732A96/TDDE15 Advanced Machine Learning

Gaussian Process Regression and Classification

Jose M. Peña
IDA, Linköping University, Sweden

Lectures 11: 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):

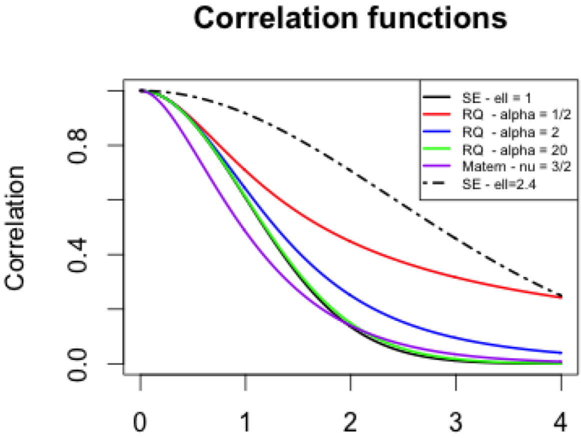
$$k_{RQ}(r) = \sigma_f^2 \left(1 + \frac{r^2}{2\alpha\ell^2} \right)^{-\alpha}$$

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

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

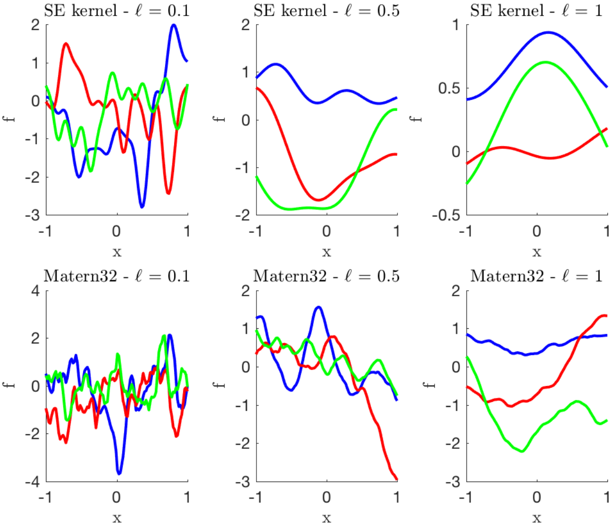
  where $\sigma_f^2 > 0, \ell > 0, \nu > 0$, and K_ν is the modified Bessel function. As $\nu \to \infty$, $k_{Matern}(r) \to k_{SE}(r)$.
- Demo of `GaussianProcesses.R` and `KernLabDemo.R`.

Three Common Covariance Functions



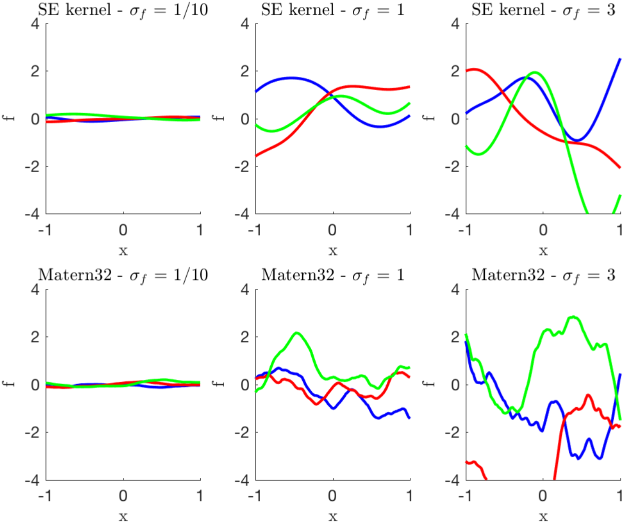
Three Common Covariance Functions

- The length scale ℓ determines the smoothness.



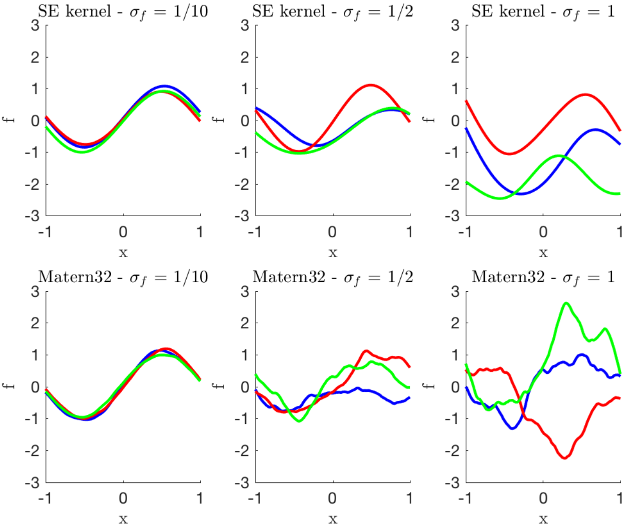
Three Common Covariance Functions

- 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:

$$p(\boldsymbol{y}|X, \theta) = \int p(\boldsymbol{y}|\boldsymbol{f}, X, \theta) p(\boldsymbol{f}|X, \theta) d\boldsymbol{f}$$

  where $\boldsymbol{f}|X, \theta \sim \mathcal{N}(0, K(X, X))$ and $\boldsymbol{y}|\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{f}, \sigma_n^2 I)$, which implies

