732A96/TDDE15 Advanced Machine Learning

Gaussian Process Regression and Classification

Jose M. Peña
IDA, Linköping University, Sweden

Lectures 10: Kernels, Hyperparameter Learning and More

Contents

Literature

- Main source
  - Rasmussen, C. E. and Williams, K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006. Chapters 2.3, 5.1-5.4.1.
- Additional source
  - Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006. Chapters 6.4.3-6.4.4.

Three Common Covariance Functions

- Let $r = ||\boldsymbol{x} - \boldsymbol{x}'||$.
- Squared exponential (SE):

$$k_{SE}(r) = \sigma_f^2 \exp\left\{ -\frac{r^2}{2\ell^2} \right\}$$

  where $\sigma_f^2 > 0, \ell > 0$. Very smooth.
- Rational quadratic (RQ):

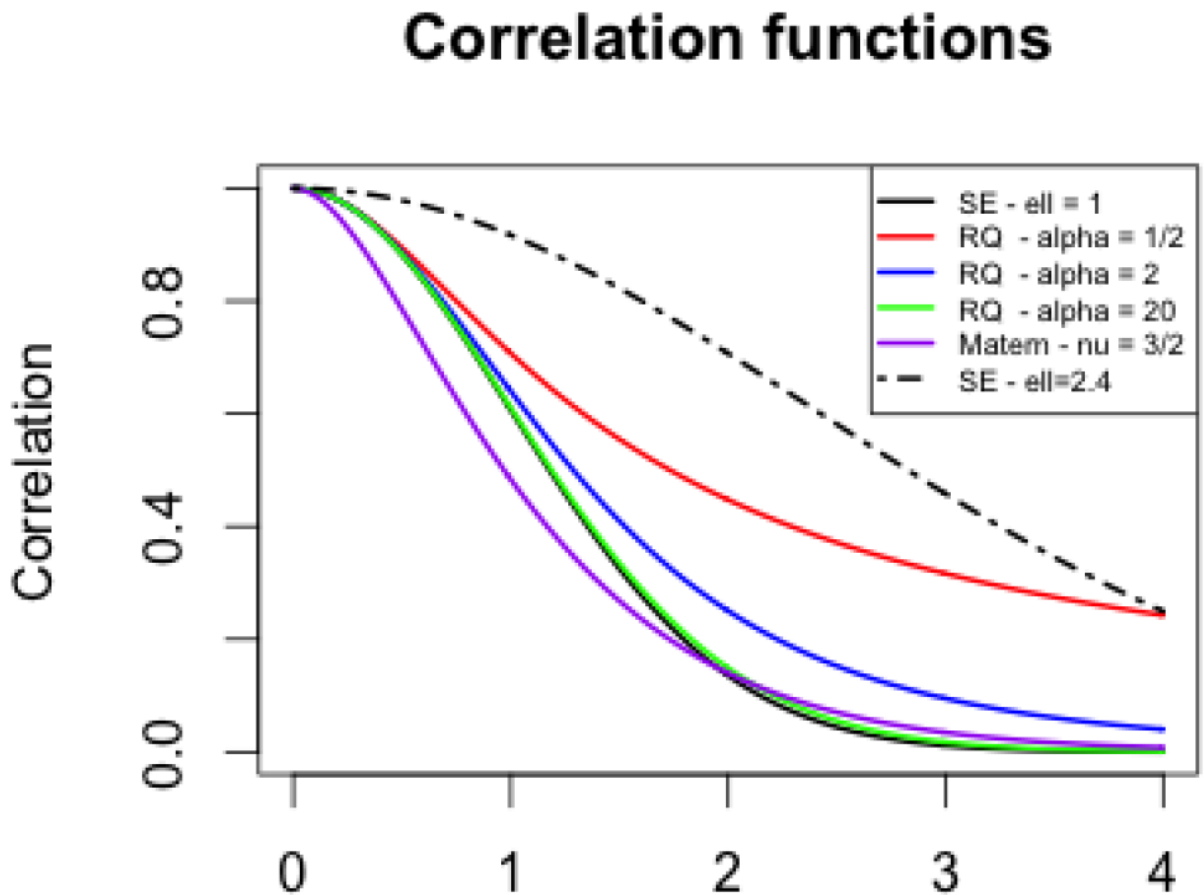
$$k_{RQ}(r) = \sigma_f^2 \left(1 + \frac{r^2}{2\alpha\ell^2} \right)^{-\alpha}$$

