Notes: Gaussian Processes for Machine learning – Ch2 Gaussian Process Regression

Yingbo Li

03/17/2021

Table of Contents

Weight-space View

Function-space View

- Prediction with noise-free observations
- Prediction with noisy observations
- Cholesky decomposition and GP regression algorithm
- Hyperparameters

Smoothing, Weight Functions, and Equivalent Kernels

Overview of Gaussian processes (GP)

- The problem is learning in GP is exactly the problem of finding suitable properties for the covariance function
- In this book, design matrix is defined slightly differently from common statistical textbooks. Rather, each column in a design matrix is a case, and each row is a covariate

A regression model with basis functions

- Basis function $\phi(\mathbf{x})$: maps a D-dimensional input vector $\mathbf{x}$ into an N-dimensional feature space
- $\mathbf{\Phi}(\mathbf{X}) \in \mathbb{R}^{N \times n}$: the aggregation of columns $\phi(\mathbf{x})$ for all n cases in the training data
- A regression model

$$f(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{w}, \quad y = f(\mathbf{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2)$$

- We use a zero mean Gaussian prior on the N-dimensional unknown weights $\mathbf{w}$ (aka regression coefficients)

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_p)$$

Predictive distribution

- For a new test point $\mathbf{x}_*$, the predictive distribution is

$$f_* \mid \mathbf{x}_*, \mathbf{X}, \mathbf{y} \sim \mathcal{N}\left(\frac{1}{\sigma_n^2}\phi_*^\top \mathbf{A}^{-1}\mathbf{\Phi}\mathbf{y}, \quad \phi_*^\top \mathbf{A}^{-1}\phi_*\right),$$

$$\phi_* = \phi(\mathbf{x}_*), \quad \mathbf{\Phi} = \mathbf{\Phi}(\mathbf{X}), \quad \mathbf{A} = \frac{1}{\sigma_n^2}\mathbf{\Phi}\mathbf{\Phi}^\top + \mathbf{\Sigma}_p^{-1}$$

- When make predictions, we need to invert the $N \times N$ matrix $\mathbf{A}$, which may not be convenient if N, the dimension of the feature space, is large

Rewriting the predictive distribution using the matrix inversion lemma

- Marix inversion lemma: $\mathbf{Z} \in \mathbb{R}^{n\times n}$, $\mathbf{W} \in \mathbb{R}^{m\times m}$, $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n\times m}$

$$\left(\mathbf{Z} + \mathbf{U}\mathbf{W}\mathbf{V}^\top\right)^{-1} = \mathbf{Z}^{-1} - \mathbf{Z}^{-1}\mathbf{U}\left(\mathbf{W}^{-1} + \mathbf{V}^\top\mathbf{Z}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^\top\mathbf{Z}^{-1}$$

- We can rewrite the predictive distribution on the previous page as

$$\begin{aligned} f_* \mid \mathbf{x}_*, \mathbf{X}, \mathbf{y} \sim \mathcal{N}\Big(&\boldsymbol{\phi}_*^\top \boldsymbol{\Sigma}_p \boldsymbol{\Phi}\left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1}\mathbf{y}, \\ &\boldsymbol{\phi}_*^\top \boldsymbol{\Sigma}_p \boldsymbol{\phi}_* - \boldsymbol{\phi}_*^\top \boldsymbol{\Sigma}_p \boldsymbol{\Phi}\left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1}\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_p \boldsymbol{\phi}_*\Big), \end{aligned} \tag{1}$$

$$\mathbf{K} = \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_p \boldsymbol{\Phi}$$

Kernel and the kernel trick

- In the predictive distribution on the previous page, the feature space always enters in the form of the kernel $k(\cdot, \cdot)$:

$$k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\Sigma}_p \boldsymbol{\phi}(\mathbf{x}'),$$

  where $\mathbf{x}, \mathbf{x}'$ are in either the training or the test sets

- Moreover, we can define

$$\boldsymbol{\psi}(\mathbf{x}) = \boldsymbol{\Sigma}_p^{1/2} \boldsymbol{\phi}(\mathbf{x}),$$

  so that the kernel has a simple dot product representation

$$k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\psi}(\mathbf{x}) \cdot \boldsymbol{\psi}(\mathbf{x}')$$

- Kernel trick: if an algorithm is defined solely in terms of inner products in input space, the it can be lifted into feature space by replacing occurrences of those inner products by $k(\mathbf{x}, \mathbf{x}')$

Gaussian process: definition

- A Gaussian process(GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution
- A GP is completely specified by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$$

- Usually the prior mean function is set to zero
- Bayesian linear regression as a Gaussian process

$$f(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{w}, \quad \mathbf{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_p)$$

Here, the GP mean function and the covariance function are

$$m(\mathbf{x}) = \mathbf{0}, \quad k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\Sigma}_p \boldsymbol{\phi}(\mathbf{x}')$$

