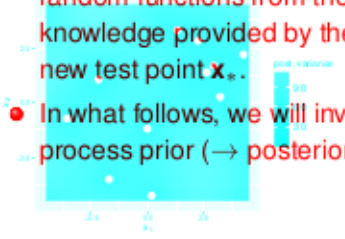
GAUSSIAN POSTERIOR PROCESS AND PREDICTION

Introduction to Machine Learning

Gaussian Process Prediction

- So far, we have learned how to **sample** from a GP prior.
- However, most of the time, we are not interested in drawing random functions from the prior. Instead, we usually like to use the knowledge provided by the training data to predict values of f at a new test point $\mathbf{x}_*$.
- In what follows, we will investigate how to update the Gaussian process prior ($\rightarrow$ posterior process) and how to make predictions.



Learning goals

- Know how to derive the posterior process
- GPs are interpolating and spatial models
- Model noise via a nugget term

GAUSSIAN POSTERIOR PROCESS AND PREDICTION

Gaussian Posterior Process and Prediction

- However, most of the time, we are not interested in drawing random functions from the prior. Instead, we usually like to use the knowledge provided by the training data to predict values of f at a new test point $\mathbf{x}_*$.
- In what follows, we will investigate how to update the Gaussian process prior ($\rightarrow$ posterior process) and how to make predictions.

POSTERIOR PROCESS

- Let us now distinguish between observed training inputs, also denote by a design matrix $\mathbf{X}$, and the corresponding observed values

$$\boldsymbol{f} = \left[f\left(\mathbf{x}^{(1)}\right), \ldots, f\left(\mathbf{x}^{(n)}\right)\right]$$

and one single **unobserved test point** $\mathbf{x}_*$ with $f_* = f(\mathbf{x}_*)$.

- We now want to infer the distribution of $f_* | \mathbf{x}_*, \boldsymbol{X}, \boldsymbol{f}$.

$$f_* = f(\mathbf{x}_*)$$

- Assuming a zero-mean GP prior $\mathcal{GP}(\mathbf{0}, k(\mathbf{x}, \mathbf{x}'))$ we know

$$\begin{bmatrix} \boldsymbol{f} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \boldsymbol{k}_* \\ \boldsymbol{k}_*^T & \boldsymbol{k}_{**} \end{bmatrix}\right).$$

Here, $\mathbf{K} = \left(k\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right)\right)_{i,j}$, $\boldsymbol{k}_* = \left[k\left(\mathbf{x}_*, \mathbf{x}^{(1)}\right), \ldots, k\left(\mathbf{x}_*, \mathbf{x}^{(n)}\right)\right]$ and $\boldsymbol{k}_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$.

POSTERIOR PROCESS / 2

- Given that $\boldsymbol{f}$ is observed, we can apply the general rule for condition $^{(*)}$ of Gaussian random variables and obtain the following formula:

$$\boldsymbol{f}_* | \mathbf{x}_*, \boldsymbol{X}, \boldsymbol{f} \sim \mathcal{N}\left(\boldsymbol{k}_*^T \mathbf{K}^{-1} \boldsymbol{f}, k_{**} - \boldsymbol{k}_*^T \mathbf{K}^{-1} \boldsymbol{k}_*\right).$$

- We now want to infer the distribution of $f_* | \mathbf{x}_*, \boldsymbol{X}, \boldsymbol{f}$.
- As the posterior is a Gaussian, the maximum a-posteriori estimate, i.e. the mode of the posterior distribution, is $\boldsymbol{k}_*^T \mathbf{K}^{-1} \boldsymbol{f}$.

$$f_* = f(\mathbf{x}_*)$$

- Assuming a zero-mean GP prior $\mathcal{GP}(\mathbf{0}, k(\mathbf{x}, \mathbf{x}'))$ we know

$$\begin{bmatrix} \boldsymbol{f} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \boldsymbol{k}_* \\ \boldsymbol{k}_*^T & k_{**} \end{bmatrix}\right).$$

Here, $\mathbf{K} = \left(k\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right)\right)_{i,j}$, $\boldsymbol{k}_* = \left[k\left(\mathbf{x}_*, \mathbf{x}^{(1)}\right), ..., k\left(\mathbf{x}_*, \mathbf{x}^{(n)}\right)\right]$ and $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$.

