Lecture 3 and 4: Gaussian Processes

Machine Learning 4F13, Michaelmas 2015

Zoubin Ghahramani

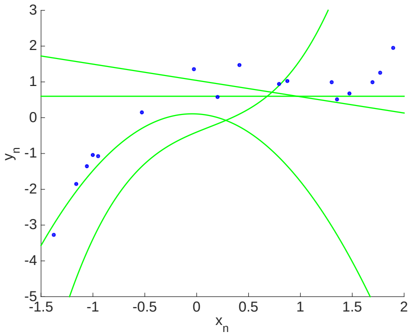
Department of Engineering
University of Cambridge

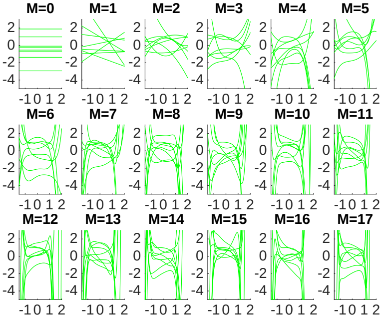
http://mlg.eng.cam.ac.uk/teaching/4f13/

Key concepts

- Distributions over parameters and over functions
  - Motivation: representaion of multiple hypothesis
  - concepts of prior over functions and over parameters
  - priors over functions are priors over long vectors
  - GP definition
  - joint generation and conditional generation
- Properties of Gaussian Processes
  - the predictive distribution
  - hyperparameters
  - the marginal likelihood for a GP
- Connections between linear in the parameters model and GPs
  - from finite linear models to GPs
  - weight space and function space views
  - infinite dimensional models and why finite dimensional models are dangerous

Old question, new marginal likelihood view





- Should we choose a polynomial? model structure
  we will address this soon
- What degree should we choose for the polynomial? model structure
  let the marginal likelihood speak
- For a given degree, how do we choose the weights? model parameters
  we consider many possible weights under the posterior
- For now, let find the single "best" polynomial: degree and weights.
  we don't do this sort of thing anymore

Marginal likelihood (Evidence) of our polynomials

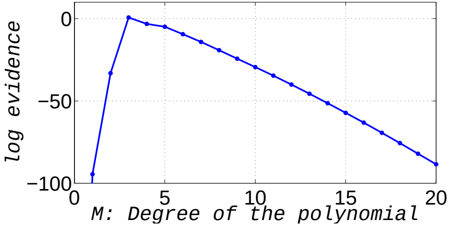
Marginal likelihood, or "evidence" of a finite linear model:

$$p(\mathbf{y}|\mathbf{x},\mathcal{M}) \;=\; \int p(\mathbf{f}|\mathbf{x},\mathcal{M})p(\mathbf{y}|\mathbf{f})d\mathbf{f} \;=\; \mathcal{N}(\mathbf{y};\,\mathbf{0},\sigma^2_{\mathbf{w}}\,\boldsymbol{\Phi}\,\boldsymbol{\Phi}^\top + \sigma^2_{\text{noise}}\,\mathbf{I})$$

For each polynomial degree, repeat the following infinitely many times:

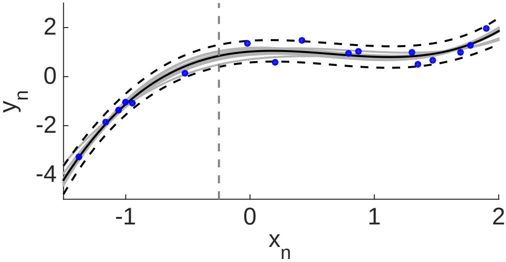
1. Sample a function $\mathbf{f}_s$ from the prior: $p(\mathbf{f}|\mathbf{x},\mathcal{M})$.
2. Compute the likelihood of that function for the data: $p(\mathbf{y}|\mathbf{f}_s)$.
3. Keep count of the number of samples so far: S.
4. The marginal likelihood is the average likelihood: $\frac{1}{S}\sum_{s=1}^{S} p(\mathbf{y}|\mathbf{f}_s)$

Luckily for Gaussian noise there is a closed-form analytical solution!



