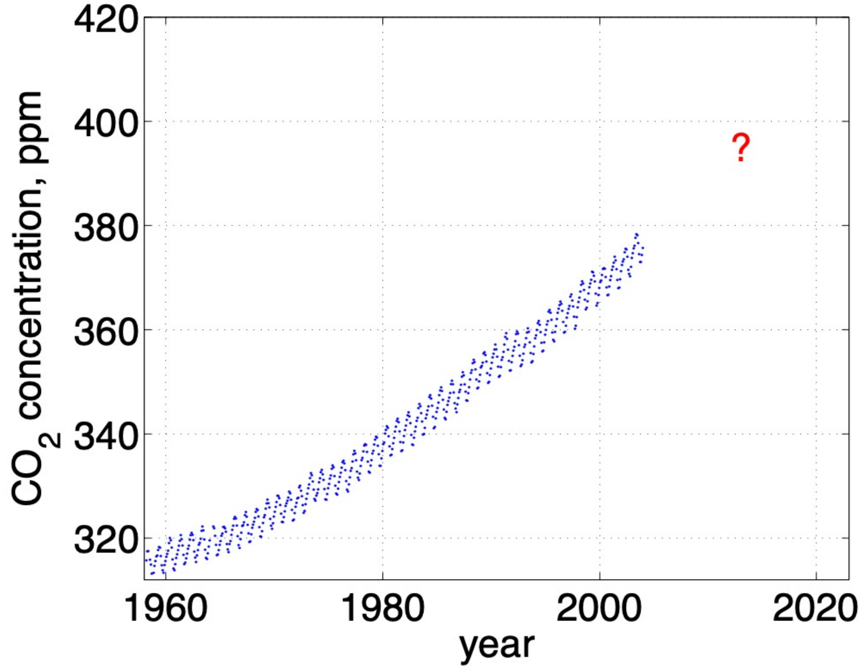
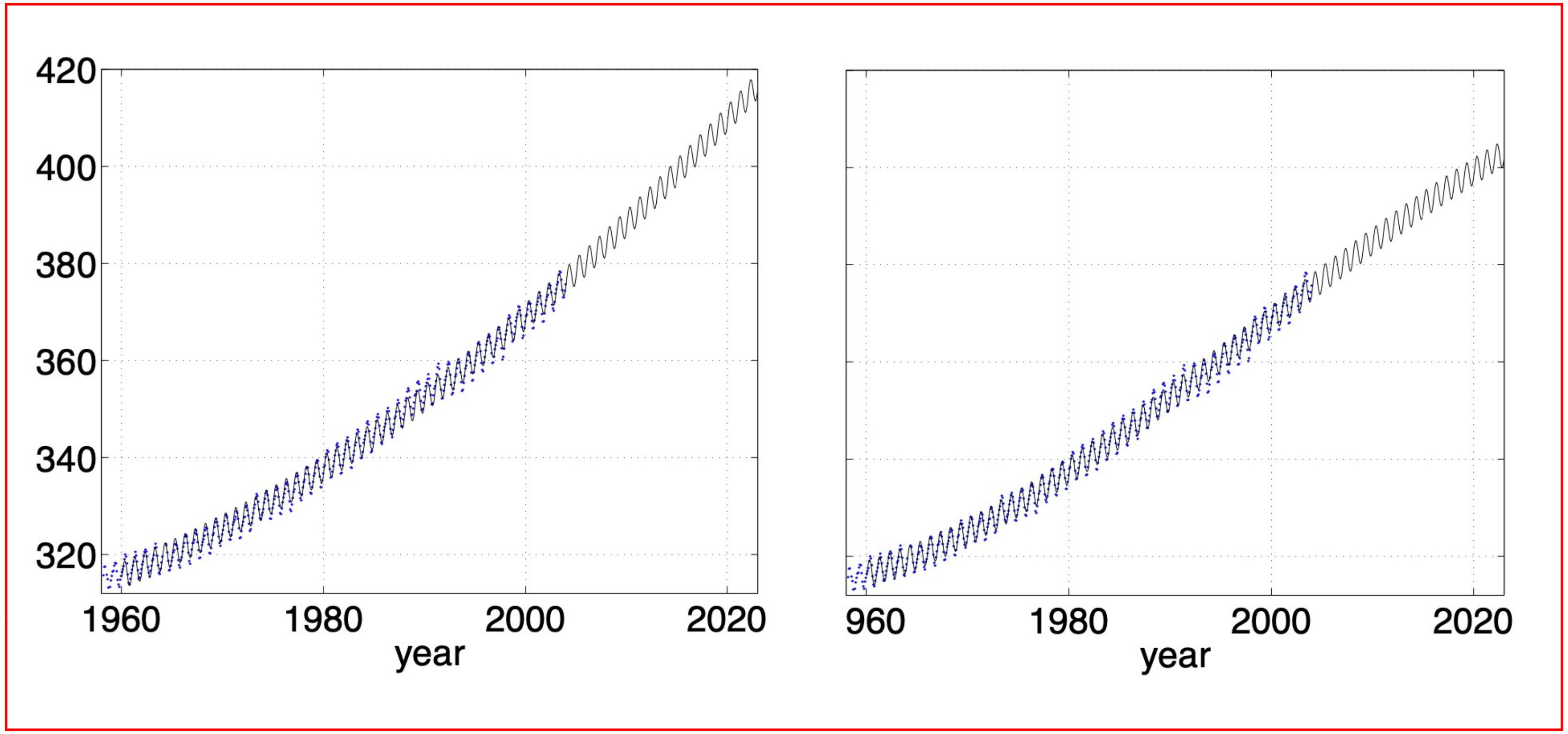
Applications of Gaussian Process

