Gaussian Process for Regression
An introduction

Unlike classical regression methods in Machine Learning, Gaussian process makes inference directly in the function space. Hence, the learned distribution is made not over weights, but over functions.

Contents

1 Weight-space view

Lets start by reminding the case of linear regression in weight-space view. We consider a training set :

$$\mathcal{D} = \{(x_i, y_i),\, i \in \{1, \dots, n\}\} \tag{1}$$

with $x_i \in \mathbb{R}^d$, $\forall i \in \{1, \dots, n\}$. The dataset is stored in the design matrix $X \in \mathcal{M}_{d,n}(\mathbb{R})$ (the i^{th} column of X is made of x_i). X is the *design matrix*.

1.1 Bayesian linear regression

▶ We consider that the observed values $(y_i)_{i\in\{1,\dots,n\}}$ differ from an underlying function f from a centered Gaussian noise :

$$y = f(x) + \varepsilon \qquad \varepsilon \sim \mathcal{N}(\varepsilon \,|\, 0, \sigma^2) \tag{2}$$

For now, we will consider that the underlying function f is linear in x. Hence we obtain that $\forall i \in \{1, \dots, n\}$:

$$y_i = w^T x_i + \varepsilon \tag{3}$$

hence $y_i \sim \mathcal{N}(y_i \,|w^T x_i, \sigma)$

▶ The likelihood of the training set D is thus, if we suppose that drawing of ε are independently distributed (hence making the $(y_i)_i$ independent) :

$$\begin{aligned} p(Y \mid X, w) &= \prod_{i=1}^{n} p(y_i \mid x_i, w) \\ &= \prod_{i=1}^{n} \mathcal{N}(y_i \mid w^T x_i, \sigma^2) \\ &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y_i - w^T x_i\right)^2\right) \\ &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2} \left(\|Y - X^T w\|\right)^2\right) \end{aligned} \tag{4}$$

▶ Hence :

$$p(Y \mid X, w) = \mathcal{N}(Y \mid X^T w, \sigma^2 I_n) \tag{5}$$

To comply with the Bayesian regression framework, we need to specify a *prior distribution* for the weight vector w. Let us suppose that it is normally distributed :

$$w \sim \mathcal{N}(w \mid 0, \Sigma_p) \tag{6}$$

Let us now remind the Bayes' rule :

Bayes' rule

$$p(A \mid B) \propto p(B \mid A) \cdot p(A)$$

Then we have that :

$$\underbrace{p(w \mid X, Y)}_{\text{posterior}} \propto \underbrace{p(Y \mid X, w)}_{\text{likelihood}} \underbrace{p(w)}_{\text{prior}} \tag{7}$$

By a standard tractation known as "completing the square" of a Gaussian distribution, it is then easy to that the *maximum-a-posteriori* (MAP) solutions is given by :

MAP Bayesian solution for linear regression

$$w_{MAP} \sim \mathcal{N}(w_{MAP} \mid \frac{1}{\sigma^2} A^{-1} X y, A^{-1}) \tag{8}$$

with

$$A = \frac{1}{\sigma_n^2} X X^T + \Sigma_p^{-1} \tag{9}$$

One can notice the presence of the term $\frac{1}{\sigma_n^2} X X^T$ that is derived in the tractation of the maximum-likelihood solution w_{ML}. This solutions is actually equivalent to *Ridge-regression* for $\Sigma_p = \lambda Id$.

1.2 Making new prediction

Of course, our motivation is not to draw weight points from the posterior distribution. We want to be able to make new predictions from new inputs vector x^*. This is done by averaging over all values for w :

$$p(f_* \mid x_*, X, Y) = \int_{w \in \mathbb{R}^n} p(f_* \mid x_*, w) \cdot p(w \mid X, Y) dw \tag{10}$$

which leads to the following expression, knowing that $f_* = w^T x^*$:

MAP prediction

$$f_* \,|\, x_*, X, Y \sim \mathcal{N}\left(f_* \,\middle|\, \frac{1}{\sigma^2}(x^*)^T A^{-1} X y,\ (x^*)^T A^{-1} x^*\right)$$

1.3 Change to feature space

Of course, simple linear regression can sometimes poorly fit our data. The trick is thus to convert the inputs vectors into vectors of feature space. Let

$$(\Phi_1, \dots, \Phi_N) : \mathbb{R}^d \to \mathbb{R} \tag{11}$$

Then $\Phi = (\Phi_1, \dots, \Phi_N)$ is a functions that map $\mathbb{R}^d$ into $\mathbb{R}^n$. We can therefore derive the exact same procedure as before but with a weight vector living in *feature space* :

$$f(x) = w^T \Phi(x) \tag{12}$$

MAP prediction with features

$$f_* \,|\, x_*, X, Y \sim \mathcal{N}\left(f_* \,\middle|\, \Phi_* \Sigma_p \Phi (K + \sigma^2 I)^{-1} y,\ \Phi_*^T K \phi_* - \Phi_* \Sigma_p \Phi (K + \sigma^2 I)^{-1} \Phi^T \Sigma_p \Phi_*\right)$$

where we used $\Phi = \Phi(X)$, $\Phi_* = \Phi(x^*)$ and $K = \Phi^T \Sigma_p \Phi$.

