Background

Jimmy Zhan

August 2025

Inspiration

I first noticed the existence of the Newton-Raphson Method (hereafter, Newton's Method) while reading the A-Level Further Math textbook. This numerical approach provides a unique perspective in solving problems.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuous and twice differentiable. Consider the unconstrained minimisation problem

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \tag{1}$$

and assume that the solution set of (1) is not empty. In general, numerical methods based on line search have the iterative formula

$$x_{k+1} = x_k + \alpha_k p_k \tag{2}$$

where x_k is the current iterative point, α_k is the step size, and p_k is the search direction. Traditionally, Newton and Newton-type optimisation methods take the step size as -1, and the direction

$$p_k = -H_k^{-1} \nabla_k \tag{3}$$

where H_k is the Hessian $\nabla^2 f(x_k)$.

Newton's Method

The method starts with the famous Taylor's Expansion. Consider $f(x) \in \mathbb{R}$ being a smooth single-variabled function. At point $x = x_k$, there is

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{f''(x_k)}{2!}(x - x_k)^2 + \frac{f'''(x_k)}{3!}(x - x_k)^3 + ... \quad (4)$$

By taking only up to the third term, and changing f(x) to a multi-variabled function (that is, take all x as matrices), there is

$$f(x) \approx f(x_k) + \nabla(x_k)^\intercal(x - x_k) + \frac{1}{2}(x - x_k)^\intercal H(x_k)(x - x_k) \quad (5)$$

Here, the quadratic approximation is chosen specifically. Taking the linear information only provides a line with no curvature information. While the Hessian term describes how the function curves in each direction. This allows Newton's method to adaptively choose the step length as well as the direction to achieve faster convergence. Calculating further terms require lots of time and effort.

To find the minima of f(x), we essentially solve $f'(x) = 0$. In terms of multi-variabled functions (equation 5), that is

$$\nabla(x_k) + H(x_k)(x_{k+1} - x_k) = 0 \quad (6)$$

Rearranging the equation we get

$$x_{k+1} = x_k - \nabla(x_k)H^{-1}(x_k) \quad (7)$$

Logistic Regression

After further investigation, I noticed that Newton's method was presented in the context of logistic regression, where it naturally arises through iteratively reweighted least squares. This unexpected connection between a classical numerical technique and a modern statistical model sparked my interest. It suggested that methods developed for solving equations could be directly adapted to optimization problems in machine learning.

Logistic regression is a popular matehmatical modeling procedure used in the analysis of epidemiologic data. Consider a hypothetical disease MAD (Mathematical Anxiety Disorder). Scientists found k independent variables, denoted as $X_1, X_2, ..., X_k$. Logistic regression can be used to describe the relationship of the Xs to a dichotomous dependent variable, such as D, in this case represents having MAD or not.

Consider function

$$f(z) = \frac{1}{1+e^{-z}} \tag{8}$$

It is obvious that $\lim_{z\to\infty} f(z) = 1$ and $\lim_{z\to-\infty} f(z) = 0$. The model is designed to describe a probability, which is always some number between 0 and 1. In epidemiologic terms, such a probability gives the risk of an individual getting a disease.

In real life however, different variables may have different effect on the probability. We can then use coefficients to simulate these effects. Let

$$z = \alpha + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k = \alpha + \beta^\intercal X \tag{9}$$

z then becomes an index that combines all the Xs.

We can then turn the whole epidemiologic problem into a math problem. Essentially we want the probability

$$\begin{aligned} P(y=1|x) = \sigma(z) = \frac{1}{1+e^{-(\alpha+\beta^\intercal x)}} \\ P(y=0|x) = 1-\sigma(z) \end{aligned} \tag{10}$$

If we have n indenpendent variables, the likelihood is the product of every variable.

$$L(\alpha,\beta) = \prod_{i=1}^{n} P(y_i|x_i) = \prod_{i=1}^{n} \sigma(z_i)^{y_i}(1-\sigma(z_i))^{1-y_i} \tag{11}$$

It is not very convenient to maximise L as it is a product. We therefore take the log of eq 11. Let

$$l(\alpha,\beta) = \ log\ L(\alpha,\beta) = \sum_{i=1}^{n}[y_i\ log\ \sigma(z_i) + (1-y_i)\ log\ (1-\sigma(z_i))] \tag{12}$$

The most commonly used method to minimise l is the gradient descent method. It gives the parameters

$$\begin{aligned} \alpha^{(t+1)} &= \alpha^t + \alpha_{lr}\frac{\partial l}{\partial \alpha}(\alpha^t,\beta^t), \\ \beta^{(t+1)} &= \beta^t + \alpha_{lr}\nabla_\beta l(\alpha^t,\beta^t) \end{aligned} \tag{13}$$

where α_{lr} is the learning rate. At each iteration, the parameters are adjusted along the gradient direction.

In the case of logistic regression, Newton's method exists practical limitations, including the computational cost and instability in high dimensions. This motivated me to explore whether a refined version of Newton's method could be designed to overcome these obstacles, making it more suitable for logistic regression and potentially competitive with gradient descent.