Logistic Regression

- Classification problem is similar to predict only a small number of discrete values instead of continuous values.

1. Logistic Function (Sigmoid Function)

$$g(z) = \frac{1}{1+e^{-z}}$$

Properties

- Bound: $g(z) \in (0,1)$
- Symmetric: $1 - g(z) = g(-z)$
- Gradient: $g'(z) = g(z)(1-g(z))$

2. Logistic Regression

$$h_\theta(x) = g(\theta^T x) = \frac{1}{(1+e^{-\theta^T x})}$$

- $\theta^T x$ is called score. h_θ is called logistic regression.
- $Pr(Y=1|X=x;\theta) = h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$
- $Pr(Y=0|X=x;\theta) = 1 - h_\theta(x) = \frac{1}{1+e^{\theta^T x}}$

Decision Boundary

- $Pr(Y=1|X=x;\theta) = Pr(Y=0|X=x;\theta) \quad \Rightarrow \quad \theta^T x = 0$
- $\therefore$ The decision boundary is a linear hyperplane.
- The score $\theta^T x$ is also a measure of distance x from the hyperplane.

Probability Mass Function

$p(y|x;\theta) = Pr(Y=y|X=x;\theta) = (h_\theta(x))^y(1-h_\theta(x))^{1-y}$, where $y \in \{0,1\}$

$p(y|x;\theta) = \dfrac{1}{1+exp(-y\theta^T x)}$, where $y \in \{-1,1\}$ instead of $y \in \{0,1\}$,

- Maximize the log likelihood $L(\theta) = \displaystyle\prod_{i=1}^{m} p(y|x;\theta)$
- $l(\theta) = \log L(\theta) = \sum_{i=1}^{m}(y^{(i)}\log h_\theta(x^{(i)}) + (1-y^{(i)})\log(1-h_\theta(x^{(i)})))$, still assume $y \in \{-1,1\}$

$$\frac{\partial}{\partial\theta_j} l(\theta) = \sum_{i=1}^{m} \frac{y^{(i)} - h_\theta(x^{(i)})}{h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))} \cdot \frac{\partial h_\theta(x^{(i)})}{\partial\theta_j} = \sum_{i=1}^{m}(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

3. Newton’s Method

Properties

- Highly dependent on initial guess
- Quadratic convergence once it is sufficiently close to x^*
- If $f' = 0$, only has linear convergence
- Is not guaranteed to convergence at all, depending on function or initial guess

Update

$$x \leftarrow x - \frac{f'(x)}{f''(x)}$$

- For $l : \mathbb{R}^n \rightarrow \mathbb{R}$,

$$\theta \leftarrow \theta - H^{-1}\nabla_\theta l(\theta), \text{ where } H_{i,j} = \frac{\partial^2 l(\theta)}{\partial\theta_i \partial\theta_j}$$

Multiclass Classification

- Transformation to binary
  - One-vs.-rest (OvR, train a single classifier per class, with the samples of that class as positive samples and all other samples as negative ones)
    - $y^* = \underset{k}{\text{argmax}}\, f_k(x)$
    - $f_k(x)$ implies hight robability that x is in class k.
  - One-vs.-one (OvO, to train $K(K-1)/2$ binary classifiers)
    - $y^* = \underset{s}{\text{argmax}}(\sum_t f_{s,t}(x))$
    - $f_{s,t}(x)$ implies that label s has higher probability than label t.
- Extension from binary
- Hierarchical classification

4. Softmax Regression

$$\begin{aligned} l(\theta) &= \sum_{i=1}^{m} \text{log} p(y^{(i)}|x^{(i)};\theta) \\ &= \sum_{i=1}^{m} \text{log} \prod_{k=1}^{K} \left(\frac{exp(\theta^{(k)^T} x^{(i)})}{\sum_{k'=1}^{K} exp(\theta^{(k')^T} x^{(i)})} \right)^{\mathbb{I}(y^{(i)}=k)} \end{aligned}$$

- where $\mathbb{I} : \{True, False\} \rightarrow \{0, 1\}$ is an indicator function.