Regression loss functions for machine learning

Overview

Loss functions take as input a set of predictions and actual values, and return a metric of the prediction error. This “prediction error metric” guides the machine learning model during training. Model training often consists of a model tuning its inner workings to minimize the output of the loss function for the training data set.

Three of the most useful loss functions for regression problems are described below: mean squared error, mean absolute error and Huber loss. Recommendations about when to apply each of them are also included.

This post focuses on regression problems (i.e. the output variable takes continuous values). If you would like us to cover classification problems (i.e. the output variable takes class labels) on another blog entrance, just email us.

Mean squared error

The most common loss function for regression problems is the mean squared error (MSE). The MSE is calculated by the sum of the squared distance between the target variable (y_i) and its predicted value (y_i^p):

$$MSE = \frac{\sum_{i=1}^{n}\left(y_i^p - y_i\right)^2}{n}.$$

The mean squared error function is widely used as it is simple, continuous and differentiable.

A key MSE characteristic is its disproportional sensitivity to large errors compared to small ones. A model trained with MSE will give the same importance to a single error of 5 units compared to 25 errors of 1 unit. In other words, the model will be biased to reduce the largest errors, even if that penalizes the predictions of many common conditions.

The above MSE characteristic is especially important when dealing with outliers that the model fails to predict. These could be outliers caused by corrupted data or random unpredictable processes. A small number of outliers very distant from other observations can impair the model's predictive ability. Figure 1 shows the prediction of a linear model trained with a mean squared error loss function. The training data is formed by 15 instances, one of them an outlier (see upper right corner in Figure 1). Even though most of the training data set can be well represented by a linear model, the outlier distorts the model prediction when MSE is applied.