  $\sigma_f^2 > 0, \ell > 0, \alpha > 0$. k_{RQ} is an infinite sum of k_{SE} with different ℓ. As $\alpha \to \infty$, $k_{RQ}(r) \to k_{SE}(r)$.
- Matérn:

$$k_{Matern} = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{\ell} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu} r}{\ell} \right)$$

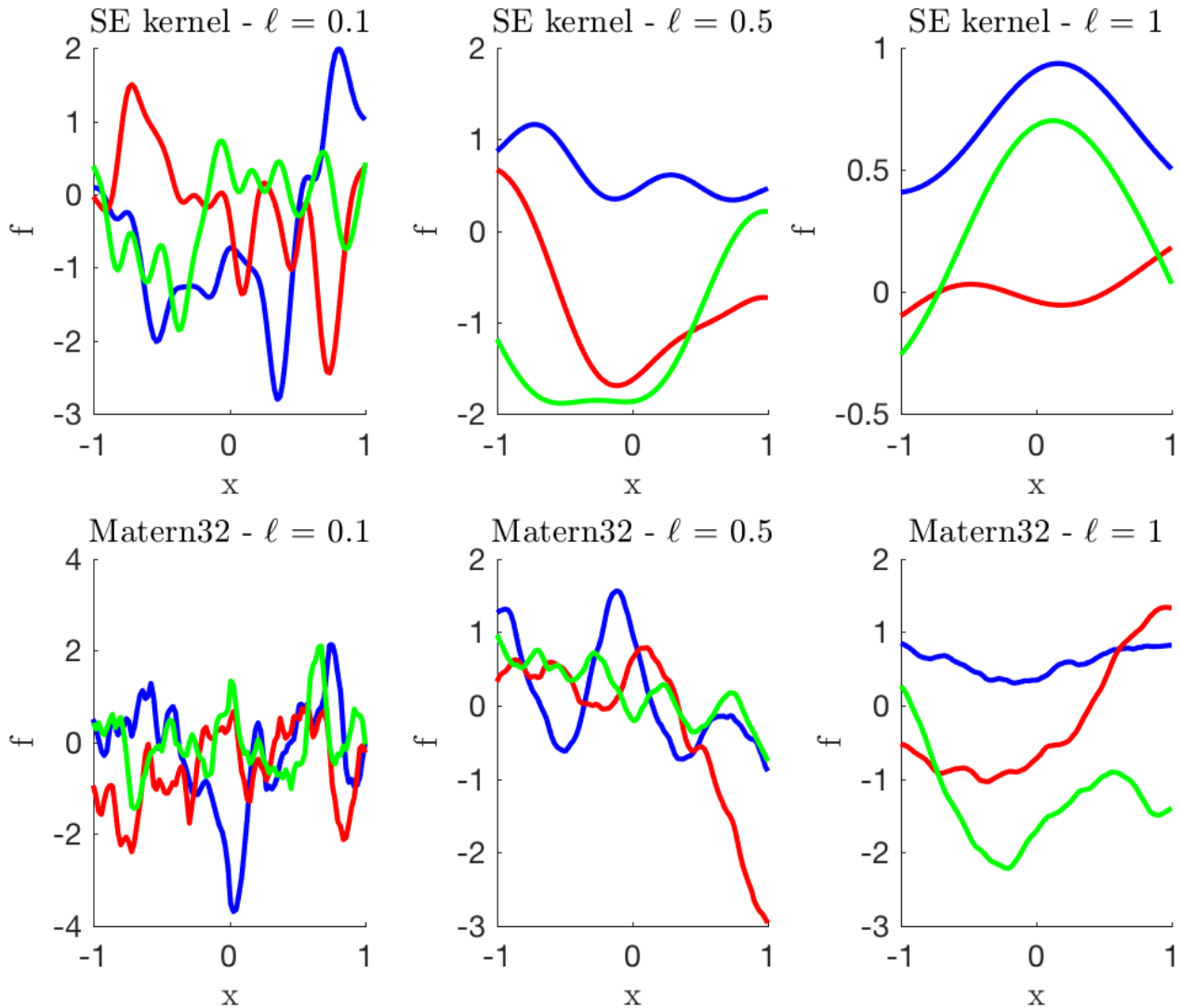
  where $\sigma_f^2 > 0, \ell > 0, \nu > 0$, and K_ν is the modified Bessel function. As $\nu \to \infty$, $k_{Matern}(r) \to k_{SE}(r)$.
- Demo of `GaussianProcesses.R` and `KernLabDemo.R`.

