Gaussian Processes for Machine Learning

Carl Edward Rasmussen
Max Planck Institute for Biological Cybernetics
Tübingen, Germany
carl@tuebingen.mpg.de

Carlos III, Madrid, May 2006

The actual science of logic is conversant at present only with things either certain, impossible, or entirely doubtful, none of which (fortunately) we have to reason on. Therefore the true logic for this world is the calculus of Probabilities, which takes account of the magnitude of the probability which is, or ought to be, in a reasonable man's mind.

— James Clerk Maxwell [1850]

Course Goals

- To understand how learning for
  - regression
  - classification

  can be based on Gaussian processes

- how GPs relate to alternative methods

- the advantages and shortcomings of GPs

- some practical familiarity using matlab functions

Todays Outline

- Brief background on probability
- Gaussian processes
- Bayesian inference
- two views
- the process view
- Occam's Razor
- the weight-space view
- on the equivalence between the two views

Why Gaussian Processes

Gaussian processes are not new!

Used in spatial statistics, geostatistics (kriging), meteorology

Why are GPs not used more often?

- computational reasons
- Bayesian

Probabilities and Probability Densities

Probabilities are non-negative $p(x) \geq 0$

Probabilities normalize: $\sum_x p(x) = 1$ (discrete) or $\int_{-\infty}^{\infty} p(x)dx = 1$ (continuous).

The joint probability of x and y is $p(x, y)$.

The marginal probability of x is $p(x) = \int p(x, y)dy$.

The conditional probability of x given y is

$$p(x|y) = \frac{p(x, y)}{p(y)}.$$

Bayes Rule is given by

$$p(x, y) = p(x|y)p(y) = p(y|x)p(x) \Rightarrow p(y|x) = \frac{p(x|y)p(y)}{p(x)} \propto p(x|y)p(y).$$

Interpretations of $p(x)$:

- long run frequencies (frequentist, classical)
- subjective degree of belief (Bayesian)

Expectation and Variance (Moments)

The expectation (or *mean* or *average*) of a random variable is:

$$\mu \ = \ \mathbb{E}[x] \ = \ \int x p(x) dx \ = \ \langle x \rangle_{p(x)}.$$

The variance (or second central moment) is:

$$\sigma^2 \ = \ \mathbb{V}[x] \ = \ \int (x-\mu)^2 p(x) dx \ = \ \mathbb{E}[x^2] - (\mathbb{E}[x])^2.$$

The covariance between x and y:

$$\mathrm{cov}(x,y) \ = \ \mathbb{E}[(x-\mathbb{E}[x])(y-\mathbb{E}[y])].$$

If x and y are independent, then their covariance is zero.

The Gaussian Distribution

A joint, or multivariate Gaussian distribution in D dimensions:

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

is fully specified by its mean vector where $\boldsymbol{\mu}$ is the mean vector and Σ the covariance matrix.

Let $\mathbf{x}$ and $\mathbf{y}$ be jointly Gaussian random vectors

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_x\\ \boldsymbol{\mu}_y\end{bmatrix}, \begin{bmatrix}A & C\\ C^\top & B\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_x\\ \boldsymbol{\mu}_y\end{bmatrix}, \begin{bmatrix}\tilde{A} & \tilde{C}\\ \tilde{C}^\top & \tilde{B}\end{bmatrix}^{-1}\right),$$

then the *marginal* distribution of $\mathbf{x}$ and the *conditional* distribution of $\mathbf{x}$ given $\mathbf{y}$ are

$$\begin{aligned}\mathbf{x} \;\sim\; \mathcal{N}(\boldsymbol{\mu}_x, A), \;\; \text{and} \;\; \mathbf{x}|\mathbf{y} \;&\sim\; \mathcal{N}\big(\boldsymbol{\mu}_x + CB^{-1}(\mathbf{y}-\boldsymbol{\mu}_y),\; A - CB^{-1}C^\top\big)\\ \text{or} \;\; \mathbf{x}|\mathbf{y} \;&\sim\; \mathcal{N}\big(\boldsymbol{\mu}_x - \tilde{A}^{-1}\tilde{C}(\mathbf{y}-\boldsymbol{\mu}_y),\; \tilde{A}^{-1}\big).\end{aligned} \tag{1}$$

