(\mathbf{z}_2;\ \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{z}_1 - \mu_1),\ \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}).$$

Standard Model for Gaussian Process Regression

Let $\mathbf{x} = \{x_i\}_{i=1}^n$ and $\mathbf{y} = [y_1, \ldots, y_n]^\top \in \mathbb{R}^n$ denote our input and output training data respectively.

We will model this with a zero-mean GP with covariance function k, such that we can say $f \sim \mathrm{GP}(0, k)$ and our predictions are $f(x)$

We will couple this with an independent Gaussian likelihood, such that

$$y_i | f \sim \mathcal{N}(f(x_i), \sigma^2)$$

Denoting the random vector $\mathbf{f} = [f(x_1), \ldots, f(x_n)]^\top \in \mathbb{R}^n$ and the covariance (kernel) matrix $(\mathbf{K_{xx}})_{ij} = k(x_i, x_j)$, this gives the following generative model for $\mathbf{y}$

$$\mathbf{f} \sim \mathcal{N}(0, \mathbf{K_{xx}}), \qquad \mathbf{y}|\mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma^2 I)$$

Here $\mathbf{f}$ and $\mathbf{y}$ are jointly Gaussian with $\mathbb{E}[\mathbf{f}] = \mathbb{E}[\mathbf{y}] = 0$.

Predictive Distribution

Suppose now we wish to make predictions at some set of new points $\mathbf{x}' = \{x'_j\}_{j=1}^m$, such that we wish to reason about the corresponding evaluations $\mathbf{f}' = [f(x'_1), \ldots, f(x'_m)]^\top \in \mathbb{R}^m$.

By the definition of a GP, $\mathbf{f}$ and $\mathbf{f}'$ are jointly Gaussian, while $\mathbf{y}$ and $\mathbf{f}'$ are conditionally independent given $\mathbf{f}$

This implies that $\mathbf{f}$, $\mathbf{f}'$, and $\mathbf{y}$ are all jointly Gaussian, and thus the marginal joint on $\mathbf{f}'$ and $\mathbf{y}$ is itself Gaussian.

We thus have

$$\begin{bmatrix} \mathbf{f}' \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathrm{Cov}(\mathbf{f}', \mathbf{f}') & \mathrm{Cov}(\mathbf{f}', \mathbf{y}) \\ \mathrm{Cov}(\mathbf{f}', \mathbf{y})^\top & \mathrm{Cov}(\mathbf{y}, \mathbf{y}) \end{bmatrix} \right) \tag{1}$$

Predictive Distribution

Defining the matrices $(\mathbf{K}_{\mathbf{x}'\mathbf{x}'})_{ij} = k(x'_i, x'_j)$, $(\mathbf{K}_{\mathbf{x}'\mathbf{x}})_{ij} = k(x'_i, x_j)$, we can now calculate each of these terms as follows by noting that we can write $\mathbf{y} = \mathbf{f} + \sigma\boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ and $\mathbf{f}$ and $\boldsymbol{\epsilon}$ are marginally independent

$$
\begin{aligned}
\text{Cov}(\mathbf{f}', \mathbf{f}') &= \mathbf{K}_{\mathbf{x}'\mathbf{x}'} \\
\text{Cov}(\mathbf{f}', \mathbf{y}) &= \mathbb{E}[\mathbf{f}'\mathbf{y}^T] \\
&= \mathbb{E}[\mathbf{f}'\mathbf{f}^T] + \sigma\,\mathbb{E}[\mathbf{f}'\boldsymbol{\epsilon}^T] \\
&= \mathbf{K}_{\mathbf{x}'\mathbf{x}} \\
\text{Cov}(\mathbf{y}, \mathbf{y}) &= \mathbb{E}[\mathbf{y}\mathbf{y}^T] \\
&= \mathbb{E}[\mathbf{f}\mathbf{f}^T] + \sigma\,\mathbb{E}[\mathbf{f}\boldsymbol{\epsilon}^T + \boldsymbol{\epsilon}\mathbf{f}^T] + \sigma^2\,\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] \\
&= \mathbf{K}_{\mathbf{x}\mathbf{x}} + \sigma^2 I
\end{aligned}
$$