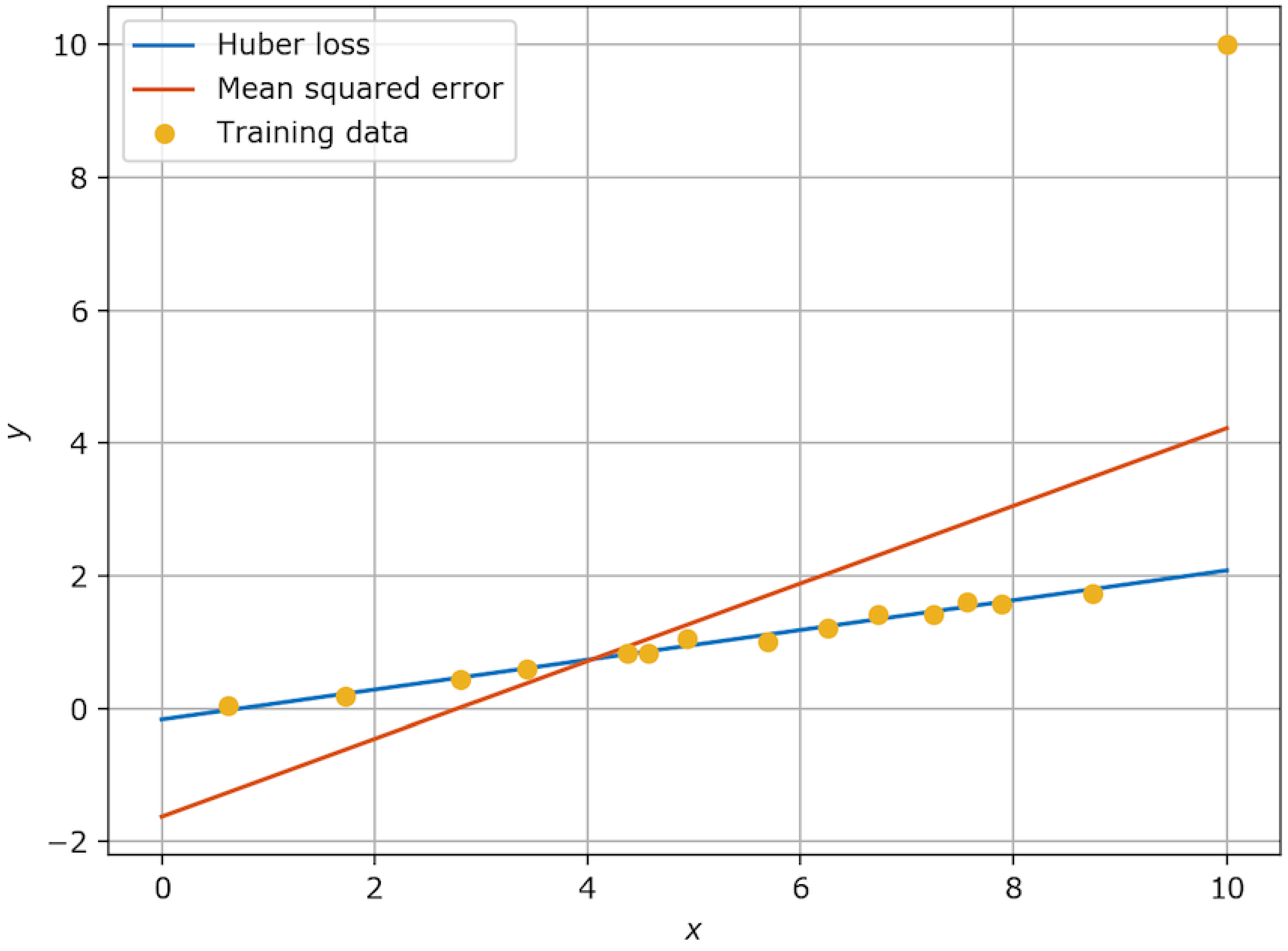

Figure 1: Model prediction with outlier in training data. Comparison of Huber loss and MSE.

Mean absolute error

The mean absolute error (MAE) is the sum of the absolute differences between the actual and the predicted values:

$$MAE = \frac{\sum_{i=1}^{n} \left| y_i^p - y_i \right|}{n}.$$

The MAE method has the advantage of not being overly affected by outliers in the training data. Using the previously discussed example, a model trained with the MSA approach will give equal importance to 1 error of 5 units and 5 errors of 1 unit.

The main issue with the MAE is that it is not differentiable at its minimum, see Figure 2. This lack of differentiability can produce convergence issues when training machine learning models.

Huber loss

The Huber loss approach combines the advantages of the mean squared error and the mean absolute error. It is a piecewise-defined function:

$$Hubber\ Loss = \begin{cases} \frac{1}{2}\left(y - y_p\right)^2, & \left|y - y_p\right| \le \delta \\ \delta\left|y - y_p\right| - \frac{1}{2}\delta^2, & \left|y - y_p\right| > \delta \end{cases},$$

where δ is a hyperparameter that controls the split between the two sub-function intervals. The sub-function for large errors, such as outliers, is the absolute error function. Hence, it avoids the excessive sensitivity to large errors that characterizes MSE. The sub-function for small errors is the squared error making the whole function continuous and differentiable, which overcomes MAE's convergence issues. Figure 2 shows the value of squared error, absolute error and the Huber loss as a function of the prediction error. The Huber loss can be seen to be proportional to the absolute value, except for small errors, where it is proportional to the square of the error.