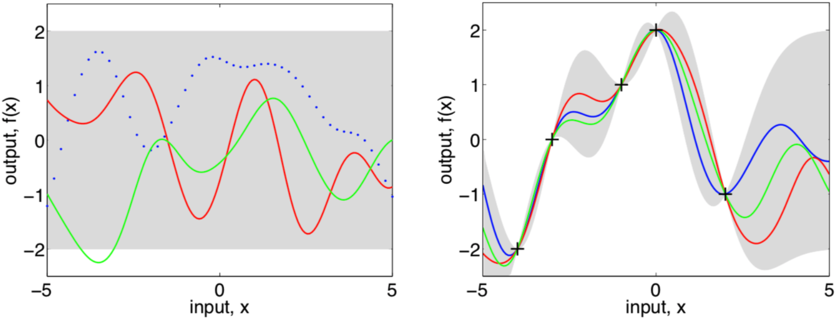
The squared exponential covariance function

- In this chapter, squared exponential (SE) covariance function will be used

$$\text{cov}\left(f(\mathbf{x}), f(\mathbf{x}')\right) = k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}\left|\mathbf{x} - \mathbf{x}'\right|^2\right)$$

  - By replacing $|\mathbf{x} - \mathbf{x}'|$ by $|\mathbf{x} - \mathbf{x}'|/\ell$ for some positive constant ℓ, we can change the characteristic length-scale of the process
  - Note that the covariance between the outputs is written as a function of the inputs
  - The squared exponential covariance function corresponds to a Bayesian linear regression model with a infinite number of basis functions
  - Actually for every positive definite covariance function $k(\cdot, \cdot)$, there exists a (possibly infinite) expansion in terms of basis functions

Three functions drawn at random from a GP prior (left) and their posteriors (right)



(a), prior

(b), posterior

- In both plots, shaded area are the pointwise mean plus and minus two times the standard deviation from each input value

Prediction with noise-free observations

- Suppose we have noise-free observations $\{(\mathbf{x}_i, f_i) : i = 1, \ldots, n\}$
- According to the GP prior, the joint distribution of the training outputs $\mathbf{f}$ and the test outputs $\mathbf{f}_*$ is

$$\left[\begin{array}{c} \mathbf{f} \\ \mathbf{f}_* \end{array}\right] \sim \mathcal{N}\left(\mathbf{0}, \left[\begin{array}{cc} K(\mathbf{X},\mathbf{X}) & K(\mathbf{X},\mathbf{X}_*) \\ K(\mathbf{X}_*,\mathbf{X}) & K(\mathbf{X}_*,\mathbf{X}_*) \end{array}\right]\right)$$

- By conditioning the joint Gaussian prior on the observations, we get the posterior distribution

$$\begin{aligned} \mathbf{f}_* \mid \mathbf{X}_*, \mathbf{X}, \mathbf{f} \sim \mathcal{N}\Big(&K(\mathbf{X}_*,\mathbf{X})K(\mathbf{X},\mathbf{X})^{-1}\mathbf{f}, \\ &K(\mathbf{X}_*,\mathbf{X}_*) - K(\mathbf{X}_*,\mathbf{X})K(\mathbf{X},\mathbf{X})^{-1}K(\mathbf{X},\mathbf{X}_*)\Big) \end{aligned}$$

Prediction with noisy observations

- With noisy observations $y = f(\mathbf{x}) + \epsilon$, the covariance becomes

$$\text{cov}(\mathbf{y}) = K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}$$

- Thus, the joint prior distribution becomes

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

- **Key predictive equation for GP regression**

$$\mathbf{f}_* \mid \mathbf{X}_*, \mathbf{X}, \mathbf{f} \sim \mathcal{N}\left(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*)\right), \quad \text{where} \tag{2}$$

$$\bar{\mathbf{f}}_* = K(\mathbf{X}_*, \mathbf{X}) \left[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2\right]^{-1} \mathbf{y}$$

$$\text{cov}(\mathbf{f}_*) = K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X}) \left[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2\right]^{-1} K(\mathbf{X}, \mathbf{X}_*)$$

Correspondence with weight-space view

- Connection between the function-space view, Eq (2), and the weight-space view, Eq (1)

$$K(C, D) = \boldsymbol{\Phi}(C)^\top \boldsymbol{\Sigma}_p \boldsymbol{\Phi}(D)$$

  where C, D stand for either $\mathbf{X}$ or $\mathbf{X}_*$

- Thus, for any set of basic functions, we can compute the corresponding covariance function as

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \boldsymbol{\Sigma}_p \phi(\mathbf{x}')$$

- On the other hand, for every positive definite covariance function k, there exists a possibly infinite expansion in terms of basis functions

