AIML - CS 337

Lecture 15: Kernels and Gaussian Processes

6th October 2022

Lecturer: Abir De *Scribe: Harsh Shah, Gutta Sindhuja, Divyeswar, Khyathi Gayathri*

1 Prologue

By now, we have studied various kernel tricks which can be for separating data having non-linear relationship by simply defining an appropriate Gram matrix representing the kernel. Further, the trick can be extended to non parametric regression[3], classification and PCA(kernel PCA[2]) as well.

2 Problem

Consider a linear regression model as follows:

$$w^{\text{regression}} \rightarrow \min \left[\sum_{i \in D} (y_i - w^T x_i)^2 \right]$$

The solution to the above problem is:

$$w^{\text{regression}} = (\sum_{i \in D} x_i x_i^T)^{-1} \cdot (\sum_{i \in D} x_i y_i)$$

The predictions are made using function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, $f(x_i) = w^T \cdot x_i$ Notice that when we substitute an input data from training set,

$$f(x_i) \neq y_i$$

We aim to devise a function f such that $\forall i \in D \ f(x_i) \sim y_i$, while having certain guarantees on accuracy on test set(assuming train and test set are from same distribution). Let's roll down to some maths...

3 Gaussian Process

Let $X_D = [X_1^d X_2^d \dots X_n^d]^T$ denote the points in train set and $X_T = [X_1^t X_2^t \dots X_n^t]^T$ denote the points in test set. $f(X_i)$ and $f(T_i)$ denote random variables depending on the input.

$$f(X) = \begin{pmatrix} f(X_1^d) \\ f(X_2^d) \\ . \\ . \\ f(X_n^t) \end{pmatrix}$$

These random variables are dependent on each other, and we need to model the dependency between them(Gaussian has got our backs!). We model $f(X)$ as multi-variate Gaussian distribution. Therefore,

$$\begin{bmatrix} f(X_D) \\ f(X_T) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(X_D) \\ m(X_T) \end{bmatrix}, \begin{bmatrix} k(X_D, X_D) & k(X_D, X_T) \\ k(X_D, X_T)^T & k(X_T, X_T) \end{bmatrix} \right)$$

Here, $m(\cdot)$ is a function denoting mean, and $k(\cdot, \cdot)$ is our beloved kernel function used for creating the covariance matrix. The reason we are using kernel function here is because we want to model some sort of similarity between the random variables, high correlation implying higher similarity. With this model, we can determine the prior distribution of the random variables $f(\cdot)$

$$P(f(X)) = P\left(\begin{bmatrix} f(X_D) \\ f(X_T) \end{bmatrix} \right) = \frac{1}{\text{(some constant)} \cdot \det(K)^{0.5}} \exp(-0.5 f(X)^T K^{-1} f(X))$$

(The above expression is just joint probability distribution of muli-variate Gaussian)
This is just the prior and we are yet to use the train predictions $\{y_i\}$.
Reminder : Our aim is to get distribution of $f(X_T)$ given the train predictions $\{y_i\}$.
Let's now use Bayes' rule to evaluate the posterior distribution.

3.1 Evaluating the posterior

We would model the predictions $Y = [y_1 y_2 \dots y_n]^T$ as

$$Y = I \cdot f(X_D)$$

A more general model would be to include additive Gaussian noise such that $Y = f(X_D) + \eta$. But for simplicity, we assume noise to be zero. Before performing any further math to get to the posterior, there are two results which will reduce much of our pain(which otherwise would have to be done using Bayes' rule and some integrations).

3.1.1 Gaussian Marginalisation Rule

If we marginalize out variables in a multivariate Gaussian distribution, the result is still a Gaussian distribution. Mathematically, if $X = [X_1, X_2, \dots X_n]^T$ is a multi-variate Gaussian random variable($\sim \mathcal{N}(\mu, \sigma)$), then any subset of X is a multi-variate Gaussian and the mean and covariance is given by $(A\mu, A\sigma A^T)$. A can be constructed by using e_i^T as rows. For example, if $n = 3$, and the subset is constructed using $[X_1\ X_3]$, then,

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

How to prove? Need to integrate the joint probability distribution over all random variables that are to be removed (eg, X_2 in the above case)

3.1.2 Conditional Rule for multi-variate Gaussian

Intuitively, if we start with a Gaussian distribution and update our knowledge given the observed value of one of its components(that is, find conditional probability distribution), then the resulting distribution is still Gaussian! Mathematically,
Let $[x\ y]$ jointly form multi variate Gaussian random variable,

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} \right)$$

The covariance is represented as a block matrix. The reason being, we would later represent covariance matrix of $f(x|y)$ in terms of the blocks in Σ. Here $f(\cdot)$ is PDF(and not a random variable).

$$f(x|y) = \frac{f(x, y)}{f(y)}$$

Now, we will substitute $f(x, y)$ with the expression for multi-variate Gaussian distribution($\mathcal{N}(\mu, \Sigma)$), and $f(y)$ with $\mathcal{N}(\mu_y, \Sigma_{yy})$. Then we will perform all sorts of manipulations and simplifications in our head to get the below result

$$f(x|y) = \mathcal{N}(\Sigma_{xy}\Sigma_{yy}^{-1}y, \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx})$$

(to get the above expression, μ is assumed to be zero)
To get the detailed derivation, please see [4]

3.2 Getting to the posterior

Remember we modelled $Y = f(X_D)$, now's the time to use it.

$$\begin{bmatrix} f(X_T) \\ Y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(X_T) \\ \mu(X_D) \end{bmatrix}, \begin{bmatrix} k(X_T, X_T) & k(X_T, X_D) \\ k(X_T, X_D)^T & k(X_D, X_D) \end{bmatrix} \right)$$

