Solving Challenging Non-linear Regression Problems by Manipulating a Gaussian Distribution

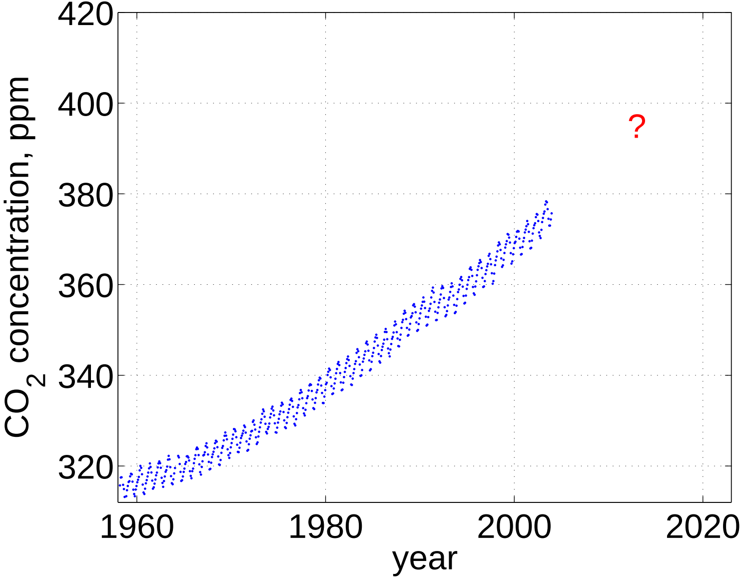
Sheffield Gaussian Process Summer School, 2014

Carl Edward Rasmussen

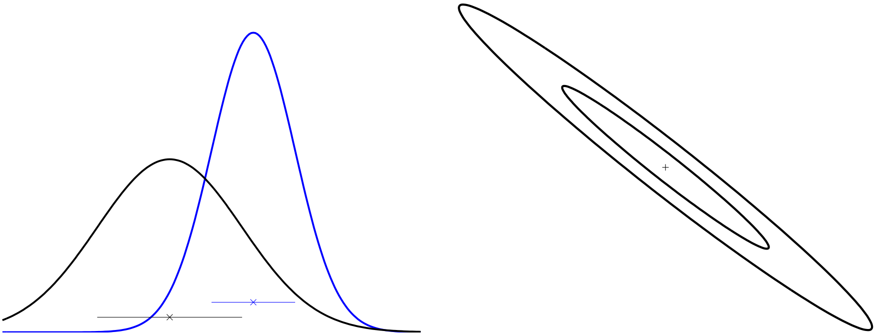
Department of Engineering, University of Cambridge

September 15-17th, 2014

The Prediction Problem



The Gaussian Distribution

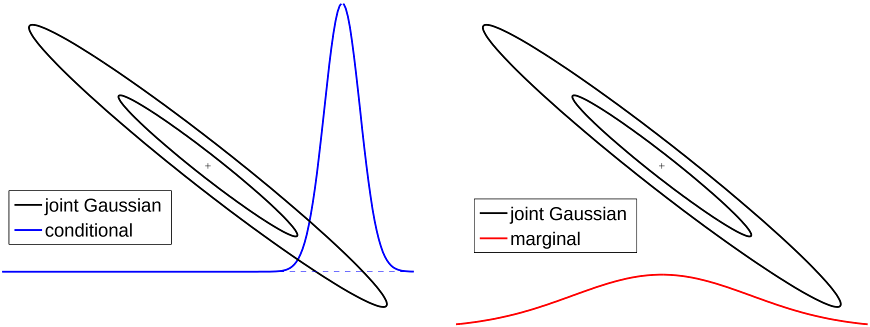


The Gaussian distribution is given by

$$p(\mathbf{x}|\mu, \Sigma) = \mathcal{N}(\mu, \Sigma) = (2\pi)^{-D/2}|\Sigma|^{-1/2} \exp\big(-\tfrac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)\big)$$

where μ is the mean vector and Σ the covariance matrix.

Conditionals and Marginals of a Gaussian



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

Conditionals and Marginals of a Gaussian

In algebra, if **x** and **y** are jointly Gaussian

$$p(\mathbf{x},\mathbf{y}) \;=\; \mathcal{N}\Big(\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}A & B\\ B^\top & C\end{bmatrix}\Big),$$

the marginal distribution of **x** is

$$p(\mathbf{x},\mathbf{y}) \;=\; \mathcal{N}\Big(\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}A & B\\ B^\top & C\end{bmatrix}\Big) \;\Longrightarrow\; p(\mathbf{x}) \;=\; \mathcal{N}(\mathbf{a},\; A),$$

and the conditional distribution of **x** given **y** is

$$p(\mathbf{x},\mathbf{y}) \;=\; \mathcal{N}\Big(\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}A & B\\ B^\top & C\end{bmatrix}\Big) \;\Longrightarrow\; p(\mathbf{x}|\mathbf{y}) \;=\; \mathcal{N}(\mathbf{a}{+}BC^{-1}(\mathbf{y}{-}\mathbf{b}),\; A{-}BC^{-1}B^\top),$$

where **x** and **y** can be scalars or vectors.

