AutoML: Gaussian Processes

Gaussian Process Prediction

Bernd Bischl Frank Hutter Lars Kotthoff
Marius Lindauer Joaquin Vanschoren

Motivation

- So far, we have learned how to **sample** from a Gaussian process prior.
- However, most of the time, we are not interested in drawing random functions from the prior. Instead, we usually like to implement 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 Gaussian process prior ($\rightarrow$ posterior process) and how to make predictions.

Gaussian Posterior Process and Prediction

Posterior Process I

- Let us distinguish between **observed training** inputs (also denoted by a design matrix $\mathbf{X}$), their corresponding values

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

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

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

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

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

Posterior Process II

$(*)$ *A General Rule of Conditioning for Gaussian Random Variables*

Let $\boldsymbol{z}_1 \in \mathbb{R}^{m_1}$, $\boldsymbol{z}_2 \in \mathbb{R}^{m_2}$, and $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2)$. If the m-dimensional Gaussian vector $\boldsymbol{z}$ can be partitioned as

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

then the conditional distribution $\boldsymbol{a} = \boldsymbol{z}_2 \mid \boldsymbol{z}_1$ will be a multivariate normal distribution:

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

Posterior Process III

- Given that $\boldsymbol{f}$ is observed, we can exploit the general rule $(*)$ to obtain the following formula:

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

- As the posterior is Gaussian, the maximum a-posteriori estimate (i.e., the mode of the posterior distribution) is:

$$\boldsymbol{k}_*^T \boldsymbol{K}^{-1} \boldsymbol{f}$$

.

GP Prediction: Two Points I

To visualize the above idea, assume that we have observed a single training point $\mathbf{x} = -0.5$. Based on this point, we intend to make a prediction at the test point $\mathbf{x}_* = 0.5$.

- Under a zero-mean $\mathcal{G}$ with $k(\mathbf{x}, \mathbf{x}') = \exp(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2)$, we assume (the covariance matrix has been computed):

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

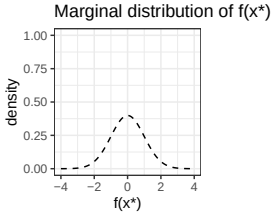
- Let us assume that we observe the point $f(\mathbf{x}) = 1$. We can compute the posterior distribution:

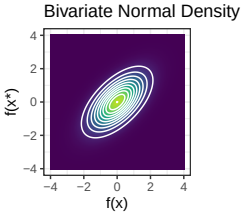
$$\begin{aligned} f_* \mid \mathbf{x}_*, \mathbf{x}, f &\sim \mathcal{N}(\boldsymbol{k}_*^T \boldsymbol{K}^{-1} f, k_{**} - \boldsymbol{k}_*^T \boldsymbol{K}^{-1} \boldsymbol{k}_*) \\ &\sim \mathcal{N}(0.61 \cdot 1 \cdot 1, 1 - 0.61 \cdot 1 \cdot 0.61) \\ &\sim \mathcal{N}(0.61, 0.6279) \end{aligned}$$

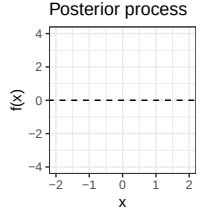
- The MAP-estimate for $\mathbf{x}_*$ is $f(\mathbf{x}_*) = 0.61$, and the uncertainty estimate is 0.6279.

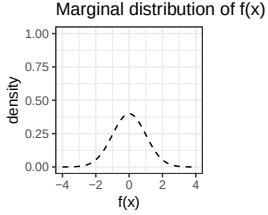
GP Prediction: Two Points II

The figures show the bivariate normal density as well as the corresponding marginals.



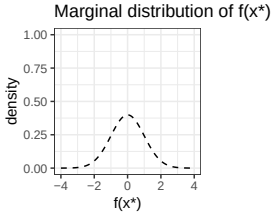


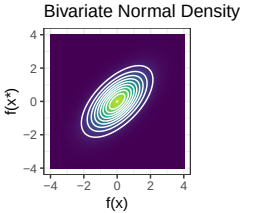


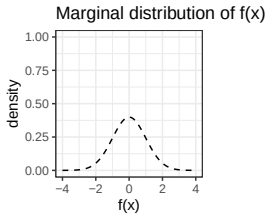


GP Prediction: Two Points III

Now, assume that we observe the value of $f(\mathbf{x}) = 1$ corresponding to the training point $\mathbf{x} = -0.5$.

