THE UNIVERSITY OF LIVERPOOL

MACHINE LEARNING SERIES

Linear Regression and Gaussian Processes

Adam Lee

in collaboration with
Dr. P.L. GREEN

January 18, 2021

Contents

Introduction

What is Machine Learning?

Simply put, machine-learning is the science of automated data-analysis through computer algorithms. In particular, we can define machine-learning as a set of methods which automatically detect patterns in data (any digitally stored information can be considered data for a machine learning algorithm) and learn from those patterns in order to predict the outcomes of 'unseen' data. In the era of **big data**, efficient and intuitive algorithms are needed to process large data-sets in such a way that predictions are accompanied with low measures of uncertainty. The idea of uncertainty is the first example of how probability can be tied into machine learning. This report aims to summarise the concepts behind **linear regression** and **Gaussian processes**, two famous methods for **supervised** machine learning. There are two main groupings for machine learning methods, the most common of which being the aforementioned supervised learning. Supervised learning is the task of learning a function that maps an input to an output based on given example pairs of input-output data. Typically, we will denote the pairs of inputs and outputs as $\varkappa = \{(x_0, y_0), \ldots, (x_n, y_n)\}$ and refer to this data set as the **training set**. Our focus will be on the machine learning methods for **regression**, and thus our input-output pairs will be of the form $(\mathbf{x}, y)$ where $\mathbf{x} \in \mathbb{R}^D$ is a real-valued, D-dimension input vector and $y \in \mathbb{R}$ is a real-valued, scalar output. However, inputs can take any form, as long as the data can be digitalised. For example, **classification** machine learning algorithms may take PNG files or strings as inputs.

Regression - A probabilistic approach to machine learning

As we mentioned before, we will be taking a probabilistic approach to machine learning. We do so in the form of regression methods. Regression predictive modelling is the task of learning a function, say f, from input variables $\mathbf{x}$ to a continuous output variable y. Our continuous output y is therefore real-valued and often scalar. The most widely known method for regression is linear regression which is the linear approach to finding our function f by modelling the response variable y to one or more input variables $\mathbf{x}_i$. The case of a single input variable $(i = 1)$ is known as simple-linear regression, which we shall consider first. Regression models incorporate probability in the form of uncertainty, we assume our regression model has a degree of random error in its predictions, and the more accurate the model, the less the uncertainty in our predictions. It is common to assume the error is a random variable which is Gaussian distributed about a zero mean and a variance σ_ϵ^2. Gaussian processes provide a method of regression which minimises uncertainty when entropy is low. While linear regression is a parametric approach to machine learning, Gaussian processes are a non-parametric, inherently **Bayesian** method and will open up a wide discussion on optimisation and computational efficiency within machine learning.

1 Background Theory

> *"Life's most important questions are, for the most part, nothing but probability problems."*
>
> **Pierre-Simon Laplace**

1.1 Introduction

This report assumes the reader has some knowledge of the underlying theory we intend to use to generate our regression models, however it is convenient for us to review the key aspects of probability theory and linear algebra which will be commonplace in this report. We will also specify frequent notations which will be used throughout.

1.2 Linear Algebra

It is convenient to introduce some standard notations which shall be followed through the course of this report. Column vectors shall be denoted by lowercase letters (e.g. $\mathbf{x}$) with row vector transposition of $\mathbf{x}$ written $\mathbf{x}^{\mathrm{T}}$. We shall refer to the i-th element of a vector $\mathbf{x}$ as $\mathbf{x}_i$. The *norm of a vector* $\mathbf{x}$ shall be denoted $\|\mathbf{x}\|$. We shall refer to the elementary unit vectors as $\mathbf{e}_i$ which contain all zero entries apart from the i-th entry which is instead one. Matrices shall be denoted by capital letters (e.g. A) and such a matrix A referred to as being $m \times n$ will have m rows and n columns. A matrix A is said to be square if $m = n$. The (i, j) element of a matrix A shall be denoted as (a_{ij}) or simply a_{ij} if the context is clear. When working with a square matrix A we will refer to the determinant of A as either $\det(A)$ or as $|A|$. If $\det(A) = 0$ the matrix A is said to be singular, and otherwise it is non-singular. The $n \times n$ identity matrix will be referred to as I_{n}. In the case of a non-singular, square matrix A we shall refer to its matrix inverse as A^{-1}.

1.2.1 Relevant Special Matrices

This subsection will introduce any special matrices we are likely to come across in this report and outline their properties.

- **The zero matrix.** A matrix whose elements are all zero is called a *zero matrix* and is written 0.
- **Diagonal matrices.** A square matrix $D = (d_{ij})$ is *diagonal* if

  $$i \neq j \Rightarrow d_{ij} = 0.$$

  That is, the matrix D's off-diagonal entries are all 0. For example:

  $$D = \begin{pmatrix} \mathbf{X} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{X} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{X} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{X} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{X} \end{pmatrix}.$$

(*N.B. in the above matrix,* $\mathbf{X}$ *represents an entry that may or may not be* 0*, following the convention of J. H. Wilkinson* [1] *and the notation shall be carried forward through this report*)

- **(Upper) Triangular matrices**. A square matrix $U = (u_{ij})$ is *upper triangular* if

$$i > j \Rightarrow u_{ij} = 0.$$

For example:

$$U = \begin{pmatrix} \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} \\ 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} \\ 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} \\ 0 & 0 & 0 & \mathbf{X} & \mathbf{X} \\ 0 & 0 & 0 & 0 & \mathbf{X} \end{pmatrix}.$$

Similarly, a square matrix $L = (l_{ij})$ is *lower triangular* if

$$i < j \Rightarrow l_{ij} = 0.$$

- **Symmetric matrices.** A square matrix $S = (s_{ij})$ is symmetric if

$$s_{ij} = s_{ji} \ \forall \ (i, j).$$

- **Positive-definite matrices.** A square matrix P is positive-definite if the scalar value $\mathbf{z}^{\mathrm{T}} P \mathbf{z}$ is strictly positive for all non-zero column vectors $\mathbf{z}$.

1.2.2 The Cholesky Decomposition

For a positive-definite matrix A, there exists a decomposition of A such that

$$A = LL^{\mathrm{T}} \tag{1.1}$$

where L is lower triangular. This decomposition is known as the Cholesky decomposition. With this, we lay out our first two theorems which will be useful for the discussions on computational efficiency of Gaussian processes.

Theorem 1. *For any real matrix* $A \in \mathbb{R}^{n \times m}$*, the product matrix* $A^T A$ *is positive definite i.e*

$$\mathbf{z}^T A^T A \mathbf{z} > 0 \ \forall \ \mathbf{z} \in \mathbb{R}^n \tag{1.2}$$

Proof.

$$\begin{aligned} \mathbf{z}^{\mathrm{T}} A^{\mathrm{T}} A \mathbf{z} &= (A\mathbf{z})^{\mathrm{T}} (A\mathbf{z}) \\ &= ||Az||^2 > 0. \end{aligned} \tag{1.3}$$

□

Theorem 2. *For a positive-definite matrix* A*, there exists a lower-triangular matrix* U *such that*

$$A^{-1} = UU^T \tag{1.4}$$

Proof. For positive-definite matrix A, we have a Cholesky decomposition such that

$$A = LL^{\mathrm{T}}, \tag{1.5}$$

then we have

$$AA^{-1} = LL^{\mathrm{T}}A^{-1} = I, \tag{1.6}$$

that is

$$A^{-1} = (L^{\mathrm{T}})^{-1}L^{-1}. \tag{1.7}$$

Since L is lower triangular, if we write $(L^{\mathrm{T}})^{-1} = U$ then we have

$$A^{-1} = UU^{\mathrm{T}}, \tag{1.8}$$

which is a positive-definite symmetric matrix such that we need only compute the upper-triangular elements of UU^{T} in order to know all elements of A^{-1}. □

1.3 Probability Theory

1.3.1 Discrete Random Variables

We use the notation $p(A)$ to denote the probability that event A is true, thus we require that $0 \leq p(A) \leq 1$. We therefore define $p(\bar{A}) = 1 - p(A)$ to be the probability of not A. We now define discrete random variables (**drv** or **rv**). A random variable is a measurable function $X : \Omega \to \chi$ from a set of outcomes Ω to a measurable space χ. The discreteness of a random variable indicates it has a countable number of possible values, as opposed to a continuous random variable (**crv**). We denote the probability that $X = x$ as $p(X = x)$ or simply $p(x)$ if the context is clear. Here, $p()$ is known as the probability mass function (**pmf**).