What is a Gaussian Process?

A *Gaussian process* is a generalization of a multivariate Gaussian distribution to infinitely many variables.

Informally: infinitely long vector $\simeq$ function

> **Definition:** *a Gaussian process is a collection of random variables, any finite number of which have (consistent) Gaussian distributions.* □

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 $k(x, x')$:

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

The marginalization property

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

…luckily we are saved by the *marginalization property*:

Recall:

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

For Gaussians:

$$p(\mathbf{x},\mathbf{y}) \;=\; \mathcal{N}\Big(\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix},\,\begin{bmatrix}A & B\\ B^\top & C\end{bmatrix}\Big) \;\Longrightarrow\; p(\mathbf{x}) \;=\; \mathcal{N}(\mathbf{a},\;A)$$

Random functions from a Gaussian Process

Example one dimensional Gaussian process:

$$p(f(x)) \sim \mathcal{GP}\big(m(x) = 0,\ k(x, x') = \exp(-\tfrac{1}{2}(x - x')^2)\big).$$

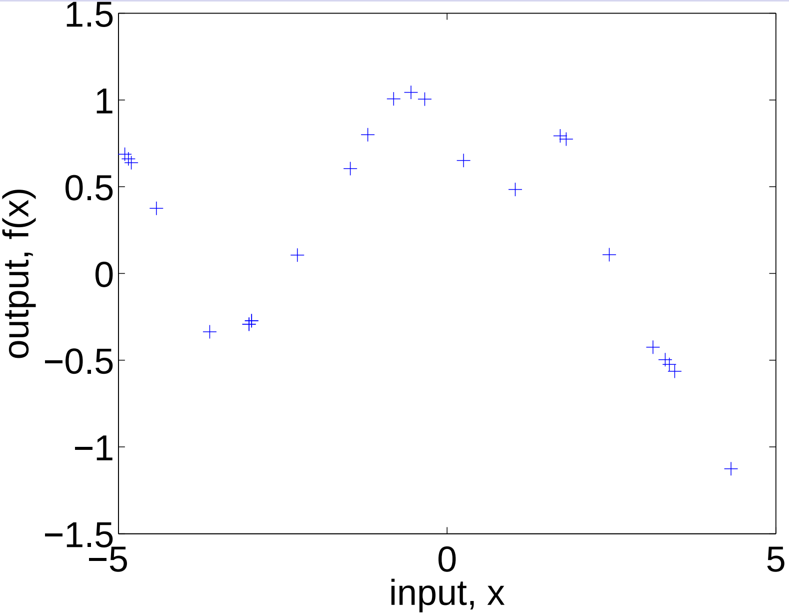
To get an indication of what this distribution over functions looks like, focus on a finite subset of function values $\mathbf{f} = (f(x_1), f(x_2), \ldots, f(x_n))^\top$, for which

$$\mathbf{f} \sim \mathcal{N}(0, \Sigma),$$

where $\Sigma_{ij} = k(x_i, x_j)$.

Then plot the coordinates of f as a function of the corresponding x values.

Some values of the random function



Joint Generation

To generate a random sample from a D dimensional joint Gaussian with covariance matrix K and mean vector $\mathbf{m}$: (in octave or matlab)

```
z = randn(D,1);
y = chol(K)'*z + m;
```

where `chol` is the Cholesky factor R such that $R^\top R = K$.

Thus, the covariance of $\mathbf{y}$ is:

$$\mathbb{E}[(\mathbf{y}-\bar{\mathbf{y}})(\mathbf{y}-\bar{\mathbf{y}})^\top] \;=\; \mathbb{E}[R^\top \mathbf{z}\mathbf{z}^\top R] \;=\; R^\top \mathbb{E}[\mathbf{z}\mathbf{z}^\top]R \;=\; R^\top I R \;=\; K.$$

Sequential Generation

Factorize the joint distribution

$$p(f_1, \ldots, f_n | \mathbf{x}_1, \ldots \mathbf{x}_n) \;=\; \prod_{i=1}^{n} p(f_i | f_{i-1}, \ldots, f_1, \mathbf{x}_i, \ldots, \mathbf{x}_1),$$

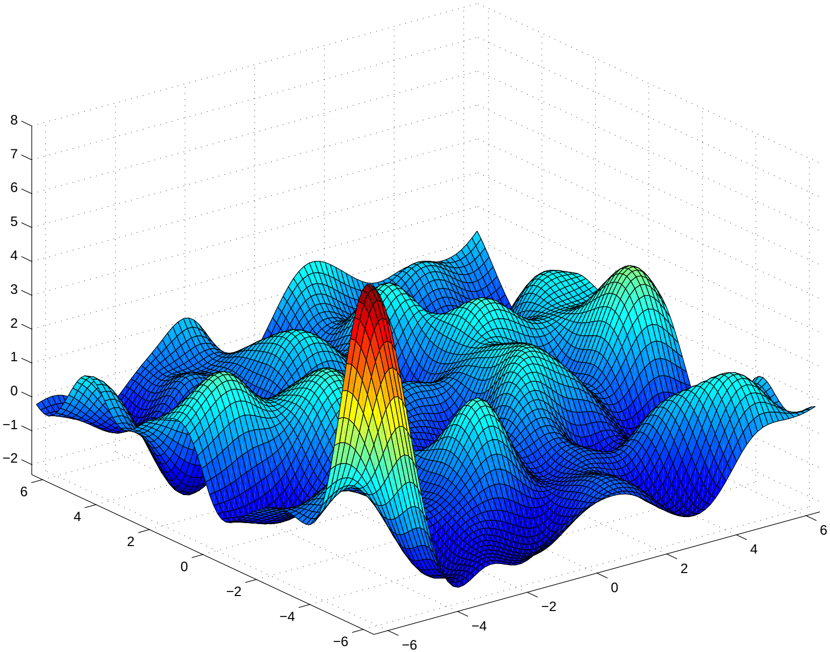
and generate function values sequentially.

