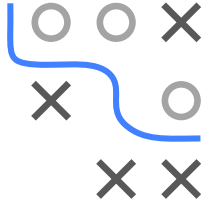
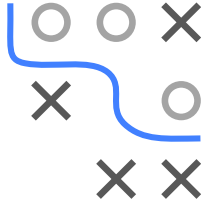
GAUSSIAN POSTERIOR PROCESS AND PREDICTION

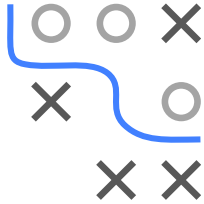


- So far, we have learned how to **sample** from a GP prior.
- However, most of the time, we are not interested in drawing random functions from the prior. Instead, we usually like to use the knowledge provided by the training data to predict values of f at a new test point $\mathbf{x}_*$.
- In what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

Gaussian Posterior Process and Prediction



POSTERIOR PROCESS



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

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

  and one single **unobserved test point** $\mathbf{x}_*$ with $f_* = f\left(\mathbf{x}_*\right)$.
- We now want to infer the distribution of $f_* | \mathbf{x}_*, \boldsymbol{X}, \boldsymbol{f}$.

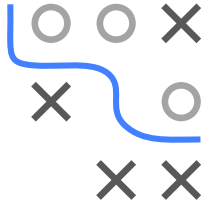
$$f_* = f\left(\mathbf{x}_*\right)$$

- Assuming a zero-mean GP prior $\mathcal{GP}\left(\mathbf{0}, k(\mathbf{x}, \mathbf{x}')\right)$ 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, $\mathbf{K} = \left(k\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right)\right)_{i,j}$, $\boldsymbol{k}_* = \left[k\left(\mathbf{x}_*, \mathbf{x}^{(1)}\right), ..., k\left(\mathbf{x}_*, \mathbf{x}^{(n)}\right)\right]$ and $\boldsymbol{k}_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$.

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 $k(\mathbf{x}, \mathbf{x}') = \exp(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2)$, we compute the cov-matrix:

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

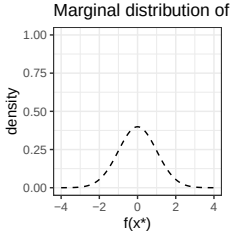
- Assume that we observe the point $f(\mathbf{x}) = 1$.
- We compute the posterior distribution:

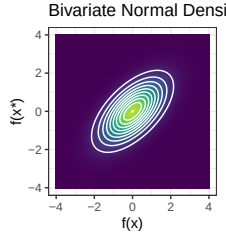
$$\begin{aligned} f_* \mid \mathbf{x}_*, \mathbf{x}, f &\sim \mathcal{N}(\boldsymbol{k}_*^T \mathbf{K}^{-1} f, k_{**} - \boldsymbol{k}_*^T \mathbf{K}^{-1} \boldsymbol{k}_*) \\ &\sim \mathcal{N}(0.61 \cdot 1 \cdot 1, 1 - 0.61 \cdot 1 \cdot 0.61) \\ &\sim \mathcal{N}(0.61, 0.6279) \end{aligned}$$

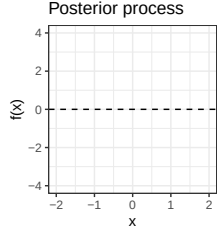
- The MAP-estimate for $\mathbf{x}_*$ is $f(\mathbf{x}_*) = 0.61$, and the uncertainty estimate is 0.6279.

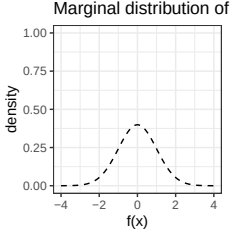
GP PREDICTION: TWO POINTS

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



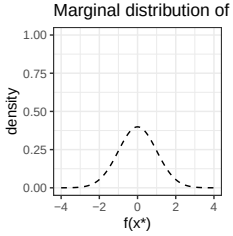


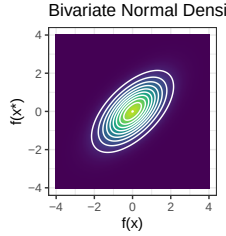


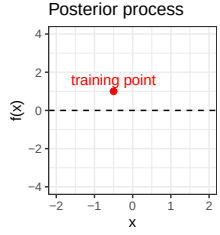


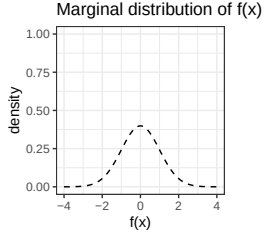
GP PREDICTION: TWO POINTS

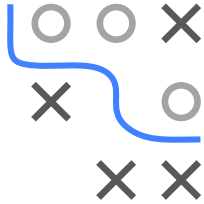
Assume we observed $f(\mathbf{x}) = 1$ for the training point $\mathbf{x} = -0.5$.