1.3.2 Fundamental Rules

1. Given two events A and B, we define the joint probability as

$$p(A, B) = p(A \wedge B) = p(A|B)p(B). \tag{1.9}$$

2. Given two events A and B, we define the union probability as

$$\begin{aligned} p(A \vee B) &= p(A) + p(B) - p(A \wedge B) & (1.10)\\ &= p(A) + p(B) \text{ if } A \text{ and } B \text{ are mutually exclusive.} & (1.11) \end{aligned}$$

3. Given a joint probability $p(A \wedge B)$, we define the marginal distribution as

$$p(A) = \sum_b p(A, B) = \sum_b p(A|B)p(B = b). \tag{1.12}$$

4. Given two events A and B, we define the conditional probability as

$$p(A|B) = \frac{p(A, B)}{p(B)} \text{ iff } p(B) > 0. \tag{1.13}$$

5. For two events X and Y we define **Bayes' Rule** as

$$p(X = x|Y = y) = \frac{p(X = x, Y = y)}{p(Y = y)} = \frac{p(X = x)p(Y = y|X = x)}{\sum_{x'} p(X = x')p(Y = y|X = x')} \tag{1.14}$$

1.3.3 Continuous Random Variables

Suppose X is some uncertain continuous quantity and we wish to compute the probability that X lies in some interval $a < X < b$. We define the events $A = (X \leq a), B = (X \leq b), W = (a < X \leq b)$ then

$$p(W) = p(B) - p(A) \tag{1.15}$$

by the sum rule. Define the function $F(x) \triangleq p(X \leq x)$, we call this the cumulative distribution function (**cdf**) of X. This is a monotonically non-decreasing function. Then we have

$$p(a < X \leq b) = F(b) - F(a). \tag{1.16}$$

Now, define $f(x) = F'(x)$ which is the probability density function (**pdf**), given such a function we have

$$p(a < X \leq b) = \int_a^b f(x)dx. \tag{1.17}$$

A point of interest therefore is that

$$\lim_{da \to 0} p(a < X \leq a + da) = \lim_{da \to 0} \int_a^{a+da} f(x)dx = p(X = a) = 0^{*}, \tag{1.18}$$

so we instead write $p(a < X \leq a + da) \approx p(a)$ for small da.

1.3.4 Mean and Variance

Two important quantities associated with a distribution is its **mean** mu, or **expected value**, and **variance** σ^2. For drvs we define the mean as $\mathbb{E} \triangleq \sum_{x \in \mathcal{X}} x\ p(x)$. For crvs, the mean is defined as $\mathbb{E} \triangleq \int_{\mathcal{X}} x\ p(x)dx$. The variance of a distribution is the measure of how spread our data is from the mean and is defined as

$$\begin{aligned} \mathrm{var}(X) \triangleq \mathbb{E}\left[(X-\mu)^2\right] &= \int_{\mathcal{X}} (x-\mu)^2\ p(x)dx & (1.19) \\ &= \mathbb{E}[X^2] - \mu^2 & (1.20) \end{aligned}$$

This gives way to the useful expression[†] $\mathbb{E}[X^2] = \mu^2 + \sigma^2$. The standard deviation of a distribution is

$$\mathrm{sd}(X) = \sqrt{\mathrm{var}(X)}, \tag{1.21}$$

which has the same units as X itself.

1.3.5 The Binomial Distribution

A common example for discrete distributions is the **binomial distribution** which is written $X \sim \mathrm{Bin}(n, \theta)$ where $X \in \{0, \ldots, n\}$ with pmf

$$\mathrm{Bin}(k|n,\theta) = \binom{n}{k} \theta^k (1-\theta)^{n-k} \tag{1.22}$$

where $\binom{n}{k} = \frac{n!}{(n-k)!k!}$. This distribution has mean $\mu = n\theta$ and variance $\sigma^2 = n\theta(1-\theta)$.

[*] would the same hold for Dirac delta function?

[†] Easily derived

1.3.6 Gaussian Distribution

The most famous example, and the focus of our probabilistic approach to machine learning, is the Gaussian distribution. We say a continuous random variable X is Gaussian distributed if $X \sim \mathcal{N}(\mu, \sigma^2)$ and has pdf

$$\mathcal{N}(x|\mu, \sigma^2) \triangleq \frac{\lambda}{\sqrt{\pi}} \exp^{-\lambda^2(x-\mu)^2}, \qquad \lambda = \frac{1}{\sqrt{2\sigma^2}} \tag{1.23}$$

Here, $\mu = \mathbb{E}[X]$ and $\mathrm{var}[X] = \sigma^2$ and $\frac{\lambda}{\sqrt{\pi}}$ is the normalisation constant which allows our distribution to integrate to one. Our notation $X \sim \mathcal{N}(\mu, \sigma^2)$ is equivalent to $p(X = x) = \mathcal{N}(x|\mu, \sigma^2)$.

The cdf for a Gaussian distribution is given by

$$\Phi(x; \mu, \sigma^2) \triangleq \int_{-\infty}^{x} \mathcal{N}(z|\mu, \sigma^2)dz = \frac{1}{2}\left[1 + \mathrm{erf}(\lambda(x - \mu))\right]. \tag{1.24}$$

We plot various pdfs and cdfs for Gaussian distribution in fig. 1 to demonstrate the shape of the a Gaussian.

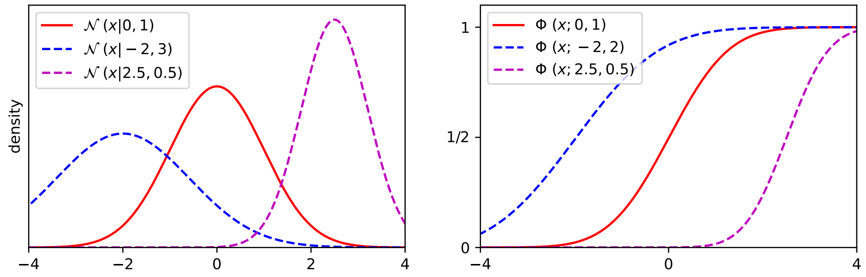


Figure 1: Left: the pdf for various Gaussian distributions. Right: the cdf for various Gaussian distribution

We can generalise the one-dimension (univariate) Gaussian distribution to higher dimension-space. We call this multivariate or joint Gaussian distribution on a continuous random vector $\mathbf{X} = [X_1, X_2, \ldots, X_k]^{\mathrm{T}} \in \mathbb{R}^k$.