Predictive Distribution

We thus have

$$\begin{bmatrix} \mathbf{f}' \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{\mathbf{x}'\mathbf{x}'} & \mathbf{K}_{\mathbf{x}'\mathbf{x}} \\ \mathbf{K}_{\mathbf{x}'\mathbf{x}}^\top & \mathbf{K}_{\mathbf{x}\mathbf{x}} + \sigma^2 I \end{bmatrix} \right) \tag{2}$$

Applying our conditional formula from earlier now gives our GP predictive distribution $\mathbf{f}'|\mathbf{y}$ as follows

Gaussian Process Regression Predictive Distribution

$$\mathbf{f}'|\mathbf{y} \sim \mathcal{N}\left(\mathbf{K}_{\mathbf{x}'\mathbf{x}}(\mathbf{K}_{\mathbf{x}\mathbf{x}} + \sigma^2 I)^{-1}\mathbf{y}\,,\, \mathbf{K}_{\mathbf{x}'\mathbf{x}'} - \mathbf{K}_{\mathbf{x}'\mathbf{x}}(\mathbf{K}_{\mathbf{x}\mathbf{x}} + \sigma^2 I)^{-1}\mathbf{K}_{\mathbf{x}'\mathbf{x}}^\top\right)$$

As $\mathbf{x}'$ was totally arbitrary, we can now use this to make predictions at any desired set of points by **jointly** drawing from $\mathbf{f}'|\mathbf{y}$

Example Predictive Distribution

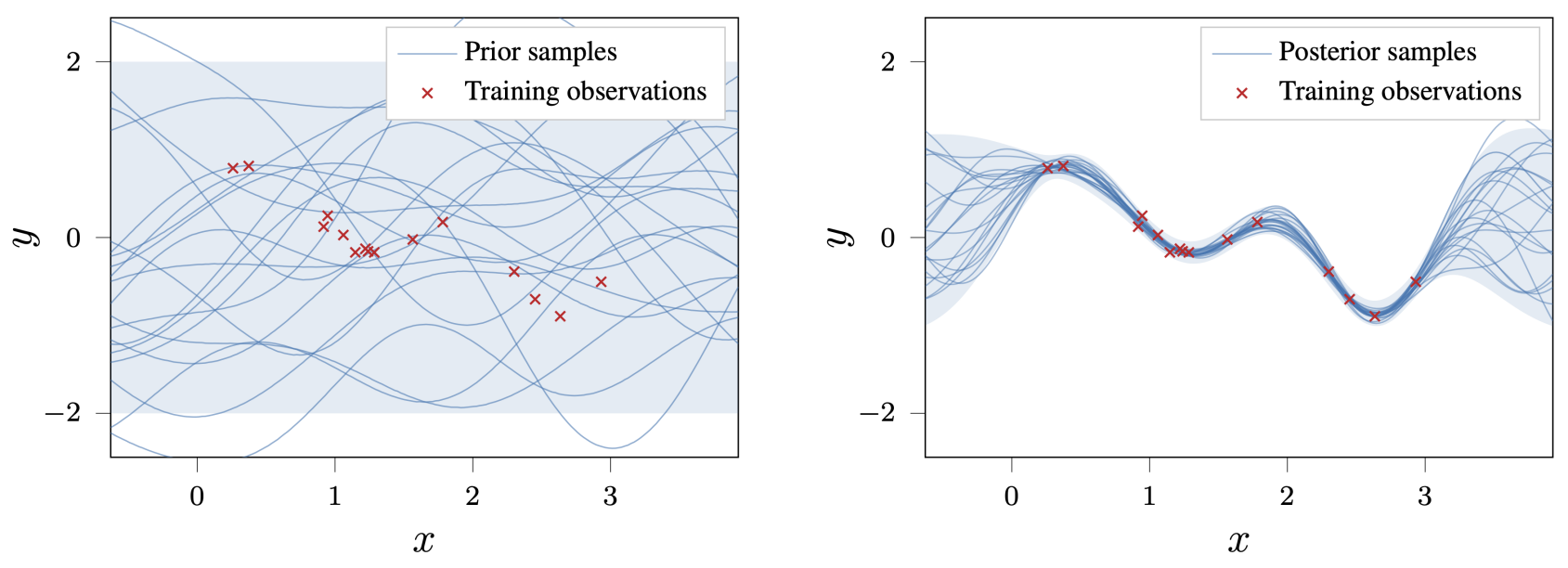


Figure 2: Samples from the prior on $\mathbf{f}'$ and the conditional distribution after observing training data $\mathbf{f}'|\mathbf{y}$ [right]. Shading represents $\pm$ 2 standard deviations. Figure credit: Gabriele Abbati.

