2 Linear Regression

In regression, our goal is to learn a mapping from one real-valued space to another. Linear regression is the simplest form of regression: it is easy to understand, often quite effective, and very efficient to learn and use.

2.1 The 1D case

We will start by considering linear regression in just 1 dimension. Here, our goal is to learn a mapping $y = f(x)$, where x and y are both real-valued scalars (i.e., $x \in \mathbb{R}, y \in \mathbb{R}$). We will take f to be an linear function of the form:

$$y = wx + b \tag{1}$$

where w is a *weight* and b is a *bias*. These two scalars are the parameters of the model, which we would like to learn from training data. n particular, we wish to estimate w and b from the N training pairs $\{(x_i, y_i)\}_{i=1}^N$. Then, once we have values for w and b, we can compute the y for a new x.

Given 2 data points (i.e., N=2), we can exactly solve for the unknown slope w and offset b. (How would you formulate this solution?) Unfortunately, this approach is extremely sensitive to noise in the training data measurements, so you cannot usually trust the resulting model. Instead, we can find much better models when the two parameters are estimated from larger data sets. When $N > 2$ we will not be able to find unique parameter values for which $y_i = wx_i + b$ for all i, since we have many more constraints than parameters. The best we can hope for is to find the parameters that minimize the residual errors, i.e., $y_i - (wx_i + b)$.

The most commonly-used way to estimate the parameters is by *least-squares regression*. We define an energy function (a.k.a. objective function):

$$E(w, b) = \sum_{i=1}^{N} (y_i - (wx_i + b))^2 \tag{2}$$

To estimate w and b, we solve for the w and b that minimize this objective function. This can be done by setting the derivatives to zero and solving.

$$\frac{dE}{db} = -2 \sum_i (y_i - (wx_i + b)) = 0 \tag{3}$$

Solving for b gives us the estimate:

$$b^* = \frac{\sum_i y_i}{N} - w \frac{\sum_i x_i}{N} \tag{4}$$

$$= \bar{y} - w\bar{x} \tag{5}$$

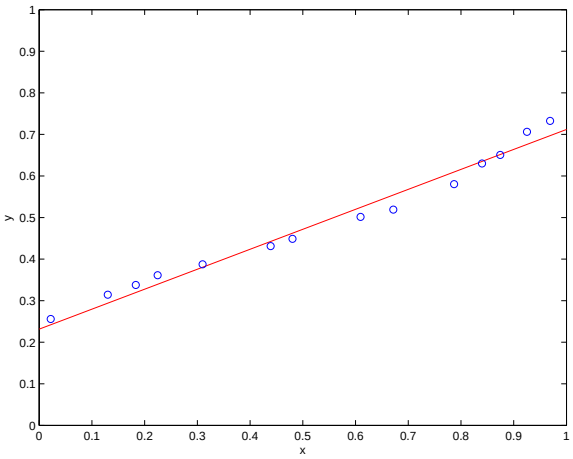

Figure 1: An example of linear regression: the red line is fit to the blue data points.

where we define $\bar{x}$ and $\bar{y}$ as the averages of the x's and y's, respectively. This equation for b^* still depends on w, but we can nevertheless substitute it back into the energy function:

$$E(w, b) = \sum_i ((y_i - \bar{y}) - w(x_i - \bar{x}))^2 \tag{6}$$

Then:

$$\frac{dE}{dw} = -2 \sum_i ((y_i - \bar{y}) - w(x_i - \bar{x}))(x_i - \bar{x}) \tag{7}$$

Solving $\frac{dE}{dw} = 0$ then gives:

$$w^* = \frac{\sum_i (y_i - \bar{y})(x_i - \bar{x})}{\sum_i (x_i - \bar{x})^2} \tag{8}$$

The values w^* and b^* are the least-squares estimates for the parameters of the linear regression.

2.2 Multidimensional inputs

Now, suppose we wish to learn a mapping from D-dimensional inputs to scalar outputs: $\mathbf{x} \in \mathbb{R}^D$, $y \in \mathbb{R}$. Now, we will learn a vector of weights $\mathbf{w}$, so that the mapping will be:[1]

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = \sum_{j=1}^{D} w_j x_j + b \quad . \tag{9}$$

[1] Above we used subscripts to index the training set, while here we are using the subscript to index the elements of the input and weight vectors. In what follows the context should make it clear what the index denotes.

For convenience, we can fold the bias b into the weights, if we augment the inputs with an additional 1. In other words, if we define

$$\tilde{\mathbf{w}} = \begin{bmatrix} w_1 \\ \vdots \\ w_D \\ b \end{bmatrix}, \quad \tilde{\mathbf{x}} = \begin{bmatrix} x_1 \\ \vdots \\ x_D \\ 1 \end{bmatrix} \tag{10}$$

then the mapping can be written:

$$f(\mathbf{x}) = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}} \,. \tag{11}$$

Given N training input-output pairs, the least-squares objective function is then:

$$E(\tilde{\mathbf{w}}) = \sum_{i=1}^{N} (y_i - \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_i)^2 \tag{12}$$

If we stack the outputs in a vector and the inputs in a matrix, then we can also write this as:

$$E(\tilde{\mathbf{w}}) = ||\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}||^2 \tag{13}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \quad \tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{x}_1^T & 1 \\ \vdots & \\ \mathbf{x}_N^T & 1 \end{bmatrix} \tag{14}$$

and $|| \cdot ||$ is the usual Euclidean norm, i.e., $||\mathbf{v}||^2 = \sum_i v_i^2$. (You should verify for yourself that Equations 12 and 13 are equivalent).

