

Logistic Regression

Quách Đình Hoàng

Outline

- Empirical Risk Minimization
- Binary Classification
- Logistic Regression
- Regularized Logistic Regression
- Hyperparameter Tuning
- Multiclass Classification
- Multi-class Logistic Regression

Empirical Risk Minimization – ERM

- ERM is a common framework for supervised learning
- Given:
 - some labelled training dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$
 - a loss function $l: Y \times Y \rightarrow R$
 - a hypothesis class or set of functions F
- The goal is to find

$$\hat{f} = \underset{f \in F}{\operatorname{argmin}} \sum_{i=1}^n l(f(x^{(i)}), y^{(i)})$$

with the hope that

$$E_{p(x,y)}[l(f(x), y)] \approx \frac{1}{n} \sum_{i=1}^n l(f(x^{(i)}), y^{(i)})$$

Empirical Risk Minimization – ERM

- ERM is a common framework for supervised learning

- Given:

- a labelled training dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^n, x^{(i)} \in \mathbb{R}^{d+1}$
- a loss function $l: Y \times Y \rightarrow R$
- a hypothesis class or set of functions F

- The goal is to find

$$\hat{f} = \underset{f \in F}{\operatorname{argmin}} \sum_{i=1}^n l(f(x^{(i)}), y^{(i)})$$

- Depending on the choice of F and l , this objective function may be **convex** (easy to optimize) or **non-convex** (hard)

Binary Classification

- Classification is a type of supervised learning

- Given:

- a labelled training dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$
- a loss function $l: Y \times Y \rightarrow R$, where $Y = \{0,1\}$
- a hypothesis class or set of functions F

- The goal is to find

$$\hat{f} = \underset{f \in F}{\operatorname{argmin}} \sum_{i=1}^n l(f(x^{(i)}), y^{(i)})$$

- Depending on the choice of F and l , this objective function may be **convex** (easy to optimize) or **non-convex** (hard)

Binary Classification with 0/1 Loss

- Classification is a type of supervised learning
- Given:

- a labelled training dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$
 - a loss function $l(y, y') = \delta(y \neq y')$, for $y, y' \in \{0, 1\}$
 - a hypothesis class or set of functions F
- $$\delta(p) = \begin{cases} 1, & \text{if } p \text{ is true} \\ 0, & \text{if } p \text{ is false} \end{cases}$$

- The goal is to find

$$\hat{f} = \underset{f \in F}{\operatorname{argmin}} \sum_{i=1}^n l(f(x^{(i)}), y^{(i)})$$

- This loss function is difficult to optimize (non-convex)

A Probabilistic Approach to Binary Classification

- Suppose we have binary labels $y \in \{0,1\}$ and $(d + 1)$ -dimensional inputs $x = (1, x_1, x_2, \dots, x_d)^T \in \mathbb{R}^{d+1}$
- Assume

$$P(Y = 1|x) = \sigma(\theta^T x) = \frac{1}{1 + e^{-w^T x}} = \frac{e^{w^T x}}{1 + e^{w^T x}}$$

- This implies two useful facts

$$P(Y = 0|x) = 1 - P(Y = 1|x) = \frac{1}{1 + e^{w^T x}}$$

$$\frac{P(Y = 1|x)}{P(Y = 0|x)} = e^{w^T x} \Rightarrow \log \frac{P(Y = 1|x)}{P(Y = 0|x)} = w^T x$$

Logistic Function

- Why use logistic function?
 - Differentiable everywhere
 - $\sigma: \mathbb{R} \rightarrow [0,1]$
 - The decision boundary is linear in x

$$y = \begin{cases} 1, & P(Y = 1|x) \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