- The evidence prefers $M = 3$, not simpler, not more complex.
- Too simple models consistently miss most data.
- Too complex models frequently miss some data.

Multiple explanations of the data



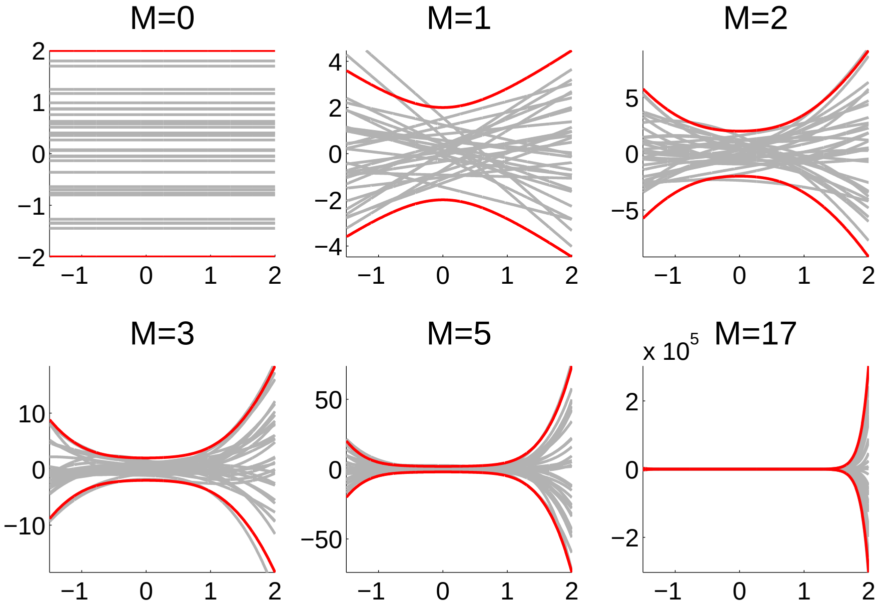
Remember that a finite linear model $f(x_n) = \boldsymbol{\phi}(x_n)^\top \mathbf{w}$ with prior on the weights $p(\mathbf{w}) = \mathcal{N}(\mathbf{w};\ \mathbf{0}, \sigma_{\mathbf{w}}^2 \mathbf{I})$ has a posterior distribution

$$p(\mathbf{w}|\mathbf{x}, \mathbf{y}, \mathcal{M}) = \mathcal{N}(\mathbf{w};\ \boldsymbol{\mu},\ \boldsymbol{\Sigma}) \quad \text{with} \quad \begin{aligned} \boldsymbol{\Sigma} &= \left(\sigma_{\text{noise}}^{-2} \boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \sigma_{\mathbf{w}}^{-2}\right)^{-1} \\ \boldsymbol{\mu} &= \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{\sigma_{\text{noise}}^2}{\sigma_{\mathbf{w}}^2} \mathrm{I}\right)^{-1} \boldsymbol{\Phi}^\top \mathbf{y} \end{aligned}$$

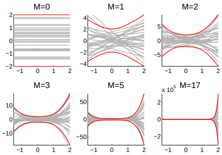
and predictive distribution

$$p(y_*|x_*, \mathbf{x}, \mathbf{y}, \mathcal{M}) \ = \ \mathcal{N}(y_*;\ \boldsymbol{\phi}(x_*)^\top \boldsymbol{\mu},\ \boldsymbol{\phi}(x_*)^\top \boldsymbol{\Sigma} \boldsymbol{\phi}(x_*) + \sigma_{\text{noise}}^2 \mathbf{I})$$

Are polynomials a good prior over functions?

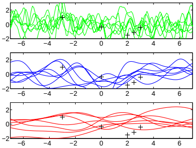


A prior over functions view



We have learnt that linear-in-the-parameter models with priors on the weights *indirectly* specify priors over functions.