What do the individual terms look like? For Gaussians:

$$p(\mathbf{x}, \mathbf{y}) \;=\; \mathcal{N}\Big(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}\Big) \;\Longrightarrow\; p(\mathbf{x}|\mathbf{y}) \;=\; \mathcal{N}(\mathbf{a}+BC^{-1}(\mathbf{y}-\mathbf{b}),\; A-BC^{-1}B^\top)$$

Do try this at home!

Function drawn at random from a Gaussian Process with Gaussian covariance



Maximum likelihood, parametric model

Supervised parametric learning:

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

Gaussian likelihood:

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

Maximize the likelihood:

$$\mathbf{w}_{\text{ML}} \;=\; \underset{\mathbf{w}}{\operatorname{argmax}}\, p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M_i).$$

Make predictions, by plugging in the ML estimate:

$$p(y^*|x^*, \mathbf{w}_{\text{ML}}, M_i)$$

Bayesian Inference, parametric model

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 $p(a|b) = p(b|a)p(a)/p(b)$:

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

Bayesian Inference, parametric model, cont.

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:

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

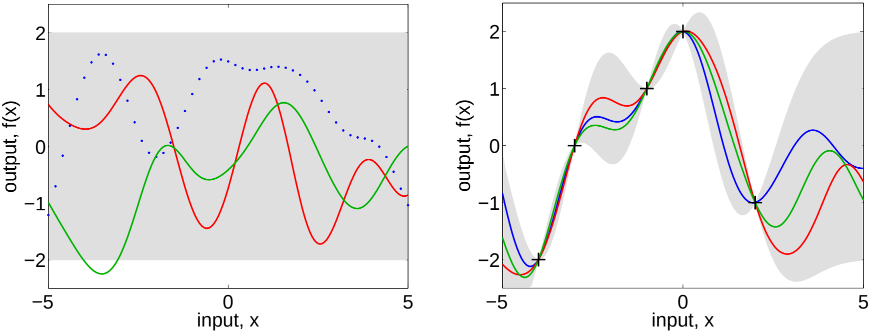
Leads to a Gaussian process posterior

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

And a Gaussian predictive distribution:

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

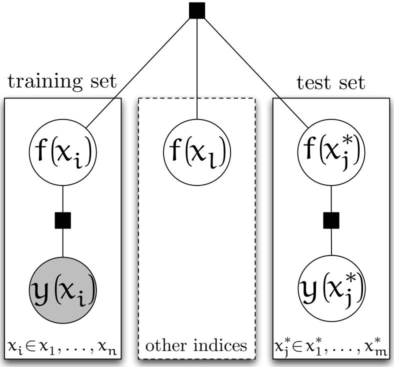
Prior and Posterior



Predictive distribution:

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

Factor Graph for Gaussian Process



A Factor Graph is a graphical representation of a multivariate distribution.

Nodes are random variables, black boxes are *factors*. The factors induce dependencies between the variables to which they have edges. Open nodes are stochastic (free) and shaded nodes are observed (clamped).*Plates* indicate repetitions.

The predictive distribution for test case $y(x_j^*)$ depends *only* on the corresponding latent variable $f(x_j^*)$.

Adding other variables (without observations) doesn't change the distributions.
This explains why we can make inference using a finite amount of computation!

Some interpretation

Recall our main result:

$$\mathbf{f}_*|X_*, X, \mathbf{y} \sim \mathcal{N}\big(K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1}\mathbf{y},\\ K(X_*, X_*) - K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1}K(X, X_*)\big).$$

The mean is linear in two ways:

$$\mu(\mathbf{x}_*) = k(\mathbf{x}_*, X)[K(X,X) + \sigma_n^2]^{-1}\mathbf{y} = \sum_{c=1}^{n} \beta_c y^{(c)} = \sum_{c=1}^{n} \alpha_c k(\mathbf{x}_*, \mathbf{x}^{(c)}).$$

The last form is most commonly encountered in the kernel literature.

The variance is the difference between two terms:

$$V(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}(\mathbf{x}_*, X)[K(X,X) + \sigma_n^2 I]^{-1}\mathbf{k}(X, \mathbf{x}_*),$$

the first term is the *prior variance*, from which we subtract a (positive) term, telling how much the data X has explained. Note, that the variance is independent of the observed outputs $\mathbf{y}$.