Function Values as Parameters

As shown in the notes, an alternative way of deriving the same predictive distribution is to view $p(\mathbf{f}|\mathbf{y})$ as our posterior and $p(\mathbf{f}'|\mathbf{y})$ as our posterior predictive by calculating

$$p(\mathbf{f}'|\mathbf{y}) = \int p(\mathbf{f}'|\mathbf{f},\mathbf{y})p(\mathbf{f}|\mathbf{y})d\mathbf{f} = \int p(\mathbf{f}'|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f} \qquad (3)$$

where we have exploited the conditional independence of $\mathbf{y}$ and $\mathbf{f}'$ given $\mathbf{f}$ and $p(\mathbf{f}|\mathbf{y}) \propto p(\mathbf{f})p(\mathbf{y}|\mathbf{f})$

This gives an alternative nonparameteric view of GPs as we can think of $\mathbf{f}$ as our model parameters such that our number of parameters increases with the number of datapoints

This formulation will still hold for non-Gaussian likelihoods

Gaussian Process Conjugacy

A GP prior is **conjugate** to a Gaussian likelihood so we can think directly in terms of f: a prior $f \sim \mathsf{GP}(0, k)$ and Gaussian likelihood $\mathbf{y}|f \sim \mathcal{N}(\mathbf{f}, \sigma^2 I)$ gives a **GP posterior** as follows:

Gaussian Process Posterior

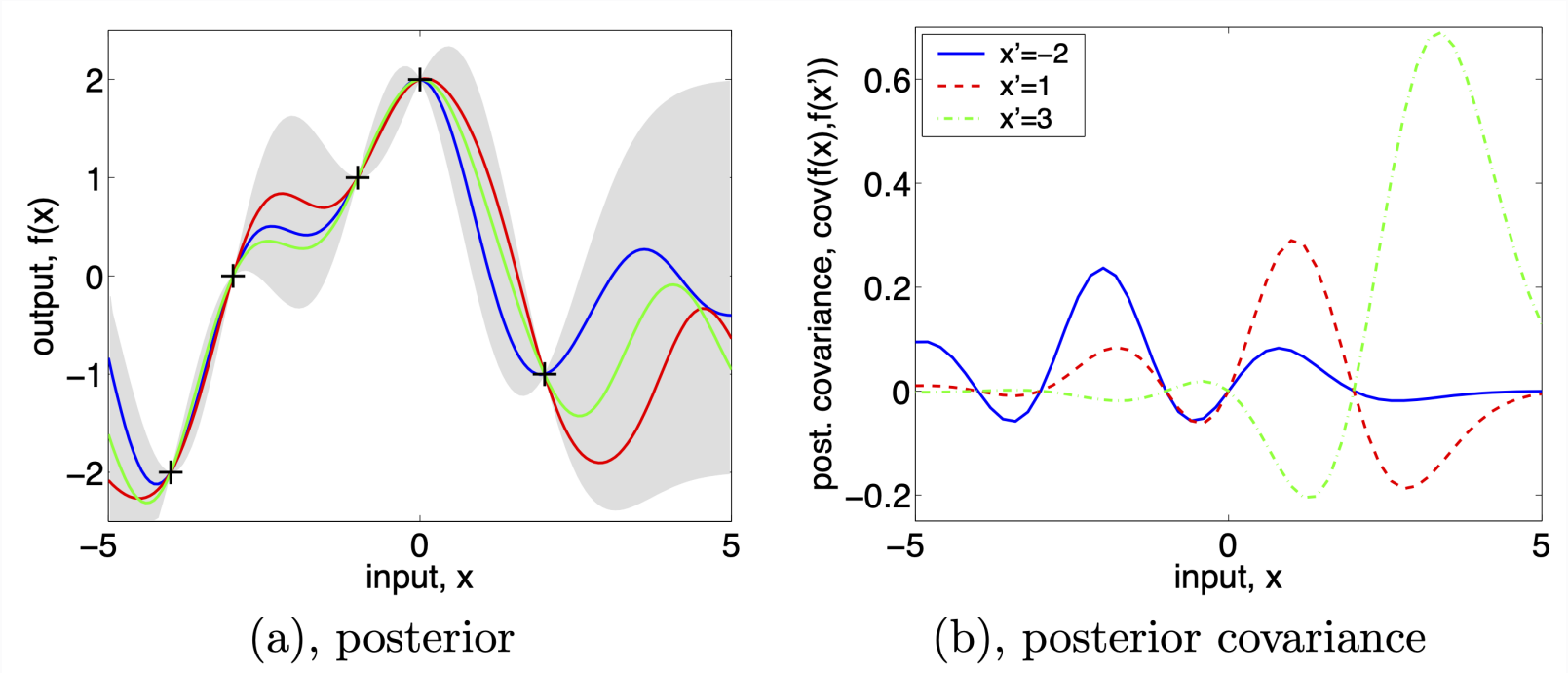
$$
\begin{aligned}
f|\mathbf{y} &\sim \mathsf{GP}(m_{\mathsf{post}}, k_{\mathsf{post}}) \quad \text{where} \\
m_{\mathsf{post}}(x) &= \mathbf{K_x}(x)^T(\mathbf{K_{xx}} + \sigma^2 I)^{-1}\mathbf{y} \\
k_{\mathsf{post}}(x, x') &= k_{\mathsf{prior}}(x, x') - \mathbf{K_x}(x)^T(\mathbf{K_{xx}} + \sigma^2 I)^{-1}\mathbf{K_x}(x') \\
\mathbf{K_x}(x) &= \begin{bmatrix} k_{\mathsf{prior}}(x, x_1) & \dots & k_{\mathsf{prior}}(x, x_N) \end{bmatrix}^T
\end{aligned}
$$

