Optimization methods in Regression

Parin Chaipunya

King Mongkut's University of Technology Thonburi
Email: parin.cha@kmutt.ac.th

1. What is regression?

In general, *regression* refers to a broad class of statistical models where one wants to fit a function to the relationship between *explanatory variables* and a *response.* Usually, the function of choice is described by several *parameters*, which are to be estimated in an error-minimizing way.

Let us study a relationship between scalar explanatory variables $x_1, \ldots, x_n$ and a response variable y, which are observed across M samples. Then we make an assumption that

$$y^i \approx f_\beta(x_1^i, \ldots, x_n^i)$$

at each sample $i = 1, \ldots, M$, where f_β belongs to a certain class (*e.g.* affine functions, polynomials, etc.) and is described with a vector β of parameters. The aim of a regression model is to solve the following optimization problem

$$\min_\beta \quad \mathrm{Err}[(f_\beta(x_1^i, \ldots, x_n^i) - y^i)_{i=1,\ldots,M}],$$

where Err is a function describing the errors between the *observed values* and the *modeled values.*

Be cautious that the term *variables* here are known values, while the unknowns are the *parameter vector* β.

In this short note, the one dimensional vector $a = (a_1, \ldots, a_n)$ is interpreted as a column vector. This is not the same as $[a_1 \ \ldots \ a_n]$, which is a row vector.

2. Linear regression

The simplest class of regression models is the *linear regression.* In linear regression, the regression function f_β belongs to the class of *affine functions.* Each affine function $f : \mathbb{R}^n \to \mathbb{R}$ is described by $n + 1$ scalar parameters, $\beta_0, \beta_1, \ldots, \beta_n$. Hence $\beta := (\beta_0, \beta_1, \ldots, \beta_n)$ and f_β is expressed with

$$f_\beta(x_1, \ldots, x_n) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n.$$

Hence, we make an assumption, at each sample i, that

$$y^i \approx \beta_0 + \beta_1 x_1^i + \dots \beta_n x_n^i.$$

To estimate the parameters β_i's, the squares error function is used. This leads to the least-squares problem

$$\min_{\beta} \quad E(\beta) := \sum_{i=1}^{M} (y^i - \beta_0 - \beta_1 x_1^i - \dots - \beta_n x_n^i)^2.$$

If we put

$$X = \begin{bmatrix} 1 & x_1^1 & \dots & x_n^1 \\ 1 & x_1^2 & \dots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^M & \dots & x_n^M \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^M \end{bmatrix},$$

then the objective function becomes

$$E(\beta) = (X\beta - y)^\top (X\beta - y) = \beta^\top (X^\top X)\beta - 2(X^\top y)^\top \beta + y^\top y.$$

Since $X^\top X$ is positive semidefinite, E is convex and so the Fermat's rule implies that

$$\overline{\beta} \in \arg\min E \iff X^\top X \overline{\beta} = X^\top y.$$

The following Figure 2.1 shows an illustration of linear regression in one dimension.

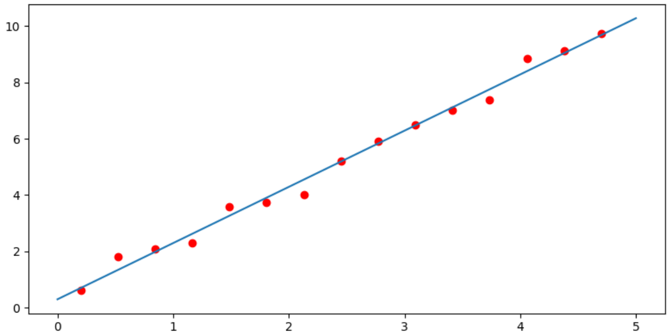

Figure 2.1: Linear regression.

3. Quadratic regression

In quadratic regression, one takes f_β from the class of quadratic functions. The parameter vector β is now decomposed into

- the intercept β_0,

○ the linear terms β_j for $j = 1, \dots, n$

○ the quadratic terms β_{jk} for $j, k = 1, \dots, n$ with $j \leq k$.

