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

- Three Common Covariance Functions
- Learning the Hyperparameters of the Covariance Function
- More on Covariance Functions
- Gaussian Process Networks
- Bayesian Optimization

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.
  - Friedman, N. and Nachman, I. Gaussian Process Networks. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, 211-219, 2000.
  - Brochu, E., Cora, V. M. and de Freitas, N. *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*. https://arxiv.org/pdf/1012.2599.
- 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 $\alpha \to \infty$, $k_{RQ}(r) \to K_{SE}(r)$.

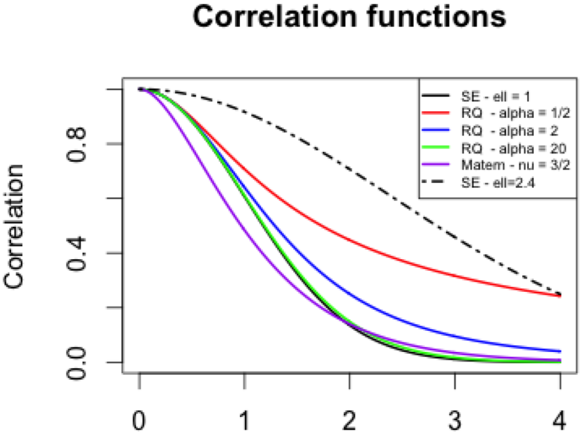
- Matérn:

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

  where $\sigma_f^2 > 0, \ell > 0, \nu > 0$, and K_ν the modified Bessel function. As $\nu \to \infty$, $k_{Matern}(r) \to K_{SE}(r)$.

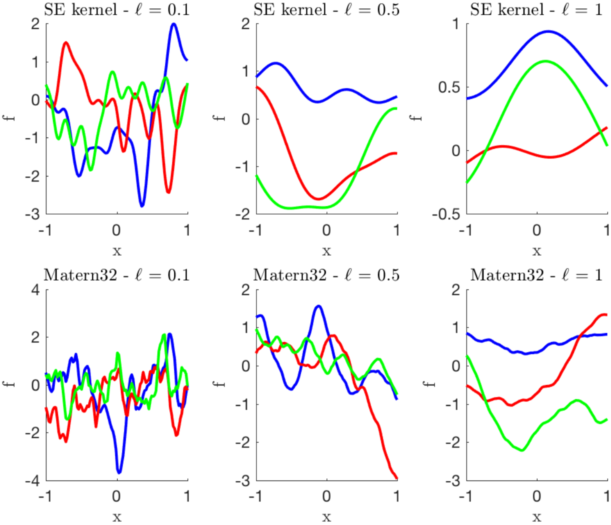
- Demo of `GaussianProcesses.R`.

Three Common Covariance Functions



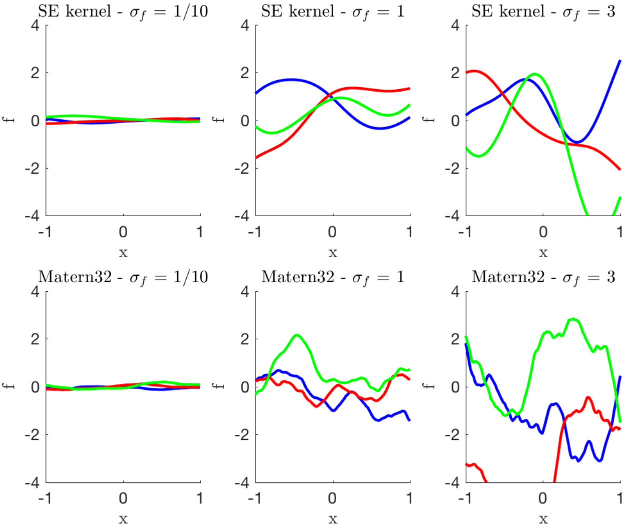
Three Common Covariance Functions

- The length scale ℓ determines the smoothness.



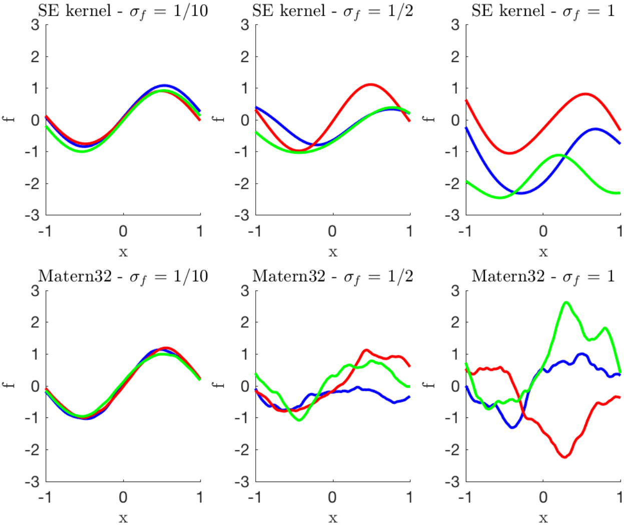
Three Common Covariance Functions

- 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 $\boldsymbol{f}|X \sim \mathcal{N}(0, K(X,X))$ and $\boldsymbol{y}|\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{f}, \sigma_f I)$, which implies