Note that this formulation satisfies the consistency of Bayes' rule under multiple observations (this is a little ugly to show algebraically but easily confirmed numerically)

Demo

http://www.tmpl.fi/gp/

The Posterior Covariance is a Non–Stationary Kernel



(a), posterior (b), posterior covariance

Figure 3: [Left] GP posterior for simple regression problem with three random function draws, shading is $m_{\text{post}}(x) \pm 2\sqrt{k_{\text{post}}(x,x)}$. [Right] posterior covariance function $k_{\text{post}}(x,x')$ for three different choices of x'. Figure credit: Rasmussen and Williams.

Links with Kernel Ridge Regression

The solution to a kernel ridge regression is identical to the GP posterior mean[2] when taking a zero prior mean function, the kernel as the prior covariance, and $\sigma^2 = \lambda$

Recall that in KRR we are solving the empirical risk minimisation

$$f^* = \arg\min_{f \in \mathcal{H}_k} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|^2_{\mathcal{H}_k}.$$

From earlier we have that there is closed form solution given by

$$f^*(x) = \mathbf{K_x}(x)^T (\mathbf{K_{xx}} + \lambda I)^{-1} \mathbf{y}$$

and so we see that $f^* = m_{\text{post}}$ when $\lambda = \sigma^2$.

[2] Note that the posterior mean is also the posterior mode/MAP because of Gaussianity.

Uncertainty Estimates

- A big advantage of GPs over KRR is that they provide a full posterior distribution over functions (and model evidence) rather than just a point estimate
- In particular, they provide **uncertainty estimates** through the posterior standard deviation $\sqrt{k_{\text{post}}(x, x)}$
- The **scaling** of this posterior standard deviation is entirely **subjective**: it depends only on k_{prior} and σ, not the data
  - If we apply the scaling $k'_{\text{prior}} = ck_{\text{prior}}$ and $\sigma' = \sqrt{c}\sigma$, this scaling carries over directly to the posterior: $k'_{\text{post}} = ck_{\text{post}}$
  - We must thus carefully chose a scaling of our covariance function to achieve appropriate uncertainty estimates
  - Our **relative** uncertainty estimates for different points are far more reliable than their **absolute** values
- Our uncertainty estimates also have no dependency on the observed output values: k_{post} only depends on the inputs

Model Evidence

The model evidence, i.e. marginal likelihood, of our model is analytically tractable and can be uncovered by taking the marginal distribution of $\mathbf{y}$ from our previous results

$$p(\mathbf{y}|k_{\text{prior}}, \sigma) = \mathcal{N}(\mathbf{y}; 0, \mathbf{K_{xx}} + \sigma^2 I)$$

We can perform model learning by maximizing this evidence.