Three Common Covariance Functions



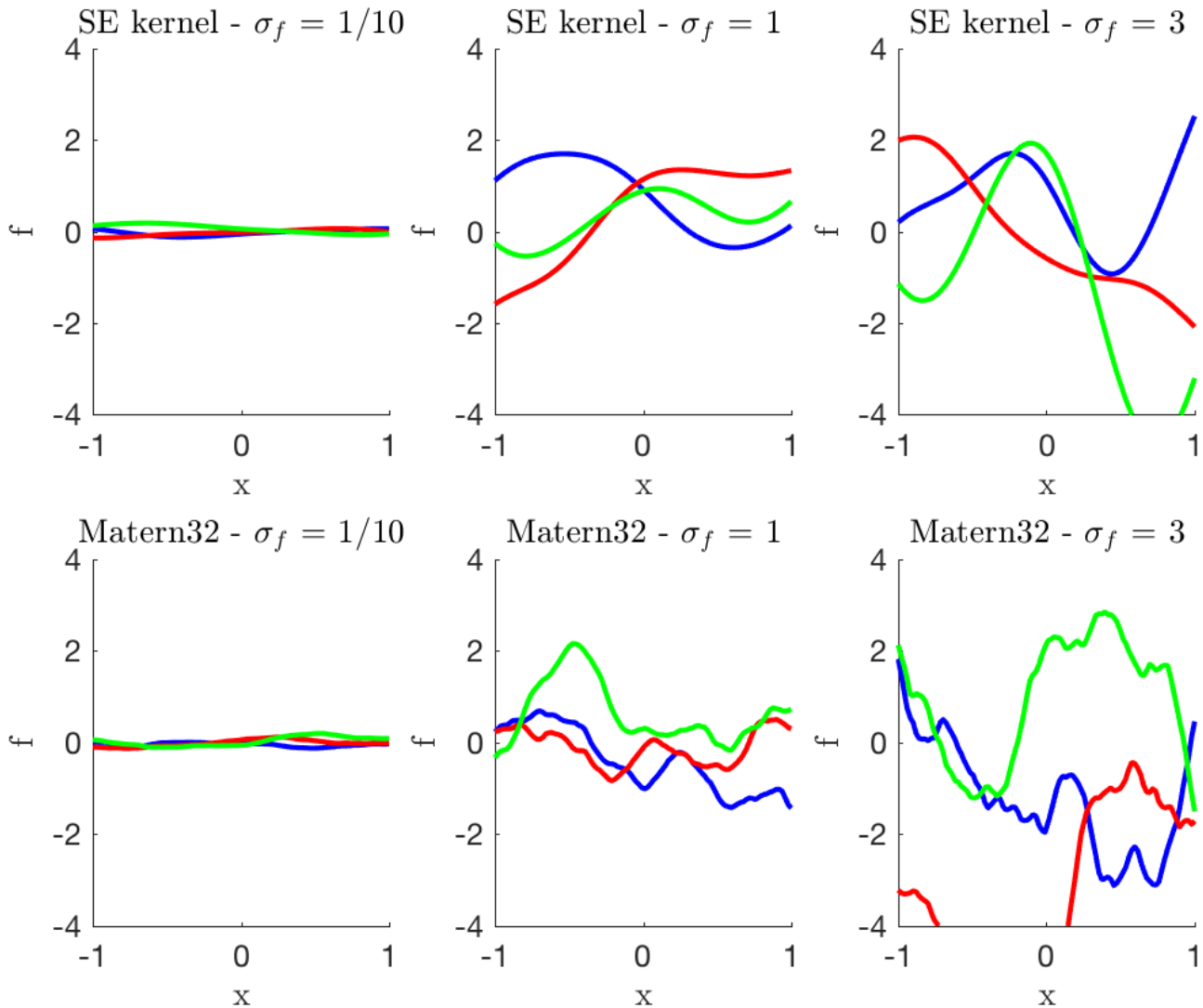
Three Common Covariance Functions

- The length scale ℓ determines the smoothness.



