A Unified View of Loss Functions in Supervised Learning

Shuiwang Ji
Texas A&M University
College Station, TX 77843
sji@tamu.edu

Lei Cai
Washington State University
Pullman, WA 99164
lei.cai@wsu.edu

1 Introduction

This note aims at providing an introduction of different loss functions for supervised learning and shedding light on their differences and similarities. This document is based on lecture notes by Shuiwang Ji at Texas A&M University and can be used for undergraduate and graduate level classes.

2 Linear Classifier

For a binary classification problem, we are given an input dataset $X = [x_1, x_2, \dots, x_n]$ with the corresponding label $Y = [y_1, y_2, \dots, y_n]$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{+1, -1\}$. For a given sample x_i, a linear classifier computes the linear score s_i as a weighted summation of all features as:

$$s_i = w^T x_i + b, \tag{1}$$

where w is the weights and b is the bias. We can predict the label of x_i based on the linear score s_i. By employing an appropriate loss function, we can train and obtain a linear classifier. In the following sections, we will introduce the loss functions for linear classifier, including zero-one loss, hinge loss, and log loss (also known as logistic regression loss or cross entropy loss).

3 Zero-one Loss

Zero-one loss aims at measuring the number of prediction errors for linear classifier. The perceptron employs zero-one loss as its loss function. For a given input x_i, if $y_i s_i > 0$, then the perceptron makes a correct prediction. Otherwise, it makes a wrong prediction. Therefore, the zero-one loss function can be described as follows:

$$\sum_{i=1}^{n} L_{0/1}(y_i, s_i), \tag{2}$$

where $L_{0/1}$ is the zero-one loss defined as

$$L_{0/1}(y_i, s_i) = \begin{cases} 1 & \text{if } y_i s_i < 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

4 Hinge Loss

The support vector machine employs hinge loss to obtain a classifier with “maximum-margin”. The loss function in the support vector machine is defined as follows:

$$\sum_{i=1}^{n} L_h(y_i, s_i), \tag{4}$$

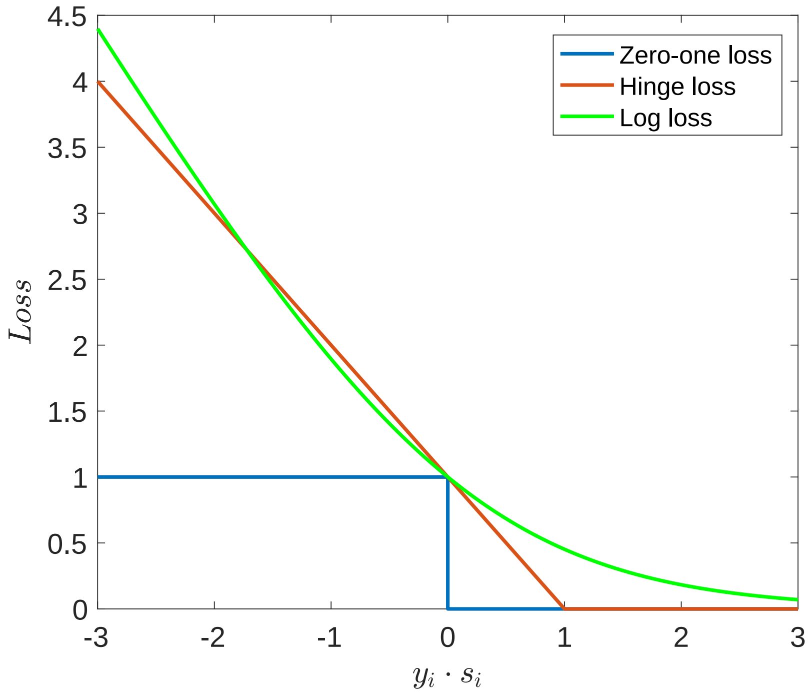

Figure 1: Comparison of different loss functions.

where L_h is the hinge loss:

$$L_h(y_i, s_i) = \max(0, 1 - y_i s_i). \tag{5}$$

Different with the zero-one loss, a data may be penalized even if it is predicted correctly.

5 Log loss

99%的时候用这个

Logistic regression employs the log loss to train classifiers. The loss function used in logistic regression can be expressed as

$$\sum_{i=1}^{n} L_{log}(y_i, s_i), \tag{6}$$

where L_{log} is the log loss, defined as

$$L_{log}(y_i, s_i) = \underline{\log(1 + e^{-y_i s_i})}. \tag{7}$$

6 Convexity

In mathematics, a function $f(\cdot)$ is convex if

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2), \text{for } t \in [0, 1].$$

A function $f(\cdot)$ is strictly convex if

$$f(tx_1 + (1-t)x_2) < tf(x_1) + (1-t)f(x_2), \boxed{\text{for } t \in (0, 1),\ x_1 \neq x_2.}$$

Intuitively, a function is convex if the line segment between any two points on the function is not below the function. A function is strictly convex if the line segment between any two distinct points on the function is strictly above the function, except for the two points on the function itself.

7 Comparison of Loss Functions

In order to compare the differences and similarities among different loss functions, we show the different loss functions in Figure 1. In the zero-one loss, if a data sample is predicted correctly ($y_i s_i > 0$), it results in zero penalties; otherwise, there is a penalty of one. For any data sample that is not predicted correctly, it receives the same penalty. However, a data sample can still incur penalty even if it is classified correctly in hinge loss. The log loss is similar to the hinge loss. But the log loss is a smooth function and it can be optimized with the gradient descent method. In comparison, the hinge loss is not smooth. Both hinge loss function and log loss function are convex functions, but zero-one loss is not convex. Also the hinge loss is not strictly convex, while the log loss is strictly convex.

Acknowledgements

This work was supported in part by National Science Foundation grants IIS-1908220, IIS-1908198, IIS-1908166, DBI-1147134, DBI-1922969, DBI-1661289, CHE-1738305, National Institutes of Health grant 1R21NS102828, and Defense Advanced Research Projects Agency grant N66001-17-2-4031.