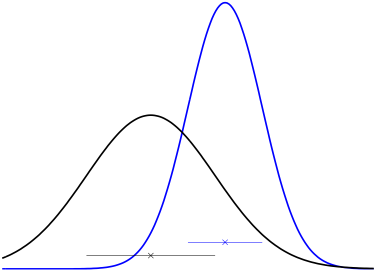
True... but those priors over functions might not be good.

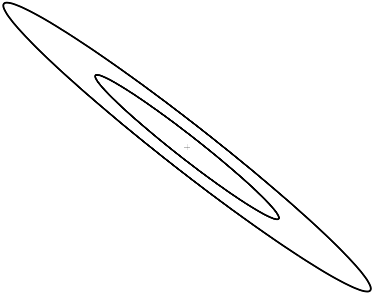


... why not try to specify priors over functions *directly*?

What? What does a probability density over functions even look like?

The Gaussian Distribution





The univariate Gaussian distribution is given by

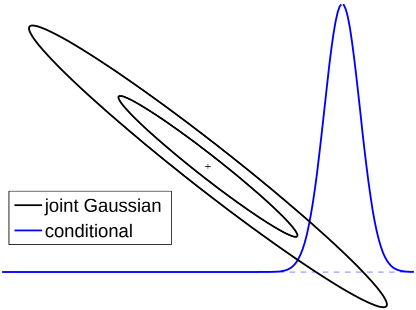
$$p(x|\mu, \sigma^2) = (2\pi\sigma^2)^{-1/2} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

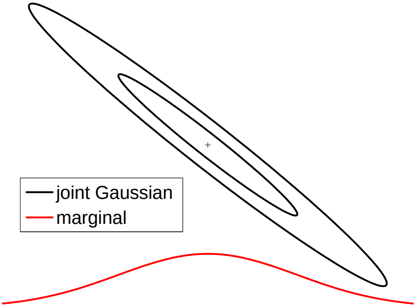
The multivariate Gaussian distribution for D-dimensional vectors is given by

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

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

Conditionals and Marginals of a Gaussian, pictorial





Both the conditionals $p(x|y)$ and the marginals $p(x)$ of a joint Gaussian $p(x, y)$ are again Gaussian.

Conditionals and Marginals of a Gaussian, algebra

If $\mathbf{x}$ and $\mathbf{y}$ are jointly Gaussian

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

we get the marginal distribution of $\mathbf{x}$, $p(\mathbf{x})$ by

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

and the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ by

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

where $\mathbf{x}$ and $\mathbf{y}$ 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:

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

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

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

here f and m are functions on $\mathcal{X}$, and k is a function on $\mathcal{X} \times \mathcal{X}$

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:

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

Random functions from a Gaussian Process

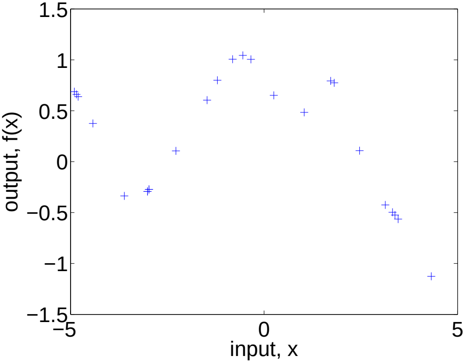
Example one dimensional Gaussian process:

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

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

$$\mathbf{f} \sim \mathcal{N}(0, \Sigma), \text{ where } \Sigma_{ij} = k(x_i, x_j).$$

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



Joint Generation

To generate a random sample from a D dimensional joint Gaussian with covariance matrix K and mean vector **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 **y** is:

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

Sequential Generation

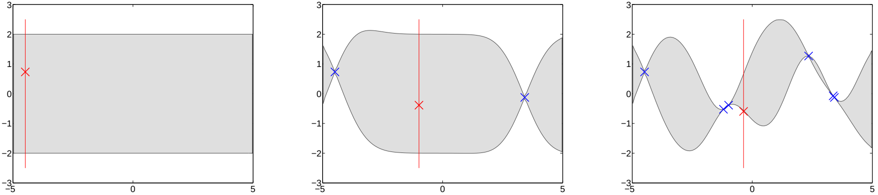
Factorize the joint distribution

$$p(f_1, \ldots, f_N | x_1, \ldots x_N) = \prod_{n=1}^{N} p(f_n | f_{n-1}, \ldots, f_1, x_n, \ldots, x_1),$$

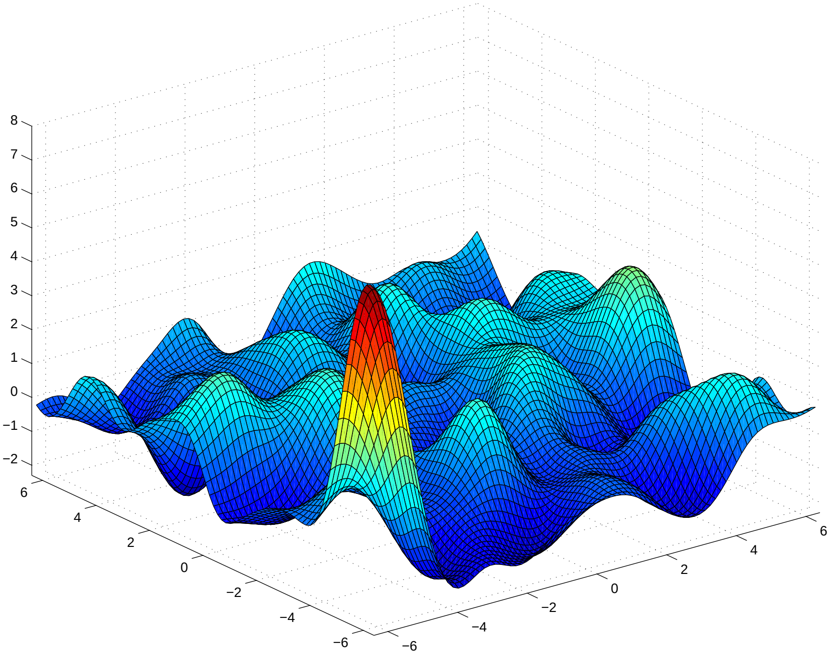
and generate function values sequentially. For Gaussians:

$$p(f_n, f_{<n}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}\right) \implies$$

$$p(f_n | f_{<n}) = \mathcal{N}(\mathbf{a} + BC^{-1}(f_{<n} - \mathbf{b}),\ A - BC^{-1}B^\top).$$



Function drawn at random from a Gaussian Process with Gaussian covariance



Non-parametric Gaussian process models

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

Gaussian likelihood, with noise variance σ^2_{noise}

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

Gaussian process prior with zero mean and covariance function k

$$p(f|\mathcal{M}_i) \sim \mathcal{GP}(m \equiv 0, \; k),$$

Leads to a Gaussian process posterior

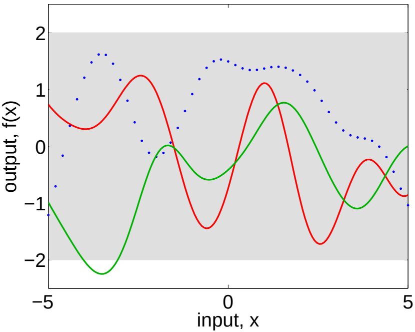
$$p(f|\mathbf{x}, \mathbf{y}, \mathcal{M}_i) \sim \mathcal{GP}(m_{\text{post}}, \; k_{\text{post}}),$$