Let $\theta := (\nu, \sigma)$ where ν are the kernel hyperparameters (including a kernel scaling factor),[3], we now have

$$\begin{aligned}\log p(\mathbf{y}|\theta) = &-\frac{1}{2}\log|\mathbf{K_{xx}} + \sigma^2 I| - \frac{1}{2}\mathbf{y}^\top \left(\mathbf{K_{xx}} + \sigma^2 I\right)^{-1}\mathbf{y} \\ &- \frac{n}{2}\log(2\pi).\end{aligned} \tag{4}$$

We can then optimize this to learn the best hyperparameters.

[3] For example, for a squared exponential / RBF kernel we have $k_{\text{prior}}(x, x') = s\exp(-0.5\|x - x'\|^2/\gamma^2)$ with scale factor s and length scale γ, such that $\nu = \{s, \gamma\}$.

Effect of Model Learning

Incorporating hyperparameter learning as part of our model actually alleviates the concerns from before:

- k_{post} becomes dependent on the output values due to their influence on hyperparameter selection
- The overall scaling of the uncertainty estimates becomes data–dependent and meaningful

Key intuition: even if its RKHS is very general, choosing a kernel with fixed hyperparameters equates to making extremely strong prior assumptions about the target function's behavior

- Some degree of hyperparameter tuning (or inference) is almost always necessary, particularly if we care about uncertainty
- Many issues translate to KRR (e.g. finding right length scale)

Demo

[Coding Demo]

Non–Gaussian Likelihoods and Link Functions

- Often a Gaussian likelihood model is not appropriate and we must use something else
- Unfortunately, this breaks our conjugacy relationships and we no longer have analytic solutions: we must perform **approximate** Bayesian inference instead
  - Our posterior is not a GP itself anymore
- We can still use our conditional independence relationships to treat the function values at our training inputs, $\mathbf{f}$, as our parameters with posterior $p(\mathbf{f}|\mathbf{y})$ and then make predictions with the posterior predictive

$$p(\mathbf{f}'|\mathbf{y}) = \int p(\mathbf{f}'|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}$$

- If $\mathcal{Y} \neq \mathbb{R}$ then we further need to introduce a **link function** that maps predictions to the correct space

Gaussian Process Classification

Classification is a setting that requires a link function as probabilities must lie in the interval $[0, 1]$