Products of Gaussians

The product of two Gaussians gives another (un-normalized) Gaussian

$$\begin{aligned}\mathcal{N}(\mathbf{x}|\mathbf{a},A)\mathcal{N}(\mathbf{x}|\mathbf{b},B) \; &= Z^{-1}\mathcal{N}(\mathbf{x}|\mathbf{c},C) \\ \text{where} \;\; \mathbf{c} \; &= C(A^{-1}\mathbf{a}+B^{-1}\mathbf{b}) \;\; \text{and} \;\; C \; = \; (A^{-1}+B^{-1})^{-1}.\end{aligned} \tag{2}$$

Notice that the resulting Gaussian has a precision (inverse variance) equal to the sum of the precisions and a mean equal to the convex sum of the means, weighted by the precisions. The normalizing constant looks itself like a Gaussian (in $\mathbf{a}$ or $\mathbf{b}$)

$$Z^{-1} \; = \; (2\pi)^{-D/2}|A+B|^{-1/2}\exp\big(-\tfrac{1}{2}(\mathbf{a}-\mathbf{b})^{\top}(A+B)^{-1}(\mathbf{a}-\mathbf{b})\big).$$

Gaussian Processes

Definition: A *Gaussian Process* is a collection of random variables any finite number of which have (consistent) joint Gaussian distributions.

Key observation: A Gaussian process is a natural generalization from vectors to functions:

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

and is fully specified by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x},\mathbf{x}')$.

This may sound very impractical...

...but we are saved by the marginalization property. Recall:

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

for Gaussians:

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

Gaussian Processes specify distributions over functions

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

```
  x = randn(D,1);
  y = chol(S)'*x + m;
```

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

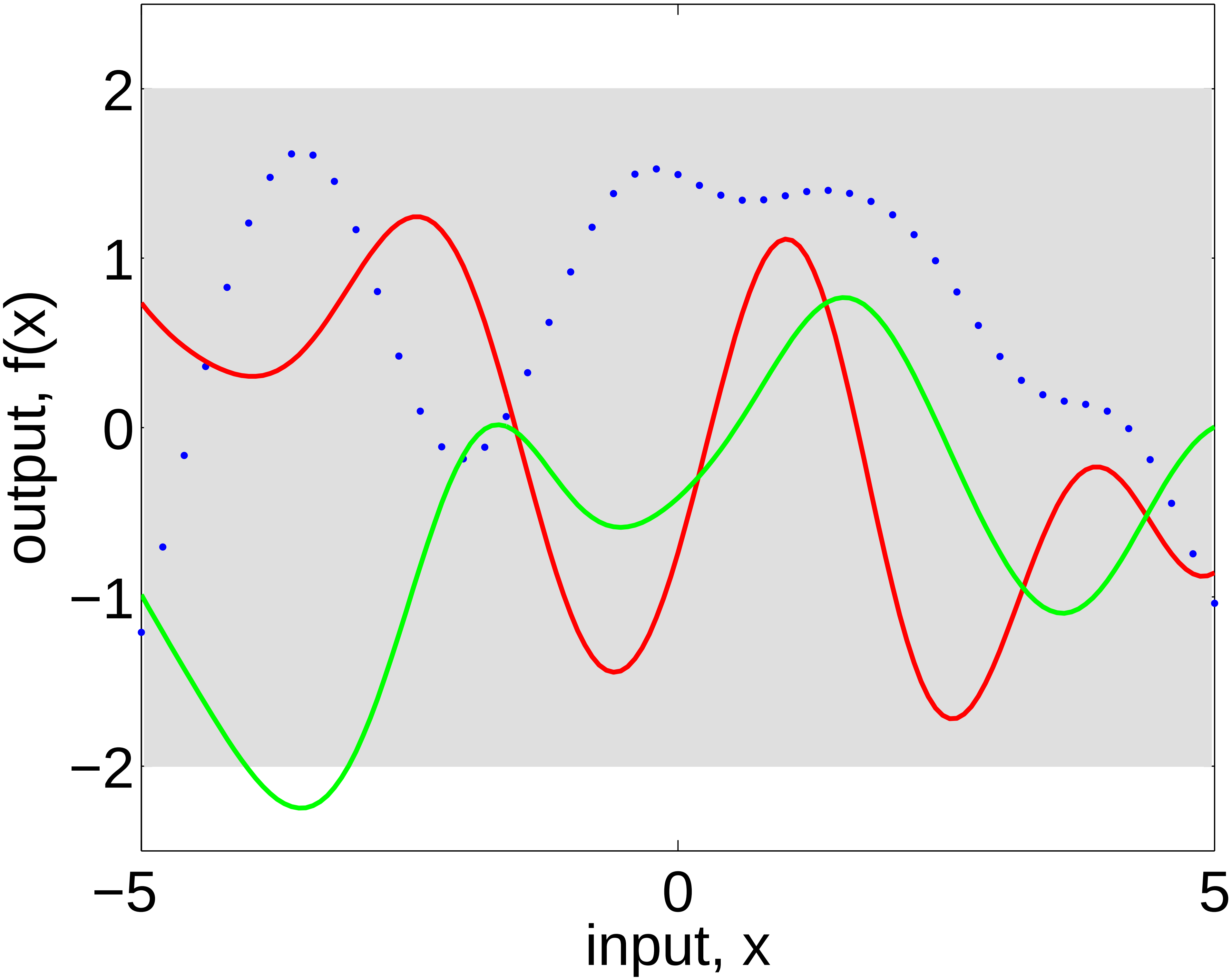
Thus, the covariance of $\mathbf{y}$ is:

$$\mathbb{E}[\mathbf{y}\mathbf{y}^\top] \;=\; \mathbb{E}[R^\top \mathbf{x}\mathbf{x}^\top R] \;=\; R^\top \mathbb{E}[\mathbf{x}\mathbf{x}^\top] R \;=\; R^\top I R \;=\; S.$$

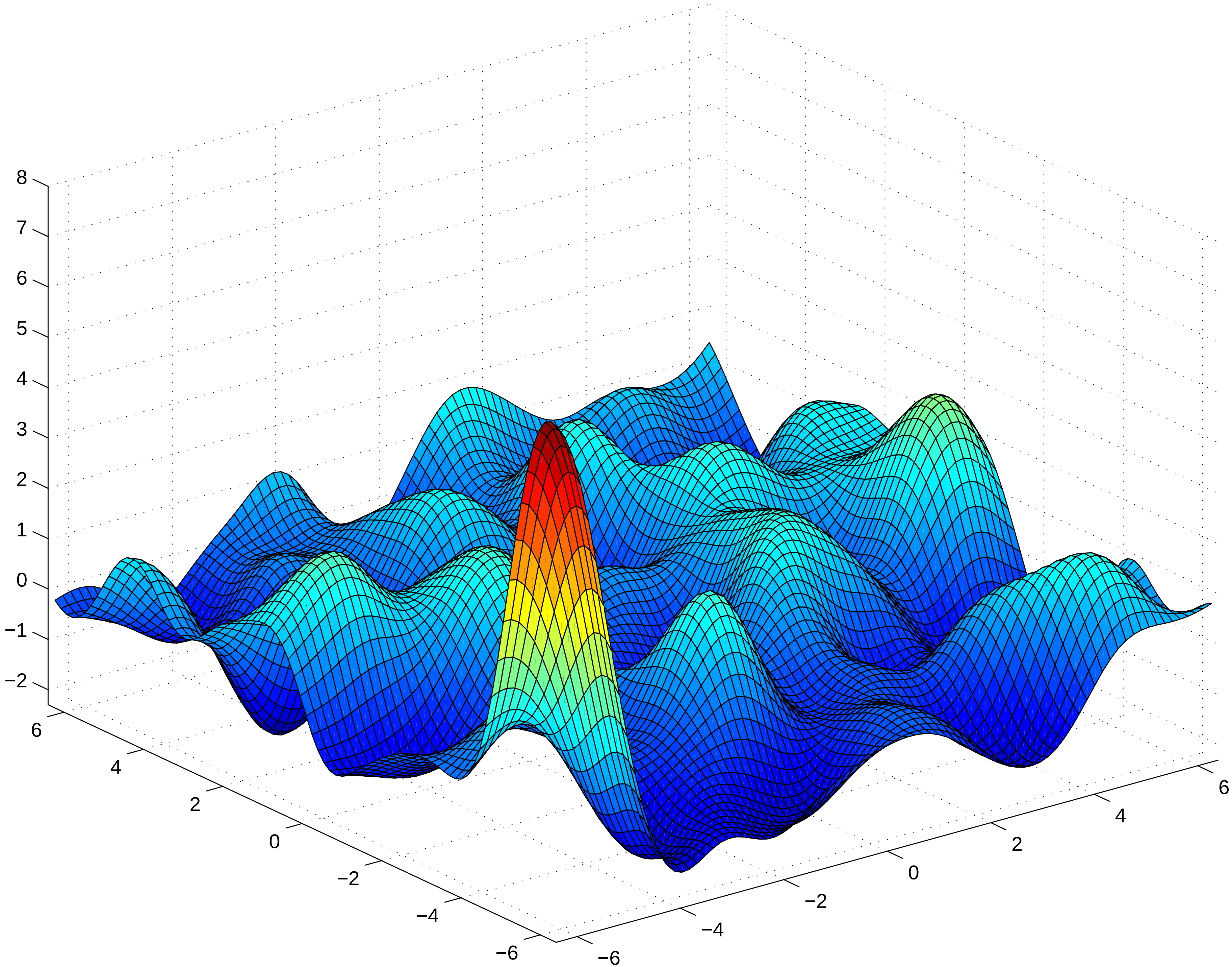
Example: smooth, stationary, Squared Exponential (or Gaussian) GP:

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

Do try this at home!