The dimension of β equals to $1 + 2n + \frac{n(n-1)}{2}$ and f_β is expressed with

$$f_\beta(x) = \beta_0 + \sum_{j=1}^{n} \beta_j x_j + \sum_{j=1}^{n} \sum_{k=j}^{n} \beta_{jk} x_j x_k.$$

We still use the least-squares error term in our least-squares, which gives

$$\min_\beta \quad E(\beta) := \sum_{i=1}^{M} \left(y^i - \beta_0 - \sum_{j=1}^{n} \beta_j x_j^i - \sum_{j=1}^{n} \sum_{k=j}^{n} \beta_{jk} x_j^i x_k^i \right)^2 .$$

We set

$$X_{\text{linear}} = [x_j^i]_{\substack{i=1,\dots,M \\ j=1,\dots,n}} \quad \text{and} \quad X^j_{\text{quadratic}} = [x_j^i x_k^i]_{\substack{i=1,\dots,M \\ k=j,\dots,n}} \quad (j = 1, \dots, n).$$

Finally, we put

$$X = [\mathbf{1} \ X_{\text{linear}} \ X^1_{\text{quadratic}} \ \cdots \ X^n_{\text{quadratic}}],$$

where $\mathbf{1}$ denotes the column vector of 1's of appropriate dimension. Then the objective function is, again, expressed by

$$E(\beta) = (X\beta - y)^\top (X\beta - y) = \beta^\top (X^\top X)\beta - 2(X^\top y)^\top \beta + y^\top y.$$

and we have

$$\overline{\beta} \in \arg\min E \iff X^\top X \overline{\beta} = X^\top y.$$

The following Figure 3.2 illustrates quadratic regression.

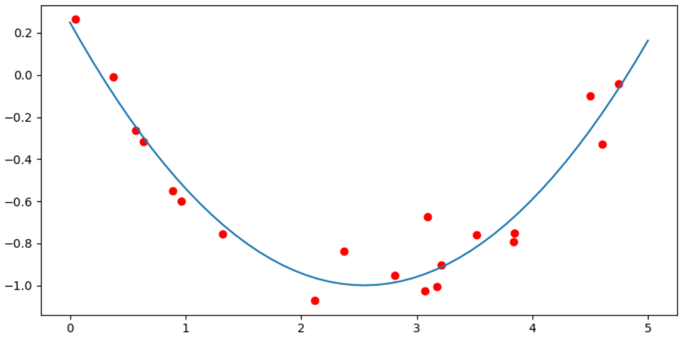

Figure 3.2: Quadratic regression

4. Logistic regression

Logistic regression is used with a different nature when y^i is either $+1$ or 0, where the two values describes whether a sample belongs to a certain class. A good example is in medical science where explanatory variables are diagnosed whether a patient has the disease.

Since it is difficult to estimate a function with discrete values, a logistic function

$$p(t) = \frac{1}{1 + e^{\mu - \lambda t}}, \quad (t \in \mathbb{R}),$$

whose graph is shown in Figure 4.3.

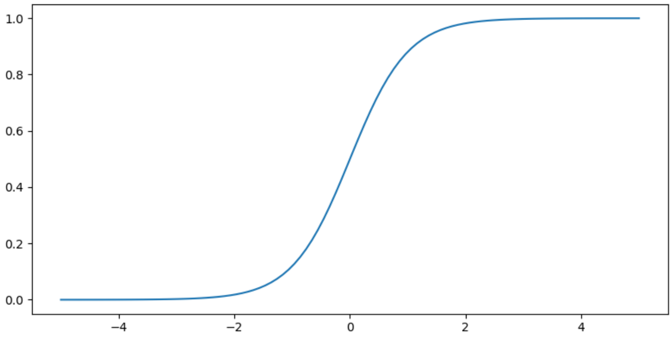

Figure 4.3: The graph of a logistic function.

One would see that $p(t) \to 1$ as $t \to +\infty$ and $p(t) \to 0$ as $t \to -\infty$. The idea is then to map to the far right the points that are likely classified to the $+1$ class, and to map to the far left the points that are likely classified to the 0 class. When passed into the logistic function (and possibly again into the Heaviside function), the misprediction should be minized through an error function.