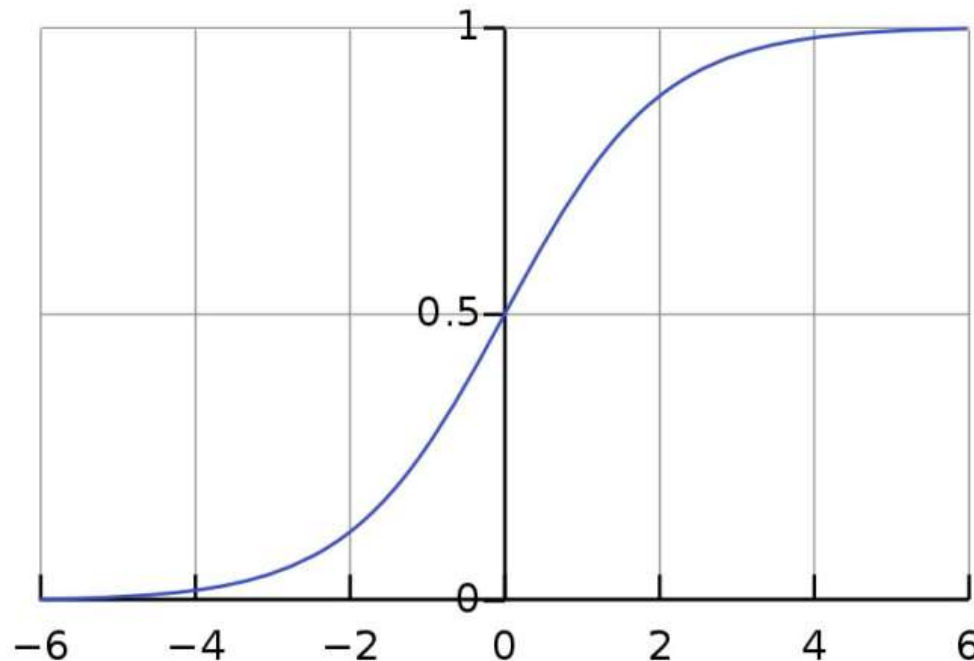
$$P(Y = 1|x) = \sigma(\theta^T x) = \frac{1}{1 + e^{-w^T x}} \geq \frac{1}{2}$$

$$\Rightarrow 2 \geq 1 + e^{-w^T x} \Rightarrow 1 \geq e^{-w^T x}$$

$$\Rightarrow 0 \geq -w^T x$$

$$\Rightarrow w^T x \geq 0$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{1 + e^x} \in \{0,1\}$$



Logistic Regression Decision Boundary



Logistic Regression – Maximum Likelihood (1)

- Goal: find the θ that maximizes the (conditional) probability of the training dataset

$$\prod_{i=1}^n P(y^{(i)}|x^{(i)}, \theta)$$

- This is equivalent to finding the θ that minimizes the negative log of this probability

$$\begin{aligned} L_D(\theta) &= -\log \left(\prod_{i=1}^n P(y^{(i)}|x^{(i)}, \theta) \right) = -\sum_{i=1}^n \log \left(P(y^{(i)}|x^{(i)}, \theta) \right) \\ &= -\sum_{i=1}^n \log \left(P(Y = 1|x^{(i)}, \theta)^{y^{(i)}} P(Y = 0|x^{(i)}, \theta)^{1-y^{(i)}} \right) \\ &= -\sum_{i=1}^n \left(y^{(i)} \log(\sigma(\theta^T x)) + (1 - y^{(i)}) \log(1 - \sigma(\theta^T x)) \right) \end{aligned}$$

Logistic Regression – Maximum Likelihood (2)

- Classification is a type of supervised learning
- Given:
 - a labelled training dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$
 - a loss function $l(y, y') = \delta(y \neq y')$, for $y, y' \in \{0, 1\}$
 - a hypothesis class or set of functions F
- The goal is to find

$$\begin{aligned} & \underset{\theta}{\operatorname{argmin}} - \sum_{i=1}^n \left(y^{(i)} \log(\sigma(\theta^T x)) + (1 - y^{(i)}) \log(1 - \sigma(\theta^T x)) \right) \\ &= \underset{\theta}{\operatorname{argmin}} - \sum_{i=1}^n \left(y^{(i)} \log \left(\frac{\sigma(\theta^T x)}{1 - \sigma(\theta^T x)} \right) + \log(1 - \sigma(\theta^T x)) \right) \\ &= \underset{\theta}{\operatorname{argmin}} - \sum_{i=1}^n \left(y^{(i)} \theta^T x^{(i)} - \log(1 + e^{\theta^T x^{(i)}}) \right) \end{aligned}$$

Logistic Regression – Maximum Likelihood (3)

$$L_D(\theta) = - \sum_{i=1}^n \left(y^{(i)} \theta^T x^{(i)} - \log \left(1 + e^{\theta^T x^{(i)}} \right) \right)$$

$$\nabla_{\theta} L_D(\theta) = - \sum_{i=1}^n \left(y^{(i)} \nabla_{\theta} (\theta^T x^{(i)}) - \nabla_{\theta} \log \left(1 + e^{\theta^T x^{(i)}} \right) \right)$$