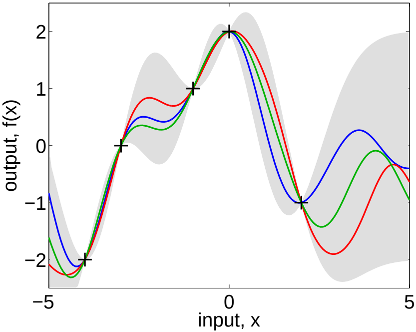
$$\text{where} \begin{cases} m_{\text{post}}(x) = k(x, \mathbf{x})[K(\mathbf{x}, \mathbf{x}) + \sigma^2_{\text{noise}} I]^{-1}\mathbf{y}, \\ k_{\text{post}}(x, x') = k(x, x') - k(x, \mathbf{x})[K(\mathbf{x}, \mathbf{x}) + \sigma^2_{\text{noise}} I]^{-1} k(\mathbf{x}, x'), \end{cases}$$

And a Gaussian predictive distribution:

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

Prior and Posterior





Predictive distribution:

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

Some interpretation

Recall our main result:

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

The mean is linear in two ways:

$$\mu(x_*) = k(x_*, \mathbf{x})[K(\mathbf{x}, \mathbf{x}) + \sigma^2_{\text{noise}} I]^{-1}\mathbf{y} = \sum_{n=1}^{N} \beta_n y_n = \sum_{n=1}^{N} \alpha_n k(x_*, x_n).$$

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

The variance is the difference between two terms:

$$V(x_*) = k(x_*, x_*) - \mathbf{k}(x_*, \mathbf{x})[K(\mathbf{x}, \mathbf{x}) + \sigma^2_{\text{noise}} I]^{-1} \mathbf{k}(\mathbf{x}, x_*),$$

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

The marginal likelihood

Log marginal likelihood:

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

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

Learning in Gaussian process models involves finding

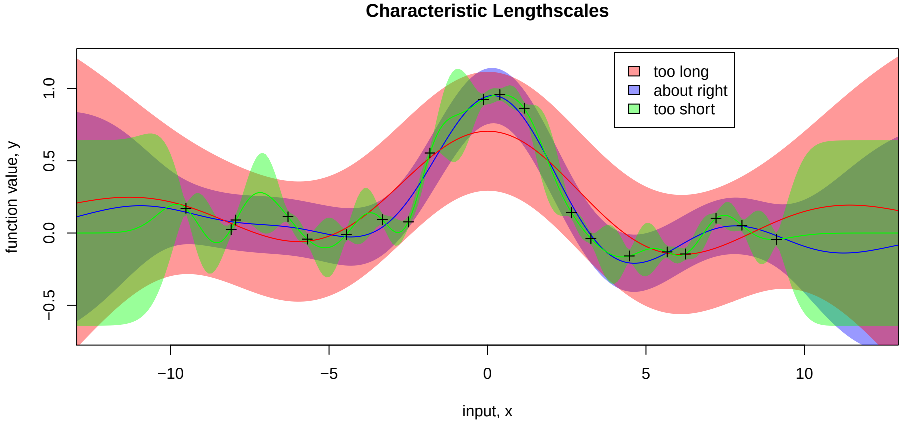
- the form of the covariance function, and
- any unknown (hyper-) parameters θ.

This can be done by optimizing the marginal likelihood:

$$\frac{\partial \log p(\mathbf{y}|\mathbf{x}, \theta, \mathcal{M}_i)}{\partial \theta_j} = \frac{1}{2}\mathbf{y}^\top K^{-1}\frac{\partial K}{\partial \theta_j}K^{-1}\mathbf{y} - \frac{1}{2}\operatorname{trace}(K^{-1}\frac{\partial K}{\partial \theta_j})$$

