

Logistic Regression

Basic Model

- use for Binary classification: $y \in \{0, 1\}$, $x \in \mathbb{R}^d$
- $f(x) = w^T x + b$, $w \in \mathbb{R}^d$, $b \in \mathbb{R}$
- sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$
 - map \mathbb{R} to $[0, 1]$
 - $1 - \sigma(z) = \sigma(-z)$
- $P(y = 1|x) = \sigma(w^T x + b)$, $P(y = 0|x) = 1 - \sigma(w^T x + b)$
 - 用概率表示误差是一种比最小二乘更好的方式
 - y等于0或1只是一种指示，表示这是两种样本，它可以是任意两个不相等的数

Maximum Likelihood Estimation (MLE)

- Likelihood of training data (i.i.d): $\prod_{i=1}^n P(y = y_i | x = x_i)$
- 我们希望likelihood尽可能大:

$$\begin{aligned} & \max_{w,b} \prod_{i=1}^n P(y = y_i | x = x_i) \\ &= \max_{w,b} \prod_{i=1}^n (\sigma(w^T x_i + b))^{y_i} (1 - \sigma(w^T x_i + b))^{1-y_i} \end{aligned}$$

- 但是连乘会导致数字变得非常小(因为都小于1)，所以我们可以使用log:

$$\max_{w,b} \sum_{i=1}^n [y_i \log(\sigma(w^T x + b)) + (1 - y_i) \log(1 - \sigma(w^T x + b))]$$

- so now we have our **loss function** for Logistic Regression:

$$\min_{w,b} - \sum_{i=1}^n [y_i \log(\sigma(w^T x + b)) + (1 - y_i) \log(1 - \sigma(w^T x + b))]$$

- negative log likelihood = **cross entropy loss (交叉熵)**

Optimization

- let $\hat{X} = \begin{bmatrix} X \\ 1 \end{bmatrix} \in \mathbb{R}^{d+1}$, $\hat{W} = \begin{bmatrix} w \\ b \end{bmatrix} \in \mathbb{R}^{d+1}$
- Cross Entropy Loss ($L(\hat{W})$):

$$\begin{aligned}
L(\hat{W}) &= - \sum_{i=1}^n [y_i \log \frac{1}{1 + e^{-\hat{W}^T \hat{X}_i}} + (1 - y_i) \log(1 - \frac{1}{1 + e^{-\hat{W}^T \hat{X}_i}})] \\
&= - \sum_{i=1}^n [-y_i \log(1 + e^{-\hat{W}^T \hat{X}_i}) + y_i \log(1 + e^{\hat{W}^T \hat{X}_i}) - \log(1 + e^{\hat{W}^T \hat{X}_i})] \\
&= - \sum_{i=1}^n [y_i \log e^{\hat{W}^T \hat{X}_i} - \log(1 + e^{\hat{W}^T \hat{X}_i})] \\
&= - \sum_{i=1}^n [y_i \hat{W}^T \hat{X}_i - \log(1 + e^{\hat{W}^T \hat{X}_i})]
\end{aligned}$$

- 求梯度:

$$\begin{aligned}
\frac{\partial L(\hat{W})}{\partial \hat{W}} &= - \sum_{i=1}^n (y_i \hat{X}_i - \frac{\hat{X}_i e^{\hat{W}^T \hat{X}_i}}{1 + e^{\hat{W}^T \hat{X}_i}}) \in \mathbb{R}^{d+1} \\
&= - \sum_{i=1}^n (y_i - P(y = 1|x_i)) \hat{X}_i
\end{aligned}$$

- when $y_i = P(y = 1|x_i), \forall i$, then $\frac{\partial L(\hat{W})}{\partial \hat{W}} = 0$

- 这时模型完全能够反映正确的分类, 说明此时所有的样本是**线性可分(linearly separable)**的, 即可以被模型这个**分割超平面(hyperplane)**恰好一分为二; 但是很显然更多的情况是样本线性不可分, 但我们可以保证一定能找到一个全局最优解

- 求Hessian矩阵:

$$\begin{aligned}
\frac{\partial^2 L(\hat{W})}{\partial \hat{W} \partial \hat{W}^T} &= \sum_{i=1}^n \frac{e^{-\hat{W}^T \hat{X}_i}}{(1 + e^{-\hat{W}^T \hat{X}_i})^2} \hat{X}_i \hat{X}_i^T \\
&= \sum_{i=1}^n P(y = 1|x_i)(1 - P(y = 1|x_i)) \hat{X}_i \hat{X}_i^T
\end{aligned}$$