2
1
0
−1
−2
−5
0
5
output, f(x)
input, x

Function drawn at random from a Gaussian Process with Gaussian covariance



Bayesian Inference, parametric model

Supervised parametric learning:

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

Gaussian likelihood:

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

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:

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

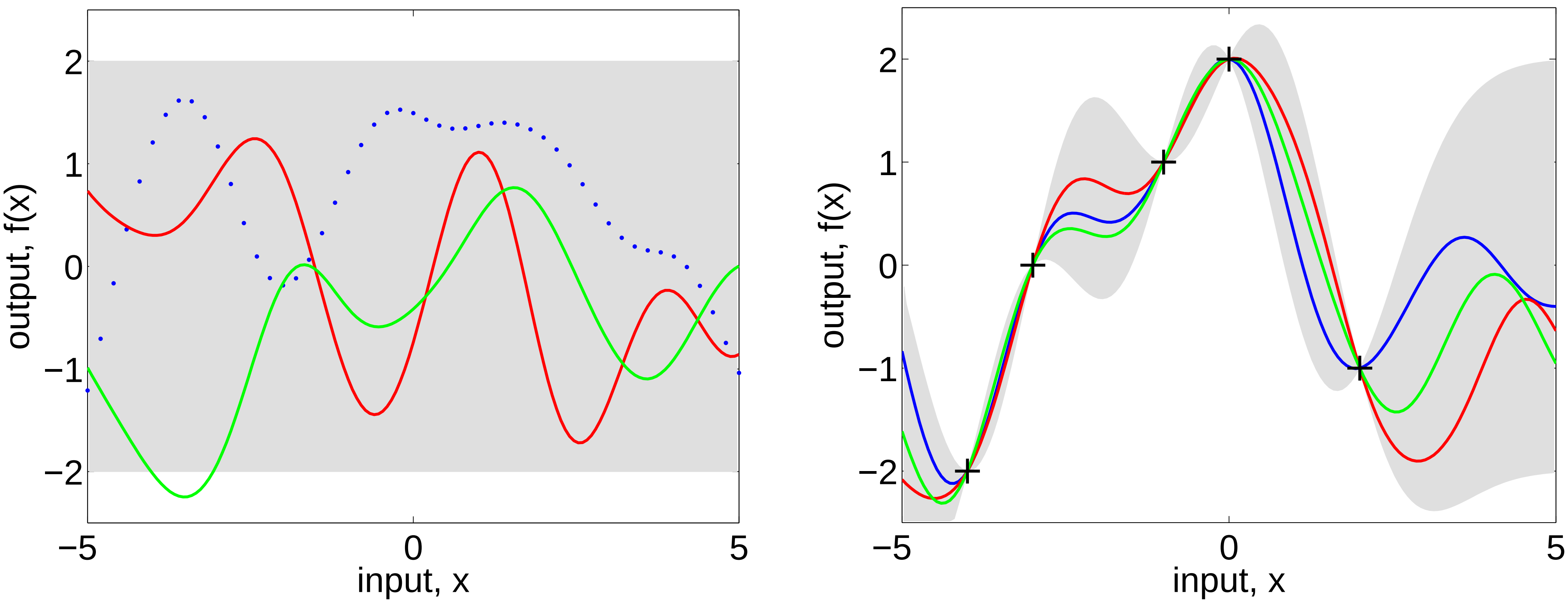
Leads to a Gaussian process posterior

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

And a Gaussian predictive distribution:

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

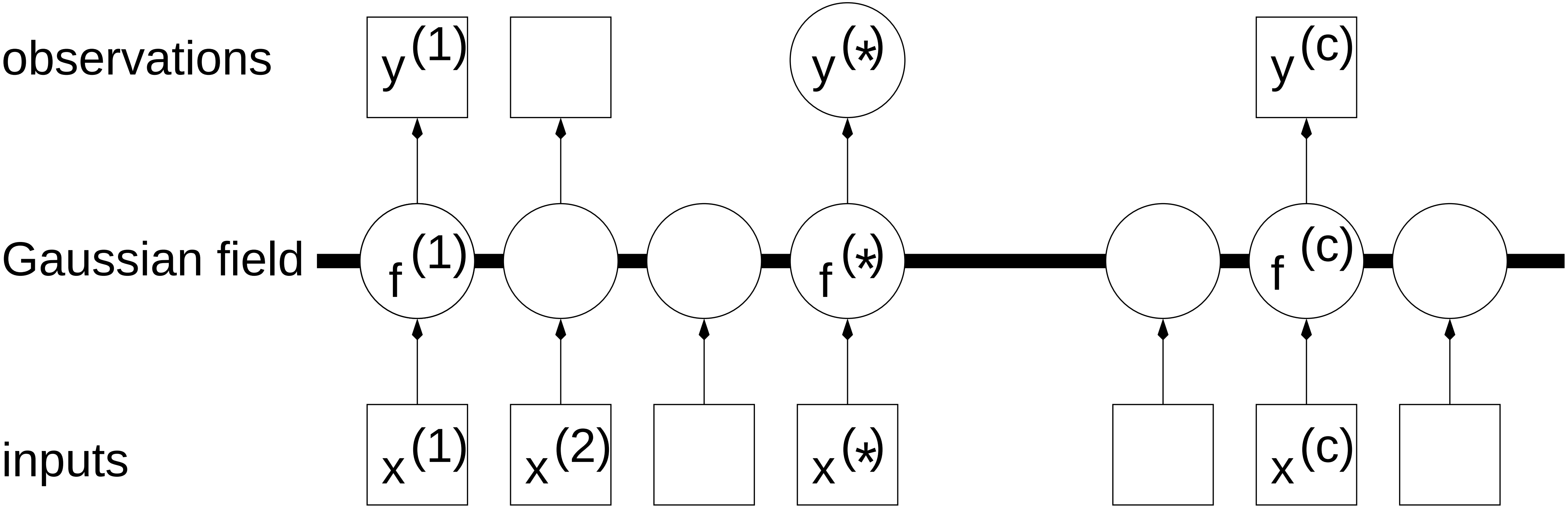
Prior and Posterior



Predictive distribution:

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

Graphical model for Gaussian Process



Square nodes are clamped, round nodes stochastic (free).

Thick black line indicates *every-one is connected to everyone*.

Notice, that adding a triplet $x^{(i)}, f^{(i)}, y^{(i)}$ (where the input $x^{(i)}$ is clamped) does not influence the distribution (as long as $y^{(i)}$ is not clamped). This is guaranteed by the consistency of the GP. This explains why we can make inference using a finite amount of computation!

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

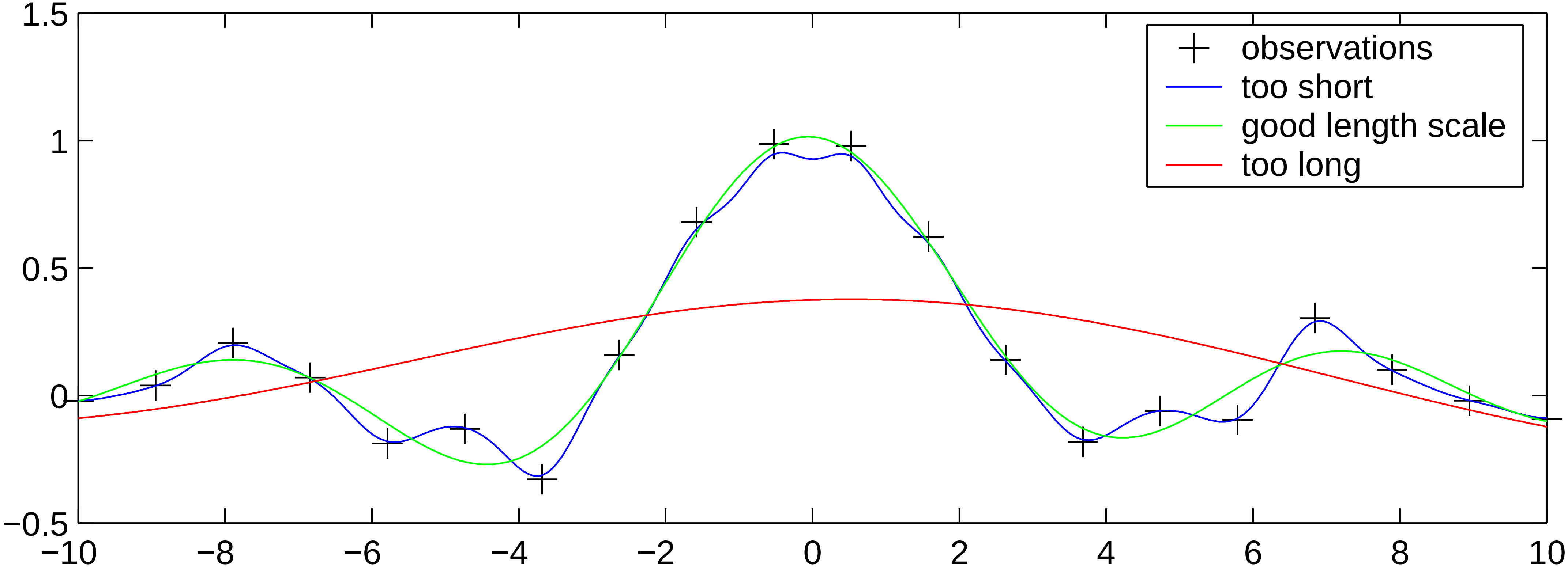
- the form of the covariance function, and
- any unknown (hyper-) parameters θ.