- Solve challenging non-linear regression problems
- Solve classification problems
- Bayesian Optimization

The Prediction Problem





Bayesian parametric inference

Supervised parametric learning:

- data: $\mathbf{x}, \mathbf{y}$
- model: $y = f_{\mathbf{w}}(x) + \varepsilon$

Gaussian likelihood:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}) \;\propto\; \prod_c \exp(-\tfrac{1}{2}(y_c - f_{\mathbf{w}}(x_c))^2/\sigma^2_{\text{noise}}).$$

Parameter prior

$$p(\mathbf{w})$$

Posterior parameter distribution by Bayes rule

$$p(\mathbf{w}|\mathbf{x}, \mathbf{y}) \;=\; \frac{p(\mathbf{w})p(\mathbf{y}|\mathbf{x}, \mathbf{w})}{p(\mathbf{y}|\mathbf{x})}$$

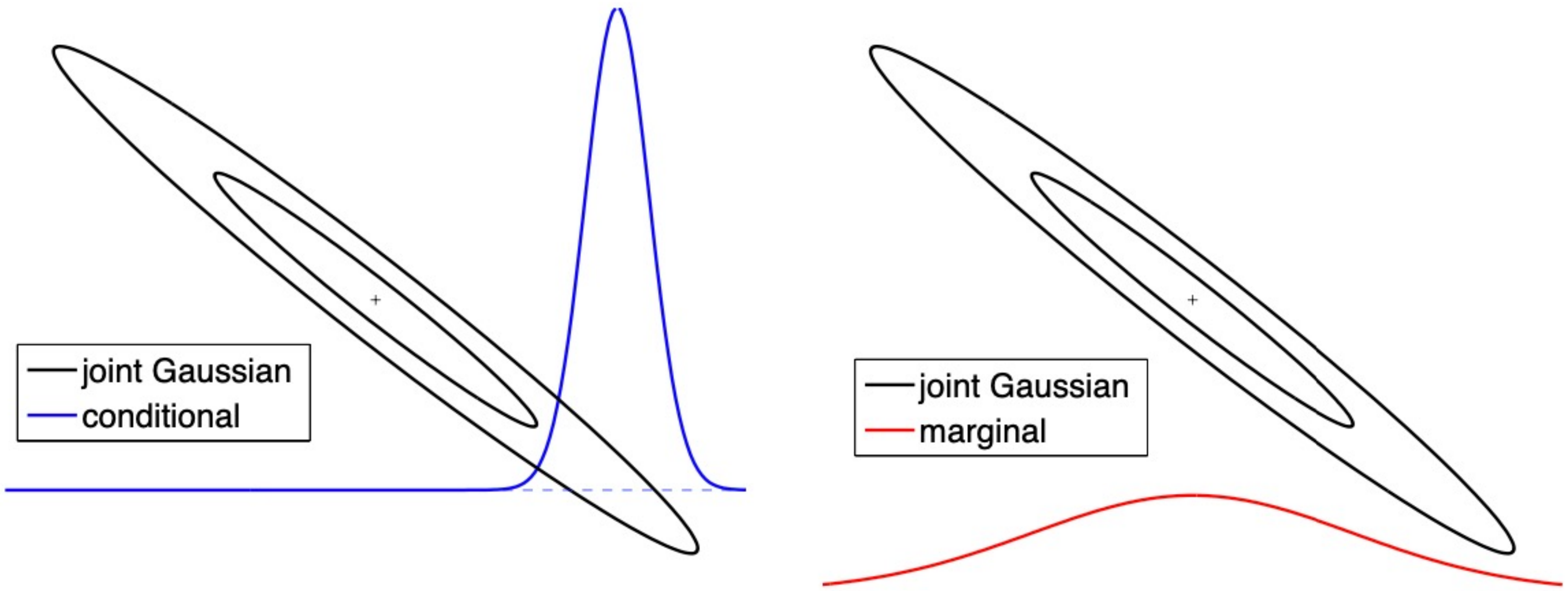
Making predictions:

$$p(y^*|x^*, \mathbf{x}, \mathbf{y}) \;=\; \int p(y^*|\mathbf{w}, x^*)p(\mathbf{w}|\mathbf{x}, \mathbf{y})d\mathbf{w}$$

Marginal Likelihood:

$$p(\mathbf{y}|\mathbf{x}) \;=\; \int p(\mathbf{w})p(\mathbf{y}|\mathbf{x}, \mathbf{w})d\mathbf{w}.$$

Once Gaussian, Always Gaussian



Both the conditionals and the marginals of a joint Gaussian are again Gaussian.

Gaussian Process vs Gaussian Distribution

A Gaussian distribution is fully specified by a mean vector μ and covariance matrix Σ:

$$\mathbf{f} = (f_1, \ldots, f_n)^\top \sim \mathcal{N}(\mu, \Sigma), \quad \text{indexes } i = 1, \ldots, n$$