POSTERIOR PROCESS / 3

General rule for condition of Gaussian random variables:

If the m-dimensional Gaussian vector $\boldsymbol{z} \sim \mathcal{N}(\mu, \Sigma)$ can be partitioned with $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2)$ where $\boldsymbol{z}_1$ is m_1-dimensional and $\boldsymbol{z}_2$ is m_2-dimensional, and:

$$(\mu_1, \mu_2), \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

then the conditioned distribution of $\boldsymbol{z}_2 \mid \boldsymbol{z}_1 = \boldsymbol{a}$ is a multivariate normal

$$\mathcal{N}\left(\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}\left(\boldsymbol{a} - \mu_1\right), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right)$$

GP PREDICTION: TWO POINTS

Let us visualize this by a simple example:

- Assume we observed a single training point $\mathbf{x} = -0.5$, and want to make a prediction at a test point $\mathbf{x}_* = 0.5$.
- Under a zero-mean GP with $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right)$, we compute the cov-matrix:

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} 1 & 0.61 \\ 0.61 & 1 \end{bmatrix}\right).$$

- Assume that we observe the point $f(\mathbf{x}) = 1$.
- We compute the posterior distribution:

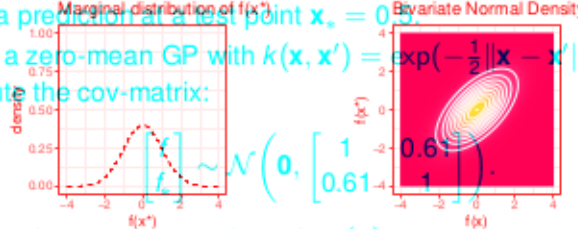
$$\begin{aligned} f_* \mid \mathbf{x}_*, \mathbf{x}, f &\sim \mathcal{N}(\boldsymbol{k}_*^T \mathbf{K}^{-1} f, k_{**} - \boldsymbol{k}_*^T \mathbf{K}^{-1} \boldsymbol{k}_*) \\ &\sim \mathcal{N}(0.61 \cdot 1 \cdot 1, 1 - 0.61 \cdot 1 \cdot 0.61) \\ &\sim \mathcal{N}(0.61, 0.6279) \end{aligned}$$

- The MAP-estimate for $\mathbf{x}_*$ is $f(\mathbf{x}_*) = 0.61$, and the uncertainty estimate is 0.6279.

GP PREDICTION: TWO POINTS

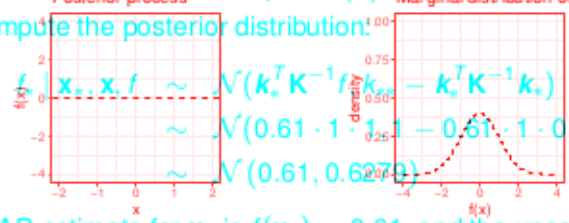
Shown is the bivariate normal density, and the respective marginals.

- Assume we observed a single training point $\mathbf{x} = -0.5$, and want to make a prediction at a test point $\mathbf{x}_* = 0.5$.
- Under a zero-mean GP with $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right)$, we compute the cov-matrix:



$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} 1 & 0.61 \\ 0.61 & 1 \end{bmatrix}\right).$$

- Assume that we observe the point $f(\mathbf{x}) = 1$.
- We compute the posterior distribution:

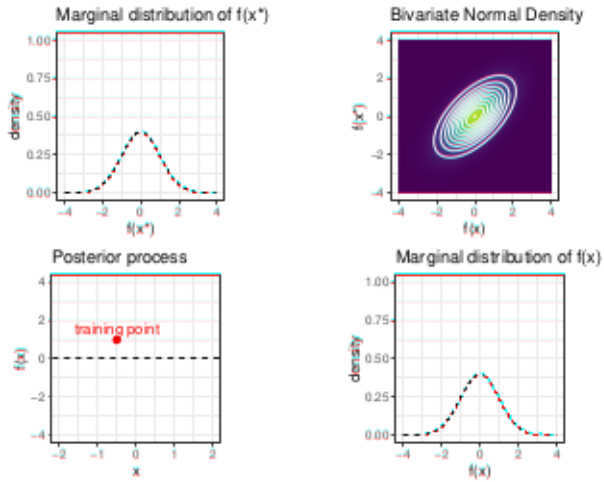


$$\begin{aligned} f_* \mid \mathbf{x}_*, \mathbf{x}, f &\sim \mathcal{N}(\boldsymbol{k}_*^T \mathbf{K}^{-1} f, k_{**} - \boldsymbol{k}_*^T \mathbf{K}^{-1} \boldsymbol{k}_*) \\ &\sim \mathcal{N}(0.61 \cdot 1 \cdot 1, 1 - 0.61 \cdot 1 \cdot 0.61) \\ &\sim \mathcal{N}(0.61, 0.6279) \end{aligned}$$