- 我们知道 $\hat{X}_i \hat{X}_i^T$ 一定是一个半正定矩阵, 那么可以说 $L(\hat{W})$ 一定是一个凸函数, 因此通过梯度下降它一定能找到一个全局最优解

标签为{-1,1}的逻辑回归

- $P(y_i = 1|x_i) = \sigma(w^T x_i + b)$
 - $\text{loss}(x_i, y_i) = -\log \sigma(w^T x_i + b) = \log(1 + e^{-y_i(w^T x_i + b)})$
- $P(y_i = -1|x_i) = \frac{1}{1 + e^{w^T x_i + b}}$
 - $\text{loss}(x_i, y_i) = \log(1 + e^{-y_i(w^T x_i + b)})$
- 我们发现, 无论 y_i 取 -1 还是 1, 损失函数的表达方式是相同的:

$$L(w, b) = \sum_{i=1}^n \log(1 + e^{-y_i(w^T x_i + b)})$$

- 梯度下降为:

$$\frac{\partial L}{\partial w} = - \sum_{i=1}^n \frac{y_i^T x_i}{1 + e^{y_i(w^T x_i + b)}}$$

- let $z_i := y_i(w^T x_i + b)$, then:

$$z_i \begin{cases} > 0 & \text{sign}(y_i) = \text{sign}(w^T x_i + b) \\ < 0 & \text{sign}(y_i) = -\text{sign}(w^T x_i + b) \end{cases}$$

- 可以发现 $\log(1 + e^{z_i})$ 是一个平滑的单调递减函数，性质很好
- $\sum_{i=1}^n \text{loss}(z_i)$ is the *number* of training point that wrongly classified
- $\frac{1}{n} \sum_{i=1}^n \text{loss}(z_i)$ is the *error rate*
- 0/1 loss function:

$$y = \begin{cases} 0 & x \geq 0 \\ 1 & x < 0 \end{cases}$$

- not differentiable
- not continuous at $x = 0$
- cannot use gradient descent
- cross-entropy loss is an *upper bound* of 0/1 loss
 - "Substitute loss function"

Multi-class Classification

- $y \in \{1, 2, \dots, K\}, x \in \mathbb{R}^d$
- **Softmax Regression**
 - 虽然叫回归，但解决的是分类问题
 - Define K linear models $f_k(x) = w_k^T x + b_k, k \in \{1, \dots, K\}$
 - Given (x_i, y_i) , find $f_k(x_i) > f_j(x_i), \forall j \neq k$, then predict $y_i = k$
- **MLE**
 - $P(y = k|x) = \frac{e^{f_k(x)}}{\sum_{j=1}^K e^{f_j(x)}}$
 - It's a probability distribution:
 - $\sum_{k=1}^K P(y = k|x) = 1$
 - $P(y = k|x) \geq 0, \forall k$
 - When $f_k(x) \gg f_j(x), \forall j \neq k$, then $P(y = k|x) \rightarrow 1$ and $P(y = j|x) \rightarrow 0$
 - MLE log-likelihood:

$$\begin{aligned} Loss &= \min_{\{(w_j, b_j) | j=1, \dots, K\}} - \sum_{i=1}^n \log P(y = y_i | x = x_i) \\ &= - \sum_{i=1}^n \log \frac{e^{w_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^K e^{w_j^T x_i + b_j}} \end{aligned}$$

- Relation between *Logistic Regression* and *Softmax Regression* with $K = 2$
 - $y = \{1, 2\}, x \in \mathbb{R}^d$
 - we define $w := w_1 - w_2$ and $b := b_1 - b_2$
 - then we have:

$$\begin{aligned} P(y = 1|x) &= \frac{e^{w_1^T x + b_1}}{e^{w_1^T x + b_1} + e^{w_2^T x + b_2}} \\ &= \frac{1}{1 + e^{(w_2 - w_1)^T x + b_2 - b_1}} \\ &= \frac{1}{1 + e^{-(w^T x + b)}} \\ &= \sigma(w^T x + b) \end{aligned}$$