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

  which alternatively can be obtained directly from

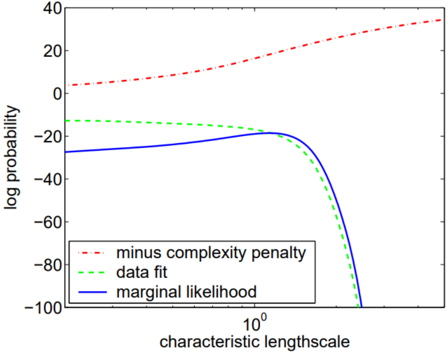
  $$\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{f}_*\end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix}K(X,X)+\sigma_n^2 I & K(X,X_*)\\ K(X_*,X) & K(X_*,X_*)\end{bmatrix}\right).$$

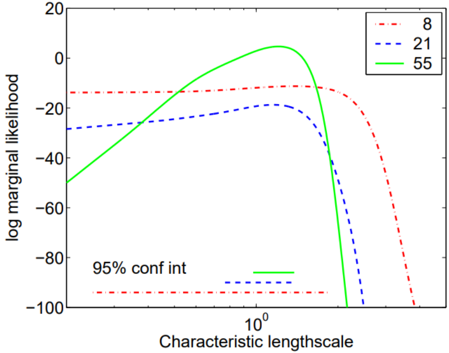
- In general, this is a non-convex optimization problem, and gradient methods are typically used. For most common covariance functions, the derivative of $K(X,X)$ wrt θ is easy to compute.
- For a Bayesian approach, choose the hyperparameters that maximize the posterior distribution $p(\theta|\boldsymbol{y},X) \propto p(\boldsymbol{y}|X,\theta)p(\theta)$. 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

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

= 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 align 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

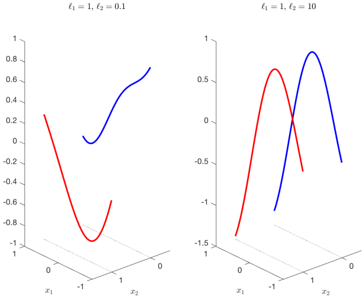
- Assume x_1 is continuous (mg/week) and x_2 is binary (0=male, 1=female). Then,

$$k_{ARD}((x_1, x_2), (x_1', x_2')) = \exp\left\{-\frac{(x_1 - x_1')^2}{2\ell_1^2}\right\} \exp\left\{-\frac{(x_2 - x_2')^2}{2\ell_2^2}\right\}$$

  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.



Gaussian Process Networks

- GPNs = BNs + GPs. In other words,

$$p(\boldsymbol{y}) = \prod_i q(y_i|pa_G(y_i)) = \prod_i q(y_i|\boldsymbol{x}_i) \text{ where } q(y_i|\boldsymbol{x}_i) = \mathcal{GP}(m(\boldsymbol{y}_i), k(\boldsymbol{x}_i, \boldsymbol{x}_i')).$$

- Learning means choosing the most likely DAG G given the data $d[1:N]$:

$$p(G|d[1:N]) = p(d[1:N]|G)p(G)/P(d[1:N]) \propto p(d[1:N]|G)p(G)$$

where

$$p(d[1:N]|G) = \prod_i p(y_i[1:N]|\boldsymbol{x}_i[1:N])$$

where

$$p(y_i[1:N]|\boldsymbol{x}_i[1:N]) = \mathcal{N}(m(\boldsymbol{y}_i[1:N]), K(\boldsymbol{x}_i[1:N], \boldsymbol{x}_i[1:N])).$$

- Note that the last term involves hyperparameter learning, i.e. not really closed-form expression.
- Problem: The Lauritzen-Spiegelhalter algorithm does not work. Solution: Approximate inference.

Bayesian Optimization

- Technique to find extreme values of an objective function. Particularly useful when the objective function is expensive to evaluate, no closed-form expression of it is available, its derivatives are not available, or it is non-convex. However, we should be able to evaluate the objective function at chosen values.
- Typical application: Hyperparameter learning.
- GPs are used to represent the posterior distribution over the objective function, which is used to choose the next point to evaluate. Specifically, the posterior mean of the GP (a.k.a. the surrogate function) can be seen an estimate of the objective function. The next point to evaluate is chosen so as to maximize the so-called acquisition function.
- BO in a nutshell:
  - Assign GP prior over the unknown objective function f.
  - Evaluate f at some values $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.
  - Update GP to the posterior $f|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)$.
  - Use the GP posterior over f to find a new evaluation point $\boldsymbol{x}_{n+1}$ by maximizing some acquisition function, which combines exploration with exploitation.
  - Iterate until the change in optimum is lower that some tolerance.