Here, the regression function is a composition of an affine function with parameters $\beta = (\beta_0, \dots, \beta_n)$, that is

$$f_\beta(x) = \frac{1}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n}} = p \circ L_\beta(x),$$

where

$$L_\beta(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n.$$

The final classification is evaluated with the Heaviside function, so that the predicted response $\hat{y}$ is computed from the variable x by

$$\hat{y} = H(f_\beta(x) - 0.5),$$

where the Heaviside function is defined by

$$H(t) = \begin{cases} 1 & \text{if } t \geq 0, \\ 0 & \text{if } t < 0. \end{cases}$$

Due to the complicate structure of the logistic regression function f_β, the squares error is no longer appropriate as it renders the error function $\mathrm{Err}(\cdot)$ nonconvex. To this end, we use the *log-loss function*

$$J(p^1, \ldots, p^M) = \sum_{i=1}^{M} \left[-y^i \ln(p^i) - (1 - y^i) \ln(1 - p^i) \right], \quad (0 < p^i < 1).$$

Applying this to the prediction $f_\beta(x)$, we consider $E(\beta) = J \circ (f_\beta(x^i))$. Hence, we have

$$E(\beta) = \sum_{i=1}^{M} \left[-y^i \ln(f_\beta(x^i)) - (1 - y^i) \ln(1 - f_\beta(x^i)) \right],$$

whose structure is better suited with the logistic regression function. In particular, the term

$$\ln(f_\beta(x))$$

puts a positive penalty when $f_\beta(x)$ is away from 1. Moreover, $\ln(f_\beta(x)) \to +\infty$ as $x \to 0^+$. Similarly, the term

$$\ln(1 - f_\beta(x))$$

puts a positive penalty when $f_\beta(x)$ is away from 0 and $\ln(1 - f_\beta(x)) \to +\infty$ as $x \to 1^-$. The prefixes y^i and $(1 - y^i)$ are then used to activate the first and second parts of the log-loss function.

Now, even though the log-loss function is used, it is still much more tricky than the least-squares loss used in polynomial regressions. To solve for the optimal parameter β, we aim to use the gradient descent method. Let us conventionally put $x_0^i := 0$ for all $i = 1, \ldots, M$. Observe, for any $j = 0, \ldots, n$, we have

$$\begin{aligned} \frac{\partial E}{\partial \beta_j} &= \sum_{i=0}^{M} \frac{\partial J}{\partial p^i} \frac{\partial p^i}{\partial \beta_j} \\ &= \sum_{i=1}^{M} \frac{p^i - y^i}{p^i(1 - p^i)} \cdot p^i(1 - p^i) x_j^i \\ &= \sum_{i=1}^{M} (p^i - y^i) x_j^i, \end{aligned}$$

with $p^i = f_\beta(x^i)$. We consequently obtain

$$\nabla_\beta E(\beta) = X^\top (p - y),$$

where $X = [x_j^i]_{\substack{i=1,\ldots,M \\ j=0,\ldots,n}}$ is the data matrix and $p = (p^1, \ldots, p^M)$.

Algorithm 1 Gradient Descent Algorithm (with fixed learning rate)

```
Require:
    Initial parameter $\beta^0 = (\beta_0^0, \ldots, \beta_n^0)$,
    explanatory matrix $X = [x_j^i]_{\substack{i=1,\ldots,M \\ j=0,\ldots,n}}$,
    response vector $y = (y^1, \ldots, y^M)$,
    learning rate $\alpha$,
    loss function $E(\beta)$.
Ensure: Optimized parameter $\overline{\beta} = (\overline{\beta}_0, \ldots, \overline{\beta}_n)$.
Initialize $\beta \leftarrow \beta^0$.
repeat
    Compute the prediction: $\forall i = 1, \ldots, M,\ p^i \leftarrow f_\beta(x^i)$.
    Update parameters: $\beta \leftarrow \beta - \alpha X^T(p - y)$.
until convergence criterion is met
return $\beta$.
```