1.3.7 Multivariate Gaussian Distribution

We say a continuous random variable $\mathbf{X} \in \mathbb{R}^k$ is jointly Gaussian if

$$\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma),$$

where $\boldsymbol{\mu} = [\mathbb{E}(X_1), \ldots, \mathbb{E}(X_k)]$ is a mean vector and $\Sigma = \Sigma_{ij} = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)]$ is a covariance matrix. The pdf of a k-dimension multivariate Gaussian distribution given $\boldsymbol{\mu}$ and Σ is as follows

$$\mathcal{N}(\mathbf{X}|\boldsymbol{\mu}, \Sigma) \triangleq \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu})^{\mathrm{T}}\Sigma^{-1}(\mathbf{X} - \boldsymbol{\mu})\right]. \tag{1.25}$$

Clearly, this representation is only viable if Σ is non-singular, however this is guaranteed by the positive-definiteness of the covariance matrix. The definition of joint Gaussian distribution holds for arbitrarily large k, however it is useful to observe the case for $k = 2$, or the bivariate Gaussian distribution.

Example 1.1: Suppose we have

$$\mathbf{X} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}\right), \tag{1.26}$$

then we can plot the pdf for this distribution 2.

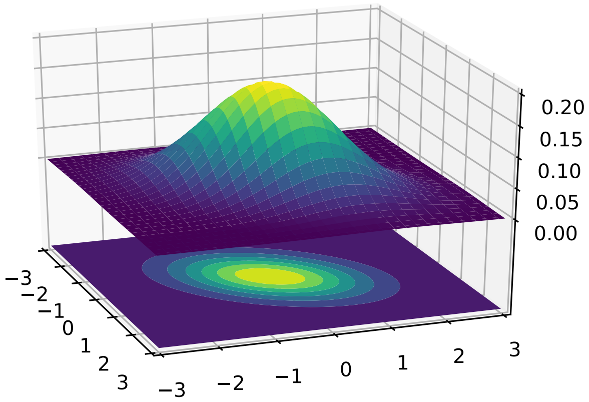


Figure 2: The pdf for $\mathbf{X}$ with accompanied contour plot

1.3.8 Conditioning a Multivariate Gaussian

Theorem 3. *Suppose now that a k-dimension continuous random vector $\mathbf{X}$ is jointly Gaussian with mean $\boldsymbol{\mu}$ and covariance Σ and can be partitioned as follows*

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}, \ \mathbf{x}_1 \in \mathbb{R}^q, \ \mathbf{x}_2 \in \mathbb{R}^{k-q}. \tag{1.27}$$

Then, we have the corresponding partitions for the mean and variance

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}, \tag{1.28}$$

where $\Sigma_{11} \in \mathbb{R}^{q\times q}$, $\Sigma_{21} = \Sigma_{12}^T \in \mathbb{R}^{(k-q)\times q}$, $\Sigma_{22} \in \mathbb{R}^{(k-q)\times(k-q)}$. *Suppose that we observe values for* $\mathbf{x}_2$, *then the conditional posterior distribution is*

$$\boxed{\begin{aligned}&\mathbf{x}_1|\mathbf{x}_2 \sim \mathcal{N}(\boldsymbol{\mu}^*, \Sigma^*),\\ &\quad \boldsymbol{\mu}^* = \boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2),\\ &\quad \Sigma^* = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}.\end{aligned}} \tag{1.29}$$

Theorem 3 is of critical importance for our purposes and we will revisit it and its implications when we begin discussing Gaussian Processes (GPs). We can observe bivariate conditional Gaussian distribution using example 1.1.

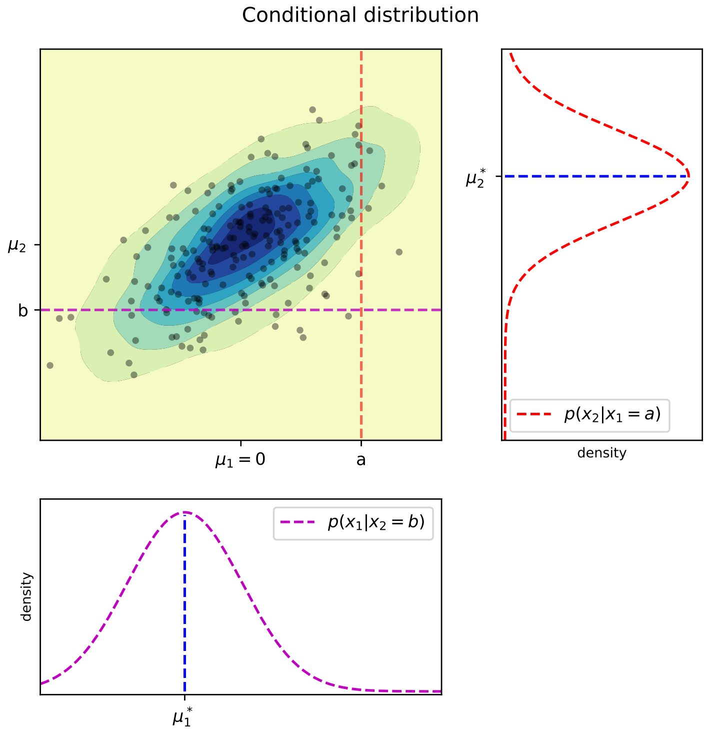


Figure 3: The conditional probabilities for a Gaussian distribution centred at (0,0) for arbitrary values a, b.

Part I
Linear Regression

2 Introduction

> *"It is not knowledge, but the act of learning, not possession but the act of getting there, which grants the greatest enjoyment."*
>
> **Carl Friedrich Gauss**

Linear regression is perhaps the most widely known regression model; here we assume a response variable y is a function of input variables $\mathbf{x}$ (or in extension a function of functions of input variables, however we will consider this later) which we can write

$$y = f(\mathbf{x}) + \varepsilon, \quad f(\mathbf{x}) = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}. \tag{2.1}$$

Here $\boldsymbol{\beta} = [\beta_0, \ldots, \beta_D] \in \mathbb{R}^{D+1}$ is a vector of weights attached to each regressor $x_i \in \mathbf{x} = [x_0, x_1, x_2, \ldots, x_D] \in \mathbb{R}^{D+1}$. This notation fails to assert that, in most cases, $x_0 = 1$ such that β_0 becomes the 'y-intercept' of the model. The model incorporates an error term ε which we often assume is Gaussian distributed such that $\varepsilon \sim \mathcal{N}(\mu, \sigma^2)$. To make the connection to probability more explicit, the same model we have defined can be rewritten

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\mu(\mathbf{x}), \sigma^2(\mathbf{x})). \tag{2.2}$$

In the form we shall examine, we take $\mu = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}$ and $\sigma^2(\mathbf{x}) = \sigma^2$ to be constant. Then, $\boldsymbol{\theta}$ becomes our vector of parameters, namely $\boldsymbol{\theta} = \left[\boldsymbol{\beta}, \sigma^2\right]^{\mathrm{T}}$.

Our model can be extended to incorporate **basis functions** $\phi_i(\mathbf{x})$ in place of just the input variables in order to model non-linear relationships between the regressors $\mathbf{x}$ and the response $\mathbf{y}$. For example, given a single regressor $x \in \mathbb{R}$ and response variable $y \in \mathbb{R}$, we can define our model

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon. \tag{2.3}$$

Here, $\boldsymbol{\beta} = [\beta_0, \beta_1, \beta_2]^{\mathrm{T}}$ are our unknown parameters and our basis functions are of the form $\boldsymbol{\Phi}^{\mathrm{T}} = [\phi_1(x), \phi_2(x), \phi_3(x)] = \left[1, x, x^2\right]$.

These extended models are still linear in the parameters $\boldsymbol{\beta}$ so is still considered linear regression and the mathematical computations remain consistent regardless of whether we use basis functions or just our input vectors $\mathbf{x}$. We will first consider the case in which $\boldsymbol{\Phi} = [1, x]$ to introduce the method of least squares estimation for determining the parameters before extending our analysis to the general case for multiple linear regression.