This can be done by optimizing the marginal likelihood:

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

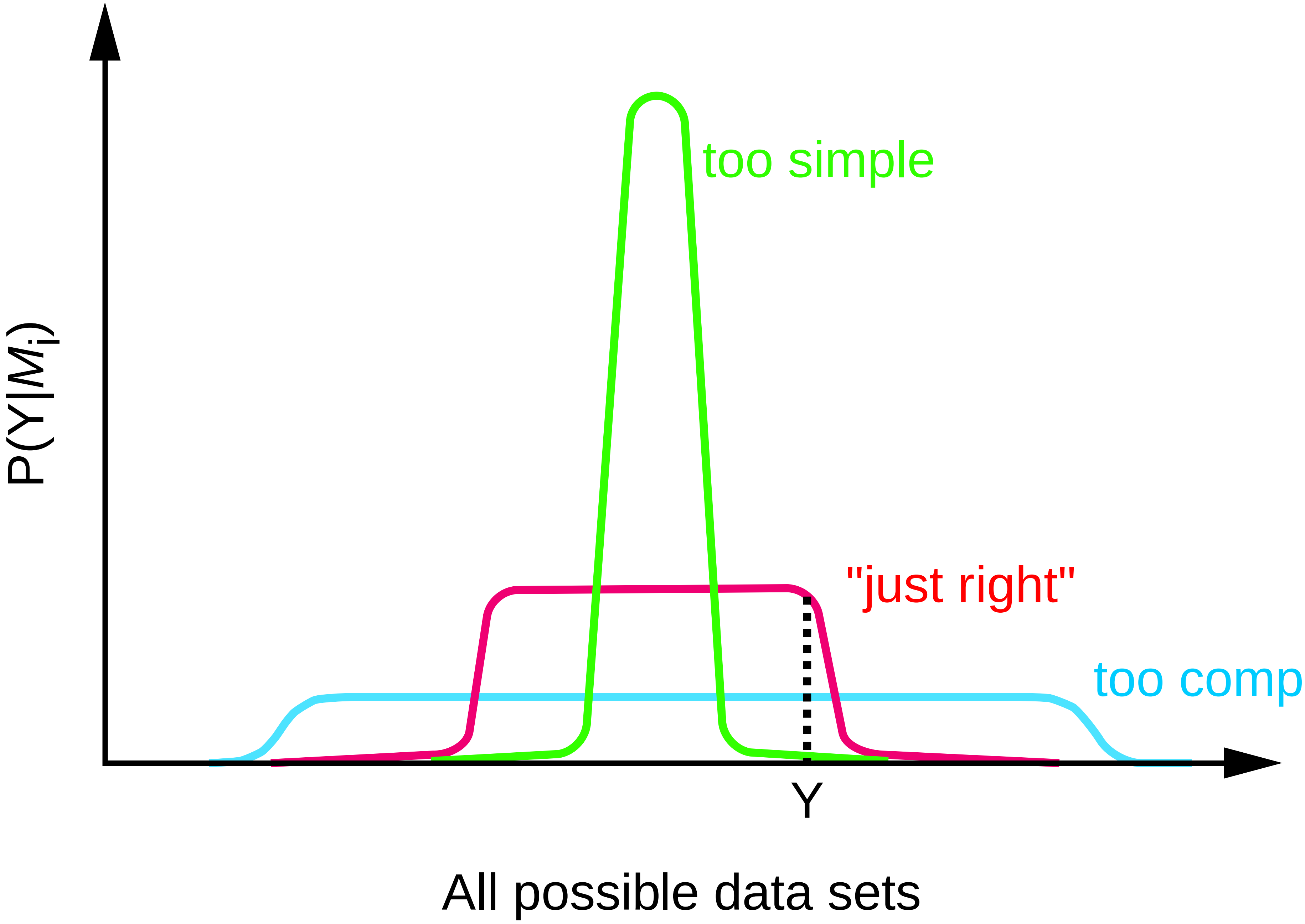
Example: Fitting the length scale parameter