- The MAP-estimate for $\mathbf{x}_*$ is $f(\mathbf{x}_*) = 0.61$, and the uncertainty estimate is 0.6279.

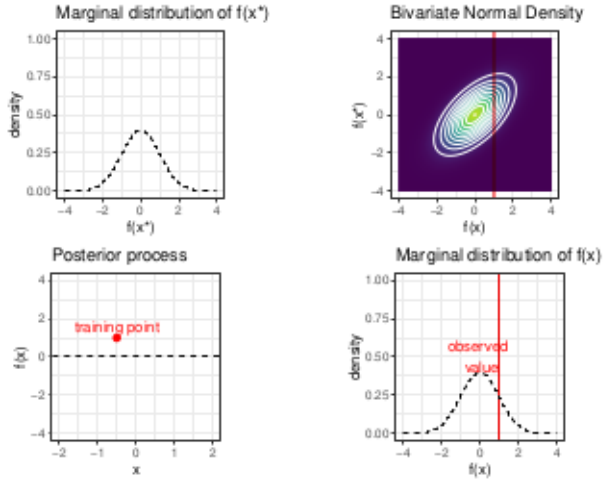
GP PREDICTION: TWO POINTS

Assume we observed $f(\mathbf{x}) = 1$ for the training point $\mathbf{x} = 0.5$.



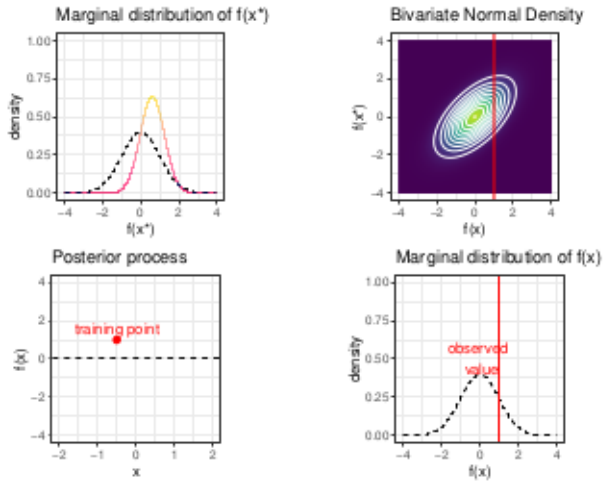
GP PREDICTION: TWO POINTS

We condition the Gaussian on $f(\mathbf{x}) = 1$ the training point $\mathbf{x} = -0.5$.



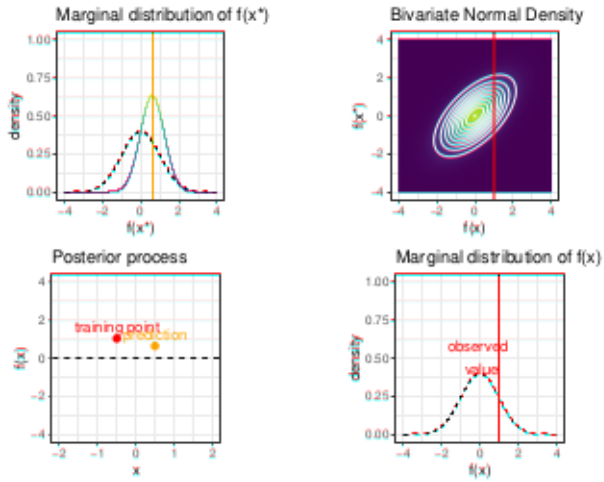
GP PREDICTION: TWO POINTS

We compute the posterior distribution of $f(\mathbf{x}_*)$ given that $f(\mathbf{x}) = 1$.