The marginal likelihood

Log marginal likelihood:

$$\log p(\mathbf{y}|\mathbf{x}, M_i) \;=\; -\frac{1}{2}\mathbf{y}^\top K^{-1}\mathbf{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log(2\pi)$$

is the combination of a data fit term and complexity penalty. Occam's Razor is automatic.

Learning in Gaussian process models involves finding

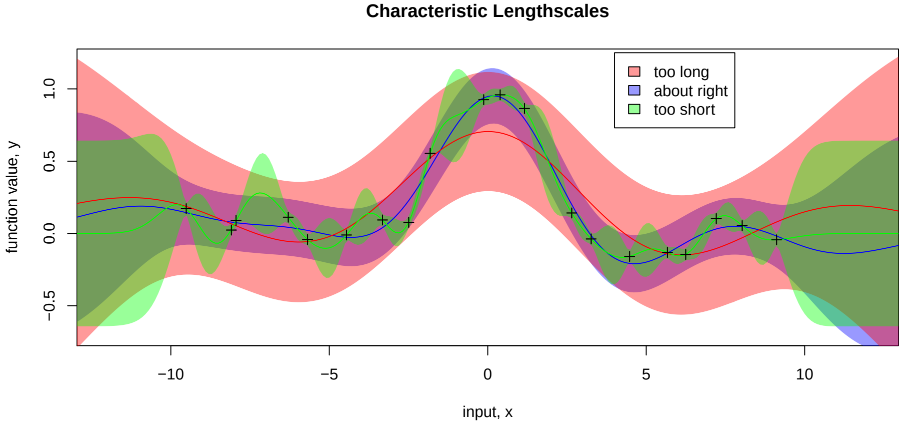
- the form of the covariance function, and
- any unknown (hyper-) parameters θ.

This can be done by optimizing the marginal likelihood:

$$\frac{\partial \log p(\mathbf{y}|\mathbf{x}, \theta, M_i)}{\partial \theta_j} \;=\; \frac{1}{2}\mathbf{y}^\top K^{-1}\frac{\partial K}{\partial \theta_j}K^{-1}\mathbf{y} - \frac{1}{2}\,\mathrm{trace}\big(K^{-1}\frac{\partial K}{\partial \theta_j}\big)$$

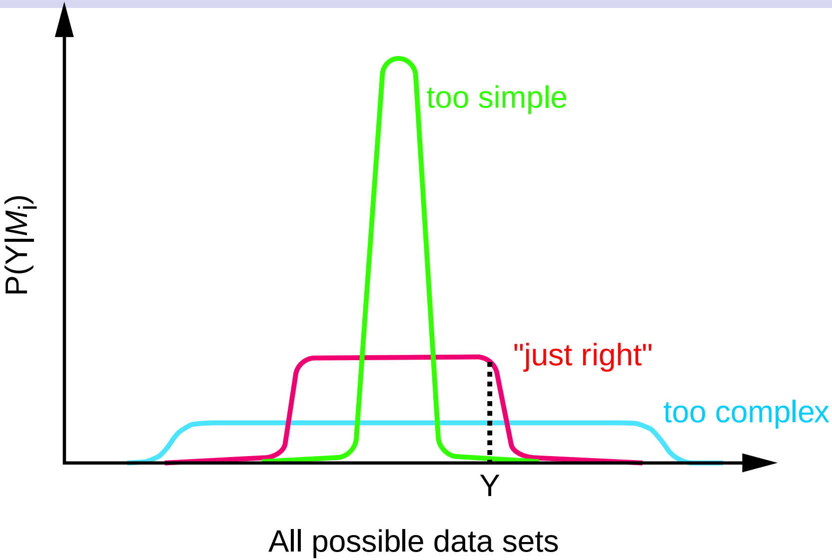
Example: Fitting the length scale parameter

Parameterized covariance function: $k(x, x') = v^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right) + \sigma_n^2 \delta_{xx'}$.



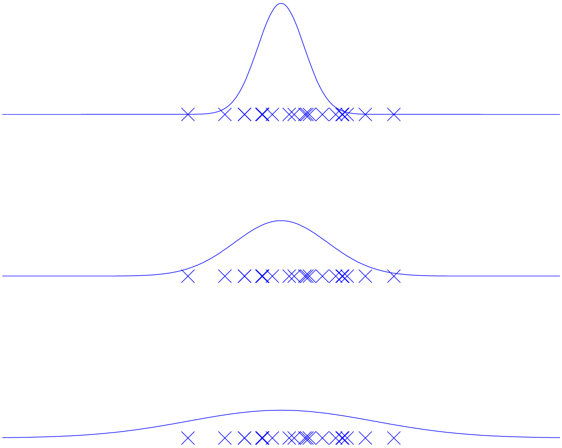
The posterior predictive density is plotted for 3 different length scales (the blue curve corresponds to optimizing the marginal likelihood). Notice, that an almost exact fit to the data can be achieved by reducing the length scale – but the marginal likelihood does not favour this!

Why, in principle, does Bayesian Inference work?
Occam's Razor



An illustrative analogous example

Imagine the simple task of fitting the variance, σ^2, of a zero-mean Gaussian to a set of n scalar observations.