A Gaussian process is fully specified by a mean function m(x) and covariance function k(x, x′):

$$f(x) \sim \mathcal{GP}\big(m(x), k(x, x')\big),$$

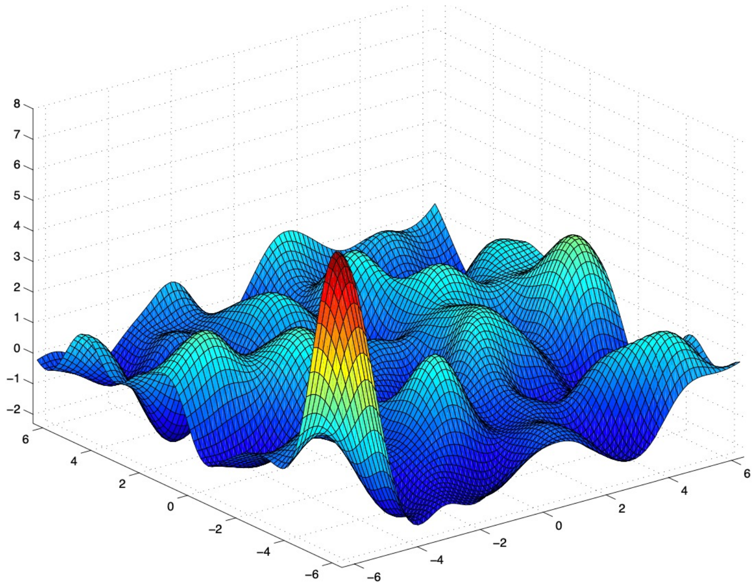
Thinking of a GP as a Gaussian distribution with an infinitely long mean vector and an infinite by infinite covariance matrix may seem impractical. . .

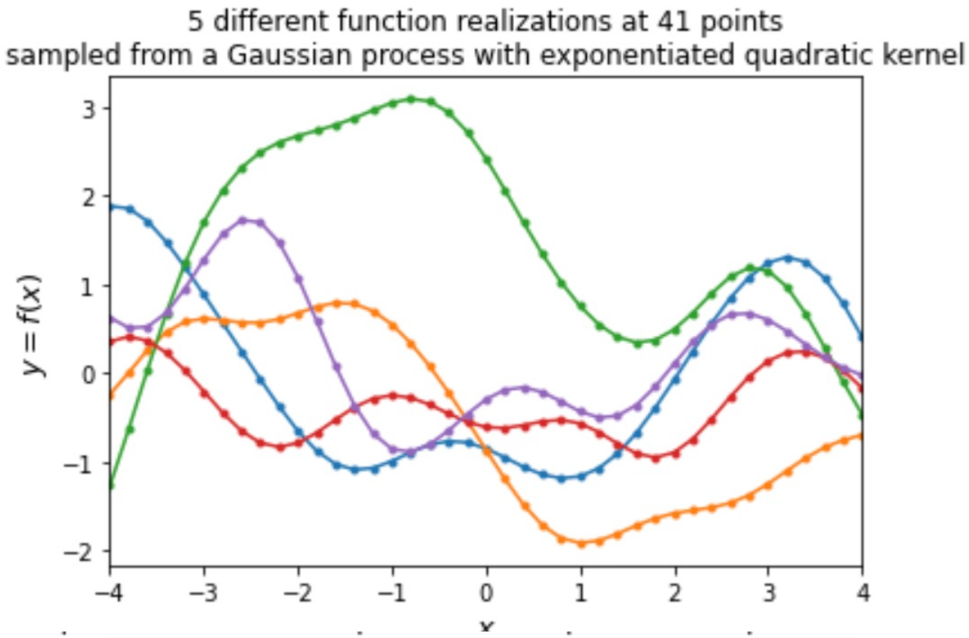
To get an indication of what this distribution over functions looks like, focus on a finite subset of function values $\mathbf{f} = (f(x_1), f(x_2), \ldots, f(x_n))^\top$, for which:

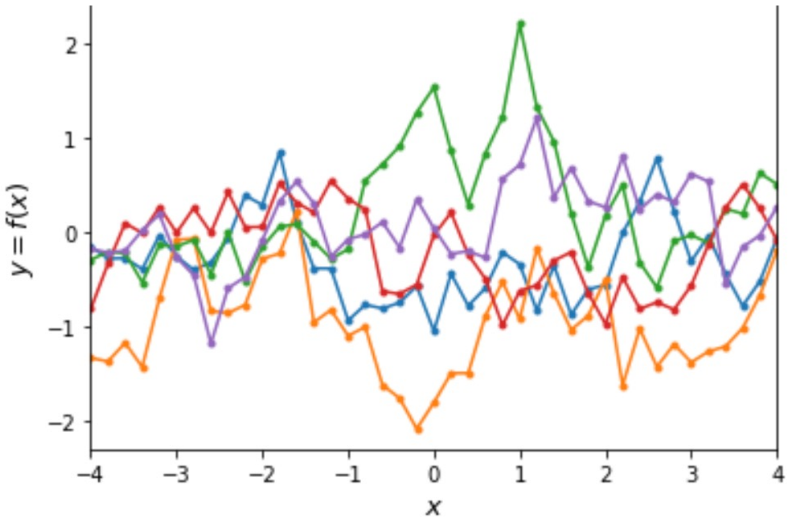
$\mathbf{f} \sim N(\mu,\Sigma)$, where $\Sigma ij = k(xi, xj)$.

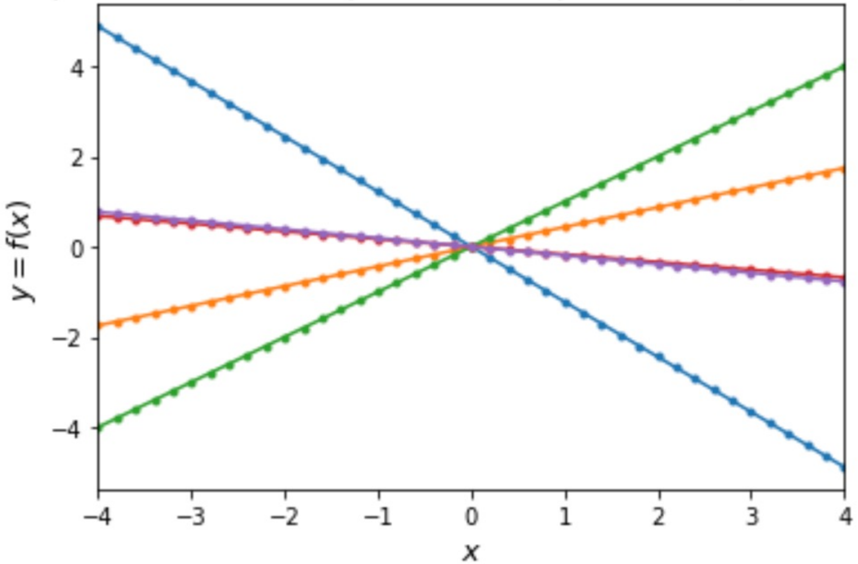
This becomes a sampling problem!