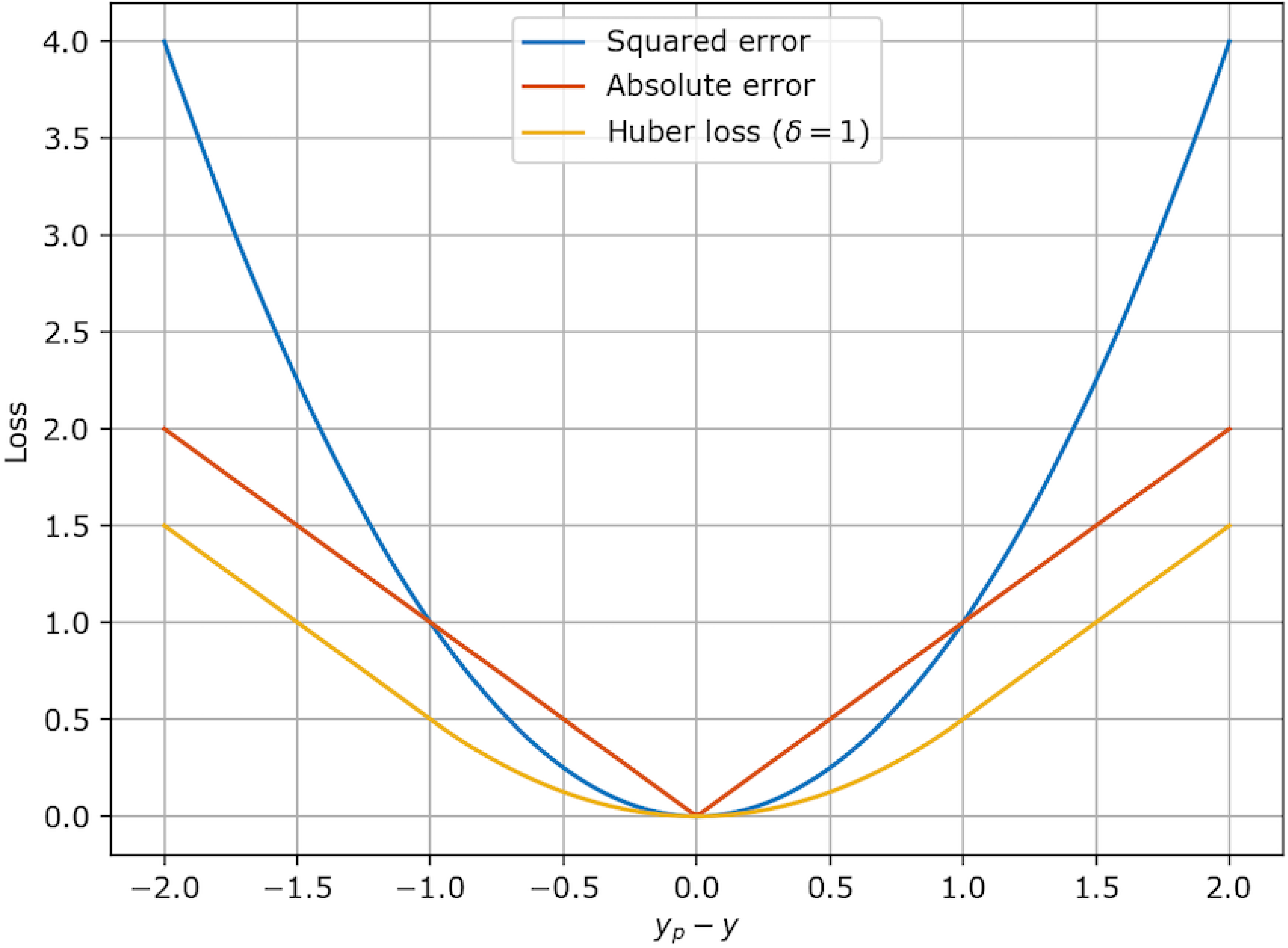

Figure 2: Squared, absolute and Huber loss functions.

No size fits all in machine learning, and Huber loss also has its drawbacks. Its main disadvantage is the associated complexity. In order to maximize model accuracy, the hyperparameter δ will also need to be optimized which increases the training requirements.

Selecting a loss function

To sum up, we recommend MSE as a default option. It works sufficiently well for the majority of machine learning problems, it is simple, mathematically robust and well supported by most machine learning libraries.

If the training data has outliers that the model fails to predict, the model accuracy with MSE may be affected. The following three options arises:

1. The most accurate approach is to apply the Huber loss function and tune its hyperparameter δ. The hyperparameter should be tuned iteratively by testing different values of δ.

2. The fastest approach is to use MAE. This should be done carefully, however, as convergence issues may appear.

3. If the outliers are not critical to the dataset, MSE can also be applied after removing the outliers from the training data.

What next

In a separate post, we will discuss the extremely powerful quantile regression loss function that allows predictions of confidence intervals, instead of just values.

If you have any questions or there any machine learning topic that you would like us to cover, just email us.