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

From random functions to covariance functions

Consider the class of linear functions:

$$f(x) \;=\; ax + b, \quad \text{where} \;\; a \sim \mathcal{N}(0, \alpha), \quad \text{and} \;\; b \sim \mathcal{N}(0, \beta).$$

We can compute the mean function:

$$\mu(x) \;=\; E[f(x)] \;=\; \iint f(x)p(a)p(b)dadb \;=\; \int axp(a)da + \int bp(b)db = 0,$$

and covariance function:

$$\begin{aligned} k(x, x') \;&= E[(f(x) - 0)(f(x') - 0)] \;=\; \iint (ax + b)(ax' + b)p(a)p(b)dadb \\ &= \int a^2 xx' p(a)da + \int b^2 p(b)db + (x + x') \int abp(a)p(b)dadb = \alpha xx' + \beta. \end{aligned}$$

From random functions to covariance functions II

Consider the class of functions (sums of squared exponentials):

$$\begin{aligned} f(x) &= \lim_{n\to\infty} \frac{1}{n}\sum_i \gamma_i \exp(-(x-i/n)^2), \quad \text{where } \gamma_i \sim \mathcal{N}(0,1),\ \forall i \\ &= \int_{-\infty}^{\infty} \gamma(u)\exp(-(x-u)^2)du, \quad \text{where } \gamma(u) \sim \mathcal{N}(0,1),\ \forall u. \end{aligned}$$

The mean function is:

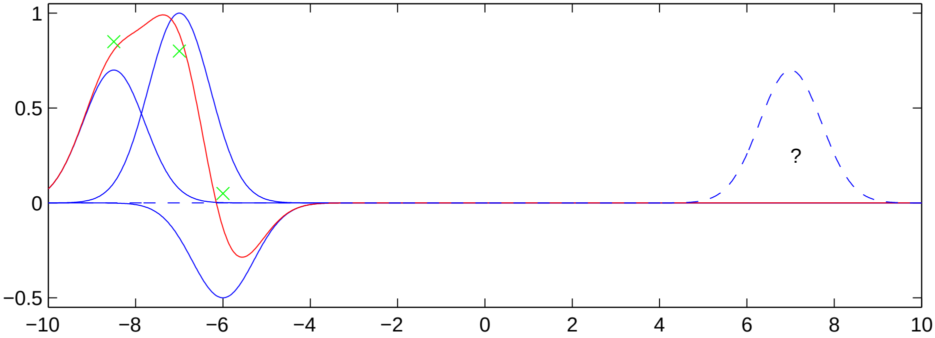
$$\mu(x) = E[f(x)] = \int_{-\infty}^{\infty} \exp(-(x-u)^2)\int_{-\infty}^{\infty} \gamma p(\gamma) d\gamma du = 0,$$

and the covariance function:

$$\begin{aligned} E[f(x)f(x')] &= \int \exp\big(-(x-u)^2 - (x'-u)^2\big)du \\ &= \int \exp\big(-2(u - \frac{x+x'}{2})^2 + \frac{(x+x')^2}{2} - x^2 - x'^2)\big)du \ \propto\ \exp\big(-\frac{(x-x')^2}{2}\big). \end{aligned}$$

Thus, the squared exponential covariance function is equivalent to regression using infinitely many Gaussian shaped basis functions placed everywhere, not just at your training points!

Using finitely many basis functions may be dangerous!

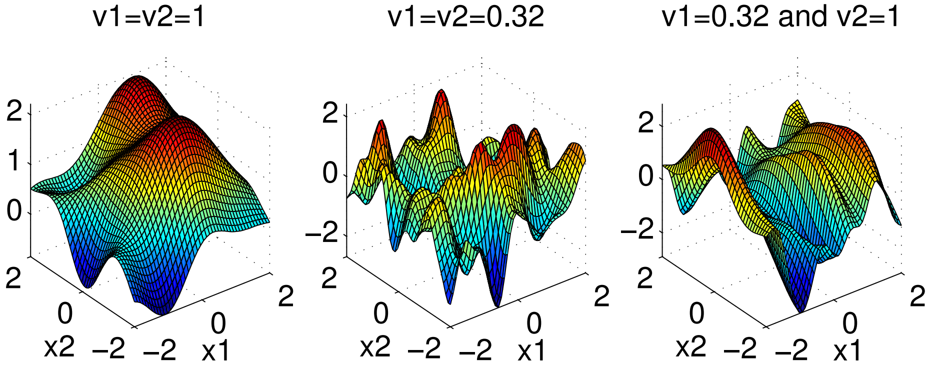


Model Selection in Practise; Hyperparameters

There are two types of task: *form* and *parameters* of the covariance function.