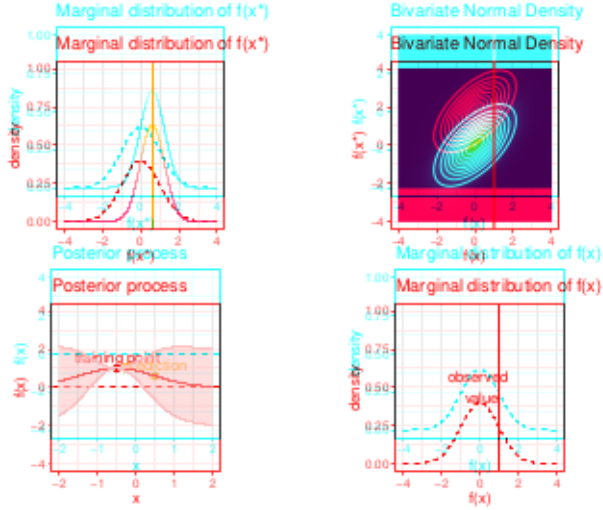
GP PREDICTION: TWO POINTS

A possible predictor for a x_* is the MAP of the posterior distribution.



GP PREDICTION: TWO POINTS

We can do this for different values x_* and show the respective mean (grey line) and standard deviations (grey area is mean $\pm 2 \cdot$ posterior standard deviation).



POSTERIOR PROCESS

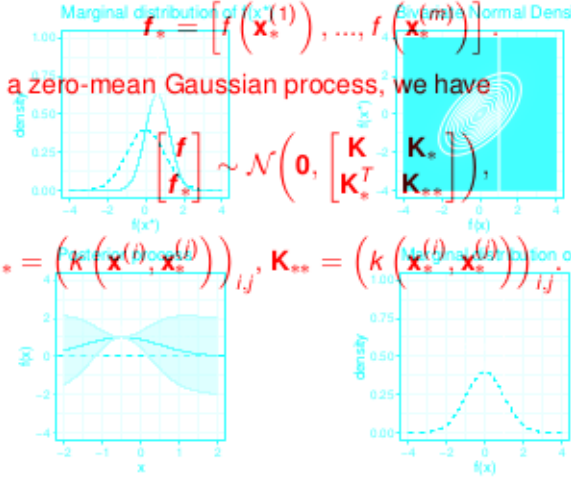
- We can generalize the formula for the posterior process for multiple unobserved test points:

$$\boldsymbol{f}_* = \left[f\left(\mathbf{x}_*^{(1)}\right), ..., f\left(\mathbf{x}_*^{(m)}\right) \right]^T.$$

- Under a zero-mean Gaussian process, we have

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_*^T \\ \mathbf{K}_* & \mathbf{K}_{**} \end{bmatrix}\right),$$

with $\mathbf{K}_* = \left(k\left(\mathbf{x}^{(i)}, \mathbf{x}_*^{(j)}\right)\right)_{i,j}$, $\mathbf{K}_{**} = \left(k\left(\mathbf{x}_*^{(i)}, \mathbf{x}_*^{(j)}\right)\right)_{i,j}$.



POSTERIOR PROCESS / 2

- Similar to the single test point situation, to get the posterior distribution, we exploit the general rule of conditioning for Gaussians:

$$\boldsymbol{f}_* = \left[f\left(\mathbf{x}_*^{(1)}\right), \ldots, f\left(\mathbf{x}_*^{(m)}\right) \right]$$

$$\boldsymbol{f}_* \mid \mathbf{X}_*, \mathbf{X}, \boldsymbol{f} \sim \mathcal{N}(\mathbf{K}_*^T \mathbf{K}^{-1} \boldsymbol{f}, \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_*).$$

- Under a zero-mean Gaussian process, we have
- This formula enables us to talk about correlations among different test points and sample functions from the posterior process.

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{bmatrix} \right),$$

with $\mathbf{K}_* = \left(k\left(\mathbf{x}^{(i)}, \mathbf{x}_*^{(j)}\right) \right)_{i,j}$, $\mathbf{K}_{**} = \left(k\left(\mathbf{x}_*^{(i)}, \mathbf{x}_*^{(j)}\right) \right)_{i,j}$.