Function drawn at random from a Gaussian Process









Bayesian parametric inference

Supervised parametric learning:

- data: $\mathbf{x}, \mathbf{y}$
- model: $y = f_{\mathbf{w}}(x) + \varepsilon$

Gaussian likelihood:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M_i) \;\propto\; \prod_c \exp(-\tfrac{1}{2}(y_c - f_{\mathbf{w}}(x_c))^2/\sigma^2_{\text{noise}}).$$

Parameter prior

$$p(\mathbf{w}|M_i)$$

Posterior parameter distribution by Bayes rule

$$p(\mathbf{w}|\mathbf{x}, \mathbf{y}, M_i) \;=\; \frac{p(\mathbf{w}|M_i)p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M_i)}{p(\mathbf{y}|\mathbf{x}, M_i)}$$

Making predictions:

$$p(y^*|x^*, \mathbf{x}, \mathbf{y}, M_i) \;=\; \int p(y^*|\mathbf{w}, x^*, M_i)p(\mathbf{w}|\mathbf{x}, \mathbf{y}, M_i)d\mathbf{w}$$

Marginal Likelihood:

$$p(\mathbf{y}|\mathbf{x}, M_i) \;=\; \int p(\mathbf{w}|M_i)p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M_i)d\mathbf{w}.$$

Model probability:

$$p(M_i|\mathbf{x}, \mathbf{y}) \;=\; \frac{p(M_i)p(\mathbf{y}|\mathbf{x}, M_i)}{p(\mathbf{y}|\mathbf{x})}$$

Problem: integrals are intractable for most interesting models!

Non-parametric Gaussian process models

In our non-parametric model, the “parameters” are the function itself!

Gaussian likelihood:

$$\mathbf{y}|\mathbf{x}, f(x), M_i \sim \mathcal{N}(\mathbf{f},\ \sigma^2_{\text{noise}} I)$$

(Zero mean) Gaussian process prior:

$$f(x)|M_i \sim \mathcal{GP}\big(m(x) \equiv 0,\ k(x, x')\big)$$

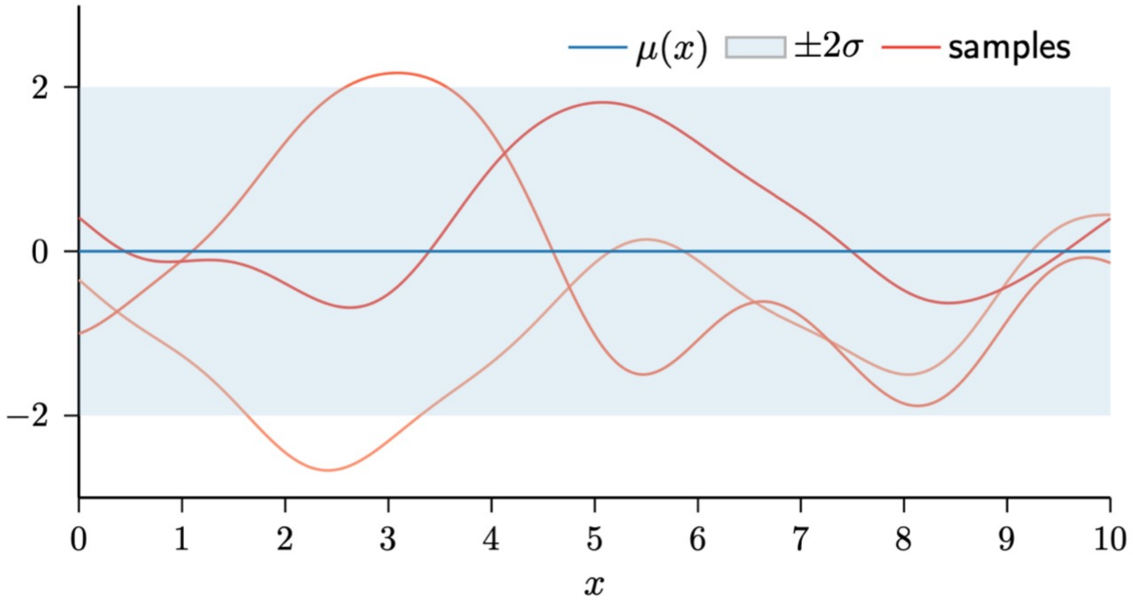
Leads to a Gaussian process posterior:

$$\begin{aligned} f(x)|\mathbf{x}, \mathbf{y}, M_i \sim \mathcal{GP}\big(&m_{\text{post}}(x) = k(x, \mathbf{x})[K(\mathbf{x}, \mathbf{x}) + \sigma^2_{\text{noise}} I]^{-1}\mathbf{y}, \\ &k_{\text{post}}(x, x') = k(x, x') - k(x, \mathbf{x})[K(\mathbf{x}, \mathbf{x}) + \sigma^2_{\text{noise}} I]^{-1} k(\mathbf{x}, x')\big). \end{aligned}$$

