Probabilistic Machine Learning

Assignment 6: Gaussian Process Regression

Ralf Herbrich, Alexander Kastius

SS 2024

1 Introduction

This file provides additional information for the 6th exercise. It introduces exact formulas for the kernels, provides an additional section to the computation of the posterior for the GP as well as an introduction to the system that you do not implement manually.

2 Definition of Kernel Functions

We define kernel functions as having additive data noise. In this assignment, please implement this version of the RBF covariance function:

$$C(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \cdot \exp\left(-(\mathbf{x}-\mathbf{x}')^{\mathrm{T}}\mathbf{M}(\mathbf{x}-\mathbf{x}')\right) + \sigma_n^2 \cdot \mathbb{I}\left(\mathbf{x}=\mathbf{x}'\right), \tag{1}$$

where we restrict the matrix $\mathbf{M}$ to have the form $M = \mathrm{diag}(\mathbf{l})$. Thus, the kernel is parameterized by $\boldsymbol{\theta} = \left(\sigma_f^2, \sigma_n^2, \mathbf{l}\right) = \left(\sigma_f^2, \sigma_n^2, l_1, \ldots, l_k\right)$.

3 GP-Regression

We know that the conditional distribution $p(y^*|\mathbf{x}^*, \mathbf{X}, \mathbf{y})$ is a Gaussian distribution with mean and covariance given by

$$m(\mathbf{x}^*) = \mathbf{k}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{y} \tag{2}$$

$$\sigma^2(\mathbf{x}^*) = C(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{k}. \tag{3}$$

where $\mathbf{k} = \left(C\left(x_1, x^*\right), \ldots, C\left(x_n, x^*\right)\right)^{\mathrm{T}}$ and $\mathbf{C}_{ij} := C\left(\mathbf{x}_i, \mathbf{x}_j\right)$. However, inverting the matrix is numerically instable. As we need to solve two separate systems of equations, we use the Cholesky decomposition $\mathbf{C} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$.

Using the Cholesky decomposition, we can now compute the mean vector. Instead of computing $\mathbf{C}^{-1}\mathbf{y}$ we first solve

To do so, we first want to solve $\mathbf{C}\boldsymbol{\alpha} = \mathbf{y}$ for $\boldsymbol{\alpha}$. We do so by first solving $\mathbf{L}\mathbf{x} = \mathbf{y}$ for $\mathbf{x}$ and then solving $\mathbf{L}^{\mathrm{T}}\boldsymbol{\alpha} = \mathbf{x}$ for $\boldsymbol{\alpha}$. We express this succinctly using the backslash operator $\backslash$:

$$\boldsymbol{\alpha} = \mathbf{L}^{\mathrm{T}}\backslash(\mathbf{L}\backslash\mathbf{y}). \tag{4}$$

Now, $m\left(\mathbf{x}^*\right) = \mathbf{k}^{\mathrm{T}}\boldsymbol{\alpha}$.

Next, let us compute $\mathbf{k}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{k}$.

$$\mathbf{k}^T\mathbf{C}^{-1}\mathbf{k} = \mathbf{k}^T(\mathbf{L}\mathbf{L}^\mathbf{T})^{-1}\mathbf{k} \tag{5}$$
$$= \mathbf{k}^T(\mathbf{L}^\mathbf{T})^{-1}\mathbf{L}^{-1}\mathbf{k} \tag{6}$$
$$= \mathbf{k}^T(\mathbf{L}^{-\mathbf{1}})^T\mathbf{L}^{-1}\mathbf{k} \tag{7}$$
$$= (\mathbf{L}^{-\mathbf{1}}\mathbf{k})^T\mathbf{L}^{-1}\mathbf{k} \tag{8}$$

Thus, we compute $\mathbf{v} = \mathbf{L}\backslash\mathbf{k}$ and then have

$$\mathbf{k}^T\mathbf{C}^{-1}\mathbf{k} = \mathbf{v}^T\mathbf{v}. \tag{9}$$

4 Finding the Hyper-parameters

For finding the hyper-parameters, we will use the *marginal likelihood* $p(\mathbf{y}|\mathbf{X})$. The marginal likelihood is the integral of the likelihood times the prior:

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{f},\mathbf{X})p(\mathbf{f}|\mathbf{X})d\mathbf{f}\,. \tag{10}$$

We are thus marginalizing over the function values $\mathbf{f}$. Under the Gaussian process model the prior is zero-mean Gaussian $\mathbf{f} \sim N(\mathbf{0}, \mathbf{C})$. We can thus use the known identities of the product of two Gaussians. For n observations, we obtain

$$\log\left(p(\mathbf{y}|\mathbf{X})\right) = -\frac{1}{2}\mathbf{y}^\mathrm{T}\mathbf{C}^{-1}\mathbf{y} - \frac{1}{2}\log\left(|\mathbf{C}|\right) - \frac{n}{2}\log\left(2\pi\right)\,. \tag{11}$$

In order to evaluate this, we can use the Cholesky decomposition of $\mathbf{C}$. Also, the fact that $\log\left(|\mathbf{C}|\right) = \sum_{i=1}^{n}\log(\mathbf{L}_{ii})$ can be helpful.

4.1 Gradient Descent

Suppose we have $f(\mathbf{x})$ to be a differentiable multivariate function. We want to find $\mathbf{x}^* = \arg\min f(\mathbf{x})$. For many functions it is impossible to so analytically. However, we can guess an initialization $\mathbf{x}_0$ and then update $\mathbf{x}$ along the gradient of the function:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \eta \cdot \underbrace{\left.\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_n}}_{\nabla f(\mathbf{x}_n)}\,. \tag{12}$$

For a small enough learning rate $\eta > 0$ we obtain a sequence of estimators $\mathbf{x}_1, \mathbf{x}_2, \ldots$ that satisfy

$$f(\mathbf{x}_0) \geq f(\mathbf{x}_1) \geq \ldots\,. \tag{13}$$

We stop when $\|\nabla f(\mathbf{x}_n)\|_2^2 < \epsilon$ for some small ϵ of our choice or when a pre-specified maximum number of iteration is reached (whichever happens first). Please complete the implementation of `gradient_descent`. Note that we added boundary parameters. The algorithm stops when the $\mathbf{x}$ is outside the boundaries.

4.2 Using Gradient Descent for Hyperparameter optimization

The gradient of the marginal log-likelihood is

$$\frac{\partial}{\partial \theta_k} \log\left(p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})\right) = \frac{1}{2}\mathrm{tr}\left(\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{T}} - \mathbf{C}^{-1}\right)\frac{\partial \mathbf{C}}{\partial \theta_k}\right). \tag{14}$$

where $\boldsymbol{\alpha} = \mathbf{C}^{-1}\mathbf{y}$ as before. Here $\frac{\partial \mathbf{C}}{\partial \theta_k}$ is a $n \times n$ matrix, where n is the number of observations. We have $\left(\frac{\partial \mathbf{C}}{\partial \theta_k}\right)_{ij} = \frac{\partial}{\partial \theta_k} k(\mathbf{x}_i, \mathbf{x}_j)$. Let us first derive the derivatives w.r.t. to the noise parameters:

In order to find the partials wrt. $\boldsymbol{\theta}$ of the RBF-kernel we note that:

$$\frac{\partial C(\mathbf{x}, \mathbf{x}')}{\partial \sigma_f^2} = \exp\left(-(\mathbf{x} - \mathbf{x}')^{\mathrm{T}}\mathbf{M}(\mathbf{x} - \mathbf{x}')\right) \tag{15}$$

$$\frac{\partial C(\mathbf{x}, \mathbf{x}')}{\partial \sigma_n^2} = \mathbb{I}(\mathbf{x} = \mathbf{x}')\,. \tag{16}$$

The derivatives of the length scales are all identical. We have

$$\frac{\partial C(\mathbf{x}, \mathbf{x}')}{\partial l_k} = -\sigma_f^2 \cdot (x_k - x_k')^2 \cdot \exp\left(-(\mathbf{x} - \mathbf{x}')^T\mathbf{M}(\mathbf{x} - \mathbf{x}')\right). \tag{17}$$