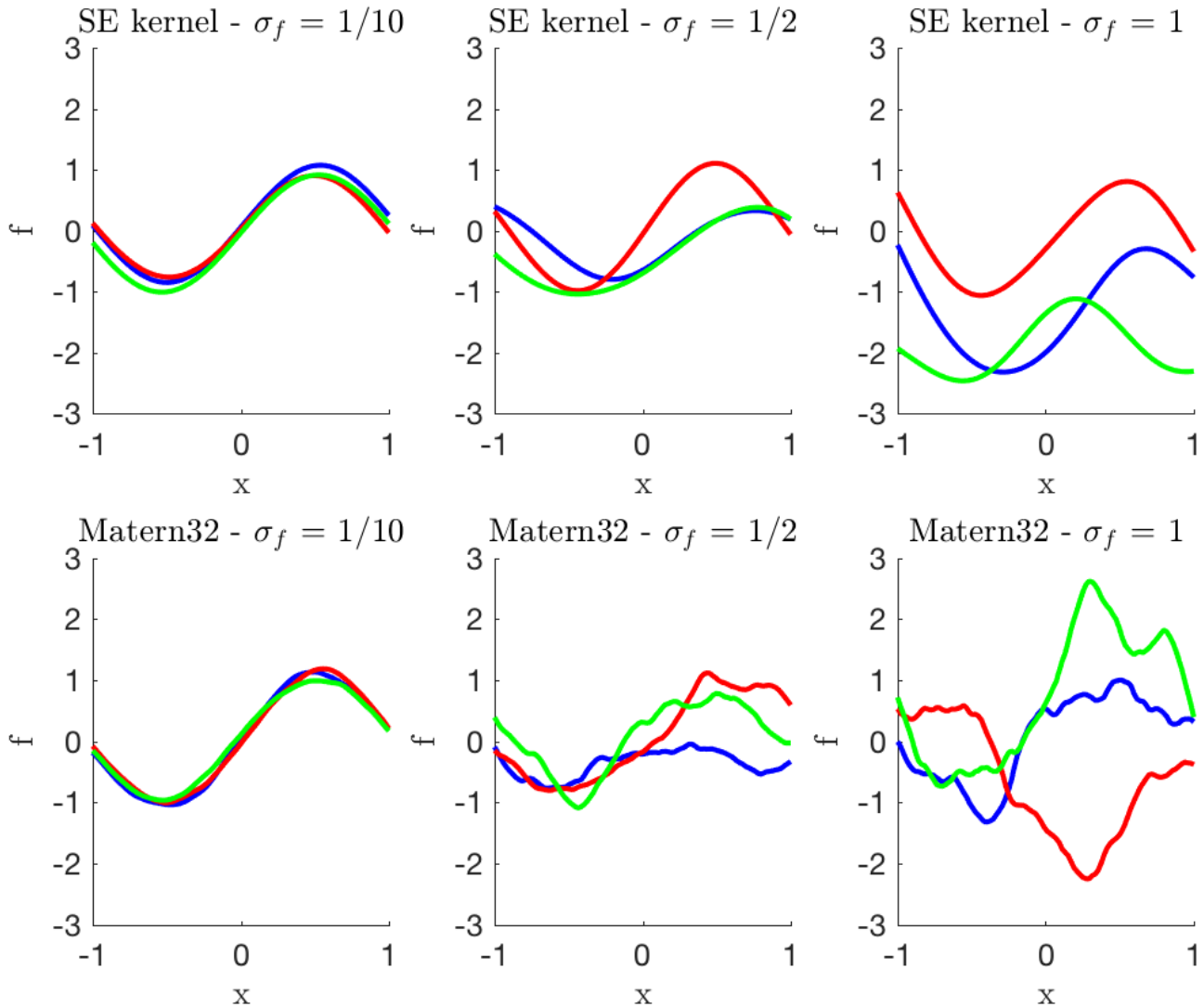
Three Common Covariance Functions

- The scale factor σ_f determines the variance.



Three Common Covariance Functions

- The mean can be arbitrary, e.g. $\sin(3x)$.