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

  which alternatively can be obtained directly from

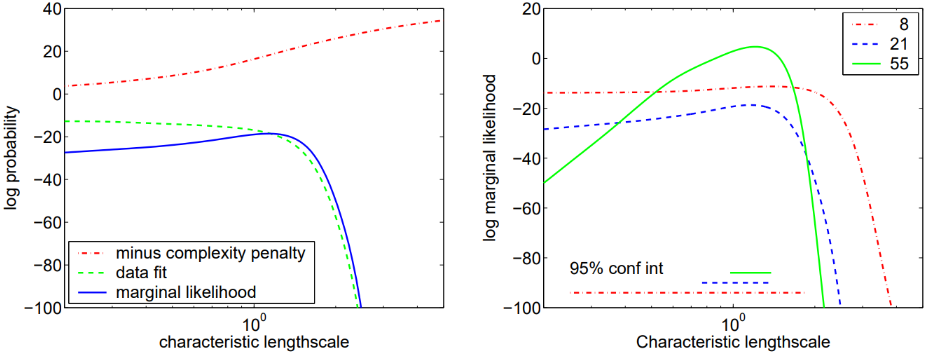
$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right).$$

- 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

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

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

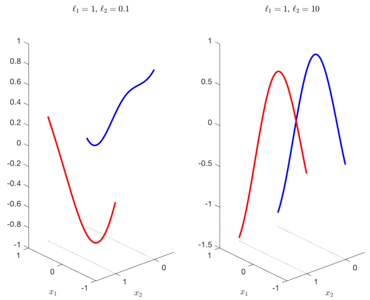


More on Covariance Functions

- Anisotropic version of isotropic covariance function (i.e., it depends only on r) by setting $r^2 = (\boldsymbol{x} - \boldsymbol{x}')^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{x}')$ where $\boldsymbol{M}$ is positive definite.
- Automatic Relevance Determination: $\boldsymbol{M} = diag(\ell_1^{-2}, \ldots, \ell_D^{-2})$, i.e. different length scales for different dimensions. In other words, ARM performs variable selection since a large ℓ_j means that the j-th dimension is essentially irrelevant for $f(\boldsymbol{x})$.
- Linear dimensionality reduction: $\boldsymbol{M} = \Lambda\Lambda^T + \Psi$ where Λ is a $D \times d$ matrix ($d < D$) whose columns define d directions of high relevance, and Ψ is a diagonal matrix capturing the axis aligned relevances.
- Periodic kernel with period d: $k(x, x') = \sigma_f^2 \exp\left\{ -\frac{2\sin^2(\pi|x - x'|/d)}{\ell^2} \right\}$.
- The sum and product of two kernels is a kernel. For instance:
  - $k_{ARD}(\boldsymbol{x}, \boldsymbol{x}') = \prod_{d=1}^{D} k_{SE,\ell_d}(x_d, x_d')$.
  - $k_{Periodic}(x, x') \times k_{SE}(x, x')$: Close peaks more dependent than distant ones.

More on Covariance Functions

- Assume x_1 is continuous (mg/week) and x_2 is binary (0=male, 1=female). Then,
$$k_{ARD}((x_1, x_2), (x_1', x_2')) = \exp\left\{-\frac{(x_1 - x_1')^2}{2\ell_1^2}\right\}\exp\left\{-\frac{(x_2 - x_2')^2}{2\ell_2^2}\right\}$$
and thus
$$cov(f(x_1, 0), f(x_1, 1)) = \exp\left\{-\frac{1}{2\ell_2^2}\right\}$$
which determines the similarity between the male and female profiles wrt x_1, i.e. large (resp. small) ℓ_2 implies similar (resp. potentially different) profiles.
- For more than two categories, use one-hot encoding.



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)

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

$$\left.\begin{array}{l}\boldsymbol{\alpha} := L^\top \backslash (L \backslash \mathbf{y}) \\ \bar{f}_* := \mathbf{k}_*^\top \boldsymbol{\alpha}\end{array}\right\} \text{ predictive mean eq. (2.25)}$$

$$\left.\begin{array}{l}\mathbf{v} := L \backslash \mathbf{k}_* \\ \mathbb{V}[f_*] := k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^\top \mathbf{v}\end{array}\right\} \text{ predictive variance eq. (2.26)}$$

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

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

- 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.
- $K = K(X, X)$.
- $K_* = K(X, X_*)$.
- $\boldsymbol{k}_* = k(\boldsymbol{x}_*) = K(X, \boldsymbol{x}_*)$.
- $L = cholesky(A) \Rightarrow A = LL^T \Rightarrow A^{-1} = (L^T)^{-1}L^{-1} = (L^{-1})^T L^{-1}$ and $|A| = det(A) = det(L)det(L^T) = 2\prod_i L_{ii}$.
- $L\ \boldsymbol{y} = solve(L, \boldsymbol{y}) = L^{-1}\boldsymbol{y}$.

Contents

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

Thank you