As one can see, feature vectors enter this expression with always the same structure, that is :

$$\begin{aligned} k(x, x') &= \Phi(x)^T \Sigma_p \Phi(x') \\ &= \Psi(x)^T \Psi(x) \end{aligned} \tag{13}$$

where $k(\cdot, \cdot)$ is the *kernel or covariance function*, and where we used that $\Sigma_p \in \mathcal{S}_n^{++}(\mathbb{R})$ to transform the initial kernel expression into a *scalar product in feature space* ! Hence, using the kernel function enables one to leave the exact expression of the feature aside, by only employing a given scalar product in feature space.

2 Function-space view

2.1 Gaussian Process

Gaussian Process

A Gaussian process can be described as a continuous process (X_t), which any finite number of which

$$(X_{t_1}, \dots, X_{t_n}, \dots, X_{t_m}) \tag{14}$$

have a joint Gaussian distribution.

Because it is normally distributed, such a random vector is entirely described by its mean and covariance. Let us now consider our regression function as a Gaussian process, meaning that any collection of samples from f will have a joint normal distribution.

Let us then write that :

$$\begin{aligned} m(x) &= \mathbb{E}\left[f(x)\right] \\ k(x, x') &= \mathbb{E}\Big[\big(f(x) - m(x)\big) \cdot \big(f(x') - m(x')\big)\Big] \end{aligned} \tag{15}$$

hence we will write the regressive Gaussian process as :

$$f(x) \sim GP\Big(m(x), k(x, x')\Big) \qquad x \in \mathcal{X} \tag{16}$$

2.2 From the Bayesian approach

Regarding the latter Bayesian development with prior $\omega \sim \mathcal{N}(\omega \,|\, 0, \Sigma_p)$. Then, given that $f = \omega^T \phi$ we have that

$$\mathbb{E}\left[f(x)\right] = 0 \tag{17}$$

$$\begin{aligned} k(x, x') &= \mathbb{E}\left[\phi(x)^T w w^T \phi(x')\right] \\ &= \phi(x)^T \Sigma_p \phi(x') \end{aligned} \tag{18}$$

meaning that two samples $f(x)$ and $f(x')$ are jointly Gaussian with zero mean and covariance given by $\phi(x)^T \Sigma_p \phi(x')$.

■ A famous example of a covariance function is given by the *square exponential* or *Radial Basis Function* :

$$cov(f(x_p), f(x_q)) = \exp\left(-\frac{1}{2}\|x_p - x_q\|_2^2\right) \tag{19}$$

It is important to understand of a covariance function implies a *distribution over functions.* If given a number of input points, stored in the matrix X_* and write out the corresponding covariance matrix, we actually consider a random Gaussian vector with :

$$f_* \overset{d}{\sim} \mathcal{N}\left(f_* \,|\, 0, K(X_*, X_*)\right) \tag{20}$$

2.3 Prediction with noise-free observation

We wish not to draw random functions from the prior distribution but to incorporate the knowledge of the training point when predicting output relative to new inputs. To do this, we write down the joint prediction of the training outputs and the test outputs :

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \overset{d}{\sim} \mathcal{N}\left(\begin{bmatrix} f \\ f_* \end{bmatrix} \middle| 0, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \tag{21}$$

The posterior distribution would then be given by the conditional distribution over f_* over f.

2.3.1 Reminder on joint Gaussian distribution

Let x be a d-dimensional vector so that

$$x \overset{d}{\sim} \mathcal{N}\left(x \,|\, \mu, \Sigma\right) \tag{22}$$

and we partition x over two subset : $x = \begin{pmatrix} x_a \\ x_b \end{pmatrix}$. Hence we have

$$\mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \Sigma_{aa} & \Sigma_{ba} \\ \Sigma_{ab} & \Sigma_{bb} \end{pmatrix} \tag{23}$$

Then a standard procedure known as *completing the square* leads to the conditional distribution :

Conditional Gaussian distribution

Given the joint Gaussian distribution of $x \overset{d}{\sim} \mathcal{N}\left(x \,\middle|\, \begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}, \begin{pmatrix} \Sigma_{aa} & \Sigma_{ba} \\ \Sigma_{ab} & \Sigma_{bb} \end{pmatrix}\right)$

$$\begin{aligned} \mu_{a|b} &= \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b) \\ \Sigma_{a|b} &= \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba} \end{aligned} \tag{24}$$

2.3.2 Application to Gaussian process

The application of the latter result allows us retrieve the posterior distribution regarding f_* :