Learning the Hyperparameters of the Covariance Function

- Let θ denote the hyperparameters of the covariance function, i.e. $\theta = (\sigma_f, \ell)$ for k_{SE}, $\theta = (\sigma_f, \ell, \alpha)$ for k_{RQ}, and $\theta = (\sigma_f, \ell, \nu)$ for k_{Matern}.
- Choose the hyperparameters that maximize the marginal likelihood:

$$\log p(\boldsymbol{y}|X,\theta) = -\frac{1}{2}\boldsymbol{y}^T(K(X,X)+\sigma_n^2 I)^{-1}\boldsymbol{y} - \frac{1}{2}\log|K(X,X)+\sigma_n^2 I| - \frac{n}{2}\log 2\pi$$

  which follows from

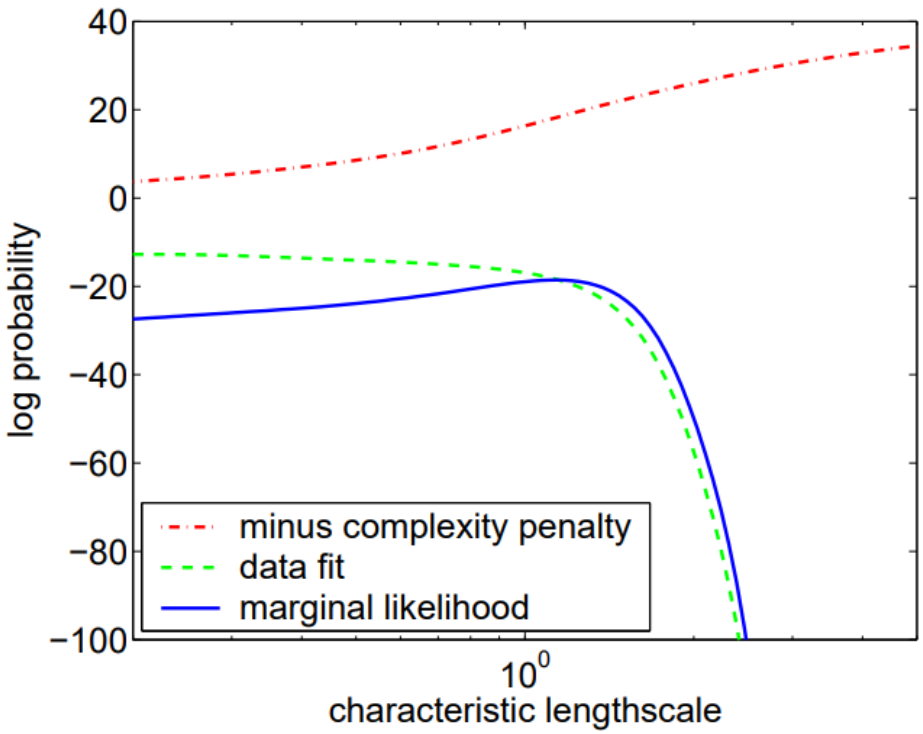
$$\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{f}_*\end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix}K(X,X)+\sigma_n^2 I & K(X,X_*)\\ K(X_*,X) & K(X_*,X_*)\end{bmatrix}\right).$$