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



The mean posterior predictive function is plotted for 3 different length scales (the green 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



The weight-space view

Linear model: assume $y = \mathbf{x}^\top \mathbf{w} + \varepsilon = f(\mathbf{x}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2_{\mathrm{noise}})$ is independent Gaussian noise.

Likelihood:

$$\begin{aligned} p(\mathbf{y}|X, \mathbf{w}) &= \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_n} \exp\big(- \frac{(y_i - \mathbf{x}_i^\top \mathbf{w})^2}{2\sigma_n^2}\big) \\ &= \frac{1}{(2\pi\sigma_n^2)^{n/2}} \exp\big(- \frac{1}{2\sigma_n^2} \|\mathbf{y} - X^\top \mathbf{w}\|^2\big) = \mathcal{N}(X^\top \mathbf{w}, \sigma_n^2 I). \end{aligned}$$

Note, that the likelihood also has a Gaussian *shape* w.r.t. the parameter $\mathbf{w}$, although not normalized.

Maximum Likelihood

Maximizing the likelihood over weights:

$$\frac{\partial \log p(\mathbf{y}|X, \mathbf{w})}{\partial \mathbf{w}} = 0 \implies \mathbf{w}_{\mathrm{ML}} \;=\; (XX^\top)^{-1}X\mathbf{y}.$$

(The solution could be ill determined if $XX^\top$ is close to singular.)

Predictive distribution:

$$p(f_*|\mathbf{x}_*, \mathcal{D}, \mathbf{w}_{\mathrm{ML}}) \;=\; \mathcal{N}(\mathbf{x}_*^\top \mathbf{w}_{\mathrm{ML}}, \sigma_n^2).$$

The Bayesian Approach

Define the prior distribution:

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

eg, $\Sigma_p = \sigma_p^2 I$, for some σ_p^2.

What does this prior mean?

Posterior $\propto$ Likelihood $\times$ Prior

$$\begin{aligned} p(\mathbf{w}|X, \mathbf{y}) \;&\propto\; \exp\big(-\frac{1}{2\sigma_n^2}(\mathbf{y} - X^\top \mathbf{w})^\top (\mathbf{y} - X^\top \mathbf{w})\big) \exp\big(-\frac{1}{2}\mathbf{w}^\top \Sigma_p^{-1} \mathbf{w}\big) \\ &\propto\; \exp\big(-\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^\top \big(\frac{1}{\sigma_n^2} X X^\top + \Sigma_p^{-1}\big)(\mathbf{w} - \bar{\mathbf{w}})\big), \end{aligned}$$

Ie, the posterior is Gaussian:

$$p(\mathbf{w}|X, \mathbf{y}) \;\sim\; \mathcal{N}(\bar{\mathbf{w}} = \frac{1}{\sigma_n^2} A^{-1} X \mathbf{y},\; A^{-1}),$$

where $A = \sigma_n^{-2} X X^\top + \Sigma_p^{-1}$.

Making Predictions

The predictive distribution is again Gaussian:

$$\begin{aligned} p(f_*|\mathbf{x}_*, X, \mathbf{y}) &= \int p(f_*|\mathbf{x}_*, \mathbf{w}) p(\mathbf{w}|X, \mathbf{y}) d\mathbf{w} = \int \mathbf{x}_*^\top \mathbf{w} \, p(\mathbf{w}|X, \mathbf{y}) d\mathbf{w} \\ &= \mathcal{N}(\frac{1}{\sigma_n^2} \mathbf{x}_*^\top A^{-1} X \mathbf{y}, \; \mathbf{x}_*^\top A^{-1} \mathbf{x}_*). \end{aligned}$$

Notice, how the error-bars increase when predicting further away from the origin. Why?

Outstanding Problems:

- The model is still too limited (only linear).
- What about σ_n^2 and Σ_p?

Nonlinear Regression

Instead of x, we can use a linear model in some fixed function basis $\phi(x) = (\varphi_1(x), \varphi_2(x), \ldots)^\top$.

