Logistic Regression

The decision boundry is where the model is equally uncertain between classes 0 & 1. In other words where probability is exactly 0.5

mathematically

$$\sigma(\omega^T x + b) = 0.5$$

$$\Rightarrow \omega^T x + b = 0$$

decision boundry (Plane, line, hyperplane)

The log loss penalizes wrong or overy confident predictions. If a point belongs to class 1 & model predicts 0.2 the loss is large.

Cost function derivation

We use -ve sign in cost function becoz we want to maximize likelihood but maximization is difficult. So we minimize is better, so we make it -ve

if $y=1$ then p

if $y=0$ then $1-p$

$$\text{Likelihood} = p^y (1-p)^{(1-y)}$$

$$\log(\text{likelihood}) = y\log(p) + (1-y)\log(1-p)$$

We want to minimize negative log loss

if n is training eg

$$J(\omega, b) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\overset{h_\theta(x)}{p_i}) + (1-y_i)\log(1-\overset{h_\theta(x)}{p_i})]$$

how wrong prediction gives big loss

Intution

if $y=0$ $p=0.8$

$$\text{cost} = -[0.\log(0.8) + (1-0)\log(1-0.8)]$$

$$-\log(0.2) = 1.609 \rightarrow \boxed{\text{high}}$$

Now if $y=1$ & $p=0.8$

$$\text{cost} = 1.\log(0.8) + (1-1)\log(1-0.8)$$

$$= -\log(0.8) = -0.223 \rightarrow \boxed{\text{lower}}$$