And a Gaussian predictive distribution:

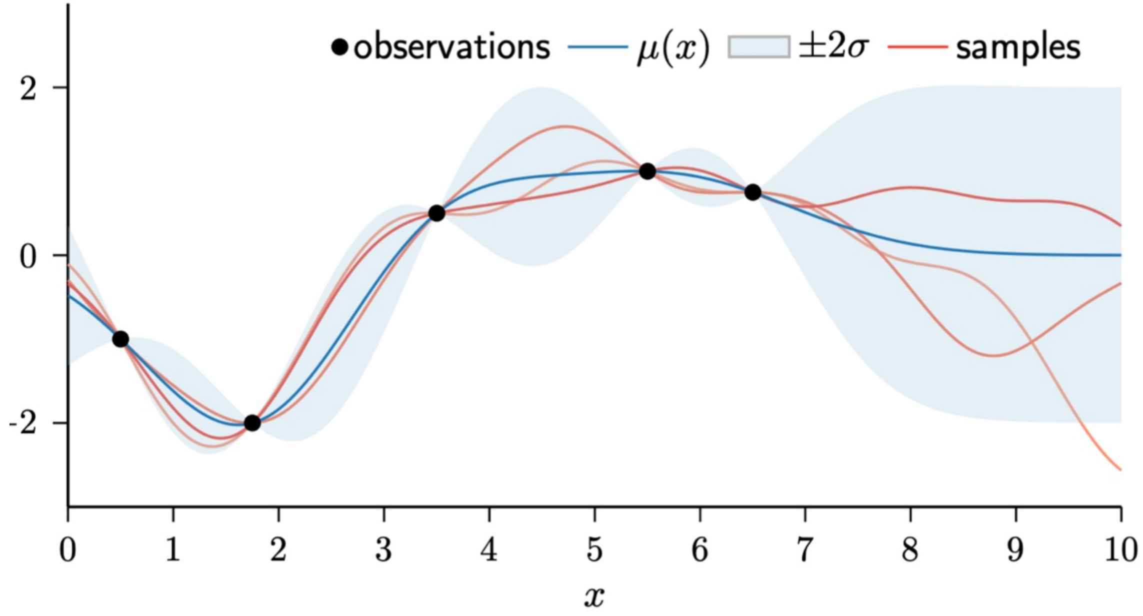
$$\begin{aligned} y^*|x^*, \mathbf{x}, \mathbf{y}, M_i \sim \mathcal{N}\big(&\mathbf{k}(x^*, \mathbf{x})^\top [K + \sigma^2_{\text{noise}} I]^{-1}\mathbf{y}, \\ &k(x^*, x^*) + \sigma^2_{\text{noise}} - \mathbf{k}(x^*, \mathbf{x})^\top [K + \sigma^2_{\text{noise}} I]^{-1} \mathbf{k}(x^*, \mathbf{x})\big) \end{aligned}$$

Prior



$$K = \exp\left(-\frac{1}{2}\|x - x'\|^2\right)$$

Posterior



Hyperparameter

$$p(\mathbf{y} \mid \mathbf{X}, \theta) = \int p(\mathbf{y} \mid \mathbf{f})\, p(\mathbf{f} \mid \mathbf{X}, \theta)\, \mathrm{d}\mathbf{f},$$

$$= \int \overbrace{\mathcal{N}(\mathbf{y}; \mathbf{f}, \sigma^2 \mathbf{I})}^{\text{iid noise}}\, \overbrace{\mathcal{N}\big(\mathbf{f}; \mu(\mathbf{X}; \theta), K(\mathbf{X}, \mathbf{X}; \theta)\big)}^{\text{GP prior}}\, \mathrm{d}\mathbf{f}$$

$$= \mathcal{N}\big(\mathbf{y}; \mu(\mathbf{X}; \theta), K(\mathbf{X}, \mathbf{X}; \theta) + \sigma^2 \mathbf{I}\big).$$

$$\log p(\mathbf{y} \mid \mathbf{X}, \theta) =$$

$$-\overbrace{\frac{(\mathbf{y} - \boldsymbol{\mu})^\top \mathbf{V}^{-1} (\mathbf{y} - \boldsymbol{\mu})}{2}}^{\text{data fit}} - \frac{\log \det \mathbf{V}}{2} - \frac{N \log 2\pi}{2}$$

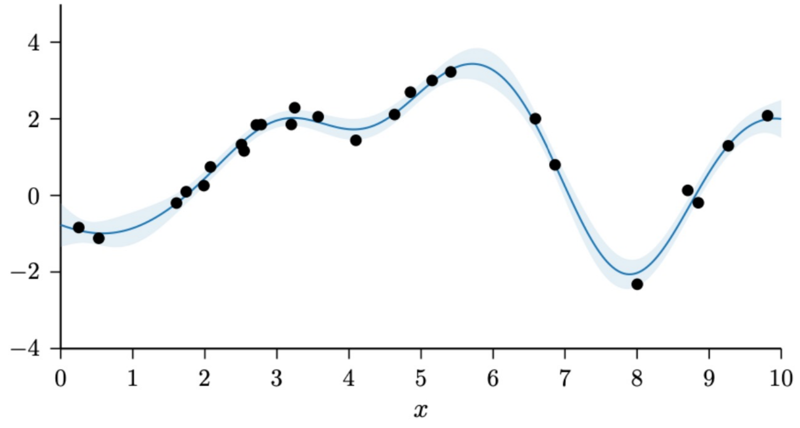
It is the combination of a data fit term and complexity penalty.