3 Simple Linear Regression

Simple linear regression is a linear regression model in which our response variable y is presumed to be dependent on a single regressor x such that our model becomes

$$y = \beta_0 + \beta_1 x + \varepsilon. \tag{3.1}$$

The practical interpretation of our parameters $\boldsymbol{\beta}$ is simple, β_0 is the 'y-intercept' of the model and β_1 determines our expected increase in y for a unit increase in x. Finally, ε, our error term, represents the uncertainty in our model to exactly fit the data, which we assume to be constant and Gaussian distributed with a 0 mean and variance σ^2. Then, our expected value for y is

$$\mathbb{E}(y|x) = \beta_0 + \beta_1 x = \mu_{y|x}, \tag{3.2}$$

with variance σ^2. Now, let's consdier the method of least-squares for the determination of our parameters β_0 and β_1.

Assume we have a set of pairs of observed values of our model

$$\varkappa = \{(x_1, y_1), \dots, (x_n, y_n)\}. \tag{3.3}$$

We can input this data into our model to obtain a system of equations

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_1, \quad i = 1, \dots, n. \tag{3.4}$$

Let us then definte the least-squares criterion function

$$\mathrm{LS}(\beta_0, \beta_1) \triangleq \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x)^2. \tag{3.5}$$

Let us further introduce real constants $\hat{\beta}_0, \hat{\beta}_1$ which satisfy the minima conditions

$$\begin{aligned}\frac{\partial \mathrm{LS}}{\partial \beta_0}\Big|_{\hat{\beta}_0, \hat{\beta}_1} &= -2 \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = 0 \\ \frac{\partial \mathrm{LS}}{\partial \beta_1}\Big|_{\hat{\beta}_0, \hat{\beta}_1} &= -2 \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) x_i = 0\end{aligned} \tag{3.6}$$

We can simplify eq. (3.6) to gain the least squares normal equations

$$\begin{aligned}(n)\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} x_i &= \sum_{i=1}^{n} y_i, \\ \hat{\beta}_0 \sum_{i=1}^{n} x_i + \hat{\beta}_1 \sum_{i=1}^{n} (x_i)^2 &= \sum_{i=1}^{n} y_i x_i.\end{aligned} \tag{3.7}$$

We can solve the first of the pair of equations for $\hat{\beta}_0$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}, \tag{3.8}$$

where $\bar{a} := \frac{1}{n} \sum_{i=1}^{n} a_i$ is the average value of all data points a_i. We now solve the second equation by substituting in eq. (3.8) as follows

$$(\bar{y} - \hat{\beta}_1 \bar{x}) \sum_{i=1}^{n} x_i + \hat{\beta}_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i, \tag{3.9}$$

and then solving with respect to $\hat{\beta}_1$ we obtain

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i - \frac{\sum_{i=1}^{n} x_i y_i}{n}}{\sum_{i=1}^{n} x_i^2 - \frac{\sum_{i=1}^{n} x_i^2}{n}}. \tag{3.10}$$

The numerator of eq. (3.10) is nothing other than the corrected sum of dot products between x_i and y_i, while the denominator is the corrected sum of squares for x_i and thus we can introduce a concise notation for $\hat{\beta}_1$, namely

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} y_i(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{S_{xy}}{S_{xx}}. \tag{3.11}$$

Then, our fitted simple linear regression model is nothing else than

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x. \tag{3.12}$$

Example 3.1: Constructing an example for the method of least squares is quite simple. Consider the data-set fig. 4, where each y_i was sampled from a normal distribution centred on x_i with uniform variance $\sigma^2 = 2$

Observation, i	x_i	y_i
1	1	0.5678
2	2	1.3519
3	3	3.1623
4	4	6.7229
5	5	5.0327
6	6	5.9015
7	7	7.1302
8	8	6.7212
9	9	9.6973
10	10	9.3017
11	11	10.7025
12	12	12.0149
13	13	12.3884
14	14	16.5306

Figure 4: Example data set

The general trend for this data is clear, and from the distribution we used to generate the samples y_i we would expect the parameter values to be $\hat{\beta}_0 \approx 0$ and $\hat{\beta}_1 \approx 1$. Quick calculation concludes that we have

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} = \frac{292.5196}{280.00} \approx 1.0447. \tag{3.13}$$

Then we have

$$\begin{aligned} \hat{\beta}_0 &= \bar{y} - \beta_1 \bar{x} \\ &= 8.2118 - 1.0447 \cdot 8 \approx -0.1459. \end{aligned} \tag{3.14}$$

We have our two parameters $\hat{\beta}_0, \hat{\beta}_1$ and can fit our regression line $\hat{y}$ accordingly. We refer back to our expectations for these values, and conclude this would be an appropriate regression line for this simple case.

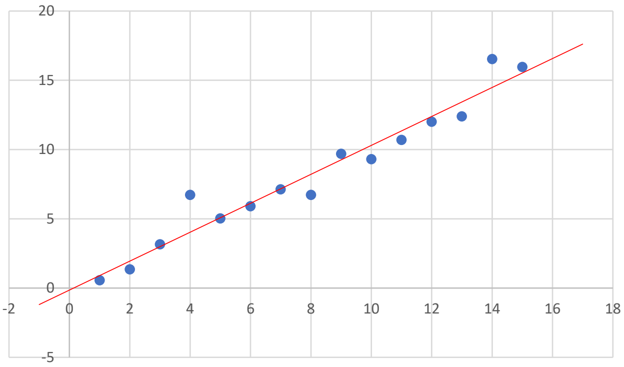


Figure 5: The scatterplot for fig. 4 with regression line $\hat{y}$ in red.

This introductory section to linear regression has outlined the principles of determining the parameters of a linear model using least-squares (Maximum Likelihood estimation) for the the most simple model with one regressor x. Let's now extend our analysis to cover models with multiple regressors x_i and generalised basis functions $\phi_i(\cdot)$.

4 Multiple Linear Regression

4.1 Model Specification

As we discussed in section 2, our multiple linear regression model takes the form

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\boldsymbol{\beta}^{\mathrm{T}}\mathbf{\Phi}(\mathbf{x}), \sigma^2), \tag{4.1}$$

where y is our response variable, $\mathbf{x}$ is our input vector, $\boldsymbol{\beta}$ is our vector of unknown parameters and $\mathbf{\Phi}$ is our vector of basis functions. Simple examples of $\mathbf{\Phi}$ include

$$\mathbf{\Phi}(x) = \left[1, x, \ldots, x^d\right], \tag{4.2}$$

and

$$\mathbf{\Phi} = \{\phi_i\}_{i=1}^d, \qquad \phi_i(x) = \exp(-\lambda^2(x - \mu_i)^2/l). \tag{4.3}$$

Using the basis function in eq. (4.2) is known as polynomial regression, of which example 3.1 covers where $\mathbf{\Phi} = [1, x]$. The basis function in eq. (4.3) is known as the radial basis function where $\boldsymbol{\mu} = \{\mu_i\}_{i=1}^d$ is a vector of centres for each function ϕ_i. An example of a basis function expansion where our input is a multi-dimension vector may be the model defined by the equation

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 + \varepsilon. \tag{4.4}$$

This model assumes the possibility of total variable interaction. We provide an example of a fully interactive model in fig. 6 which plots the model with expected response value $\mathbb{E}(y|x_1, x_2) = x_1 - x_2 - 0.2x_1^2 + 0.8x_2^2 + 0.1x_1x_2$.