Typically, our prior is too weak to quantify aspects of the covariance function. We use a hierarchical model using hyperparameters. Eg, in ARD:

$$k(\mathbf{x}, \mathbf{x}') = v_0^2 \exp\big(-\sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{2v_d^2}\big), \qquad \text{hyperparameters } \theta = (v_0, v_1, \ldots, v_d, \sigma_n^2).$$



Rational quadratic covariance function

The *rational quadratic* (RQ) covariance function:

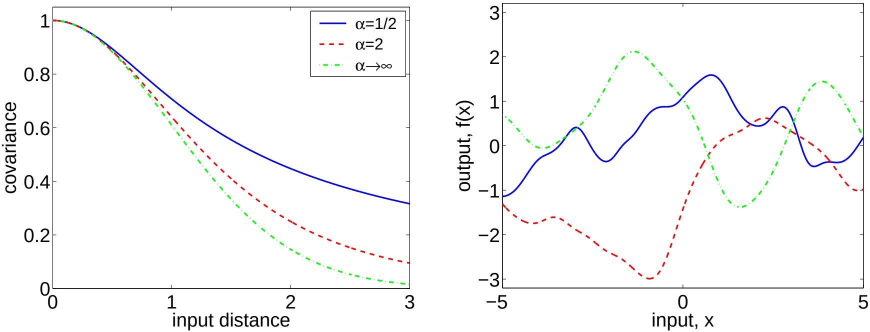
$$k_{\mathrm{RQ}}(r) \;=\; \Big(1 + \frac{r^2}{2\alpha\ell^2}\Big)^{-\alpha}$$

with $\alpha,\ \ell > 0$ can be seen as a *scale mixture* (an infinite sum) of squared exponential (SE) covariance functions with different characteristic length-scales.

Using $\tau = \ell^{-2}$ and $p(\tau|\alpha, \beta) \propto \tau^{\alpha-1}\exp(-\alpha\tau/\beta)$:

$$\begin{aligned} k_{\mathrm{RQ}}(r) \;&=\; \int p(\tau|\alpha, \beta) k_{\mathrm{SE}}(r|\tau) d\tau \\ &\propto \int \tau^{\alpha-1}\exp\Big(-\frac{\alpha\tau}{\beta}\Big)\exp\Big(-\frac{\tau r^2}{2}\Big) d\tau \;\propto\; \Big(1 + \frac{r^2}{2\alpha\ell^2}\Big)^{-\alpha}, \end{aligned}$$

Rational quadratic covariance function II



The limit $\alpha \to \infty$ of the RQ covariance function is the SE.

Matérn covariance functions

Stationary covariance functions can be based on the Matérn form:

$$k(\mathbf{x}, \mathbf{x}') = \frac{1}{\Gamma(\nu)2^{\nu-1}} \Big[\frac{\sqrt{2\nu}}{\ell}|\mathbf{x}-\mathbf{x}'|\Big]^{\nu} K_{\nu}\Big(\frac{\sqrt{2\nu}}{\ell}|\mathbf{x}-\mathbf{x}'|\Big),$$

where K_ν is the modified Bessel function of second kind of order ν, and ℓ is the characteristic length scale.

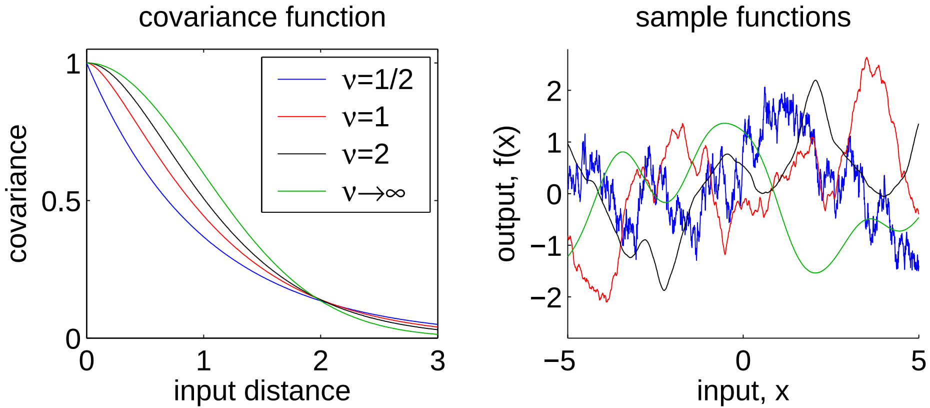
Sample functions from Matérn forms are $\lfloor \nu - 1 \rfloor$ times differentiable. Thus, the hyperparameter ν can control the degree of smoothness

Special cases:

- $k_{\nu=1/2}(r) = \exp(-\frac{r}{\ell})$: Laplacian covariance function, Browninan motion (Ornstein-Uhlenbeck)
- $k_{\nu=3/2}(r) = \big(1 + \frac{\sqrt{3}r}{\ell}\big) \exp\big(-\frac{\sqrt{3}r}{\ell}\big)$ (once differentiable)
- $k_{\nu=5/2}(r) = \big(1 + \frac{\sqrt{5}r}{\ell} + \frac{5r^2}{3\ell^2}\big) \exp\big(-\frac{\sqrt{5}r}{\ell}\big)$ (twice differentiable)
- $k_{\nu\to\infty} = \exp(-\frac{r^2}{2\ell^2})$: smooth (infinitely differentiable)

