Machine Learning

Regression

Mayson Ma

Courant Institute of Mathematical Sciences - New York University

June 2021

Linear Regression
Introduction
Solution
Pseudoinverse
The Least Norm Solution
Logistic Regression
Introduction
Solution
Newton's Method

Table of Contents

Linear Regression
  Introduction
  Solution
    Pseudoinverse
    The Least Norm Solution

Logistic Regression
  Introduction
  Solution
    Newton's Method

Linear Regression

Symbols

Sample points $x_1, x_2, ...x_n \in \mathbb{R}$ and $X \in \mathbb{R}^{n \times d}$ is the design matrix of sample points. The associated labels are $y_1, y_2, ..., y_n \in \mathbb{R}$, and $y = [y_1 \quad \cdots \quad y_n]^\top \in \mathbb{R}^n$.

Hypothesis set

Linear functions

$$\left\{ x \mapsto w^\top x + \alpha : x \in \mathbb{R}^d, \alpha \in \mathbb{R} \right\}$$

Optimization Problems

Empirical risk minimization

$$\min_{w,\alpha} F(w, \alpha) = ||(Xw + \alpha) - y||^2$$

Machine Learning

Mayson Ma

Linear Regression
Introduction
Solution
Pseudoinverse
The Least Norm Solution

Logistic Regression
Introduction
Solution
Newton's Method

Solution

Use the fictitious dimension trick, rewrite the objective functions:

$$\min_{w} F(w) = ||Xw - y||^2$$

where $X = \begin{bmatrix} x_1^\top & 1 \\ \vdots & \vdots \\ x_n^\top & 1 \end{bmatrix} \in \mathbb{R}^{n\times(d+1)}$, $w = \begin{bmatrix} w_1 \\ \vdots \\ w_d \\ \alpha \end{bmatrix} \in \mathbb{R}^{d+1}$

Solve by calculus

The objective is a convex and differentiable function

$$F(w) = w^\top X^\top Xw - 2y^\top Xw + y^\top y$$
$$\Delta_w F = 2X^\top Xw - 2X^\top y$$

Set $\Delta_w F = 0 \Leftrightarrow X^T Xw = X^T y$

Solution

The normal equation

$$X^T X w = X^T y$$

Solution to the normal equation

If $X^\top X$ is not singular, $w = (X^T X)^{-1} X^T y$.

If $X^\top X$ is singular, the problem is underconstrained, $w = (X^T X)^+ X^T y$ in general, where $(X^T X)^+$ is the pseudoinverse of $X^T X$.

Pseudoinverse

Pseudoinverse

The pseudoinverse of $X = U\Sigma V^\top$ is
$X^+ = V\Sigma^+ U^\top \in \mathbb{R}^{(d+1)\times n}$

SVD

Pieces of the SVD: $X = U\Sigma V^\top = \sigma_1 u_1 v_1^\top + \cdots + \sigma_r u_r v_r^\top$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.
The reduced form of the SVD of X is $X = U_r \Sigma_r V_r^\top$, where $U_r \in \mathbb{R}^{n\times r}, \Sigma_r \in \mathbb{R}^{r\times r}, V_r \in \mathbb{R}^{r\times d}$.
The columns of U_r is the basis for the column space of X and the columns of V_r is the basis for the row space of X.

The Least Norm Solution

One solution to this system is $w^+ = A^+ b$. It has two properties:

- $w = w^+$ is a minimizer of $||Xw - y||^2$
- If another $\hat{w}$ achieves that minimum, then $||w^+|| < ||\hat{w}||$

Proof

In other words, w^+ is a minimum norm least squares solution. When X has independent columns ($r = d$), it is the unique least square solution. If there are nonzero vectors δ in the nullspace of X ($r < d$), then they can be added to w^+, the error $||X(w^+ + \delta) - y||^2$ is not affected, since $X\delta = 0$. However, the length $||w^+ + \delta||^2$ will grow to $||w^+||^2 + ||\delta||^2$. Note that $w^+ \perp \delta$ (w^+ is in the rowspace of X and δ is in the null space of X)

Linear Regression
Introduction
Solution
Pseudoinverse
The Least Norm Solution
Logistic Regression
Introduction
Solution
Newton's Method

Logistic Regression

Linear Regression
Introduction
Solution
Pseudoinverse
The Least Norm Solution
Logistic Regression
Introduction
Solution
Newton's Method

Symbols

Sample points $x_1, x_2, ...x_n \in \mathbb{R}$ and $X \in \mathbb{R}^{n\times d}$ is the design matrix of sample points. The associated labels are $y_1, y_2, ..., y_n \in \{0, 1\}$, and $y = [y_1 \quad \cdots \quad y_n]^\top \in \{0, 1\}^n$.