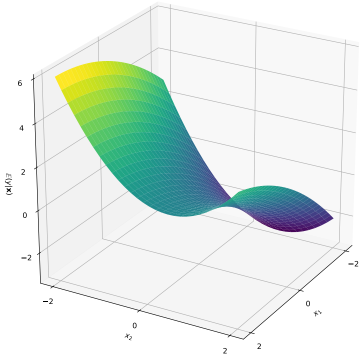


Figure 6: The expected value of a regressor y for a fully interactive model with input vector $\mathbf{x} = [x_1, x_2]$

4.2 Maximum Likelihood Estimation/Least Squares

The **maximum likelihood estimate** (MLE) of the parameters $\boldsymbol{\theta}$ for a statistical model is given by

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \log(\mathbf{y}|\boldsymbol{\theta}), \tag{4.5}$$

where $\mathbf{y} = [y_1, \ldots, y_n]$ is the vector of observed response variable y values. We can write the *log-likelihood* as

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log(y_i|\mathbf{x}_i, \boldsymbol{\theta}). \tag{4.6}$$

Maximising the log-likelihood is exactly equivalent to minimising the negative log-likelihood or **NLL**:

$$\mathrm{NLL}(\boldsymbol{\theta}) = -\sum_{i=1}^{n} \log(y_i|\mathbf{x}_i, \boldsymbol{\theta}). \tag{4.7}$$

If we apply the MLE to linear regression using our model specification form eq. (4.1), then we obtain the following

$$\begin{aligned} \mathrm{NLL}_{LR}(\boldsymbol{\theta}) &= -\sum_{i=1}^{n} \log\left[\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}(y_i - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i)^2\right)\right] && (4.8) \\ &= \frac{1}{2\sigma^2}\mathrm{SSE}(\boldsymbol{\beta}) + \frac{n}{2}\log(2\pi\sigma^2), && (4.9) \end{aligned}$$

where

$$\mathrm{SSE} \triangleq \sum_{i=1}^{n}(y_i - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x})^2,$$

is the **sum of squared errors** for our data. Our objective is to minimise NLL, which is equivalent to minimising SSE, therefore this is the least squares approach to estimating the parameters of our model.

4.2.1 Method of Least Squares

Let us define the multivariate least-squares criterion function

$$\mathrm{MLS}(\boldsymbol{\beta}) \triangleq \sum_{i=}^{n} \varepsilon_i^2 = \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon} = (\mathbf{y} - X\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y} - X\boldsymbol{\beta}), \tag{4.10}$$

where we define the matrix $X \in \mathbb{R}^{n \times k}$ as

$$X = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_k(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \cdots & \phi_k(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_n) & \phi_1(x_n) & \cdots & \phi_k(x_n) \end{bmatrix}. \tag{4.11}$$

Expanding the quadratic MLS we obtain

$$MLS(\boldsymbol{\beta}) = \mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\boldsymbol{\beta}^{\mathrm{T}}X^{\mathrm{T}}\mathbf{y} + \boldsymbol{\beta}^{\mathrm{T}}X^{\mathrm{T}}X\boldsymbol{\beta}, \tag{4.12}$$

then we search for $\hat{\boldsymbol{\beta}}$ such that

$$\left.\frac{\partial \mathbf{E}}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}} = -2X^{\mathrm{T}}\mathbf{y} + 2X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} = 0. \tag{4.13}$$

The solution to which is

$$\boxed{\hat{\boldsymbol{\beta}} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\mathbf{y},} \tag{4.14}$$

where $X^{\mathrm{T}}X$ is the *positive-definite* sum of squares matrix and is always invertible using Cholesky arithmetic 2 should it be of full rank.

Once the parameter estimates $\hat{\boldsymbol{\beta}}$ have been determined, predictions $\mathbf{y}^\star$ for response values of new data points $\mathbf{x}^\star$ can be computed with

$$\mathbf{y}^\star = \hat{X}\hat{\boldsymbol{\beta}} = \hat{X}(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\mathbf{y}, \tag{4.15}$$

where $\hat{X}$ is the corresponding matrix X for new inputs $\mathbf{x}^\star$.

Example 4.1 (Polynomial Regression)**:** Polynomial regression uses basis functions

$$\boldsymbol{\Phi} = \begin{bmatrix} 1 & x & x^2 & \cdots & x^k \end{bmatrix}^{\mathrm{T}}, \tag{4.16}$$

then our matrix X would become

$$X = \begin{pmatrix} 1 & x_0 & \cdots & x_0^k \\ 1 & x_1 & \cdots & x_1^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^k \end{pmatrix}. \tag{4.17}$$

Restricting our basis functions to $\mathbf{\Phi} = [1 \quad x]^{\mathrm{T}}$ gives us simple linear regression. In fig. 7, we generated 100 data-points $y_i, i = 1, \dots, 100$ of the form

$$y_i = f(x_i) + \varepsilon, \qquad f(x) = 2 + x + x^2 - \frac{1}{100}x^3, \qquad \varepsilon \sim \mathcal{N}(0, 1000), \tag{4.18}$$

which are plotted in blue. The red curve is our regression line computed using MLE fitting a model of the form

$$y = \beta_0 + \beta_1\phi_1(x) + \beta_2\phi_2(x) + \beta_3\phi_3(x), \quad \phi_i(x) = x^i, \tag{4.19}$$

to our observations.

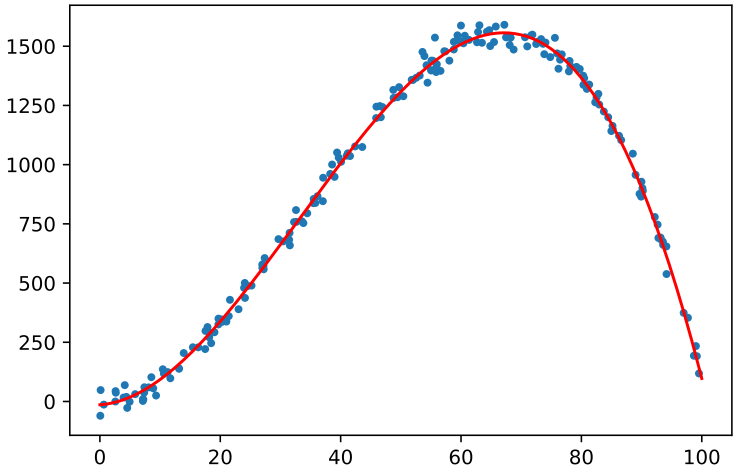


Figure 7: Example of polynomial regression

Example 4.2 (Gaussian Radial Basis Functions)**:** Let us now consider Gaussian radial basis functions

$$\phi_i(x) = e^{-\lambda^2(x-\mu_i)^2} \tag{4.20}$$

where λ is some coefficient representing the variance of our basis functions and $\boldsymbol{\mu} = \begin{bmatrix} \mu_1 & \cdots & \mu_n \end{bmatrix}^{\mathrm{T}}$ are centres for each Gaussian. Then our predictive function becomes a linear combination of weighted Gaussians where our parameters λ and μ_i determine the smoothness of this function on our test interval. There arises the problem of under or over-fitting our data should we include poor hyper-parameters or attempt to fit too many/few Gaussians to our data.

Consider **??**, here our data was generated in a similar fashion to fig. 8. Our generating function in this case was

$$y_i = f(x_i) + \varepsilon, \qquad f(x) = x\sin(0.5x), \qquad \varepsilon \sim \mathcal{N}(0, 1). \tag{4.21}$$

Our basis functions were of the form

$$\phi_i = e^{-0.5(x-i)^2}, \quad i = -10, \dots, 10, \tag{4.22}$$

that is, we fit a RBF centred at each integer within the domain of our data.