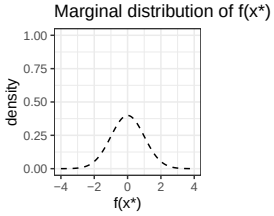


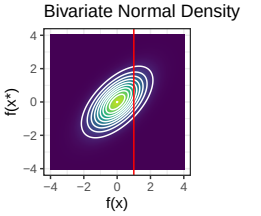


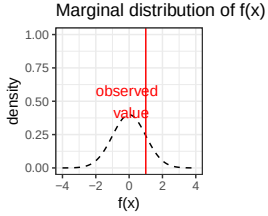


GP Prediction: Two Points IV

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

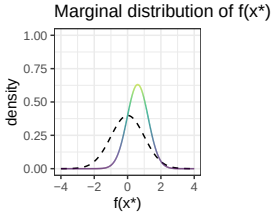


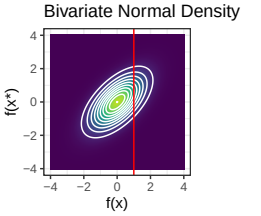


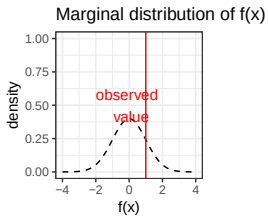


GP Prediction: Two Points V

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

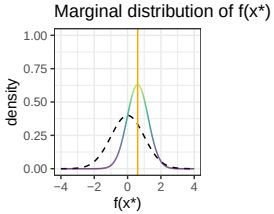


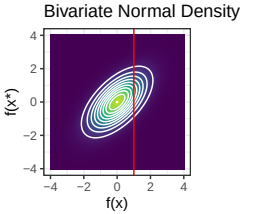


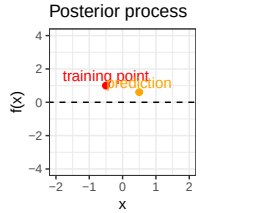


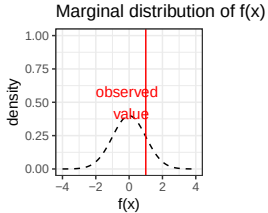
GP Prediction: Two Points VI

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



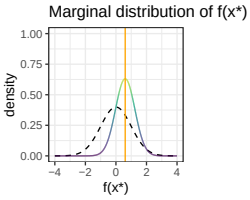


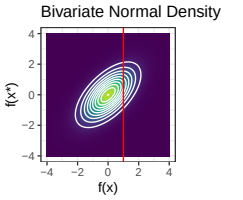


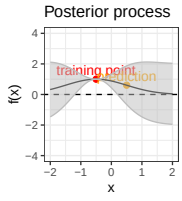


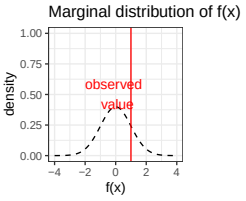
GP Prediction: Two Points VII

We can repeat this process for different $\mathbf{x}_*$ and find the respective mean (grey line) and standard deviation (grey area). Note that the grey area is mean $\pm 2\times$ standard deviation.









Posterior Process I

- The previous discussion was restricted to a single test point. However, one can generalize it to posterior processes with multiple unobserved test points:

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

- Under a zero-mean Gaussian process, we have:

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

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

Posterior Process II

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

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

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

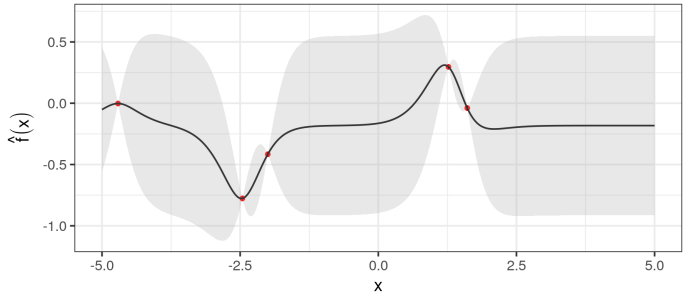
Properties of a Gaussian Process

GP as an Interpolator

- The "prediction" for a training point $\mathbf{x}^{(i)}$ is the exact function value $f(\mathbf{x}^{(i)})$. That is,

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

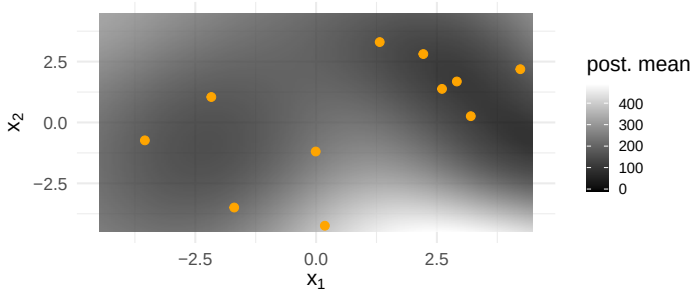
- Thus, a Gaussian process is a function **interpolator**.



