Learning Machine Learning

Pt

October 13, 2020

Contents

1 Glossary

1.1 Odds and Logit

In *Binary Classification Problem*, the probability of $label = 1$ divided by the probability of $label = 0$ is called **Odds**.

$$y = P(label = 1)$$
$$odds = \frac{y}{1-y}$$

Further, take the logarithm of both sides, we got **Log Odds/Logit:**

$$logit = \ln \frac{y}{1-y}$$

2 Loss Functions and Objective Functions

2.1 Minimum Square Loss (MSE)

2.2 Cross-Entropy/Log Loss

For *Binary Classification Problem.* Given that x_i is one of the sample training data, y_i is the corresponding label, then

$$\hat{y_i} = \sigma(h(x_i|\theta)) \in \mathbb{R}$$
$$\mathcal{L}(y_i, \hat{y_i} \mid \theta) = \begin{cases} -\log(\hat{y_i}) & y_i = 1 \\ -\log(1-\hat{y_i}) & y_i = 0 \end{cases}$$

Compress into one equation, then

$$\mathcal{L}(y_i, \hat{y_i} \mid \theta) = -[y_i * \log(\hat{y_i}) + (1-y_i) * \log(1-\hat{y_i})]$$

More generally, for *Multi-Classification Problem*, given that

- x_i is one of the sample training data, which will be classified into one of k categories,
- $y_i \in \mathbb{R}^k$ is the **One-Hot Representation** of the corresponding label

the **Cross-Entropy Loss** is

$$\hat{y_i} = softmax(h(x_i|\theta)) \in \mathbb{R}^k$$
$$\mathcal{L}(y_i, \hat{y_i} \mid \theta) = -\sum_{j}^{n} y_i[j] * log_2(\hat{y_i}[j])$$

2.3 Maximum (Log-)Likelihood

Given the probability of a series of accidents $A_i, i \in 1, 2, \ldots, k$ is $P(A_i)$, then we want to **maximize** the probability that all of these accidents happen, thus

$$max\{\prod_{i=0}^{k} P(A_i)\}$$

To simplify, we can take the logarithm of both sides, then

$$max\{\ln \sum_{i=0}^{k} P(A_i)\}$$

which is namely **Maximum (Log) Likelihood**. While the **Loss Function** derived from above is natually:

$$\mathcal{L}(y_i, \hat{y} \mid \theta) = -\ln \sum_{i=0}^{k} P(A_i)$$

which we want to **Minimize**.

3 Probability

3.1 Basis

- Priori Probability: the probability which can be empirically inferred
- Posterior Probability: after A happening, sought the probability of the reason of A
- Bayesian Equation

3.2 Conditional Indipendent

$$p(A \mid C) * p(B \mid C) = p(AB \mid C)$$

3.3 Distributions

3.3.1 Bernoulli distribution

Defiened as is the discrete probability distribution of a random variable which takes the value 1 with probability p and the value 0 with probability $q = 1 - p$. Denote $B(1, p)$ as *Bernoulli distribution*, then

$$\begin{aligned} Given\, X &\sim B(1, p) \\ E(x) &= p; \\ D(x) &= p(1-p); \end{aligned}$$

4 Matrix

4.1 Differentiate/Derivation

$$\begin{aligned} Y &= A \cdot X \cdot B \\ \frac{\partial Y}{\partial X} &= A^T \cdot B^T \\ \frac{\partial Y}{\partial X^T} &= B \cdot A \end{aligned}$$

Another scenario,

$$\begin{aligned} Y &= X^T \cdot A \cdot X \\ \frac{\partial Y}{\partial X} &= (A + A^T) \cdot X \end{aligned}$$

5 Normalization

5.1 Scale

Given $p \in \mathbb{R}^k$, where p is the result of *Dot-Product* in *Dot-Product Attention Mechanism*, in order to counteract gradient vanishing in *softmax*, scale p by

$$p_{norm} = \frac{p}{\sqrt{k}}$$

Given that $x \in \mathbb{R}^k$ is one of the record of the datasets, which is a **real-valued input vector**, in order to make *Gradient Descent* faster and more efficient, we'd better scale the value of input features to $-1 \leq x_i \leq 1$, or at least the scale of different input features is similar.

$$\bar{x_i} = \frac{x_i - \mu}{max(x_i) - min(x_i)}$$

where μ is the mean value of x_i over the datasets;