Example: $\phi(x) = (1, x, x^2, \ldots)^\top$, turns:

$$f(x) \;=\; \phi(x)^\top \mathbf{w},$$

into polynomial regression, which is more flexible.

Note: the analysis from the previous slides can be reused by replacing x by $\phi(x)$.

The maximum likelihood parameters become:

$$\mathbf{w}_{\mathrm{ML}} \;=\; (\Phi(X)\Phi(X)^\top)^{-1}\Phi(X)\mathbf{y}.$$

with design matrix $\Phi(X) = (\phi(x_1), \phi(x_2), \ldots)$.

Re-introducing the Feature Space

The predictive distribution:

$$\begin{aligned} p(f_*|\mathbf{x}_*, X, \mathbf{y}) &= \int p(f_*|\mathbf{x}_*, \mathbf{w}) p(\mathbf{w}|X, \mathbf{y}) d\mathbf{w} \\ &= \mathcal{N}\big(\frac{1}{\sigma_n^2}\phi(\mathbf{x}_*)^\top A^{-1}\Phi(X)\mathbf{y},\ \phi(\mathbf{x}_*)^\top A^{-1}\phi(\mathbf{x}_*)\big), \end{aligned}$$

where $A = \sigma_n^{-2}\Phi(X)\Phi(X)^\top + \Sigma_p^{-1}$. This can be re-written as:

$$\begin{aligned} f_*|\mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}\big(&\boldsymbol{\phi}_*^\top \Sigma_p \Phi(K + \sigma_n^2 I)^{-1}\mathbf{y}, \\ &\boldsymbol{\phi}_*^\top \Sigma_p \boldsymbol{\phi}_* - \boldsymbol{\phi}_*^\top \Sigma_p \Phi(K + \sigma_n^2 I)^{-1}\Phi^\top \Sigma_p \boldsymbol{\phi}_*\big), \end{aligned}$$

where $K = \Phi^\top \Sigma_p \Phi$.

Notice, that this last expression contains $\phi(x)$ only in the form of inner products ⇒ can be possibly replaced by kernel functions $k(x, x') = \phi(x)^\top \Sigma_p \phi(x')$!

On the correspondence between weight space and function space views

There is an exact correspondence between the two views:

- Given a set of m basis functions, $\phi(\mathbf{x})$, it is obvious that we can construct the corresponding covariance function:
$$k(\mathbf{x}, \mathbf{x}') \;=\; \phi(\mathbf{x})^\top \Sigma_p \phi(\mathbf{x}').$$

- conversely, given a positive definite covariance function $k(\mathbf{x}, \mathbf{x}')$ we can decompose it (Mercer's theorem):
$$k(\mathbf{x}, \mathbf{x}') \;=\; \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{x}'),$$
where λ_i and ϕ_i are eigenvalues and eigenfunctions respectively, obeying:
$$\int k(\mathbf{x}, \mathbf{x}') \phi_i(\mathbf{x}) d\mathbf{x} \;=\; \lambda \phi_i(\mathbf{x}').$$

The function space view is valid irrespective of whether the corresponding eigendecomposition is finite.

The squared exponential covariance function does not have a finite decomposistion.

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:

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

Predictions from a noise free Gaussian Process

Imagine that we have observed the value of the function at some locations $\{X, \mathbf{f}\}$, and seek to make predictions $\mathbf{f}_*$ at (one or more) test inputs X_*.

The joint distribution under the GP is:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\big(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\big).$$

Now, we can *condition* on the obsevations, to obtain

$$\mathbf{f}_*|X_*, X, \mathbf{f} \ \sim \mathcal{N}\big(\mathbf{f}_*;\ K(X,X_*)K(X,X)^{-1}\mathbf{f},\ K(X_*,X_*) - K(X,X_*)K(X,X)^{-1}K(X_*,X)\big).$$

Remark: Conditioning a Gaussian gives another Gaussian:

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} A & C^{\mathrm{T}} \\ C & B \end{bmatrix}\right) \Longrightarrow x|y \sim \mathcal{N}(C^{\mathrm{T}}B^{-1}y, A - C^{\mathrm{T}}B^{-1}C)$$

Predictions from a noisy Gaussian Process

More commonly, we have access to *noisy* observations $\{X, \mathbf{y}\}$.

Assuming independent Gaussian noise, the joint distribution of observations and test predictions is:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\Big(\mathbf{0}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\Big),$$

Conditioning on the observations:

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

Note: Equating $K(X,Z)$ with $\Phi(X)^\top \Sigma_p \Phi(Z)$ for the Bayesian linear model, gives exactly identical results!

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 $\mathbf{y}$.