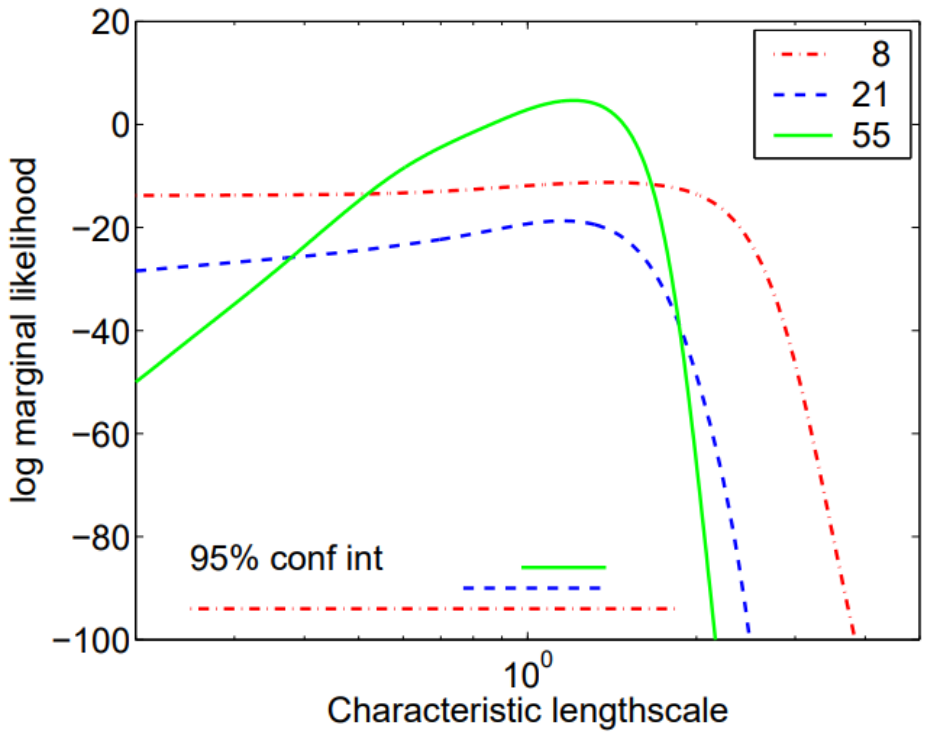
- In general, this is a non-convex optimization problem, and gradient methods are typically used. For most common covariance functions, the derivative of $K(X,X)$ wrt θ is easy to compute.
- For a Bayesian approach, choose the hyperparameters that maximize the posterior distribution $p(\theta|\boldsymbol{y},X) \propto p(\boldsymbol{y}|X,\theta)p(\theta)$. It typically requires MCMC sampling or Laplace approximation.
- The methods above can also be used to select among covariance functions, i.e. simply include them as hyperparameters. Cross-validation is also an option.

Learning the Hyperparameters of the Covariance Function

$$\log p(\boldsymbol{y}|X,\theta) = -\frac{1}{2}\boldsymbol{y}^T(K(X,X)+\sigma_n^2 I)^{-1}\boldsymbol{y} - \frac{1}{2}\log|K(X,X)+\sigma_n^2 I| - \frac{n}{2}\log 2\pi$$

$$= \text{data fit - model complexity - normalization constant.}$$





Lab: Algorithm 2.1 in Rasmussen and Williams

input: X (inputs), $\mathbf{y}$ (targets), k (covariance function), σ_n^2 (noise level), $\mathbf{x}_*$ (test input)

2: $L := \text{cholesky}(K + \sigma_n^2 I)$

$\boldsymbol{\alpha} := L^\top \backslash (L \backslash \mathbf{y})$

4: $\bar{f}_* := \mathbf{k}_*^\top \boldsymbol{\alpha}$ } predictive mean eq. (2.25)

$\mathbf{v} := L \backslash \mathbf{k}_*$

6: $\mathbb{V}[f_*] := k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^\top \mathbf{v}$ } predictive variance eq. (2.26)

$\log p(\mathbf{y}|X) := -\frac{1}{2}\mathbf{y}^\top \boldsymbol{\alpha} - \sum_i \log L_{ii} - \frac{n}{2}\log 2\pi$ eq. (2.30)

8: **return**: $\bar{f}_*$ (mean), $\mathbb{V}[f_*]$ (variance), $\log p(\mathbf{y}|X)$ (log marginal likelihood)

- $K = K(X, X)$.
- $\boldsymbol{k}_* = K(X, X_*)$.
- $L = cholesky(A) \Rightarrow A = LL^T \Rightarrow A^{-1} = (L^T)^{-1}L^{-1} = (L^{-1})^T L^{-1}$ and $|A| = det(A) = det(L)det(L^T) = (\prod_i L_{ii})^2$.
- $L \backslash \boldsymbol{y} = solve(L, \boldsymbol{y}) = L^{-1}\boldsymbol{y}$.
- The algorithm uses Cholesky decomposition instead of matrix inversion because it is faster and numerically more stable.
- It returns the predictive distribution for noise-free test data, i.e. $\boldsymbol{f}_*$. Add σ_n^2 to the predictive variances to obtain the distribution for noisy test data, i.e. $\boldsymbol{y}_*$
- It is presented for a single test case but it also works for several test cases.

Contents

- Three Common Covariance Functions
- Learning the Hyperparameters of the Covariance Function
- Lab: Algorithm 2.1 in Rasmussen and Williams

Thank you