Using the conditional rule described above,

$$P(f(X_D)|Y) = \mathcal{N}(\mu_{posterior}, \sigma^2_{posterior})$$

where

$$\begin{aligned} \mu_{posterior} &= \mu X_T + k(X_T, X_D)(k(X_D, X_D))^{-1}(Y - \mu(X_D)) \\ \sigma^2_{posterior} &= k(X_T, X_T) - k(X_T, X_D)(k(X_D, X_D))^{-1}k(X_T, X_D)^T \end{aligned}$$

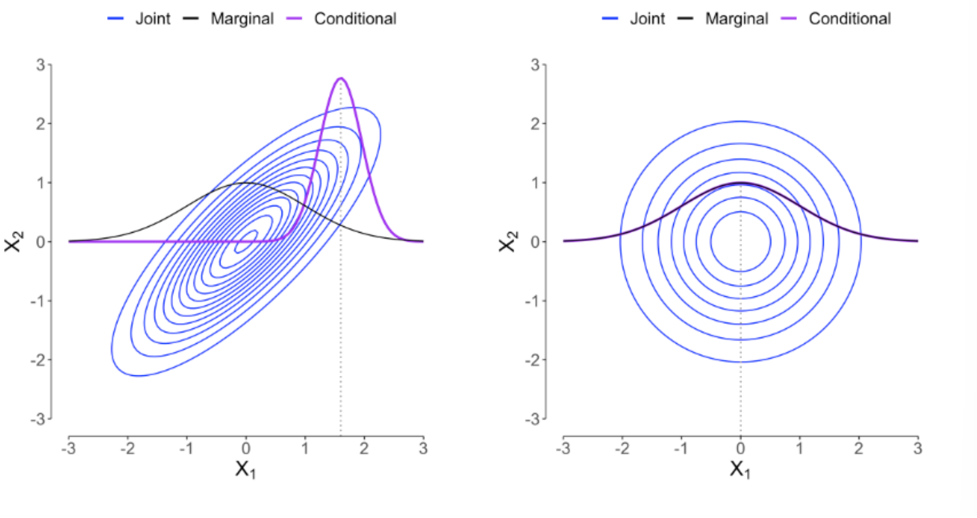

Figure 1: Joint, Marginal, Conditional for bivariate Gaussian. Source[4]

4 Aftermath[1]

We now have these long expressions, but what do they mean? Let's investigate them.

4.1 Posterior mean

For the sake of investigation, let's assume $\mu(\cdot) = 0$.
Now, if $X_D = X_T$, $\mu_{posterior} = Y$ which is desirable because we want the mean for predictions at training points to be the same as the given predictions in train set. The mean value at a single test location, say x_i^t, is a weighted sum of all the observations Y. The weights are defined by the kernel between the test location x_i^t and all training locations in X.

4.2 Posterior variance

Observe that if $X_D = X_T$, we get $\sigma_{posterior} = 0$. This means that the prediction for a point in train set is exactly the mean, which in turn is the Y of the training set.

4.3 Example

This example is taken from here [1]. The train set is generated by random sampling x ~U($[0, 2\pi]$), and $y = Sin(x) + \epsilon(noise)$. The blue bands represent a 95% confidence interval. Notice that for test points near the train points, variance is quite low and ultimate zero at exactly the train points.

[1] pun intended

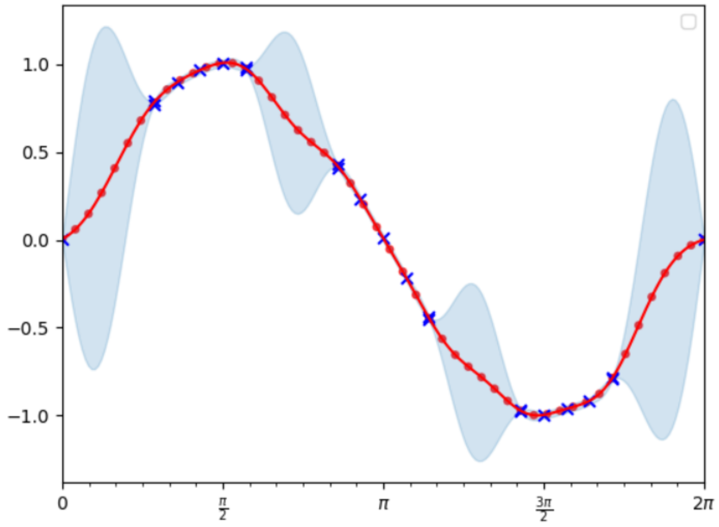

Figure 2: Red points are for test points, and blue ones for train points

5 What have we done?

We started with a train set $(\{(x_i^d, y_i^d)\})$ and test inputs $\{x_i^t\}$, with the aim to devise a function that would yield close to 0 error for inputs which are in train set, and low errors on other points(the test set). We modelled the functions as multi-variate Gaussian and applied some math to get the posterior distribution of the set of the functions$(f(\cdot))$ given the predictions Y, i.e.,$P(f(X)|Y)$. The predictions on test set can be made using the distribution acquired, either by random sampling(undeterministic) or by using the mean of the distribution. The described method is particularly useful for low data situations.

References

[1] *Gaussian Process*. URL: https://towardsdatascience.com/understanding-gaussian-process-the-socratic-way-ba02369d804.

[2] *Kernel PCA*. URL: https://www.geeksforgeeks.org/ml-introduction-to-kernel-pca/.

[3] *Non-parametric regression*. URL: https://en.wikipedia.org/wiki/Nonparametric_regression.

[4] *Properties of multi-variate Gaussian*. URL: https://fabiandablander.com/statistics/Two-Properties.html.