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 I

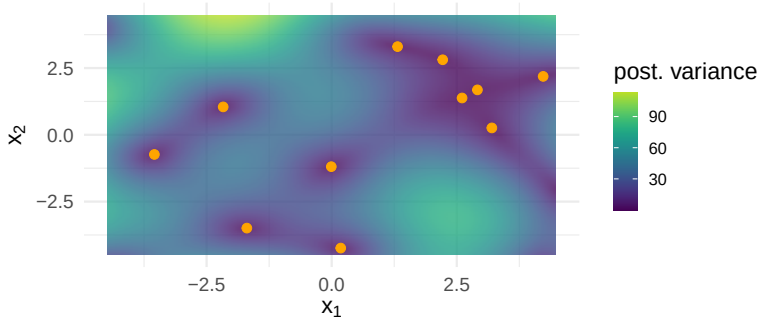
- The correlation among two outputs depends on the distance of the coresponding input points $\mathbf{x}$ and $\mathbf{x}'$. For instance, the Gaussian covariance kernel is $k(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-\|\mathbf{x}-\mathbf{x}'\|^2}{2\ell^2}\right)$.
- Hence, close data points with high 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: the posterior mean of a GP that is fitted with the Gaussian covariance kernel with $\ell = 1$.

GP as a Spatial Model II

- 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.

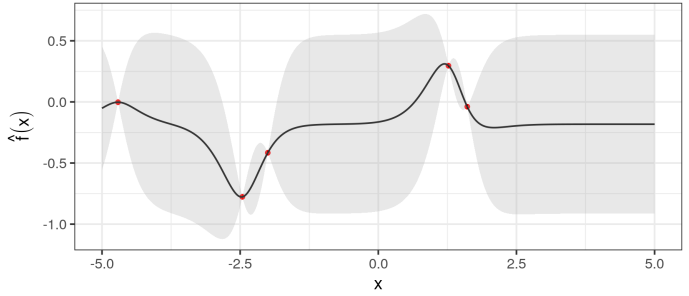


Example (continued): posterior variance

Noisy Gaussian Process

Noisy Gaussian Process I

- So far, we have implicitely assumed that we access the true function values $f(\mathbf{x})$.
- For the squared exponential kernel, for example, we had $cov\left(f(\mathbf{x}^{(i)}), f(\mathbf{x}^{(j)})\right) = 1$.
- Consequently, the posterior Gaussian process was 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)

Noisy Gaussian Process II

- However, in reality that is not often the case. Rather, we often only have access to a noisy version of the true function values:

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

- Let us assume that $f(\mathbf{x})$ is still a Gaussian process. Then, we would have the following:

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

- σ^2 is called **nugget**.

Noisy Gaussian Process III

- We can now derive the predictive distribution for the case of noisy observations.
- Assuming that f is modeled by a Gaussian process, the prior distribution of y is

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

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 IV

We distinguish again between:

- Observed training points and their corresponding values, i.e, $\mathbf{X}$ and y.
- Unobserved test points and their corresponding values, i.e, $\mathbf{X}_*$ and $\boldsymbol{f}_*$.

and get:

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

Noisy Gaussian Process V

- Similar 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 $\mathbf{X}_*$:

$$\boldsymbol{f}_* \mid \mathbf{X}_*, \mathbf{X}, \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{m}_{\text{post}}, \boldsymbol{K}_{\text{post}}),$$

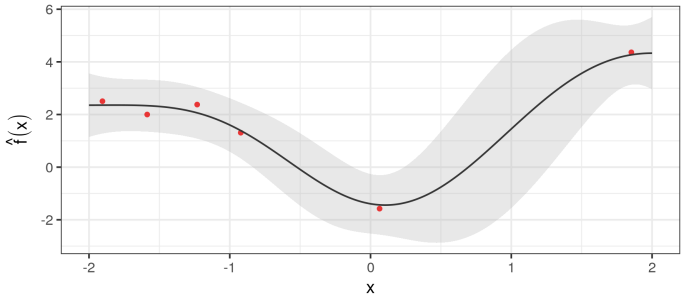
with

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

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

Noisy Gaussian Process VI

- The noisy Gaussian process is not an interpolator any more.
- A larger nugget term leads to a wider "band" around the observed training points.
- In general, the effect of the nugget term is estimated during training (see the next section).



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

Decision Theory for Gaussian Processes

Risk Minimization for Gaussian Processes I

In machine learning, we usually choose a loss function and try to minimize the empirical risk:

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

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 scenario?
- What if we were looking for predictions that atr optimal w.r.t. a certain loss function?

Risk Minimization for Gaussian Processes II

- The theory of Gaussian process provides us with a posterior distribution: $p(y \mid \mathcal{D})$.
- To make a prediction at a test point $\boldsymbol{x}_*$, we can approximate the theoretical risk by explointing 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:

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