Matérn covariance functions II

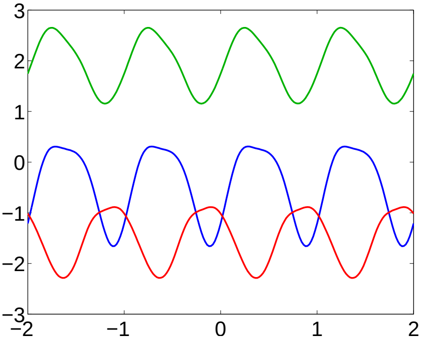
Univariate Matérn covariance function with unit characteristic length scale and unit variance:

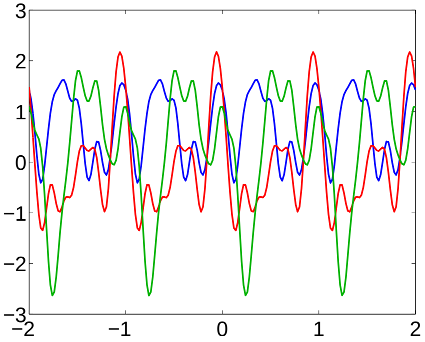


Periodic, smooth functions

To create a distribution over periodic functions of x, we can first map the inputs to $u = (\sin(x), \cos(x))^\top$, and then measure distances in the u space. Combined with the SE covariance function, which characteristic length scale ℓ, we get:

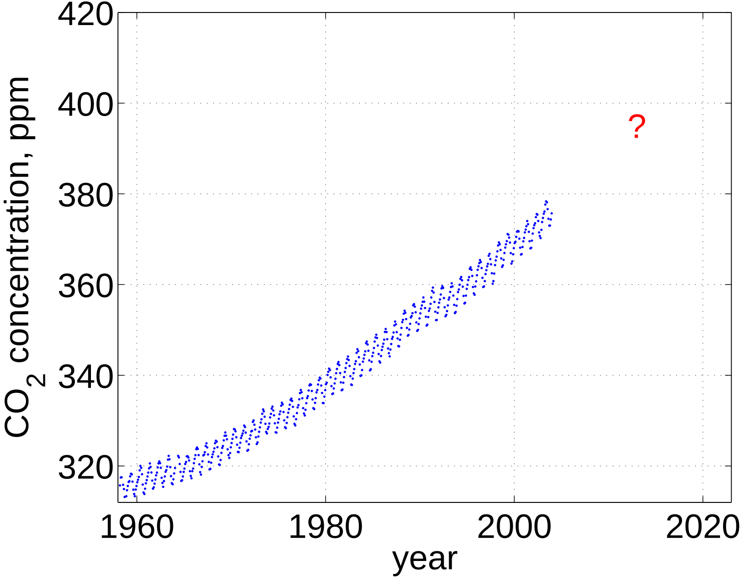
$$k_{\text{periodic}}(x, x') \;=\; \exp(-2\sin^2(\pi(x-x'))/\ell^2)$$





Three functions drawn at random; left $\ell > 1$, and right $\ell < 1$.

The Prediction Problem



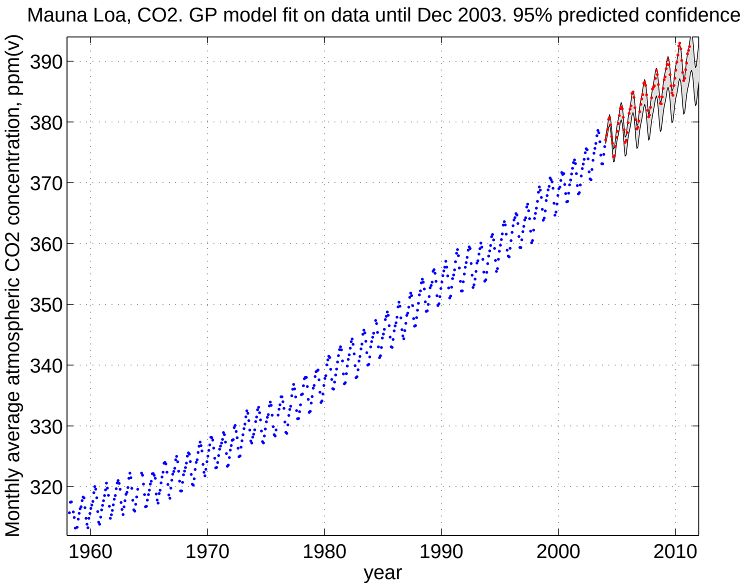
Covariance Function

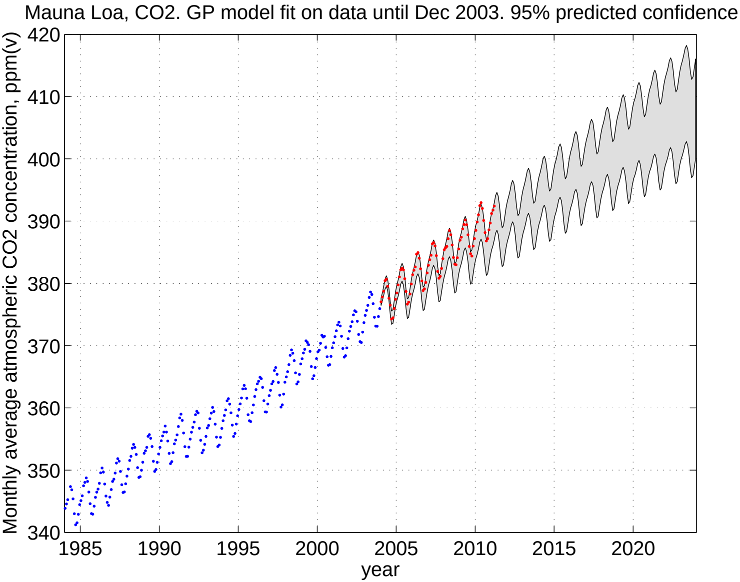
The covariance function consists of several terms, parameterized by a total of 11 *hyperparameters*:

- 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)
  $k_2(x, x') = \theta_3^2 \exp\left(-2\sin^2(\pi(x - x'))/\theta_5^2\right) \times \exp\left(-\frac{1}{2}(x - x')^2/\theta_4^2\right),$
- 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)
  $k_4(x, x') = \theta_9^2 \exp\left(-\frac{(x-x')^2}{2\theta_{10}^2}\right) + \theta_{11}^2\delta_{xx'}.$

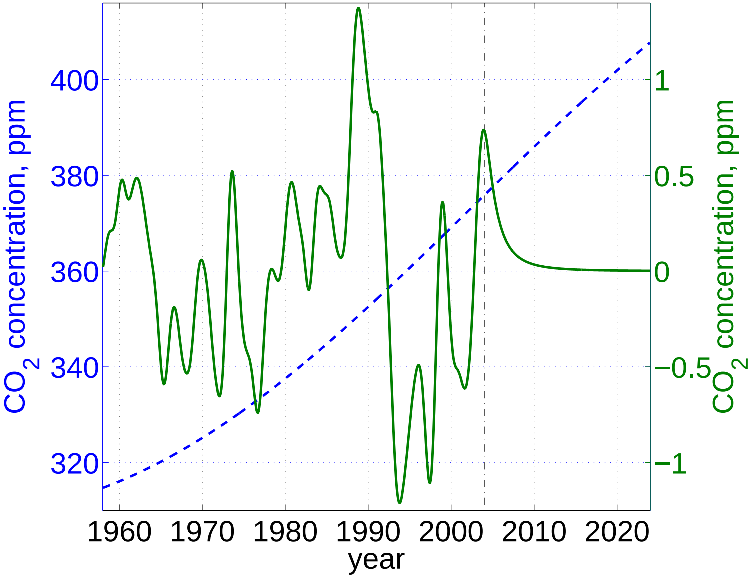
$$k(x, x') = k_1(x, x') + k_2(x, x') + k_3(x, x') + k_4(x, x')$$

Let's try this with the `gpml` software (`http://www.gaussianprocess.org/gpml`).

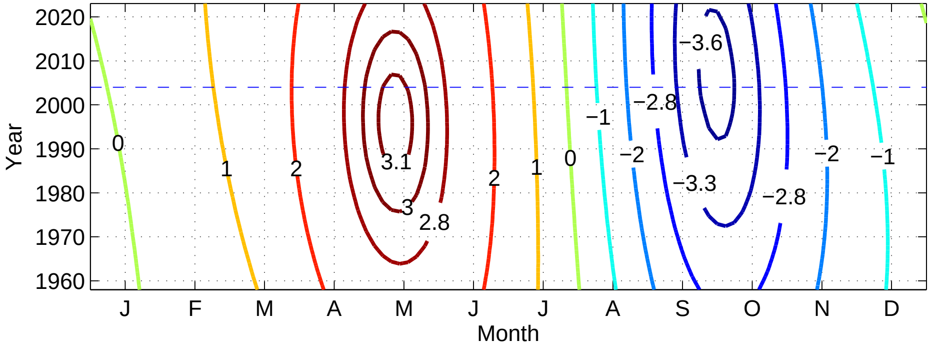

Mauna Loa, CO2. GP model fit on data until Dec 2003. 95% predicted confidence
Monthly average atmospheric CO2 concentration, ppm(v)
390
380
370
360
350
340
330
320
1960
1970
1980
1990
2000
2010
year


Mauna Loa, CO2. GP model fit on data until Dec 2003. 95% predicted confidence
Monthly average atmospheric CO2 concentration, ppm(v)
420
410
400
390
380
370
360
350
340
1985
1990
1995
2000
2005
2010
2015
2020
year

Long- and medium-term mean predictions



Mean Seasonal Component



Seasonal component: magnitude $\theta_3 = 2.4$ ppm, decay-time $\theta_4 = 90$ years.

Dependent noise, magnitude $\theta_9 = 0.18$ ppm, decay $\theta_{10} = 1.6$ months.
Independent noise, magnitude $\theta_{11} = 0.19$ ppm.

Optimize or integrate out? See MacKay [?].

Conclusions

Gaussian processes are intuitive, powerful and practical approach to inference, learning and prediction.

Bayesian inference is tractable, neatly addressing model complexity issues.

Predictions contain sensible error-bars, reflecting their confidence.

Many other models are (crippled versions) of GPs: Relevance Vector Machines (RVMs), Radial Basis Function (RBF) networks, splines, neural networks.