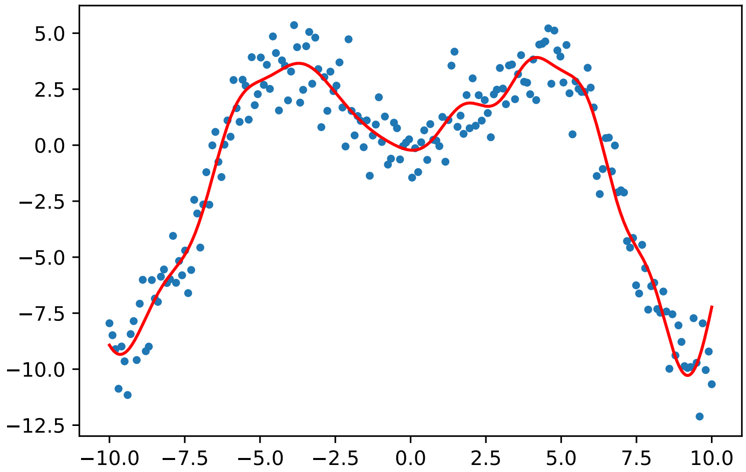


Figure 8: Example of linear regression using Gaussian radial basis functions

5 Bayesian Linear Regression

5.1 Introduction

Our standard linear regression model is a frequentist approach to machine learning, and assumes data is generated from the model

$$y = f(\mathbf{x}) + \varepsilon, \quad f(\mathbf{x}) = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}. \tag{5.1}$$

where the only uncertainty is represented in the Gaussian distributed error term ε. In contrast, Bayesian linear regression assumes the responses are sampled from a Gaussian distribution

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\beta}^{\mathrm{T}}X, \sigma^2 I). \tag{5.2}$$

We continue our analysis taking the Bayesian approach to better interpret uncertainty around the given data and furthermore predictive uncertainty.

5.2 Computing the Posterior

As we have mentioned, the likelihood for the standard linear regression model is given by

$$p(\mathbf{y}|X, \boldsymbol{\beta}) = \mathcal{N}(y|\boldsymbol{\beta}^{\mathrm{T}}X, \sigma^2 I). \tag{5.3}$$

For a Bayesian approach, we need to specify a prior over the parameters which incorporates our beliefs about $\boldsymbol{\beta}$ before having seen the data. We put a zero-mean Gaussian with covariance Σ_p on the parameters such that

$$\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \Sigma_p). \tag{5.4}$$

Then, we can use Bayes' rule to determine an expression for a posterior on the parameters as follows

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}} \quad \Rightarrow \quad p(\boldsymbol{\beta}|\mathbf{y}, X) = \frac{p(\mathbf{y}|\boldsymbol{\beta}^{\mathrm{T}}X, \sigma^2 I)p(\boldsymbol{\beta})}{p(\mathbf{y}|X)}. \tag{5.5}$$

The normalizing constant is independent of the weights, so let us consider the terms of the posterior which are dependent on the weights and complete the square

$$\begin{aligned} p(\boldsymbol{\beta}|\mathbf{y}, X) &\propto \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y} - \boldsymbol{\beta}^{\mathrm{T}}X)^{\mathrm{T}}(y - \boldsymbol{\beta}^{\mathrm{T}}X)\right)\exp\left(-\frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\Sigma_p^{-1}\boldsymbol{\beta}\right) && (5.6) \\ &\propto \exp\left(-\frac{1}{2}(\boldsymbol{\beta} - \bar{\boldsymbol{\beta}})(\frac{1}{\sigma^2}XX^{\mathrm{T}} + \Sigma_p^{-1})(\boldsymbol{\beta} - \bar{\boldsymbol{\beta}})^{\mathrm{T}})\right), && (5.7) \end{aligned}$$

where $\boldsymbol{\beta}^{\mathrm{T}} = \frac{1}{\sigma^2}\left(\sigma^{-2}XX^{\mathrm{T}} + \Sigma_p^{-1}\right)^{-1}X\mathbf{y}$. We recognise this as nothing other than a Gaussian

$$\boxed{p(\boldsymbol{\beta}|\mathbf{y}, X) \sim \mathcal{N}(\sigma^{-2}A^{-1}X\mathbf{y}, A^{-1}), \quad A = \sigma^{-2}XX^{\mathrm{T}} + \Sigma_p^{-1}.} \tag{5.8}$$

5.3 Prediction using the Posterior

Often, our concern is with using distribution models to predict the value of the response variable at a given point $\mathbf{x}_*$. Within the Bayesian outlook, our task for prediction is to average over all possible parameter values, weighted by the probability we calculated for the posterior 5.8. If we denote the predictive distribution as y_* corresponding to $\mathbf{x}^*$ then we have

$$\begin{aligned} p(y_*|\mathbf{x}^*, X, \mathbf{y}) &= \int p(y_*|\mathbf{x}_*, \boldsymbol{\beta})p(\boldsymbol{\beta}|X, \mathbf{y})d\boldsymbol{\beta} && (5.9) \\ &= \mathcal{N}(\frac{1}{\sigma^2}\mathbf{x}_*^{\mathrm{T}}A^{-1}X\mathbf{y}, \mathbf{x}_*^{\mathrm{T}}A^{-1}\mathbf{x}_*). && (5.10) \end{aligned}$$

5.4 Conclusion

Bayesian linear regression provides deeper uncertainty quantification for regression problems by placing a prior on the parameters of the model and updating the likelihood of our model with our prior beliefs. The imposition of further probability measures on linear regression makes the Bayesian outlook a powerful regression tool.

Part II

Gaussian Processes

6 Introduction

Bayesian linear regression allows for greater quantification of our uncertainty of the underlying distribution of a response variable $y(\mathbf{x})$, however still has a focus on parametrizing y with respect to $\mathbf{x}$ and the vector of parameters $\boldsymbol{\beta}$ such that

$$y = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x} + \varepsilon, \tag{6.1}$$

where ε is some noise. Suppose we wish to bypass the parametrization of y and instead approach regression non-parametrically. Gaussian processes, or just GPs, allow us to do this by placing a prior on the underlying function which can be updated as we observe data. We can consider GPs to be an extension of the multivariate Gaussian distribution over an infinite number of variables, and much like a Gaussian our GP prior consists of a mean function $m(\mathbf{x})$ (often taken to be 0) and a kernel function $\kappa(\mathbf{x}, \mathbf{x}')$. If an underlying function f has a GP prior with mean m and kernel κ we assume the notation

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}')). \tag{6.2}$$

In order to perform GP regression, we require only these two functions. The mean function provides little insight into our prior understanding of underlying function, therefore we place a significant degree of importance onto the co-variance function. The covariance function, also referred to as the kernel function, describes our beliefs regarding the smoothness, periodicity and shape of the function we wish to uncover and two different kernels can yield wildly different predictive distributions for some given new point $\mathbf{x}_*$.

7 Gaussian Process Prior

In this section, we discuss the GP prior and its implications. We begin by providing a formal definition for GPs

Definition 1 (Gaussian Processes)**.** A potentially infinite collection of random-variables $\varkappa$ describes a Gaussian Process *iff* any finite collection of elements $\mathcal{S} \subset \varkappa$ describe a Gaussian random variable, *i.e.*

$$\mathcal{S} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathcal{S}}, \Sigma_{\mathcal{S}}), \tag{7.1}$$

where $(\boldsymbol{\mu}_{\mathcal{S}})_i = \mathbb{E}((\mathcal{S})_i)$ is the mean of the RV and $(\Sigma_{\mathcal{S}})_{ij} = \kappa\left((\mathcal{S})_i, (\mathcal{S})_j\right)$ is a symmetric positive semi-definite co-variance matrix generated by the kernel function κ.