POSTERIOR PROCESS

- Similar to the single test point situation, to get the posterior distribution, we exploit the general rule of conditioning for Gaussians:

$$f_* \mid \mathbf{x}_*, \mathbf{X}, \mathbf{f} \sim \mathcal{N}(\mathbf{k}_*' \mathbf{K}^{-1} \mathbf{f}, k_{**} - \mathbf{k}_*' \mathbf{K}^{-1} \mathbf{k}_*).$$

Properties of a Gaussian Process

- This formula enables us to talk about correlations among different test points and sample functions from the posterior process.

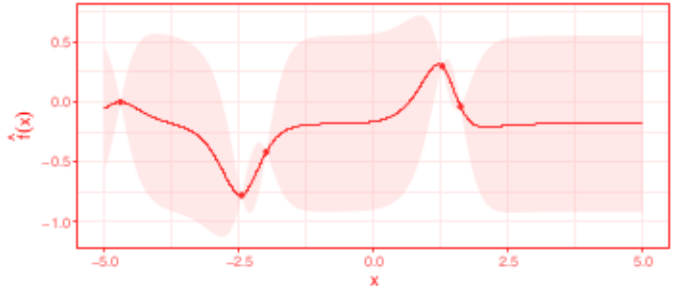
GP AS INTERPOLATOR

The "prediction" for a training point $\mathbf{x}^{(i)}$ is the exact function value $f\left(\mathbf{x}^{(i)}\right)$

$$\boldsymbol{f} \mid \mathbf{X}, \boldsymbol{f} \sim \mathcal{N}(\mathbf{K}\mathbf{K}^{-1}\boldsymbol{f}, \mathbf{K} - \mathbf{K}^{T}\mathbf{K}^{-1}\mathbf{K}) = \mathcal{N}(\boldsymbol{f}, \mathbf{0}).$$

Properties of a Gaussian Process

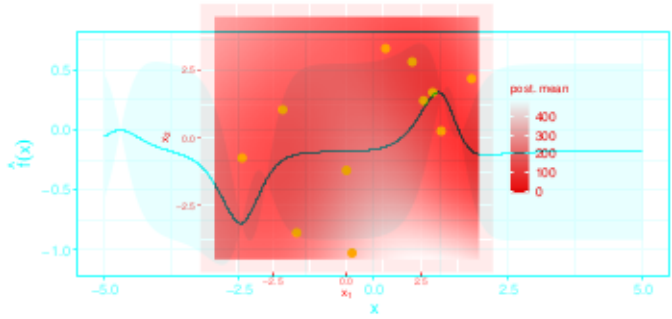
Thus, a Gaussian process is a function **interpolator**.



After observing the training points (red), the posterior process (black) interpolates the training points. (k(x,x') is Matérn with nu = 2.5, the default for DiceKriging::km)

GP AS A SPATIAL MODEL

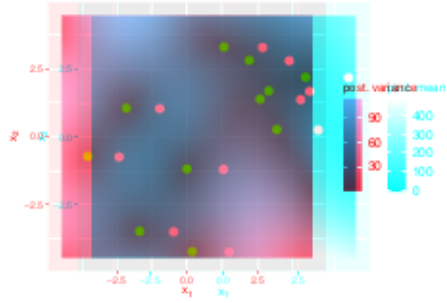
- The correlation among two outputs depends on distance of the corresponding input points $\mathbf{x}$ and $\mathbf{x}'$ (e.g. Gaussian covariance kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-\|\mathbf{x}-\mathbf{x}'\|^2}{2l^2}\right)$).
- Hence, close data points with high spatial similarity $k(\mathbf{x}, \mathbf{x}')$ enter into more strongly correlated predictions: $\boldsymbol{k}_*^\top \boldsymbol{K}^{-1} \boldsymbol{f}$ ($\boldsymbol{k}_* := \left(k(\mathbf{x}, \mathbf{x}^{(1)}), \ldots, k(\mathbf{x}, \mathbf{x}^{(n)})\right)$).



Example: Posterior mean of a GP that was fitted with the Gaussian covariance kernel with $l = 1$.

GP AS A SPATIAL MODEL / 2

- The correlation among two outputs depends on distance of the corresponding input points $\mathbf{x}$ and $\mathbf{x}'$ (e.g. Gaussian covariance kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}' \|^2}{2\ell^2}\right)$).
- Hence, close data points in high spatial similarity $k(\mathbf{x}, \mathbf{x}')$ offer little more strongly correlated predictions $\mathbf{K}_*^\top \mathbf{K}^{-1} \mathbf{f}$ ($\mathbf{k}_* := \left(k(\mathbf{x}, \mathbf{x}^{(1)}), ..., k(\mathbf{x}, \mathbf{x}^{(n)})\right)$).
- Posterior uncertainty increases if the new data points are far from the design points.
- The uncertainty is minimal at the design points, since the posterior variance is zero at these points.