For binary classification, a common choice is the **logistic** function

$$p(y_i = +1|f(x_i)) = s(f(x_i)) = \frac{1}{1 + e^{-f(x_i)}}$$

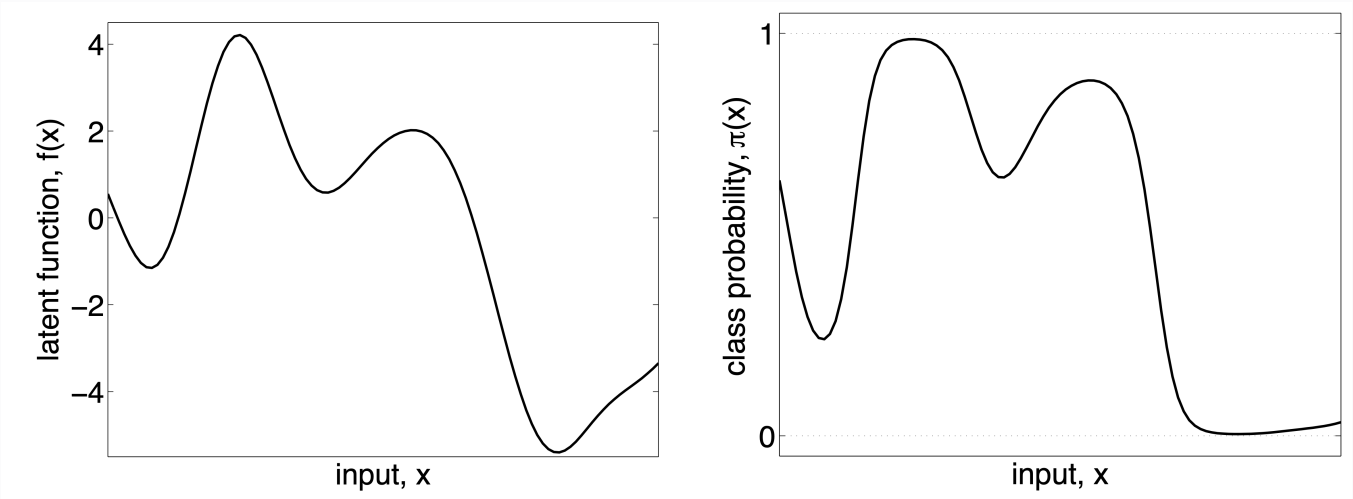


Figure 4: Effect of link function in GP classification were outputs are "squished" to $(0, 1)$. Figure credit: Rasmussen and Williams.

Inference for Gaussian Process Classification

Here the corresponding likelihood function (taking $\mathcal{Y} = \{+1, -1\}$) is

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{n} s(y_i f(x_i)) = \prod_{i=1}^{n} \frac{1}{1 + e^{-y_i f(x_i)}}.$$

Applying Bayes rule with $p(\mathbf{f}) = \mathcal{N}(\mathbf{f}; 0, \mathbf{K_{xx}})$ by our GP prior,

$$\log p(\mathbf{f}|\mathbf{y}) = \text{const} - \frac{1}{2}\mathbf{f}^\top \mathbf{K}_{\mathbf{xx}}^{-1}\mathbf{f} + \sum_{i=1}^{n} \log s(y_i f(x_i)).$$

There are a range of different approximation schemes that have been suggested for this problem such as variational methods, expectation propagation, and the Laplace approximation. See the notes for a derivation of the Laplace approximation and the further reading slide for links to other examples.

Gaussian Process Strengths

- Can provide a very general, powerful, and flexible modeling framework if we can construct an appropriate kernel
- Allow for **prior beliefs** to be expressed in the space of functions themselves
- Complexity of posterior grows with the size of the data available: **no saturation** of information that can be extracted
- **Analytic** solutions in common case of Gaussian likelihoods
- Can provide effective and well–calibrated **uncertainty estimates** with careful model setup and appropriate hyperparameter tuning or inference

Limitations

- Any GP/kernel method is only as good as the kernel
  - Represent the data only through pairwise kernel evaluations
  - Standard kernels lead to much **stronger** prior preferences/assumptions than one might initially expect
- The problem of constructing good kernels in **high dimensions** can be harder than the supervised learning problem itself
- Most kernels result in inherently **local** models such that behavior in extrapolation is dominated by choices in the prior
- **Expensive** except for small datasets with $O(n^3)$ scaling meaning cost quickly becomes prohibitive
- Difficult to use and even more expensive with **non–Gaussian** outputs (each iteration of inference algorithm can be $O(n^3)$)
- Generally require significant **hyperparameter tuning** (or inference over the hyperparameters)

Recap

- Gaussian processes (GPs) provide a powerful method for conducting **nonparametric Bayesian regression**
- GP priors are **conjugate** with Gaussian likelihoods: we get a GP posterior that can be calculated **analytically**
- The **posterior mean** of the GP is identical to the solution of kernel ridge regression, but the GP posterior provides additional information through **uncertainty estimates**
- Kernel and hyperparameter choice can make a big difference to performance
- We can maximize the **model evidence** to get the "best" kernel/hyperparameters
- For non-Gaussian likelihoods, we approximate $p(\mathbf{f}|\mathbf{y})$ and then estimate $p(\mathbf{f}'|\mathbf{y}) = \int p(\mathbf{f}'|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}$

Further Reading

- Textbook reference: Chapters 2-5 of Carl Edward Rasmussen and Christopher Williams. **Gaussian Processes for Machine Learning**. The MIT Press, 2005
- Some useful visualizations:

  `https://distill.pub/2019/visual-exploration-gaussian-processes/`
- Richard Turner's GP tutorial: `https://youtu.be/92-98SYOdlY`
- Neil Lawrence's GP tutorial with accompanying notes and code:

  `https://inverseprobability.com/talks/notes/gaussian-processes.html`
- Lecture by Philipp Hennig on GP classification

  `https://youtu.be/iatPLQd7qcg`