We note again that the mean function m is often taken to be 0, however this is not a requirement and we will later consider the case where $m(\mathbf{x}) \neq 0$. Let us briefly discuss the kernel function for GP regression $\kappa(\mathbf{x}, \mathbf{x}')$. The kernel function describes our prior beliefs regarding the smoothness, periodicity and general shape of the underlying distributive function f. The kernel is used to generate the covariance functions for the subsets of data observed as explained in our definition for GPs 1, and for the time being we will use the most popular kernel function, the Squared Exponential or SE function. The SE function is defined as follows

$$\kappa_{\scriptscriptstyle SE}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{1}{2l^2}(\mathbf{x} - \mathbf{x}')^{\mathrm{T}}(\mathbf{x} - \mathbf{x}')\right), \tag{7.2}$$

where σ^2 controls the vertical scale of the function and l is a horizontal length scale which controls the smoothness of the function. The SE kernel is infinitely differentiable therefore in general results in a distribution over smooth curves when used for the GP prior, however we can alter the smoothness of our sample functions by changing the hyper-parameter l. We present three values for l in fig. 9, and display a corresponding function drawn from a GP with a SE kernel prior equipped with such an l. It can be shown [2] that the SE kernel can be obtained by extending linear regression to incorporate Gaussian radial basis functions centred at infinitely many points on a given domain.

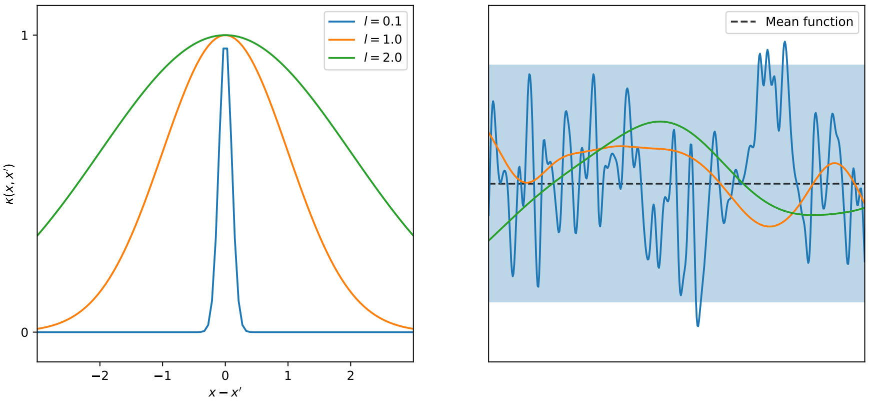


Figure 9: Left: plots of the SE kernel equipped with various values of the characteristic length scale l. Right: functions for the corresponding SE kernels drawn using multivariate Gaussian distribution

While the SE kernel is certainly the most popular with regards to GPs, there are many more well-known kernels commonly used which reflect the beliefs regarding an underlying function's periodicity for example. For now, we will continue to use the SE kernel.

8 Prediction with GP Priors

Simply drawing functions from a GP prior is not of much interest aside from comparing kernels and their hyperparameters. Instead, we are concerned with making predictions for unseen points using the GP prior and a set of observed data, forming a posterior distribution for the underlying function. Initially, we will consider the case where we assume our observations are noise free. That is, assume we have a set of observed data $\{(\mathbf{x}_i, y_i) | i = 1, \ldots, n\}$ and we wish to predict the values $\mathbf{y}_* \in \mathbb{R}^m$ of the underlying function at a set of test points

X_*. Then, the joint distribution of the training outputs $\mathbf{y}$ and the test outputs $\mathbf{y}_*$ is

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*), \end{bmatrix}\right) \tag{8.1}$$

where $K(X, X_*) \in \mathbb{R}^{n\times m}$ is the covariance matrix generated by the kernel function applied to all ordered pairs of the set of observed data X and the test data X_*, with similar definitions for $K(X, X), K(X_*, X)$ and $K(X_*, X_*)$. To determine a posterior distribution over the functions which agree with the observed data, we need now only perform conditioning 1.3.8 on the multivariate Gaussian in eq. (8.1) given X, X_* and $\mathbf{y}$

$$\boxed{\mathbf{y}_*|X, X_*, \mathbf{y} \sim \mathcal{N}\left(\boldsymbol{\mu}_*, \Sigma_*\right),} \tag{8.2}$$

where

$$\boxed{\begin{aligned} \boldsymbol{\mu}_* &= K(X_*, X)K(X, X)^{-1}\mathbf{y} && (8.3) \\ \Sigma_* &= K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*). && (8.4) \end{aligned}}$$

The closed-form solution to this prediction problem is a remarkable property of GPs, providing not only a mean prediction of the underlying function our data is drawn from but also quantifying uncertainty regarding this prediction in the form of the posterior covariance matrix Σ_*. However, we see immediately the computational bottle-neck for GP prediction in the inversion of an $n \times n$ SPSD matrix, or at the very least the solution to the linear system $K(X, X)^{-1}\mathbf{y}$. Even by taking full advantage of the SPSD nature of the Gram matrix by performing the Cholesky decomposition $K = LL^{\mathrm{T}}$, inverting this $n \times n$ matrix has a computational cost $\mathcal{O}(n^3)$ making it practically inaccessible for data-set sizes of order larger than 10^4. We will later introduce methods for combating this bottle-neck through low-rank approximations and sparse alternatives to the matrix K which can reduce the computational cost of GP regression at the expense of predictive accuracy and assurance of the uncertainty. First, let us examine the standard method kernel function and associated matrices and how model selection can drastically improve GP accuracy and measures for quantifying a GPs usefulness.

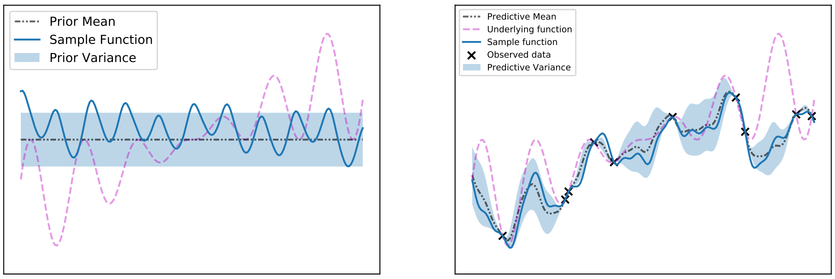


Figure 10: An example of conditioning a GP prior after observing data from an underlying function.

8.1 Noisy Data GPs

The procedure described in 8 assumes our observed data is noiseless however we can easily implement a model which assumes our data was observed with some random Gaussian noise i.e. $y(\mathbf{x}_i) = f(\mathbf{x_i}) + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$. Then we instead consider the Gram matrix

$$K_\varepsilon(X, X) = K(X, X) + \sigma_\varepsilon^2 I$$

and perform the same analysis as before but with this new matrix K_ε in place of K. The computational cost of noisy GP prediction is still $\mathcal{O}(n^3)$.

8.2 Shorthand notation

We will proceed with some shorthand notation for the general form Gram matrices, which we will introduce now. We will refer to the Gram matrix $K(X, X) \in \mathbb{R}^{n\times n}$ as K_{nn} or simply as K if the context is clear. Therefore, we will refer to the noisy Gram matrix $K_\varepsilon(X, X)$ as K_ε. The matrix $K(X, X^*) \in \mathbb{R}^{n\times m}$ will be referred to as K_* and $K(X^*, X)$ as K_*^{T}. In similar fashion, we have $K(X_*, X_*) = K_{**} \in \mathbb{R}^{m\times m}$.