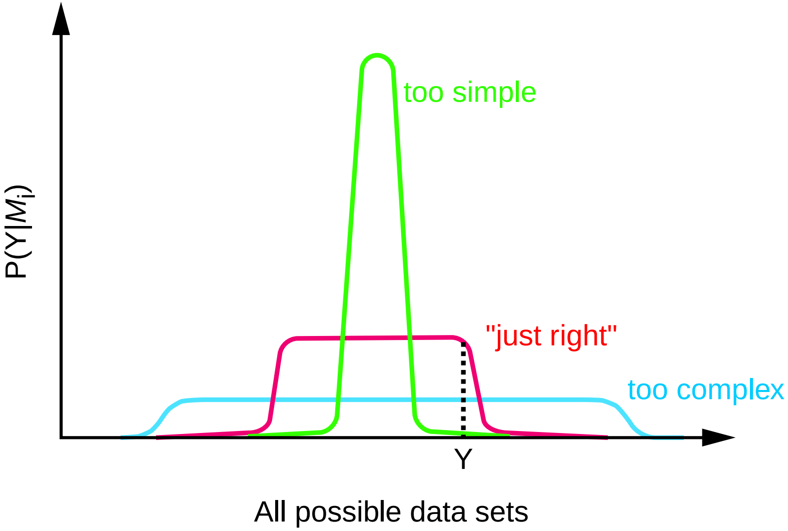
Example: Fitting the length scale parameter

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



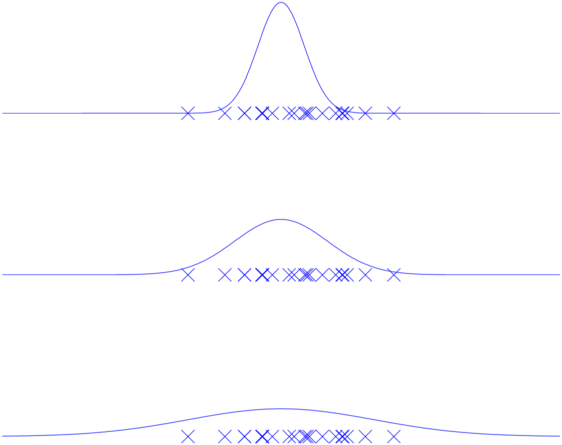
The mean posterior predictive function is plotted for 3 different length scales (the blue curve corresponds to optimizing the marginal likelihood). Notice, that an almost exact fit to the data can be achieved by reducing the length scale – but the marginal likelihood does not favour this!

How can Bayes rule help find the right model complexity? Marginal likelihoods and 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 finite linear models to Gaussian processes (1)

Finite linear model with Gaussian priors on the weights:

$$f(x) = \sum_{m=1}^{M} w_m \, \phi_m(x) \qquad\qquad p(\mathbf{w}) = \mathcal{N}(\mathbf{w};\ \mathbf{0}, A)$$

The joint distribution of any $\mathbf{f} = [f(x_1), \ldots, f(x_N)]^\top$ is a multivariate Gaussian – this looks like a Gaussian Process!

The prior $p(\mathbf{f})$ is fully characterized by the *mean* and *covariance* functions.

$$m(x) = E_{\mathbf{w}}\big(f(x)\big) = \int \Big(\sum_{m=1}^{M} w_k \phi_m(x)\Big) p(\mathbf{w}) d\mathbf{w} = \sum_{m=1}^{M} \phi_m(x) \int w_m p(\mathbf{w}) d\mathbf{w}$$

$$= \sum_{m=1}^{M} \phi_m(x) \int w_m p(w_m) dw_m = 0$$

The *mean function* is zero.

From finite linear models to Gaussian processes (2)

Covariance function of a finite linear model

$$f(x) = \sum_{m=1}^{M} w_m \phi_m(x) = \mathbf{w}^\top \boldsymbol{\phi}(x) \qquad \boldsymbol{\phi}(x) = [\phi_1(x), \ldots, \phi_M(x)]^\top_{(M\times 1)}$$