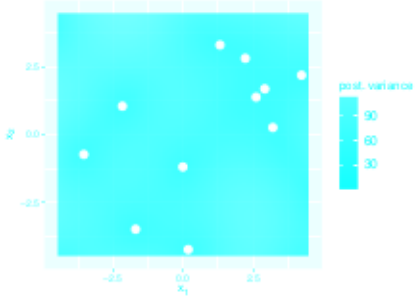
Example: Posterior mean of a GP that was fitted with the Gaussian covariance kernel with $\ell = 1$.

Example (continued): Posterior variance.

GP AS A SPATIAL MODEL

- Posterior uncertainty increases if the new data points are far from the design points.
- The uncertainty is minimal at the design points, since the posterior variance is zero (noise-free data).

Noisy Gaussian Process



Example (continued): Posterior variance.

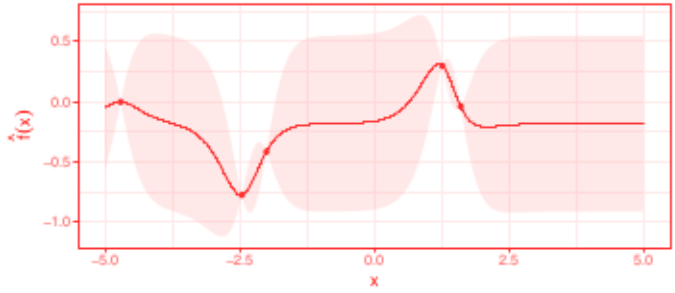
NOISY GAUSSIAN PROCESS

- So far, we implicitly assumed that we had access to the true function value $f(\mathbf{x})$.
- For the squared exponential kernel, for example, we have

Noisy Gaussian Process

$$\text{Cov}\left(f(\mathbf{x}^{(i)}), f(\mathbf{x}^{(i)})\right) = 1.$$

- As a result, the posterior Gaussian process is an interpolator:



After observing the training points (red), the posterior process (black) interpolates the training points. (k(x,x') is Matérn with nu = 2.5, the default for DiceKriging::km)

NOISY GAUSSIAN PROCESS / 2

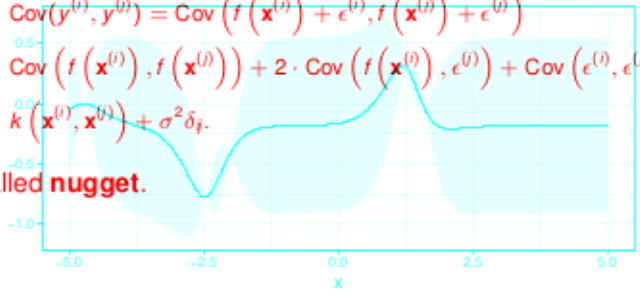
- In reality, however, this is often not the case.
- So far, we implicitly assumed that we had access to the true function value $f(\mathbf{x})$.
- We often only have access to a noisy version of the true function value.
- For the squared exponential kernel, for example, we have

$$y = f(\mathbf{x}) + \epsilon, \epsilon \sim \mathcal{N}\left(0, \sigma^2\right).$$

$$\text{Cov}\left(f\left(\mathbf{x}^{(i)}\right) f\left(\mathbf{x}^{(i)}\right)\right) = 1.$$

- Let us still assume that $f(\mathbf{x})$ is a Gaussian process.
- Then,
- As a result, the posterior Gaussian process is an interpolator:

$$\begin{aligned}
\text{Cov}(y^{(i)}, y^{(j)}) &= \text{Cov}\left(f\left(\mathbf{x}^{(i)}\right) + \epsilon^{(i)}, f\left(\mathbf{x}^{(j)}\right) + \epsilon^{(j)}\right) \\
&= \text{Cov}\left(f\left(\mathbf{x}^{(i)}\right), f\left(\mathbf{x}^{(j)}\right)\right) + 2 \cdot \text{Cov}\left(f\left(\mathbf{x}^{(i)}\right), \epsilon^{(j)}\right) + \text{Cov}\left(\epsilon^{(i)}, \epsilon^{(j)}\right) \\
&= k\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) + \sigma^2 \delta_{ij}.
\end{aligned}$$



After observing the training points (red), the posterior process (black) interpolates the training points. (k(x,x') is Matérn with nu = 2.5, the default for DiceKriging::km)

- σ^2 is called **nugget**.