Bayesian Optimization

- Assume a maximization problem. Some acquisition functions follow.
- Probability of improvement:

  $$p(f(\boldsymbol{x}) \geq f(\boldsymbol{x}^{+})) = \Phi(z)$$

  where $\Phi(\cdot)$ is the standard normal cumulative distribution function, $z = (\mu(\boldsymbol{x}) - f(\boldsymbol{x}^{+}))/\sigma(\boldsymbol{x})$, $\boldsymbol{x}^{+}$ is the current maximum, and $\mu(\boldsymbol{x})$ and $\sigma(\boldsymbol{x})$ are the GP posterior's mean and variance. Pure exploitation (?).
- Expected improvement:

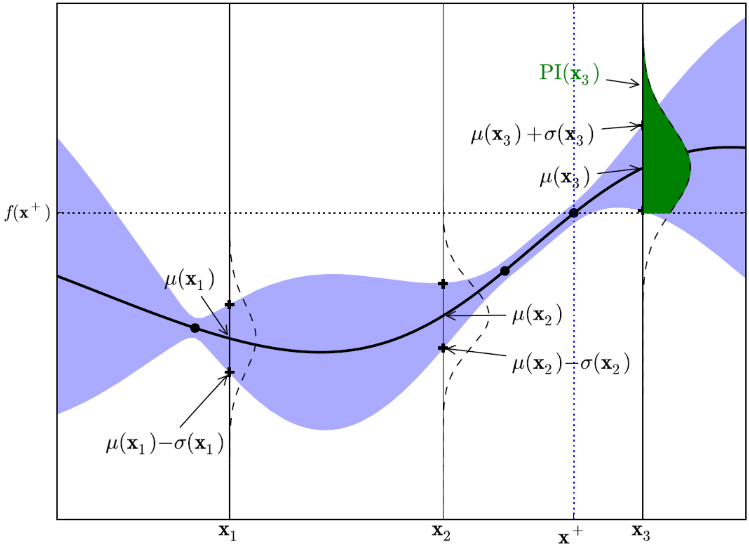
  $$(\mu(\boldsymbol{x}) - f(\boldsymbol{x}^{+}))\Phi(z) + \sigma(\boldsymbol{x})\mathcal{N}(z|0,1)$$

  which combines exploration (choose points with high surrogate variance) with exploitation (choose points with high surrogate mean).
- Upper confidence bound:

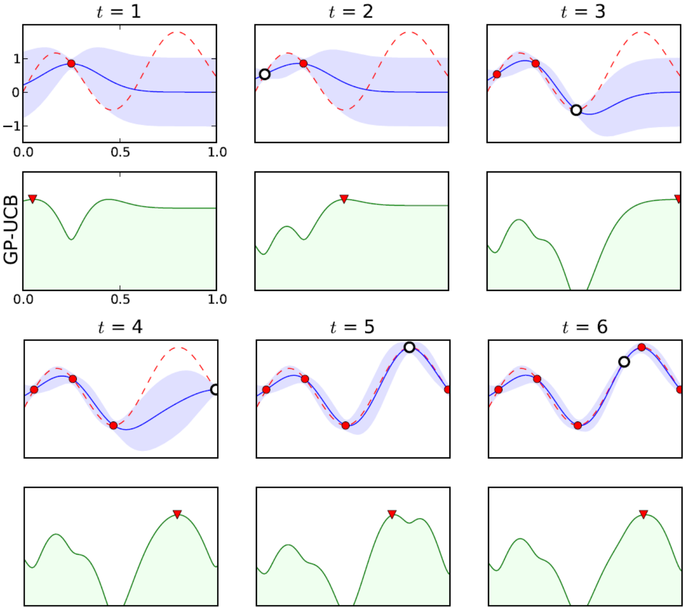
  $$\mu(\boldsymbol{x}) + \kappa\sigma(\boldsymbol{x})$$

  where κ is a user-defined parameter. Trade-off between exploration and exploitation.
- Note that we still need to maximize the acquisition function. Despite being non-convex, this function is cheaper and simpler to maximize than the original one.

Bayesian Optimization



Bayesian Optimization



Contents

- Three Common Covariance Functions
- Learning the Hyperparameters of the Covariance Function
- More on Covariance Functions
- Gaussian Process Networks
- Bayesian Optimization

Thank you