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w};\ \mathbf{0}, A)$$

$$k(x_i, x_j) = \mathrm{Cov}_{\mathbf{w}}\big(f(x_i), f(x_j)\big) = E_{\mathbf{w}}\big(f(x_i)f(x_j)\big) - \underbrace{E_{\mathbf{w}}\big(f(x_i)\big)E_{\mathbf{w}}\big(f(x_j)\big)}_{0}$$

$$= \int \ldots \int \Big(\sum_{k=1}^{M}\sum_{l=1}^{M} w_k w_l \phi_k(x_i)\phi_l(x_j)\Big) p(\mathbf{w})\, d\mathbf{w}$$

$$= \sum_{k=1}^{M}\sum_{l=1}^{M} \phi_k(x_i)\phi_l(x_j) \underbrace{\iint w_k w_l p(w_k, w_l) dw_k dw_l}_{A_{kl}} = \sum_{k=1}^{M}\sum_{l=1}^{M} A_{kl}\phi_k(x_i)\phi_l(x_j)$$

$$\boxed{k(x_i, x_j) = \boldsymbol{\phi}(x_i)^\top A \boldsymbol{\phi}(x_j)}$$

Note: If $A = \sigma_{\mathbf{w}}^2 \mathbf{I}$ then $k(x_i, x_j) = \sigma_{\mathbf{w}}^2 \sum_{k=1}^{M} \phi_k(x_i)\phi_k(x_j) = \sigma_{\mathbf{w}}^2 \boldsymbol{\phi}(x_i)^\top \boldsymbol{\phi}(x_j)$

GPs and Linear in the parameters models are equivalent

We've seen that a Linear in the parameters model, with a Gaussian prior on the weights is also a GP.

Note the different computational complexity: GP: $\mathcal{O}(N^3)$, linear model $\mathcal{O}(NM^2)$ where M is the number of basis functions and N the number of training cases.

So, which representation is most efficient?

Might it also be the case that every GP corresponds to a Linear in the parameters model? (Mercer's theorem.)

From infinite linear models to Gaussian processes

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

$$f(x) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=-\infty}^{\infty} \gamma_n \exp(-(x - \tfrac{n}{N})^2), \quad \text{where } \gamma_n \sim \mathcal{N}(0,1),\ \forall n$$
$$= \int_{-\infty}^{\infty} \gamma(u) \exp(-(x-u)^2)\,du, \quad \text{where } \gamma(u) \sim \mathcal{N}(0,1),\ \forall u.$$

The mean function is:

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

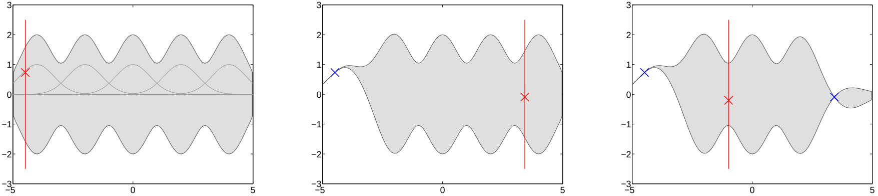
and the covariance function:

$$E[f(x)f(x')] = \int \exp\left(-(x-u)^2 - (x'-u)^2\right) du$$
$$= \int \exp\left(-2(u - \frac{x+x'}{2})^2 + \frac{(x+x')^2}{2} - x^2 - x'^2\right) du \ \propto \ \exp\left(-\frac{(x-x')^2}{2}\right).$$

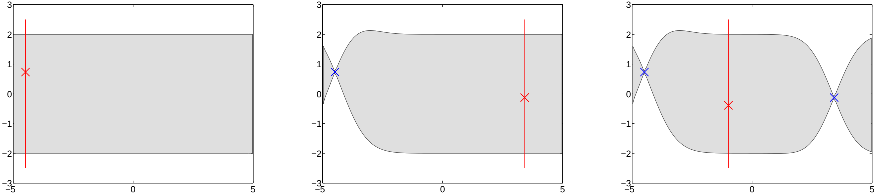
Thus, the squared exponential covariance function is equivalent to regression using infinitely many Gaussian shaped basis functions placed everywhere, not just at your training points!