NOISY GAUSSIAN PROCESS / 3

- Let us now derive the predictive distribution for the case of noisy observations.
- We often only have access to a noisy version of the true function f.
- The prior distribution of y, assuming that f is modeled by a Gaussian process is then $y(\mathbf{x}) = f(\mathbf{x}) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)$.
- Let us still assume that $f(\mathbf{x})$ is a Gaussian process.
- Then,

$$\boldsymbol{y} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{pmatrix} \sim \mathcal{N}\left(\boldsymbol{m}, \boldsymbol{K} + \sigma^2 \boldsymbol{I}_n\right),$$

$$\begin{aligned} \operatorname{Cov}(y^{(i)}, y^{(j)}) &= \operatorname{Cov}\left(f\left(\mathbf{x}^{(i)}\right) + \epsilon^{(i)}, f\left(\mathbf{x}^{(j)}\right) + \epsilon^{(j)}\right) \\ &= \operatorname{Cov}\left(f\left(\mathbf{x}^{(i)}\right), f\left(\mathbf{x}^{(j)}\right)\right) + 2 \cdot \operatorname{Cov}\left(f\left(\mathbf{x}^{(i)}\right), \epsilon^{(j)}\right) + \operatorname{Cov}\left(\epsilon^{(i)}, \epsilon^{(j)}\right) \\ \text{with} &= k\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) + \sigma^2 \delta_{ij}. \end{aligned}$$

- σ^2 is called nugget

$$\boldsymbol{m} := \left(m\left(\mathbf{x}^{(i)}\right)\right)_i, \quad \boldsymbol{K} := \left(k\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right)\right)_{i,j}.$$

NOISY GAUSSIAN PROCESS / 4

- We distinguish again between
  - observed training points $\mathbf{X}, \mathbf{y}$, and
  - unobserved test inputs $\mathbf{X}_*$ with unobserved values $\boldsymbol{f}_*$

and get

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} + \sigma^2 \boldsymbol{I}_n & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{bmatrix}\right).$$

$$\boldsymbol{y} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{pmatrix} \sim \mathcal{N}\left(\boldsymbol{m}, \boldsymbol{K} + \sigma^2 \boldsymbol{I}_n\right),$$

with

$$\mathbf{m} := \left(m\left(\mathbf{x}^{(i)}\right)\right)_i, \quad \mathbf{K} := \left(k\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right)\right)_{i,j}.$$

NOISY GAUSSIAN PROCESS / 5

- Similarly to the noise-free case, we condition according to the rule of conditioning for Gaussians to get the posterior distribution for the test outputs $\boldsymbol{f}_*$ at $\mathbf{X}_*$:
  - observed training inputs $\mathbf{X}$, $\boldsymbol{y}$, and
  - unobserved test inputs $\mathbf{X}_*$ with unobserved values $\boldsymbol{f}_*$

  and get

$$\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{f}_* \end{pmatrix} \mid \mathbf{X}, \mathbf{X}_*, \boldsymbol{y} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{0}, \begin{pmatrix} \mathbf{K} + \sigma^2 \boldsymbol{I}_n & \mathbf{K}_* \\ \mathbf{K}_* & \mathbf{K}_{**} \end{pmatrix} \right) = \mathcal{N}(\boldsymbol{m}_{\text{post}}, \boldsymbol{K}_{\text{post}}).$$