Equation 13 is known as a linear least-squares problem, and can be solved by methods from linear algebra. We can rewrite the objective function as:

$$\begin{aligned} E(\mathbf{w}) &= (\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}})^T (\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}) && (15) \\ &= \tilde{\mathbf{w}}^T \tilde{\mathbf{X}}^T \tilde{\mathbf{X}} \tilde{\mathbf{w}} - 2\mathbf{y}^T \tilde{\mathbf{X}} \tilde{\mathbf{w}} + \mathbf{y}^T \mathbf{y} && (16) \end{aligned}$$

We can optimize this by setting all values of $dE/dw_i = 0$ and solving the resulting system of equations (we will cover this in more detail later in Chapter **??**). In the meantime, if this is unclear, start by reviewing your linear algebra and vector calculus). The solution is given by:

$$\mathbf{w}^* = (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \mathbf{y} \tag{17}$$

(You may wish to verify for yourself that this reduces to the solution for the 1D case in Section 2.1; however, this takes quite a lot of linear algebra and a little cleverness). The matrix $\tilde{\mathbf{X}}^+ \equiv (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T$ is called the *pseudoinverse* of $\tilde{\mathbf{X}}$, and so the solution can also be written:

$$\tilde{\mathbf{w}}^* = \tilde{\mathbf{X}}^+ \mathbf{y} \tag{18}$$

In MATLAB, one can directly solve the system of equations using the slash operator:

$$\tilde{\mathbf{w}}^* = \tilde{\mathbf{X}} \backslash \mathbf{y} \tag{19}$$

There are some subtle differences between these two ways of solving the system of equations. We will not concern ourselves with these here except to say that I recommend using the slash operator rather than the pseudoinverse.

2.3 Multidimensional outputs

In the most general case, both the inputs and outputs may be multidimensional. For example, with D-dimensional inputs, and K-dimensional outputs $\mathbf{y} \in \mathbb{R}^K$, a linear mapping from input to output can be written as

$$\mathbf{y} = \tilde{\mathbf{W}}^T \tilde{\mathbf{x}} \tag{20}$$

where $\tilde{\mathbf{W}} \in \mathbb{R}^{(D+1)\times K}$. It is convenient to express $\tilde{\mathbf{W}}$ in terms of its column vectors, i.e.,

$$\tilde{\mathbf{W}} = [\tilde{\mathbf{w}}_1 \ \ldots \ \tilde{\mathbf{w}}_K] \equiv \begin{bmatrix} \mathbf{w}_1 & \ldots & \mathbf{w}_K \\ b_1 & \ldots & b_K \end{bmatrix} . \tag{21}$$

In this way we can then express the mapping from the input $\tilde{\mathbf{x}}$ to the j_{th} element of $\mathbf{y}$ as $y_j = \tilde{\mathbf{w}}_j^T \mathbf{x}$. Now, given N training samples, denoted $\{\tilde{\mathbf{x}}_i, \mathbf{y}_i\}_{i=1}^N$ a natural energy function to minimize in order to estimate $\tilde{\mathbf{W}}$ is just the squared residual error over all training samples and all output dimensions, i.e.,

$$E(\tilde{\mathbf{W}}) = \sum_{i=1}^{N} \sum_{j=1}^{K} (y_{i,j} - \tilde{\mathbf{w}}_j^T \tilde{\mathbf{x}}_i)^2 . \tag{22}$$

There are several ways to conveniently *vectorize* this energy function. One way is to express E solely as a sum over output dimensions. That is, let $\mathbf{y}'_j$ be the N-dimensional vector comprising the j^{th} component of each output training vector, i.e., $\mathbf{y}'_j = [y_{1,j}, y_{2,j}, ..., y_{N,j}]^T$. Then we can write

$$E(\tilde{\mathbf{W}}) = \sum_{j=1}^{K} ||\mathbf{y}'_j - \tilde{\mathbf{X}} \tilde{\mathbf{w}}_j||^2 \tag{23}$$

where $\tilde{\mathbf{X}}^T = [\tilde{\mathbf{x}}_1 \ \tilde{\mathbf{x}}_2 \ \ldots \ \tilde{\mathbf{x}}_N]$. With a little thought you can see that this really amounts to K distinct estimation problems, the solutions for which are given by $\tilde{\mathbf{w}}_j^* = \tilde{\mathbf{X}}^+ \mathbf{y}'_j$.

Another common convention is to stack up everything into a matrix equation, i.e.,

$$E(\tilde{\mathbf{W}}) = ||\mathbf{Y} - \tilde{\mathbf{X}} \tilde{\mathbf{W}}||_F^2 \tag{24}$$

where $\mathbf{Y} = [\mathbf{y}'_1 \ \ldots \mathbf{y}'_K]$, and $|| \cdot ||_F$ denotes the Frobenius norm: $||\mathbf{Y}||_F^2 = \sum_{i,j} \mathbf{Y}_{i,j}^2$. You should verify that Equations (23) and (24) are equivalent representations of the energy function in Equation (22). Finally, the solution is again provided by the pseudoinverse:

$$\tilde{\mathbf{W}}^* = \tilde{\mathbf{X}}^+ \mathbf{Y} \tag{25}$$

or, in MATLAB, $\tilde{\mathbf{W}}^* = \tilde{\mathbf{X}} \backslash \mathbf{Y}$.

