

# AutoML: Gaussian Processes

## Gaussian Processes: Additional Material

Bernd Bischl   Frank Hutter   Lars Kotthoff  
Marius Lindauer   Joaquin Vanschoren

# Notation

In this part,

- $(\mathbf{x}_*, y_*)$  denotes a single test observation, excluded from the training data.
- $\mathbf{X}_* \in \mathbb{R}^{n_* \times p}$  denotes a set of  $n_*$  test observations.
- $\mathbf{y}_* \in \mathbb{R}^{n_* \times p}$  denotes the corresponding outcomes, excluded from the training data.

# Noisy Gaussian Processes

# Noisy Gaussian Processes

- In the previous slides, we implicitly assumed that we access the true function values  $f(\mathbf{x})$ . However, in many practical cases, we only have a noisy version of the values:

$$y = f(\mathbf{x}) + \epsilon.$$

- By assuming an additive i.i.d. Gaussian noise, the covariance function becomes:

$$\text{cov}(y^{(i)}, y^{(j)}) = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) + \sigma_n^2 \delta_{ij}, \text{ where } \delta_{ij} = 1 \text{ if } i = j.$$

- In the matrix notation, this becomes:

$$\text{cov}(\mathbf{y}) = \mathbf{K} + \sigma_n^2 \mathbf{I} =: \mathbf{K}_y, \text{ where } \sigma_n^2 \text{ is called } \mathbf{nugget}.$$

# GP vs. Kernelized Ridge Regression

- The predictive function is then

$$\mathbf{f}_* | \mathbf{X}_*, \mathbf{X}, \mathbf{y} \sim \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\bar{\mathbf{f}}_*)),$$

with  $\bar{\mathbf{f}}_* = \mathbf{K}_*^T \mathbf{K}_y^{-1} \mathbf{y}$  and  $\text{cov}(\bar{\mathbf{f}}_*) = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}_y^{-1} \mathbf{K}_*$ .

- The predicted mean value at the training points  $\bar{\mathbf{f}} = \mathbf{K} \mathbf{K}_y^{-1} \mathbf{y}$  is a **linear combination** of the  $\mathbf{y}$  values.

💡 Predicting the posterior mean corresponds exactly to the predictions obtained by kernelized Ridge regression. However, a GP as a Bayesian model provides us with much more information (i.e., a posterior distribution), whilst the kernelized Ridge regression does not.

# **Bayesian Linear Regression as a GP**

# Bayesian Linear Regression as a GP

- One example for a Gaussian process is the Bayesian linear regression model, and we already discuss it.
- For  $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \tau^2 \mathbf{I})$ , the joint distribution of any set of function values is Gaussian:

$$f(\mathbf{x}^{(i)}) = \boldsymbol{\theta}^\top \mathbf{x}^{(i)} + \epsilon.$$

- The corresponding mean function is  $m(\mathbf{x}) = \mathbf{0}$ , and the covariance function is

$$\begin{aligned} \text{cov}(f(\mathbf{x}), f(\mathbf{x}')) &= \mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] - \underbrace{\mathbb{E}[f(\mathbf{x})]\mathbb{E}[f(\mathbf{x}')]_{=0}} \\ &= \mathbb{E}[(\boldsymbol{\theta}^\top \mathbf{x} + \epsilon)^\top (\boldsymbol{\theta}^\top \mathbf{x}' + \epsilon)] \\ &= \tau^2 \mathbf{x}^\top \mathbf{x}' + \sigma^2 =: k(\mathbf{x}, \mathbf{x}'). \end{aligned}$$

# Feature Spaces and the Kernel Trick I

- If one relaxes the linearity assumption by projecting the features into a higher dimensional feature space  $\mathcal{Z}$  using a basis function  $\phi : \mathcal{X} \rightarrow \mathcal{Z}$ , the corresponding covariance function becomes:

$$k(\mathbf{x}, \mathbf{x}') = \tau^2 \phi(\mathbf{x})^\top \phi(\mathbf{x}') + \sigma^2.$$

- To get arbitrarily complicated functions, we would have to handle high-dimensional feature vectors  $\phi(\mathbf{x})$ .
- Fortunately, all we need to know is the inner product  $\phi(\mathbf{x})^\top \phi(\mathbf{x}')$ . That is, the feature vector itself never occurs in calculations.



# Feature Spaces and the Kernel Trick II

💡 If we can get the inner product directly and without calculating the infinite feature vectors, we can infer an infinitely complicated model with a finite amount of computation. This idea is known as **kernel trick**.

- A Gaussian process can then be defined by either:
  - ▶ deriving the covariance function from the inner products of the basis functions evaluations, or
  - ▶ choosing a positive definite kernel function (Mercer Kernel), which- according to Mercer's theorem - corresponds to taking the inner products in some (possibly infinite) feature space.

## Summary: Gaussian Process Regression

- The Gaussian process regression is equivalent to the **kernelized** Bayesian linear regression.
- The covariance function describes the shape of the Gaussian process. Hence, with the right choice of covariance function, remarkably flexible models can be built.
- Naive implementations of Gaussian process models scale poorly with large datasets, as
  - ▶ the kernel matrix has to be inverted / factorized, which is  $\mathcal{O}(n^3)$ ,
  - ▶ computing the kernel matrix uses  $\mathcal{O}(n^2)$  memory - running out of memory places a hard limit on the size of problems
  - ▶ generating predictions is  $\mathcal{O}(n)$  for the mean, but  $\mathcal{O}(n^2)$  for the variance.(...special tricks are needed.)