Introduction to Gaussian Processes

Filipe P. Farias
Teleinformatics Engineering Department
Federal University of Ceará
filipepfarias@fisica.ufc.br

Abstract

A wide variety of methods exists to deal with supervised learning, as restrict a class of linear functions of the inputs, as linear regression, or give a prior probability to every possible function, giving high probability to the functions we consider more likely. The second approach is a way to Gaussian process itself. We will make the pathway through a intuitive construction of this framework.

1 Introduction

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2 Linear Regression

Starting with a simple regression problem. Be the dataset $\mathcal{D} = \{x_i, y_i | i = 0, \dots, N-1\}$, where we observe a real-valued input variable x and a measured real-valued variable y. Then, we'll use synthetically generated data for comparison against any learned *model*. And N will be the number of observations of the value y. Our objective is make predictions of the new value $\hat{y}$ for some new input $\hat{x}$.

For this example, we'll use a simple approach based on curve fitting by the polynomial model, i.e., being the function

$$f(x, \mathbf{w}) = \sum_{j=0}^{M-1} w_j x^j \tag{2.1}$$

where M is the order of the polynomial and $\mathbf{w} = [w_0, \dots, w_M]$ its coefficients. It's important to note that the f isn't linear in x but in $\mathbf{w}$. These functions which are linear on the unknown parameters are called *linear models*. Section 1.1 - Bishop (pg 4).

We can extend the class of models considering linear combinations of nonlinear functions of the input variables, i.e.

$$f(x, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(x) \tag{2.2}$$

where $\phi_j(x)$ are known as *basis functions*, and then the total number of parameters for this model will be M. We can evaluate the same operation of (2.2) in the matrix form by

$$f(x, \mathbf{w}) = \mathbf{w}^\top \boldsymbol{\phi}(x) \tag{2.3}$$

where $\boldsymbol{\phi}(x) = [\phi_0(x), \ldots, \phi_{M-1}(x)]^\top$. In the example of the curve fitting, the polynomial regression implies that $\phi_j(x) = x^j$. It's important to note that these linear models are needed to define its basis functions before the training dataset is observed. Section 1.4 - Bishop (pg 33).

The values of $\mathbf{w}$ are obtained by minimizing the *error function*, a measure of the distance between the training dataset and f, given values of $\mathbf{w}$. By the way, the chosen error function will be

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{f(x_i, \mathbf{w}) - y_i\}^2 \tag{2.4}$$

This indicate that if E is zero, f passes exactly through each training data point. Observe that E do not assume negative values because of its quadratic form, then we can find $\mathbf{w}$ by finding the minimum value of E, denoted $\mathbf{w}^*$, by

$$\frac{\partial E}{\partial \mathbf{w}} = 0 \tag{2.5}$$

We can rewrite (2.4) in the matrix form, considering $\mathbf{f} = f(\mathbf{x}, \mathbf{w})$, where $\mathbf{x} = [x_0, \ldots, x_{N-1}]^\top$, i.e. f evaluated for all input variables (See Appendix A.1). Then we have

$$\mathbf{f} = \Phi \mathbf{w} \tag{2.6}$$

where Φ is the *design matrix* such that $\boldsymbol{\phi}(x)$ is evaluated for all $\mathbf{x}$. Proceeding with the minimization, we obtain

$$\mathbf{w}^* = (\Phi^\top \Phi)^{-1} \Phi^\top \mathbf{y} \tag{2.7}$$

There are many example of choices for basis functions, as

$$\phi_j(x) = \exp\left\{ -\frac{(x - \mu_j)^2}{2s^2} \right\} \tag{2.8}$$

known as *squared exponential*, where μ_j controls the location of the basis function in the *input space*, and s the spatial scale. It's usually referred as 'Gaussian' basis function because of its similarity with the Gaussian distribution function, although there is no probabilistic interpretation here.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Appendix

A Derivations

A.1 Matrix Form

Be the linear model $f(x, \mathbf{w}) = \mathbf{w}^\top \boldsymbol{\phi}(x)$. Suppose $\Phi = [\boldsymbol{\phi}(x_1), \dots, \boldsymbol{\phi}(x_N)]^\top$, then Φ will be of the form

$$\Phi = \begin{bmatrix} \phi_0(x_0) & \dots & \phi_{M-1}(x_0) \\ \vdots & \ddots & \vdots \\ \phi_0(x_{N-1}) & \dots & \phi_{M-1}(x_{N-1}) \end{bmatrix} \tag{A.1}$$

called *design matrix*. Then the model turns to $\mathbf{f} = \Phi\mathbf{w}$. This will lead us to the matrix form for the quadratic error function

$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2}(\mathbf{f} - \mathbf{y})^\top(\mathbf{f} - \mathbf{y}) \\ &= \frac{1}{2}(\Phi\mathbf{w} - \mathbf{y})^\top(\Phi\mathbf{w} - \mathbf{y}) \\ &= \frac{1}{2}(\mathbf{w}^\top\Phi^\top\Phi\mathbf{w} - \mathbf{y}^\top\Phi\mathbf{w} - \mathbf{w}^\top\Phi^\top\mathbf{y} + \mathbf{y}^\top\mathbf{y}) \end{aligned}$$

Observe that even in the matrix form, the error function remains scalar, which implies that $\mathbf{y}^\top\Phi\mathbf{w} = \mathbf{w}^\top\Phi^\top\mathbf{y}$ by the transpose of the product rule. Then

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{w}^\top\Phi^\top\Phi\mathbf{w} - 2\mathbf{y}^\top\Phi\mathbf{w} + \mathbf{y}^\top\mathbf{y})$$

Then we proceed by the minimization by $\frac{\partial E}{\partial \mathbf{w}} = 0$

$$\begin{aligned} 0 &= \frac{1}{2}(2\mathbf{w}^\top\Phi^\top\Phi - 2\mathbf{y}^\top\Phi)^\dagger \\ \mathbf{w}^{*\top} &= \mathbf{y}^\top\Phi(\Phi^\top\Phi)^{-1} \\ \mathbf{w}^* &= (\Phi^\top\Phi)^{-1}\Phi^\top\mathbf{y} \end{aligned} \tag{A.2}$$

B Mauris euismod

[†] Using two facts. First, if $\alpha = \mathbf{x}^\mathrm{T}\mathbf{A}\mathbf{x}$, then $\frac{\partial\alpha}{\partial\mathbf{x}} = 2\mathbf{x}^\top\mathbf{A}$, being α scalar. Second, if $\alpha = \mathbf{y}^\mathrm{T}\mathbf{A}\mathbf{x}$, then $\frac{\partial\alpha}{\partial\mathbf{x}} = \mathbf{y}^\mathrm{T}\mathbf{A}$. For both, $\mathbf{A}$ is independent of $\mathbf{x}$ and $\mathbf{y}$ [Gra83].

References

[Gra83] F.A. Graybill. Matrices with applications in statistics. Wadsworth statistics - probability series. Wadsworth International Group, 1983.