Loss Functions

Dr. Parmalik Kumar (CSE)

PGOI (PCST, BHOPAL)

A loss function, or cost function, is a wrapper around our model's predict function that tells us "how good" the model is at making predictions for a given set of parameters. The loss function has its own curve and its own derivatives. The slope of this curve tells us how to change our parameters to make the model more accurate! We use the model to make predictions. We use the cost function to update our parameters. Our cost function can take a variety of forms as there are many different cost functions available. Popular loss functions include: MSE (L2) and Cross-entropy Loss.

Cross-Entropy

Hinge

Huber

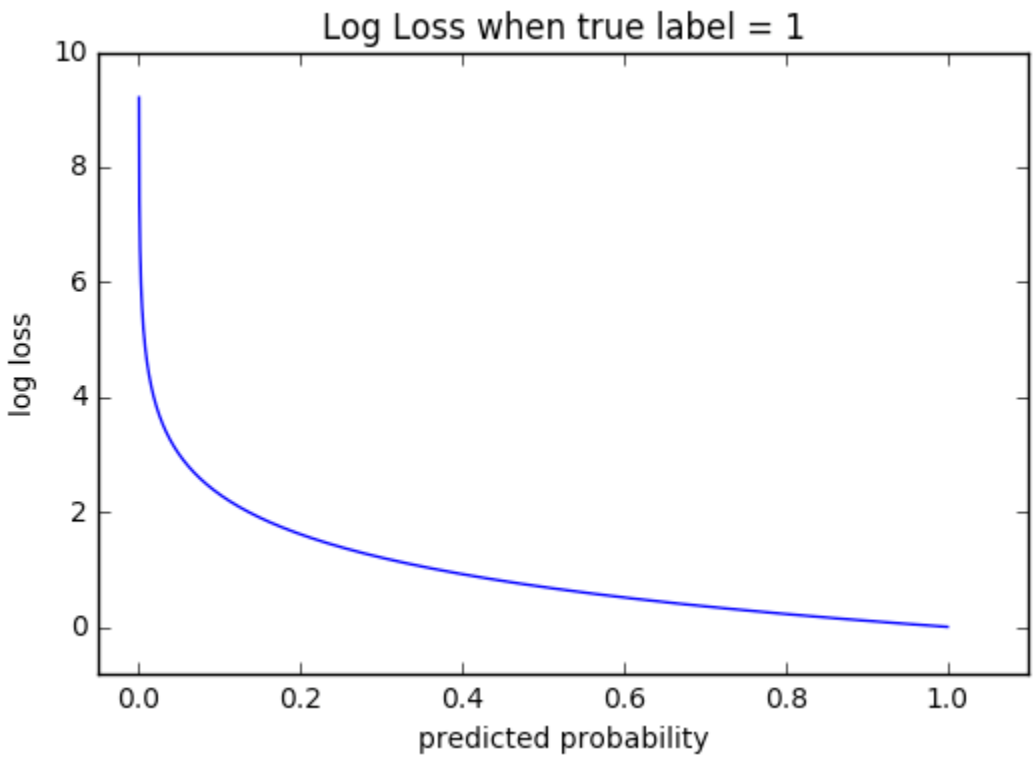
Kullback-Leibler

MAE (L1)

MSE (L2)

Cross-Entropy

Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. So, predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high loss value. A perfect model would have a log loss of 0.



The graph above shows the range of possible loss values given a true observation (isDog = 1). As the predicted probability approaches 1, log loss slowly decreases. As the predicted probability decreases, however, the log loss increases rapidly. Log loss penalizes both types of errors, but especially those predictions that are confident and wrong!

Cross-entropy and log loss are slightly different depending on context, but in machine learning when calculating error rates between 0 and 1 they resolve to the same thing.

In binary classification, where the number of classes M equals 2, cross-entropy can be calculated as:

$$-(y\log(p)+(1-y)\log(1-p))$$

If $M>2$ (i.e. multiclass classification), we calculate a separate loss for each class label per observation and sum the result.

$$-\sum_{c=1}^{M} yo,clog(po,c)$$

- M - number of classes (dog, cat, fish)
- log - the natural log
- y - binary indicator (0 or 1) if class label cc is the correct classification for observation oo
- p - predicted probability observation o is of class c

Hinge

In machine learning, the hinge loss is a loss function used for training classifiers. The hinge loss is used for maximum-margin classification. The expression of hinge loss function used in support vector machine.

The predict output t=+-1 and a classifier score y, the hinge loss of the prediction Y is defined as

L(y)=max(0,1-t.y)

Huber

In statistics, the Huber loss is a loss function used in robust regression, that is less sensitive to outliers in data than the squared error loss. A variant for classification is also sometimes used. The Huber loss function describes the penalty incurred by an estimation procedure *f*. Huber defines the loss function piecewise by

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \le \delta, \\ \delta(|a| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases}$$

This function is quadratic for small values of *a*, and linear for large values, with equal values and slopes of the different sections at the two points where . The variable *a* often refers to the residuals,

that is to the difference between the observed and predicted values , so the former can be expanded to

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for} |y - f(x)| \leq \delta, \\ \delta\,|y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

Kullback-Leibler

The Kullback-Leibler Divergence score, or KL divergence score, quantifies how much one probability distribution differs from another probability distribution. The KL divergence between two distributions Q and P is often stated using the following notation:

KL(P || Q)

Where the "||" operator indicates "divergence" or Ps divergence from Q.

KL divergence can be calculated as the negative sum of probability of each event in P multiplied by the log of the probability of the event in Q over the probability of the event in P.

KL(P || Q) = – sum x in X P(x) * log(Q(x) / P(x))

The value within the sum is the divergence for a given event.

This is the same as the positive sum of probability of each event in P multiplied by the log of the probability of the event in P over the probability of the event in Q (e.g. the terms in the fraction are flipped). This is the more common implementation used in practice.

KL(P || Q) = sum x in X P(x) * log(P(x) / Q(x))

The intuition for the KL divergence score is that when the probability for an event from P is large, but the probability for the same event in Q is small, there is a large divergence. When the probability from P is small and the probability from Q is large, there is also a large divergence, but not as large as the first case.

It can be used to measure the divergence between discrete and continuous probability distributions, where in the latter case the integral of the events is calculated instead of the sum of the probabilities of the discrete events.

One way to measure the dissimilarity of two probability distributions, p and q, is known as the Kullback-Leibler divergence (KL divergence) or relative entropy.

MAE(L1)

In statistics, mean absolute error (MAE) is a measure of errors between paired observations expressing the same phenomenon. Examples of Y versus X include comparisons of predicted versus observed, subsequent time versus initial time, and one technique of measurement versus an alternative technique of measurement. MAE is calculated as:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}.$$

MSE(L2)

L2-norm loss function is also known as least squares error (LSE). It is basically minimizing the sum of the square of the differences (S) between the target value (Yi) and the estimated values (f(xi)

$$S = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

The differences of L1-norm and L2-norm as a loss function can be promptly summarized as follows:

L2 loss function	L1 loss function
Not very robust	Robust
Stable solution	Unstable solution
Always one solution	Possibly multiple solutions