⋮

Note: $\frac{\partial \sigma(z)}{\partial z} = \sigma(z)(1 - \sigma(z))$

$$= \sum_{i=1}^n (\sigma(\theta^T x^{(i)}) - y^{(i)}) x^{(i)}$$

Gradient Descent for Logistic Regression

- Dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$
- 1. Initialize $\theta^{(0)} = 0$ (zero vector) and set $t = 0$
- 2. While **not converged**
 - Compute the gradient

$$\nabla_{\theta} L_D(\theta^{(t)}) = 2 \sum_{i=1}^n \left(\sigma(\theta^{(t)T} x^{(i)}) - y^{(i)} \right) x^{(i)}$$

- Update the weights

$$\theta^{(t+1)} = \theta^{(t)} - \eta \sum_{i=1}^n \left(\sigma(\theta^{(t)T} x^{(i)}) - y^{(i)} \right) x^{(i)}$$

- Increment t : $t = t + 1$
- Output $\theta^{(t)}$

Regularized Logistic Regression

- Logistic regression

$$L_D(\theta) = \sum_{i=1}^n (\sigma(\theta^T x^{(i)}) - y^{(i)}) x^{(i)}$$

- Ridge logistic regression (L_2 regularization)

$$\underset{\theta}{\text{minimize}} \left(L_D(\theta) + \frac{\lambda}{2} \|\theta\|_2^2 \right)$$

λ is hyperparameter

- Lasso logistic regression (L_1 regularization)

$$\underset{\theta}{\text{minimize}} \left(L_D(\theta) + \frac{\lambda}{2} \|\theta\|_1 \right)$$

λ is hyperparameter

- ElasticNet logistic regression ($L_1 + L_2$ regularization)

$$\underset{\theta}{\text{minimize}} \left(L_D(\theta) + \lambda \left(\alpha \|\theta\|_1 + \frac{1-\alpha}{2} \|\theta\|_2^2 \right) \right)$$

λ and α are hyperparameters

Hyperparameter Tuning

- Suppose we want to compare multiple hyperparameter settings $\lambda_1, \lambda_2, \dots, \lambda_k$
- For $i = 1, 2, \dots, k$
 - Train a model on D_{train} using λ_i
- Evaluate each model on D_{val} and find the best hyperparameter setting, λ_{i^*}
- Compute the error of a model trained with λ_{i^*} on D_{test}



Multi-class Classification

- Classification is a type of supervised learning
- Given:
 - a labelled training dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$
 - a loss function $l: Y \times Y \rightarrow R$, where $Y = \{1, 2, \dots, k\}$
 - a hypothesis class or set of functions F
- The goal is to find

$$\hat{f} = \underset{f \in F}{\operatorname{argmin}} \sum_{i=1}^n l(f(x^{(i)}), y^{(i)})$$

- Depending on the choice of F and l , this objective function may be **convex** (easy to optimize) or **non-convex** (hard)

Multi-class Logistic Regression

- Suppose we have k classes $y \in \{1, 2, \dots, k\}$ and $(d + 1)$ -dimensional inputs $x = (1, x_1, x_2, \dots, x_d)^T \in \mathbb{R}^{d+1}$
- For each class i , we have a binary classifier with parameter $\theta^{(i)}$
- Then

$$P(Y = i|x) = \frac{e^{\theta^{(i)T} x}}{\sum_{j=1}^k e^{\theta^{(j)T} x}} = \frac{e^{\theta_0^{(i)} + \theta_1^{(i)} x_1 + \dots + \theta_d^{(i)} x_d}}{\sum_{j=1}^k e^{\theta_0^{(j)} + \theta_1^{(j)} x_1 + \dots + \theta_d^{(j)} x_d}}$$

- Assign x to the class that maximizes the (conditional) probability

$$\underset{i}{\operatorname{argmax}} P(Y = i|x) = \underset{i}{\operatorname{argmax}} \theta^{(i)T} x$$

- Multi-class logistic regression is also called multinomial logistic regression or softmax regression

Summay

- Empirical Risk Minimization
- Binary Classification
- Logistic Regression
- Regularized Logistic Regression
- Hyperparameter Tuning
- Multiclass Classification
- Multi-class Logistic Regression