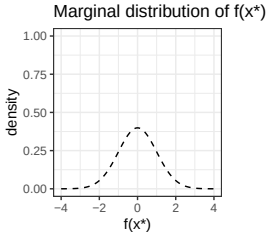


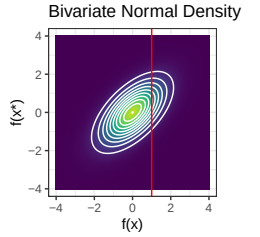


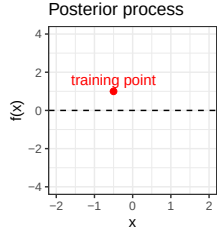


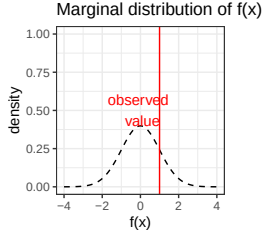
GP PREDICTION: TWO POINTS

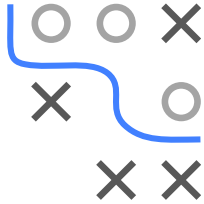
We condition the Gaussian on $f(\mathbf{x}) = 1$.





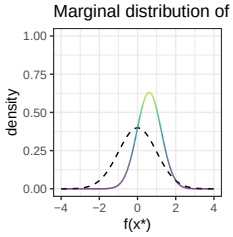


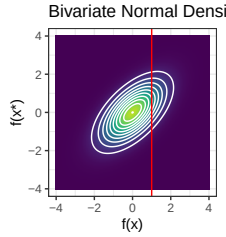


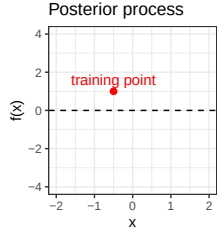


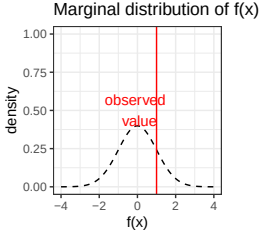
GP PREDICTION: TWO POINTS

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



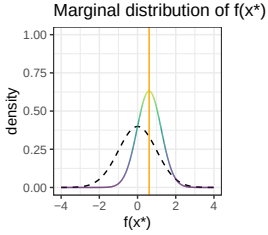


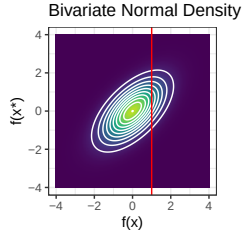


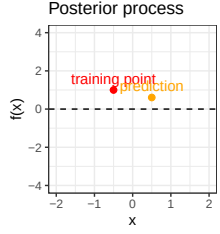


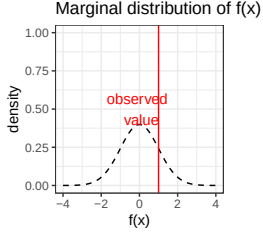
GP PREDICTION: TWO POINTS

A possible predictor for f at $\mathbf{x}_*$ is the MAP of the posterior distribution.



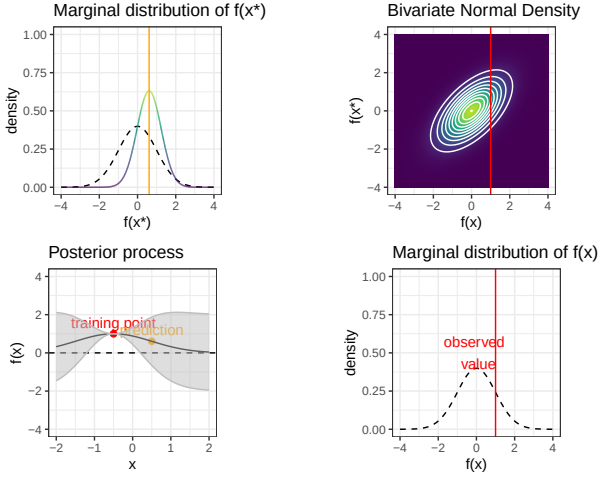




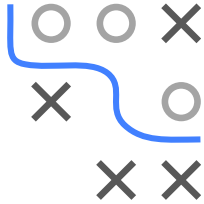


GP PREDICTION: TWO POINTS

We can do this for different values $\mathbf{x}_*$, 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:

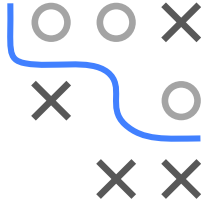
$$\boldsymbol{f}_* = \left[f\left(\mathbf{x}_*^{(1)}\right), ..., f\left(\mathbf{x}_*^{(m)}\right) \right].$$

- Under a zero-mean Gaussian process, we have

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

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

Properties of a Gaussian Process

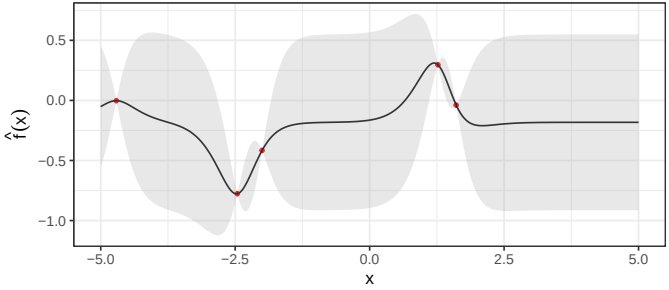


GP AS INTERPOLATOR

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

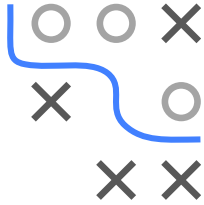
$$\boldsymbol{f} \mid \mathbf{X}, \boldsymbol{f} \sim \mathcal{N}(\mathbf{K}\mathbf{K}^{-1}\boldsymbol{f}, \mathbf{K} - \mathbf{K}^{T}\mathbf{K}^{-1}\mathbf{K}) = \mathcal{N}(\boldsymbol{f}, \mathbf{0}).$$

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

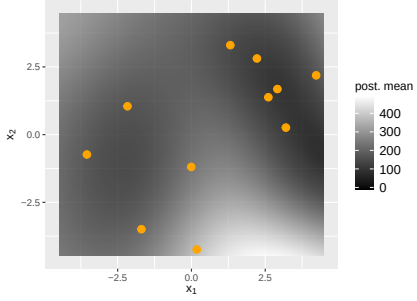


After observing the training points (red), the posterior process (black) interpolates the training points.
(k(x,x') is Matèrn with nu = 2.5, the default for DiceKriging::km)

GP AS A SPATIAL MODEL

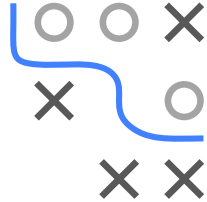


- The correlation among two outputs depends on distance of the corresponding input points $\mathbf{x}$ and $\mathbf{x}'$ (e.g. Gaussian covariance kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-\|\mathbf{x}-\mathbf{x}'\|^2}{2l^2}\right)$)
- Hence, close data points with high spatial similarity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

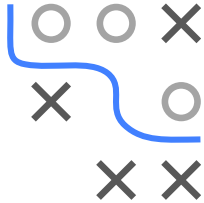


Example: Posterior mean of a GP that was fitted with the Gaussian covariance kernel with $l = 1$.

Noisy Gaussian Process



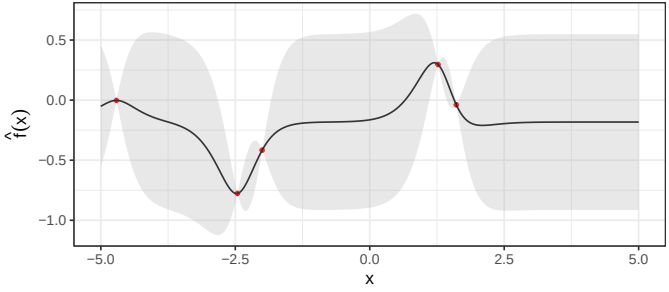
NOISY GAUSSIAN PROCESS



- 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

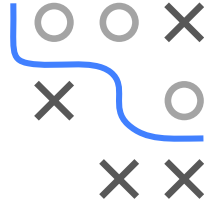
$$\text{Cov}\left(f(\mathbf{x}^{(i)}), f(\mathbf{x}^{(i)})\right) = 1.$$

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

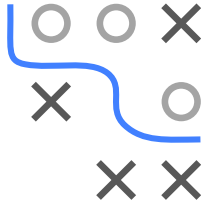


After observing the training points (red), the posterior process (black) interpolates the training points.
(k(x,x') is Matèrn with nu = 2.5, the default for DiceKriging::km)

Decision Theory for Gaussian Processes



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?