Using finitely many basis functions may be dangerous!(1)

Finite linear model with 5 localized basis functions)

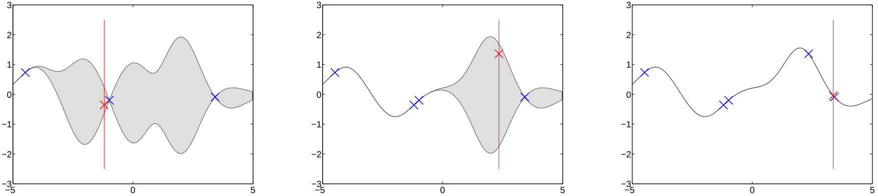


Gaussian process with infinitely many localized basis functions

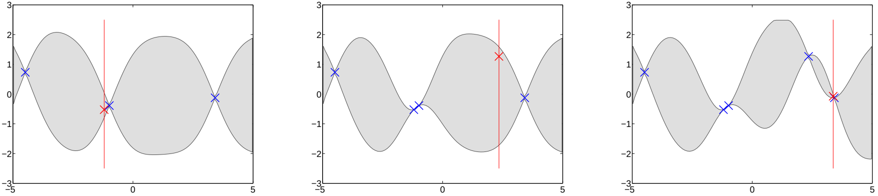


Using finitely many basis functions may be dangerous!(2)

Finite linear model with 5 localized basis functions)

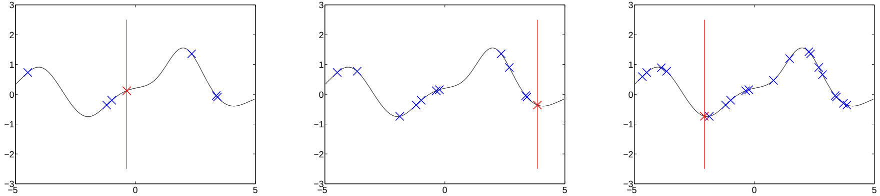


Gaussian process with infinitely many localized basis functions

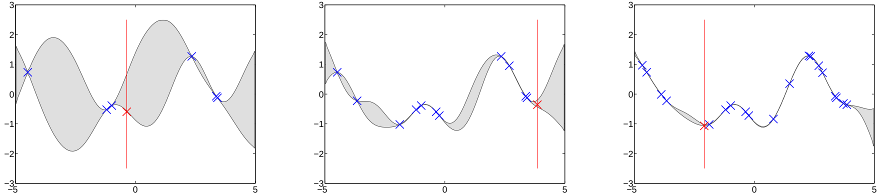


Using finitely many basis functions may be dangerous!(3)

Finite linear model with 5 localized basis functions)



Gaussian process with infinitely many localized basis functions



Matrix and Gaussian identities cheat sheet

Matrix identities

- Matrix inversion lemma (Woodbury, Sherman & Morrison formula)

$$(Z + UWV^\top)^{-1} = Z^{-1} - Z^{-1}U(W^{-1} + V^\top Z^{-1}U)^{-1}V^\top Z^{-1}$$

- A similar equation exists for determinants

$$|Z + UWV^\top| = |Z|\,|W|\,|W^{-1} + V^\top Z^{-1}U|$$

The product of two Gaussian density functions

$$\mathcal{N}(\mathbf{x}|\mathbf{a}, A)\,\mathcal{N}(P^\top\,\mathbf{x}|\mathbf{b}, B) = z_c\,\mathcal{N}(\mathbf{x}|\mathbf{c}, C)$$

- is proportional to a Gaussian density function with covariance and mean

$$C = \left(A^{-1} + P\,B^{-1}P^\top\right)^{-1} \qquad \mathbf{c} = C\,\left(A^{-1}\mathbf{a} + P\,B^{-1}\,\mathbf{b}\right)$$

- and has a normalizing constant z_c that is Gaussian both in $\mathbf{a}$ and in $\mathbf{b}$

$$z_c = (2\,\pi)^{-\frac{m}{2}}|B + P^\top A\,P|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\mathbf{b} - P^\top\,\mathbf{a})^\top\left(B + P^\top A\,P\right)^{-1}(\mathbf{b} - P^\top\,\mathbf{a})\right)$$