8.3 Prediction with Non-Zero Mean

As mentioned, it is common in practice to form a GP prior with zero-mean, however it is not difficult to incorporate a mean function into the predictions using a GP posterior. For example, assume we have a GP prior

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}')), \tag{8.5}$$

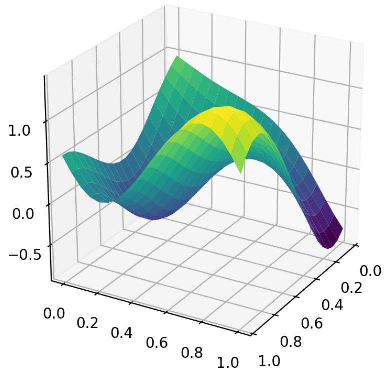
then the posterior distribution as per 8.1 is

$$\boldsymbol{\mu}_* = \boldsymbol{\mu}(X_*) + K(X_*, X)K_\varepsilon(X, X)^{-1}(\mathbf{y} - \boldsymbol{\mu}(X)) \tag{8.6}$$

$$\Sigma_* = K(X_*, X_*) - K(X_*, X)K_\varepsilon(X, X)^{-1}K(X, X_*). \tag{8.7}$$

8.4 High Dimension Inputs

Our figures thus far have considered one-dimensional inputs however GPs are defined up to arbitrarily high dimension input vectors $\mathbf{x}$. The squared exponential covariance function can be extended to higher dimensions by equipping automatic relevance determination to give each dimension of the input a weighted length scale. We demonstrate a 2-D input GP in fig. 11.



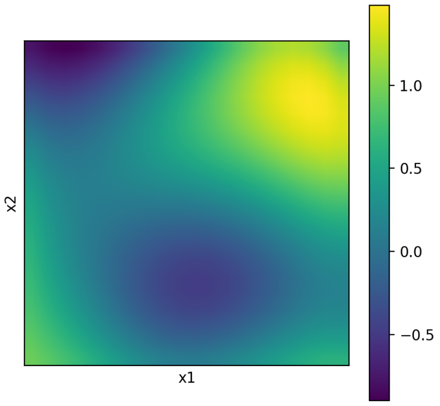


Figure 11: Sample function from a 2-D input GP with mean 0 and SE-ARD covariance function

Figure 12: Heat map for the drawn function

9 Covariance Functions and Kernel Parameters

Covariance functions are of such paramount importance to GP regression that it warrants its own discussion. First, we will introduce some commonly used covariance functions and examine some functions sampled from GP priors with such covariances functions. We will discuss the properties of such functions and why they are so popular in practice. The second half of this section will consider forming new covariance functions and forming a space of functions using covariance mixture models.

9.1 Common Covariance Functions

In this section we introduce and discuss the most common covariance functions.

9.1.1 Squared Exponential

We have already come across the squared exponential kernel $\kappa_{\textit{SE}}(\mathbf{x}, \mathbf{x}')$, however it is relevant for us to introduce a modification to the SE kernel by equipping it with automatic relevance determination (ARD). Simply put, ARD allows us to attach a separate length scale l_j for element $x_j \in \mathbf{x}_i$. The SE-ARD kernel is as follows

$$\kappa_{\textit{SE-ARD}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{1}{2}\sum_{j=1}^{D}\left(\frac{x_j - x'_j}{l_j}\right)^2\right) = \sigma^2 \exp\left(-0.5(\mathbf{x}-\mathbf{x}')^{\mathrm{T}}\boldsymbol{\Delta}^{-1}(\mathbf{x}-\mathbf{x}')\right), \tag{9.1}$$

where $\boldsymbol{\Delta} = \mathrm{diag}(l_1^2, l_2^2, \ldots, l_D^2)$. Performing hyperparameter optimization for this kernel will now tune the values of l_i such that we can select the right weight for each dimension of the inputs.

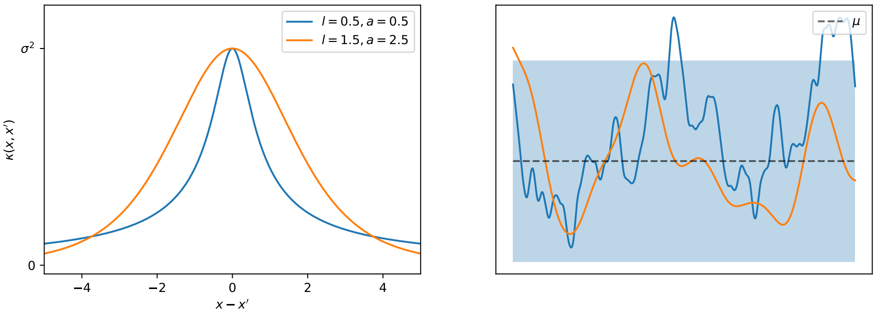


Figure 13: Plot of the RQ kernel with two different length scales and scale weights, and draws from GPs with RQ priors.

9.1.2 Rational Quadratic

The rational quadratic kernel is somewhat similar to the squared exponential in it produces sampled functions which are somewhat smooth as determined by the kernel hyperparameters. The RQ kernel is as follows

$$\kappa_{_{RQ}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \left(1 + \frac{(\mathbf{x} - \mathbf{x}')^{\mathrm{T}}(\mathbf{x} - \mathbf{x}')}{2al^2}\right)^{-a}. \tag{9.2}$$

This kernel can be interpreted as an infinite sum of different SE kernels, each equipped with different length scales. Here a determines the weighting between the length scales and $\lim_{a\to\infty} \kappa_{_{RQ}}(a; \mathbf{x}, \mathbf{x}', l, \sigma) = \kappa_{_{SE}}(\mathbf{x}, \mathbf{x}'; l, \sigma)$. We observe the form of the RQ kernel in fig. 13 and sample some functions drawn from a GP with an RQ kernel prior.

9.1.3 Periodic

A periodic kernel reflects our prior knowledge about the repetitive qualities of an underlying function. The orthodox periodic kernel, derived by David Mackay, produces sampled functions which are exactly periodic and is of the form

$$\kappa_{_{PER}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{2}{l^2} \sin^2\left(\pi \frac{|\mathbf{x} - \mathbf{x}'|}{p}\right)\right), \tag{9.3}$$

where $|\cdot|$ represents the Euclidean distance function. The components of this kernel are easily interpretable,

- σ the overall variance.
- l the length scale.
- p the period, dictating the distance between the start of each repetition.

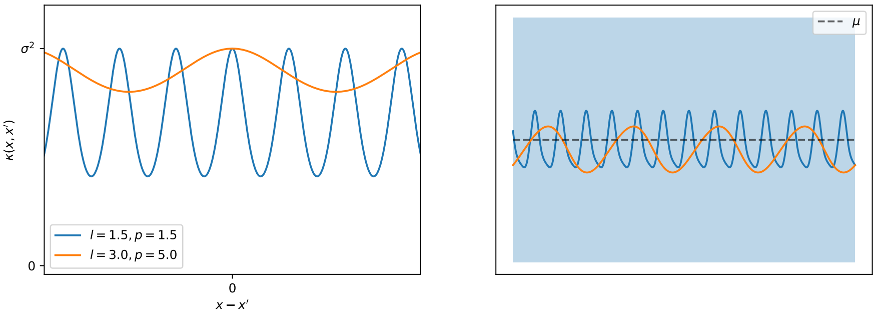


Figure 14: Plot of the periodic kernel with varying hyperparameters, and draws from GPs with periodic priors

9.1.4 Linear

A GP with a linear kernel prior is nothing other than simple Bayesian linear regression. At a cost scaling with $\mathcal{O}(n^3)$, this is a very inefficient way of performing linear regression and thus this kernel is rarely used in practice on its own. It has the form

$$\kappa_{PER}(\mathbf{x}, \mathbf{x}') = \sigma_1^2 + \sigma_2^2(\mathbf{x} - c \cdot \mathbf{1}) \cdot (\mathbf{x}' - d \cdot \mathbf{1})^{\mathrm{T}}, \tag{9.4}$$

where $c, d \in \mathbb{R}$ and $\mathbf{1}$ is a column vector of ones.

9.2 Combining Kernels

References

[1] J. H. Wilkinson, *The algebraic eigenvalue problem*, Clarendon Press, Oxford, 1965. MR0184422

[2] Carl Edward Rasmussen and Christopher K. I. Williams, *Gaussian processes for machine learning*, Adaptive Computation and Machine Learning, MIT Press, Cambridge, MA, 2006. MR2514435