Noise-free prediction with GP

$$f_* \overset{d}{\sim} \mathcal{N}\left(f_* \,\middle|\, K(X_*,X)K(X,X)^{-1}f, K(X_*,X_*) - K(X_*,X)K(X,X)^{-1}K(X,X_*)\right) \tag{25}$$

2.4 Prediction with noisy observation

Let us consider that y is a noisy observation of the regression function f :

$$y = f(x) + \varepsilon, \quad \varepsilon \overset{d}{\sim} \mathcal{N}\left(\varepsilon \,\middle|\, 0, \sigma_n^2\right) \tag{26}$$

Assuming the noise is independent from sample to sample, we thus have that :

$$\text{cov}(x_p, x_q) = k(x_p, x_q) + \sigma_n^2 \delta_{p-q}, \; \forall p, q \in \mathbb{N} \tag{27}$$

so that :

$$f \overset{d}{\sim} \mathcal{N}\left(f \,\middle|\, 0, K(X,X) + \sigma_n^2 I_n\right) \tag{28}$$

Using the exact same reasoning as before we obtain the following result.

Prediction with noisy training-set

$$f_* \overset{d}{\sim} \mathcal{N}\left(f_* \,\middle|\, \bar{f}_*, \text{cov}(f_*)\right) \tag{29}$$

with :

$$\bar{f}_* = K(X,X_*)\left[K(X,X) + \sigma_n^2 I_n\right]^{-1} f \tag{30}$$

and

$$\text{cov}(f_*) = K(X_*,X_*) - K(X_*,X)\left[K(X,X) + \sigma_n^2 I_n\right]^{-1} K(X,X_*) \tag{31}$$

■ When we wish to compute the output for only one request, we will use the notation :

$$\begin{aligned} K(X,X) &= K \\ K(X,X_*) &= k_* \end{aligned} \tag{32}$$

and thus we have that

$$\begin{aligned} \bar{f}_* &= k_*^T \left[K + \sigma_n^2 I_n\right]^{-1} f \\ &= \sum_{i=1}^{n} \alpha_i k(x_*, x_i) \qquad \text{with } \alpha = \left[K + \sigma_n^2 I_n\right]^{-1} f \end{aligned} \tag{33}$$

This means that the mean prediction for f_* can be written as a finite linear of evaluation of the covariance function, despite the fact that the GP can be represented in terms of a possibly infinite number of basis functions. This is a manifestation of the *representer theorem*.

3 Weight functions and equivalent kernels

It is pretty clear, given equation (33) that the GP procedure computes a weighted average of the noisy observations y as the guess given b y decision theory (mean of the distribution) is :

$$\bar{f}(x_*) = k(x_*)^T \left[K + \sigma_n^2 I_N\right]^{-1} y \tag{34}$$

This makes the Gaussian process regression a *linear smoother method*.

Let us now define the vector field h by :

$$h(x_*) = \Big[K + \sigma_n^2 I_N\Big]^{-1} k(x_*) \tag{35}$$

so that we have :

$$\bar{f}(x_*) = h(x_*)^T y \tag{36}$$

giving h the name of *weight function*.

There is a clear analogy with the kernel smoothing methods. Hence the weight function also takes the name of *equivalent kernel*. As a reminder, a kernel smoother is given by :

$$\kappa_i(x) = \kappa\left(\frac{|x_i - x|}{l}\right) \tag{37}$$

and we compute the prediction by the **Nadayara-Watson** estimator :

$$f(x) = \sum_{i=1}^{n} \frac{\kappa_i(x)}{\sum_{j=1}^{n} \kappa_j(x)} y_i \tag{38}$$

However, one must keep in mind that the equivalent kernel is really different from the original kernel (squared-exponential, ..).

4 Incorporation of explicit basis function

It is common but non necessary to consider GPs with a zero mean function - remember, no loss of generality ! Yet one can decide to explicitly model a mean function for a Gaussian Process.

4.1 Deterministic mean

The derivation from the previous considerations is pretty straight forward. Let :

$$f \sim GP\Big(m(x), k(x, x')\Big) \tag{39}$$

Then the prediction is made by using the following :

$$\bar{f}_* = m(X_*) + K(X_*, X)\Big(K(X, X) + \sigma_n^2 I_N\Big)(y - m(X)) \tag{40}$$

The mean function being deterministic, the posterior covariance remains the same.

4.2 Inferring the mean

It is actually more convenient to specify only a few-fixed basis functions (based on understanding of the underlying model for instance), which weights β will be inferred from the data. Let us define β's prior :

$$\beta \overset{d}{\sim} \mathcal{N}(\beta \mid b, B) \tag{41}$$

We then obtain a new Gaussian Process :

$$g(x) \sim GP(h(x)^T b, k(x, x') + h(x)^T B h(x')) \tag{42}$$

making us able to make new guesses. To infer the optimal values for b and B, one can study the marginal likelihood of the model.