Model

Let $\sigma(z) = \frac{1}{1+e^{-z}}$ be the sigmoid function. Then,

$$p_1 = \Pr(y = 1|x) = \sigma(w^\top x) = \frac{1}{1 + e^{-w^\top x}}$$

$$p_0 = \Pr(y = 0|x) = 1 - \sigma(w^\top x) = \frac{e^{-w^\top x}}{1 + e^{-w^\top x}}$$

Combine p_0 and p_1 gives $\Pr(y|x) = p_1^y \cdot p_0^{(1-y)}$.

Logistic Regression

Linear Regression
Introduction
Solution
Pseudoinverse
The Least Norm Solution
Logistic Regression
Introduction
Solution
Newton's Method

MLE

$$
\begin{aligned}
\hat{w} &= \operatorname*{argmax}_w \ln \prod_{i=1}^{n} \Pr(y_i|x_i) \\
&= \operatorname*{argmax}_w \ln \prod_{i=1}^{n} \Pr(y_i = 1|x_i)^{y_i} \cdot \Pr(y_i = 0|x_i)^{(1-y_i)} \\
&= \operatorname*{argmax}_w \sum_{i=1}^{n} y_i \ln \sigma(w^\top x_i) + (1 - y_i) \ln(1 - \sigma(w^\top x_i))
\end{aligned}
$$

Optimization Problem

$\min_w F(w) = -\sum_{i=1}^{n} y_i \ln \sigma(w^\top x_i) + (1 - y_i) \ln(1 - \sigma(w^\top x_i))$

Solution

Note that $\sigma'(z) = \sigma(z)(1 - \sigma(z))$.
Let $z_i = \sigma(w^\top x_i)$

$$\begin{aligned}
\Delta_w F &= -\sum_{i=1}^{n} \left(\frac{y_i}{z_i} - \frac{1 - y_i}{1 - z_i} \right) z_i(1 - z_i)x_i \\
&= -\sum_{i=1}^{n} (y_i - z_i)x_i \\
&= -X^\top (y - \sigma(Xw))
\end{aligned}$$

Solve by gradient descent

$w \leftarrow w + \epsilon \cdot X^\top (y - \sigma(Xw))$, where ϵ is the learning rate.

Linear Regression
Introduction
Solution
Pseudoinverse
The Least Norm Solution
Logistic Regression
Introduction
Solution
Newton's Method

Newton's Method

Idea

You are at point v. Approximate $F(w)$ near v by a quadratic function. Jump to its unique critical point. Repeat until bored.

Math

Taylor series at v:

$$F(v+d) \approx F(v) + \Delta F(v)^\top d + \frac{1}{2} d^\top \Delta^2 F(v) d$$

where $\Delta^2 F(v)$ is the **Hessian matrix** of F at point v.

Take derivative w.r.t d, set it to 0 and solve for d, i.e. find the critical point:

$$\Delta F(v) + \Delta^2 F(v) d = 0 \Rightarrow d = -(\Delta^2 F(v))^{-1} \Delta F(v)$$

Linear Regression
Introduction
Solution
Pseudoinverse
The Least Norm Solution
Logistic Regression
Introduction
Solution
Newton's Method

Newton's Method

Linear Regression
Introduction
Solution
Pseudoinverse
The Least Norm Solution
Logistic Regression
Introduction
Solution
Newton's Method

Algorithm

pick starting point w
repeat
  $e \leftarrow$ solution to linear system $(\Delta^2 F(w))^{-1} e = -\Delta F(w)$
  $w \leftarrow w + e$
until convergence

Comments

- Iterative optimization method for smooth $F(w)$
- Often much faster than gradient descent
- Does not know the difference between minima, maxima or saddle points
- Starting point must be "close enough" to desired solution

Solution

Solve by Newton's method

- Recall that $\Delta_w F = X^\intercal(z - y)$, where $z \in \mathbb{R}^n$ and $z_i = \sigma(w^\intercal x_i)$
- So $\Delta_W^2 F$ resolves to $X^\intercal(\Delta_w z)$
- Note that the i-th row of $\Delta_w z$ will be $z_i(1 - z_i)x_i^\intercal$. So we can rewrite this term as

$$\Delta_w z = \Omega X$$

where $\Omega = \begin{bmatrix} z_1(1-z_1) & & & \\ & z_2(1-z_2) & & \\ & & \ddots & \\ & & & z_n(1-z_n) \end{bmatrix}$

- Finally, $\Delta_W^2 F = X^\intercal \Omega X$

Solve by Newton's method

Algorithm

$w \leftarrow 0$
repeat
  $e \leftarrow$ solution to normal equations $(X^\top \Omega X)e = X^\top (y - z)$
  $w \leftarrow w + e$
until convergence

An example of iteratively reweighted least squares

- Ω is positive definite $\Rightarrow X^\top \Omega X$ is positive semidefinite $\Rightarrow F(w)$ is convex
- Ω prioritizes points with z near 0.5; tunes out points near 0 or 1.
- If n very large, save time by using a random subsample of the points per iteration. Increase sample size as you go.