Predictive distribution for a single test point x_*

- Denote $K = K(\mathbf{X}, \mathbf{X})$ and $\mathbf{k}_* = K(\mathbf{X}, \mathbf{x}_*)$, then the mean and variance of the posterior predictive distribution are

$$\bar{\mathbf{f}}_* = \mathbf{k}_*^\top \left(K + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{y}, \tag{3}$$

$$\mathbb{V}(\mathbf{f}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top \left(K + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{k}_* \tag{4}$$

Predictive distribution mean as a linear predictor

- The mean prediction Eq (3) is a linear predictor, i.e., it's a linear combination of observations $\mathbf{y}$
- Another way to look at this equation is to see it as a linear combination of n kernel functions

$$\bar{f}(\mathbf{x}_*) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}_*), \quad \boldsymbol{\alpha} = \left(K + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{y}$$

About the predictive distribution variance

- The predictive variance Eq (4) does not depend on the observed targets $\mathbf{y}$, but only the inputs. This is a property of the Gaussian distribution
- The noisy prediction of $\mathbf{y}_*$: simply add $\sigma_n^2\mathbf{I}$ to the variance

$$\mathbf{y}_* \mid \mathbf{x}_*, \mathbf{X}, \mathbf{y} \sim \mathcal{N}\left(\bar{\mathbf{f}}_*, \mathbb{V}(\mathbf{f}_*) + \sigma_n^2\mathbf{I}\right)$$

Cholesky decomposition

- Cholesky decomposition of a symmetric, positive definite matrix $\mathbf{A}$

$$\mathbf{A} = \mathbf{L}\mathbf{L}^\top,$$

  where $\mathbf{L}$ is a lower triangular matrix, called the Cholesky factor

- Cholesky decomposition is useful for solving linear systems with symmetric, positive definite coefficient matrix: to solve $\mathbf{A}\mathbf{x} = \mathbf{b}$
  - First solve the triangular system $\mathbf{L}\mathbf{y} = \mathbf{b}$ by forward substitution
  - Then the triangular system $\mathbf{L}^\top\mathbf{x} = \mathbf{y}$ by back substitution

- Backslash operator: $\mathbf{A}\backslash\mathbf{b}$ is the vector $\mathbf{x}$ which solves $\mathbf{A}\mathbf{x} = \mathbf{b}$
  - Under Cholesky decomposition,

$$\mathbf{x} = \mathbf{A}\backslash\mathbf{b} = \mathbf{L}^\top\backslash\left(\mathbf{L}\backslash\mathbf{b}\right)$$

- The computation of the Cholesky factor $\mathbf{L}$ is considered numerically extremely stable, and takes time' $n^3/6$

Algorithm: predictions and log marginal likelihood for GP regression

- **Input**: $\mathbf{X}, \mathbf{y}, k, \sigma_n^2, \mathbf{x}_*$

1. $\mathbf{L} = \text{cholesky}\left(K + \sigma_n^2 \mathbf{I}\right)$
2. $\boldsymbol{\alpha} = \mathbf{L}^\top \backslash \left(\mathbf{L} \backslash \mathbf{y}\right)$
3. $\bar{f}_* = \mathbf{k}_*^\top \boldsymbol{\alpha}$
4. $\mathbf{v} = \mathbf{L} \backslash \mathbf{k}_*$
5. $\mathbb{V}(\mathbf{f}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^\top \mathbf{v}$
6. $\log p(\mathbf{y} \mid \mathbf{X}) = -\frac{1}{2}\mathbf{y}^\top \boldsymbol{\alpha} - \sum_i \log L_{ii} - \frac{n}{2} \log 2\pi$

- **Return**: $\bar{f}_*, \mathbb{V}(\mathbf{f}_*), \log p(\mathbf{y} \mid \mathbf{X})$
- Computational complexity: $n^3/6$ for the Cholesky decomposition in Line 1, and $n^2/2$ for solving triangular systems in Line 2, 4

Hyperparameters

- One-dimensional squared-exponential covariance function

$$k_y(x_p, x_q) = \sigma_f^2 \exp\left[-\frac{1}{2\ell^2}(x_p - x_q)^2\right] + \sigma_n^2 \delta_{pq}$$

- It has three hyperparameters
  - Length-scale ℓ
  - Signal variance σ_f^2
  - Noise variance σ_n^2

- After selected ℓ, the rest two hyperparameters are set by optimizing the marginal likelihood

$$\log p(\mathbf{y} \mid \mathbf{X}) = -\frac{1}{2}\mathbf{y}^\top \left(K + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{y} - \frac{1}{2}\log\left|K + \sigma_n^2 \mathbf{I}\right| - \frac{n}{2}\log 2\pi$$

References

- Rasmussen, C. E. and Williams, C. K. I. (2006). Gaussian Processes for Machine learning, MIT press.
  - http://www.gaussianprocess.org/gpml/chapters/RW.pdf