Learning in Gaussian process models involves finding

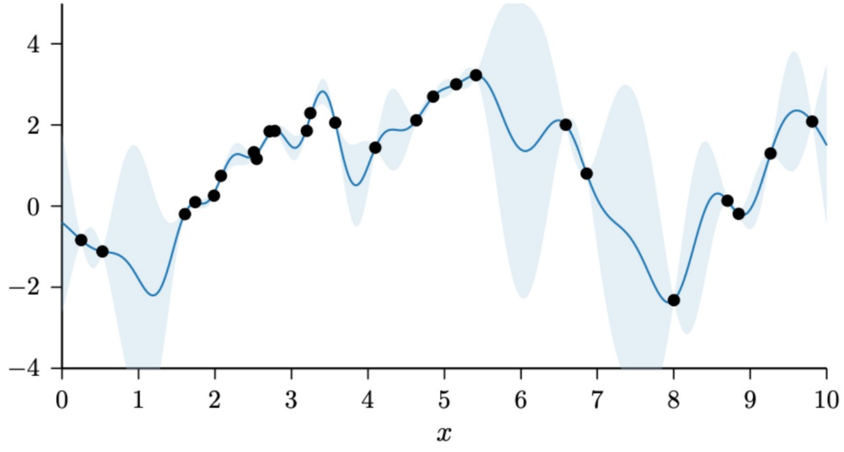
- the form of the covariance function, and
- any unknown (hyper-) parameters θ.

Hyperparameters can be found by optimizing the marginal likelihood:

$$\frac{\partial \log P(y|X,\theta)}{\partial \theta_j} = \frac{1}{2}(y-\mu)^T V^{-1}\frac{\partial V}{\partial \theta_j}V^{-1}(y-\mu) - \frac{1}{2} trace(V^{-1}\frac{\partial V}{\partial \theta_j})$$



$\theta = (\lambda, \ell, \sigma) = (1, 1, \frac{1}{5}), \quad \log p(\mathbf{y} \mid \mathbf{X}, \theta) = -27.6$

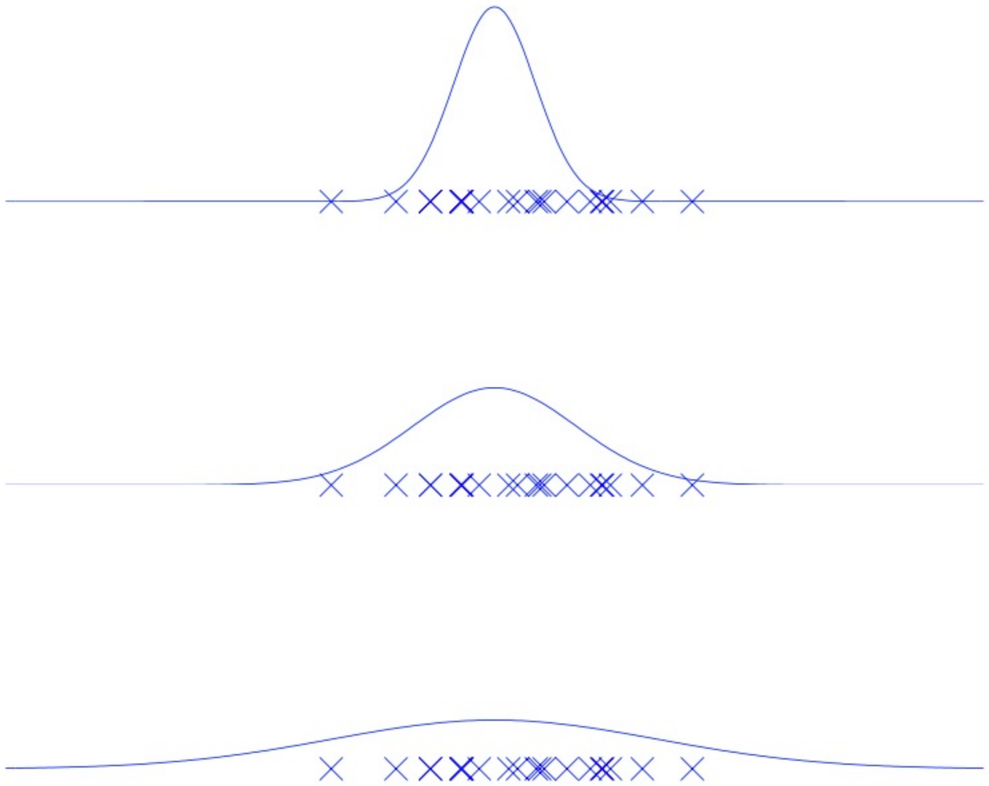


$\theta = (\lambda, \ell, \sigma) = (2, \frac{1}{3}, \frac{1}{20}), \quad \log p(\mathbf{y} \mid \mathbf{X}, \theta) = -46.5$

Notice, that an almost exact fit to the data can be achieved by reducing the length scale – but the marginal likelihood does not favor this!