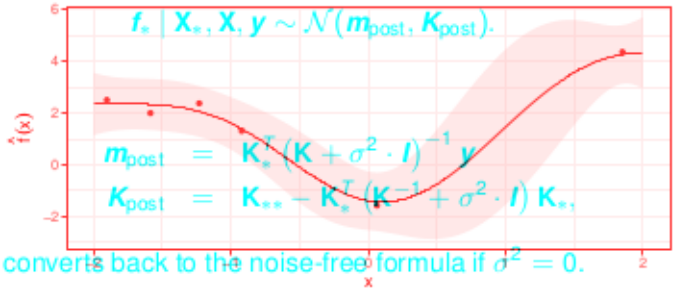
  with

$$\begin{aligned} \boldsymbol{m}_{\text{post}} &= \mathbf{K}_*^T \left(\mathbf{K} + \sigma^2 \cdot \boldsymbol{I} \right)^{-1} \boldsymbol{y} \\ \boldsymbol{K}_{\text{post}} &= \mathbf{K}_{**} - \mathbf{K}_*^T \left(\mathbf{K}^{-1} + \sigma^2 \cdot \boldsymbol{I} \right) \mathbf{K}_*, \end{aligned}$$

- This converts back to the noise-free formula if $\sigma^2 = 0$.

NOISY GAUSSIAN PROCESS / 6

- The noisy Gaussian process is not an interpolator anymore.
- A larger nugget term leads to a wider "band" around the observed training points.
- The nugget term is estimated during training.



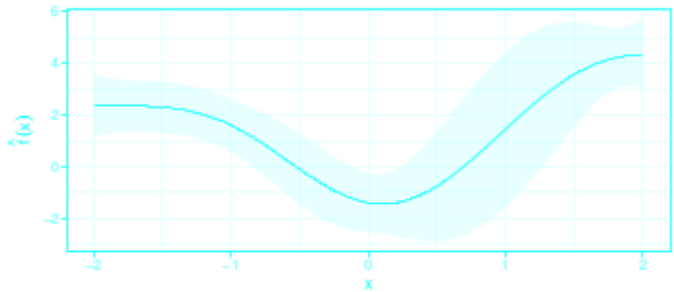
After observing the training points (red), we have a nugget-band around the oberved points. (k(x,x') is the squared exponential)

with

- This converts back to the noise-free formula if $\sigma^2 = 0$.

NOISY GAUSSIAN PROCESS

- The noisy Gaussian process is not an interpolator any more.
- A larger nugget term leads to a wider “band” around the observed training points.
- The nugget term is estimated during training.



After observing the training points (red), we have a nugget-band around the oberved points. (k(x,x') is the squared exponential)

RISK MINIMIZATION FOR GAUSSIAN PROCESSES

In machine learning, we learned about risk minimization. We usually choose a loss function and minimize the empirical risk

$$\mathcal{R}_{\text{emp}}(f) := \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right)$$

as an approximation to the theoretical risk

$$\mathcal{R}(f) := \mathbb{E}_{xy}[L(y, f(\mathbf{x}))] = \int L(y, f(\mathbf{x}))\, \mathrm{d}\mathbb{P}_{xy}.$$

- How does the theory of Gaussian processes fit into this theory?
- What if we want to make a prediction which is optimal w.r.t. a certain loss function?

RISK MINIMIZATION FOR GAUSSIAN PROCESSES

/ 2

- The theory of Gaussian process gives us a posterior distribution

$$p(y \mid \mathcal{D})$$

- If we now want to make a prediction at a test point $\boldsymbol{x}_*$, we approximate the theoretical risk in a different way, by using the posterior distribution:

$$\mathcal{R}(y_* \mid \boldsymbol{x}_*) = \int L(\tilde{y}_*, y_*) p(\tilde{y}_* \mid \boldsymbol{x}_*, \mathcal{D}) d\tilde{y}_*.$$

- The optimal prediction w.r.t the loss function is then:

$$\hat{y}_* \mid \boldsymbol{x}_* = \arg\min_{y_*} \mathcal{R}(y_* \mid \boldsymbol{x}_*).$$

RISK MINIMIZATION FOR GAUSSIAN PROCESSES

- The theory of Gaussian process gives us a posterior distribution

$$p(y \mid \mathcal{D})$$

- If we now want to make a prediction at a test point $\boldsymbol{x}_*$, we approximate the theoretical risk in a different way, by using the posterior distribution:

$$\mathcal{R}(y_* \mid \boldsymbol{x}_*) \approx \int L(\tilde{y}_*, y_*) p(\tilde{y}_* \mid \boldsymbol{x}_*, \mathcal{D}) d\tilde{y}_*.$$

- The optimal prediciton w.r.t the loss function is then:

$$\hat{y}_* | \boldsymbol{x}_* = \arg\min_{y_*} \mathcal{R}(y_* \mid \boldsymbol{x}_*).$$