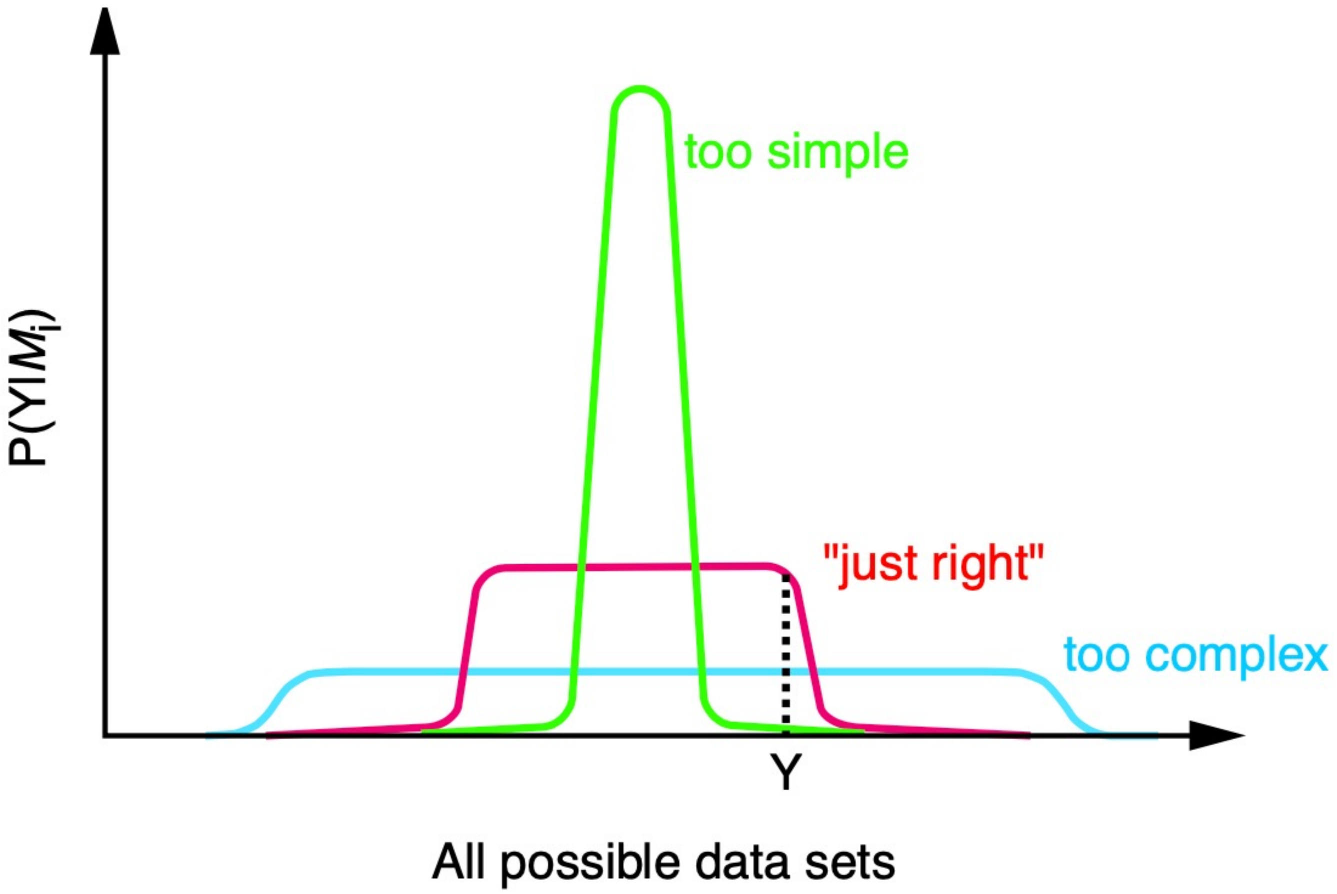
Model Complexity: An illustrative analogous example

- Imagine the simple task of fitting the variance, of a zero-mean Gaussian to a set of n scalar observations.

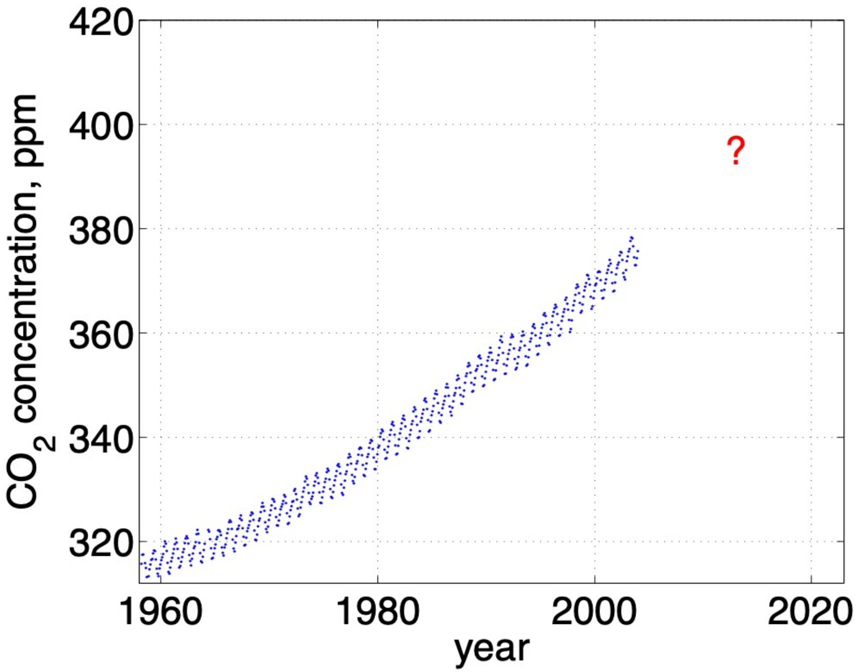


The log likelihood is $\log p(\mathbf{y}|\mu, \sigma^2) = -\frac{1}{2}\mathbf{y}^\top I\mathbf{y}/\sigma^2 - \frac{1}{2}\log|I\sigma^2| - \frac{n}{2}\log(2\pi)$

Occam's Razor



The prediction Problem



The covariance function consists of several terms, parameterized by a total of 11 *hyperparameters*:

- long-term smooth trend (squared exponential)
  $$k_1(x, x') = \theta_1^2 \exp(-(x - x')^2/\theta_2^2),$$
- seasonal trend (quasi-periodic smooth)
  $$k_2(x, x') = \theta_3^2 \exp\left(-2\sin^2(\pi(x - x'))/\theta_5^2\right) \times \exp\left(-\tfrac{1}{2}(x - x')^2/\theta_4^2\right),$$
- short- and medium-term anomaly (rational quadratic)
  $$k_3(x, x') = \theta_6^2\left(1 + \frac{(x-x')^2}{2\theta_8\theta_7^2}\right)^{-\theta_8}$$
- noise (independent Gaussian, and dependent)
  $$k_4(x, x') = \theta_9^2 \exp\left(-\frac{(x-x')^2}{2\theta_{10}^2}\right) + \theta_{11}^2\delta_{xx'}.$$

$$k(x, x') = k_1(x, x') + k_2(x, x') + k_3(x, x') + k_4(x, x')$$

Binary Gaussian Process Classification

The class probability is related to the *latent* function, f, through:

$$p(y = 1|f(\mathbf{x})) \;=\; \pi(\mathbf{x}) \;=\; \Phi\big(f(\mathbf{x})\big)$$

where Φ is a sigmoid function, such as the logistic regression
Observations are independent given f, so the likelihood is :

$$p(\mathbf{y}|\mathbf{f}) \;=\; \prod_{i=1}^{n} p(y_i|f_i) \;=\; \prod_{i=1}^{n} \Phi(y_i f_i).$$

We use a Gaussian process prior for the latent function:

$$\mathbf{f}|X, \theta \;\sim\; \mathcal{N}(\mathbf{0},\; K)$$

The posterior becomes:

$$p(\mathbf{f}|\mathcal{D}, \theta) \;=\; \frac{p(\mathbf{y}|\mathbf{f})\, p(\mathbf{f}|X, \theta)}{p(\mathcal{D}|\theta)} \;=\; \frac{\mathcal{N}(\mathbf{f}|\mathbf{0},\; K)}{p(\mathcal{D}|\theta)} \prod_{i=1}^{m} \Phi(y_i f_i)$$

which is non-Gaussian. This makes predictive class probability and latent value at the test point intractable to compute.

Gaussian Approximation to the Posterior

The latent value at the test point, $f(\mathbf{x}^*)$ is

$$p(f_*|\mathcal{D},\theta,\mathbf{x}_*) = \int p(f_*|\mathbf{f},X,\theta,\mathbf{x}_*)p(\mathbf{f}|\mathcal{D},\theta)d\mathbf{f},$$

and the predictive class probability becomes

$$p(y_*|\mathcal{D},\theta,\mathbf{x}_*) = \int p(y_*|f_*)p(f_*|\mathcal{D},\theta,\mathbf{x}_*)df_*,$$

We approximate the non-Gaussian posterior by a Gaussian:

$$p(\mathbf{f}|\mathcal{D},\theta) \simeq q(\mathbf{f}|\mathcal{D},\theta) = \mathcal{N}(\mathbf{m}, A)$$

then $q(f_*|\mathcal{D},\theta,\mathbf{x}_*) = \mathcal{N}(f_*|\mu_*,\sigma_*^2)$, where

$$\begin{aligned}\mu_* &= \mathbf{k}_*^\top K^{-1}\mathbf{m}\\ \sigma_*^2 &= k(\mathbf{x}_*,\mathbf{x}_*) - \mathbf{k}_*^\top(K^{-1} - K^{-1}AK^{-1})\mathbf{k}_*.\end{aligned}$$

Using this approximation with the cumulative Gaussian likelihood

$$q(y_*=1|\mathcal{D},\theta,\mathbf{x}_*) = \int \Phi(f_*)\,\mathcal{N}(f_*|\mu_*,\sigma_*^2)df_*$$

How to find **m** and A:

- Laplace's method: Find the Maximum A Posteriori (MAP) for latent values and use a local expansion (Gaussian) around this point. (Williams and Barber)
- Variational bounds: bound the likelihood by some tractable expression.(Gibbs and Mckay, Seeger)

Bayesian Optimization

Bayesian optimization.

Algorithm 1 Bayesian optimization with Gaussian process prior

```
input: loss function f, kernel K, acquisition function a, loop counts N_warmup and N
▷ warmup phase
y_best ← ∞
for i = 1 to N_warmup do
    select x_i via some method (usually random sampling)
    compute exact loss function y_i ← f(x_i)
    if y_i ≤ y_best then
        x_best ← x_i
        y_best ← y_i
    end if
end for
for i = N_warmup + 1 to N do
    update kernel matrix Σ ∈ ℝ^{i×i} according to (1)
    let μ(x_*) and σ(x_*) denote the expected value and standard deviation, respectively, of f(x_*) under the
Gaussian process model, conditioned on all the previous observations of f(x_i) = y_i
    x_i ← arg min_{x_*} a(μ(x_*), σ(x_*), y_best)
    compute exact loss function y_i ← f(x_i)
    if y_i ≤ y_best then
        x_best ← x_i
        y_best ← y_i
    end if
end for
return x_best
```

Acquisition function:

- Probability of Improvement
- Expected Improvement
- Lower confidence bound

References

- Williams, C. K. I. and Barber, D. (1998). Bayesian Classification with Gaussian Processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351.
- Gibbs, M. N. and MacKay, D. J. C. (2000). Variational Gaussian Process Classifiers. *IEEE Transactions on Neural Networks*, 11(6):1458–1464.
- Seeger, M. (2003). *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations*. PhD thesis, School of Informatics, University of Edinburgh. http://www.cs.berkeley.edu/∼mseeger.