Differences between L1 and L2 as Loss Function and Regularization

POSTED ON DEC 18, 2013 • LO

[2014/11/30: Updated the L1-norm vs L2-norm loss function via a programmatic validated diagram. Thanks readers for the pointing out the confusing diagram. Next time I will not draw mspaint but actually plot it out.]

While practicing machine learning, you may have come upon a choice of the mysterious L1 vs L2. Usually the two decisions are : 1) L1-norm vs L2-norm loss function; and 2) L1-regularization vs L2-regularization.

As An Error Function

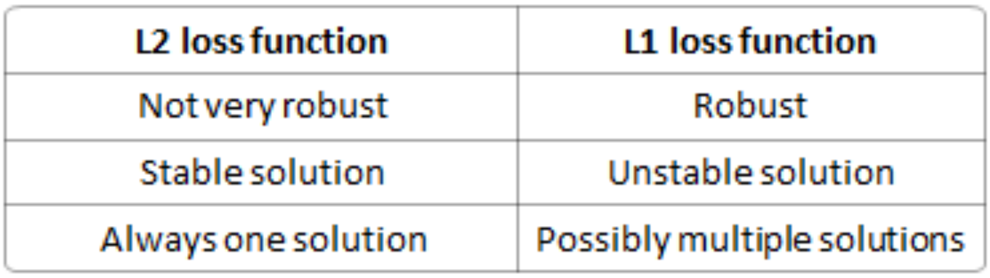
L1-norm loss function is also known as least absolute deviations (LAD), least absolute errors (LAE). It is basically minimizing the sum of the absolute differences **(S)** between the target value (**Y**$_i$) and the estimated values (**f(x**$_i$**)**):

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

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 (**Y**$_i$) and the estimated values (**f(x**$_i$**):**

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

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

Robustness, per wikipedia, is explained as:

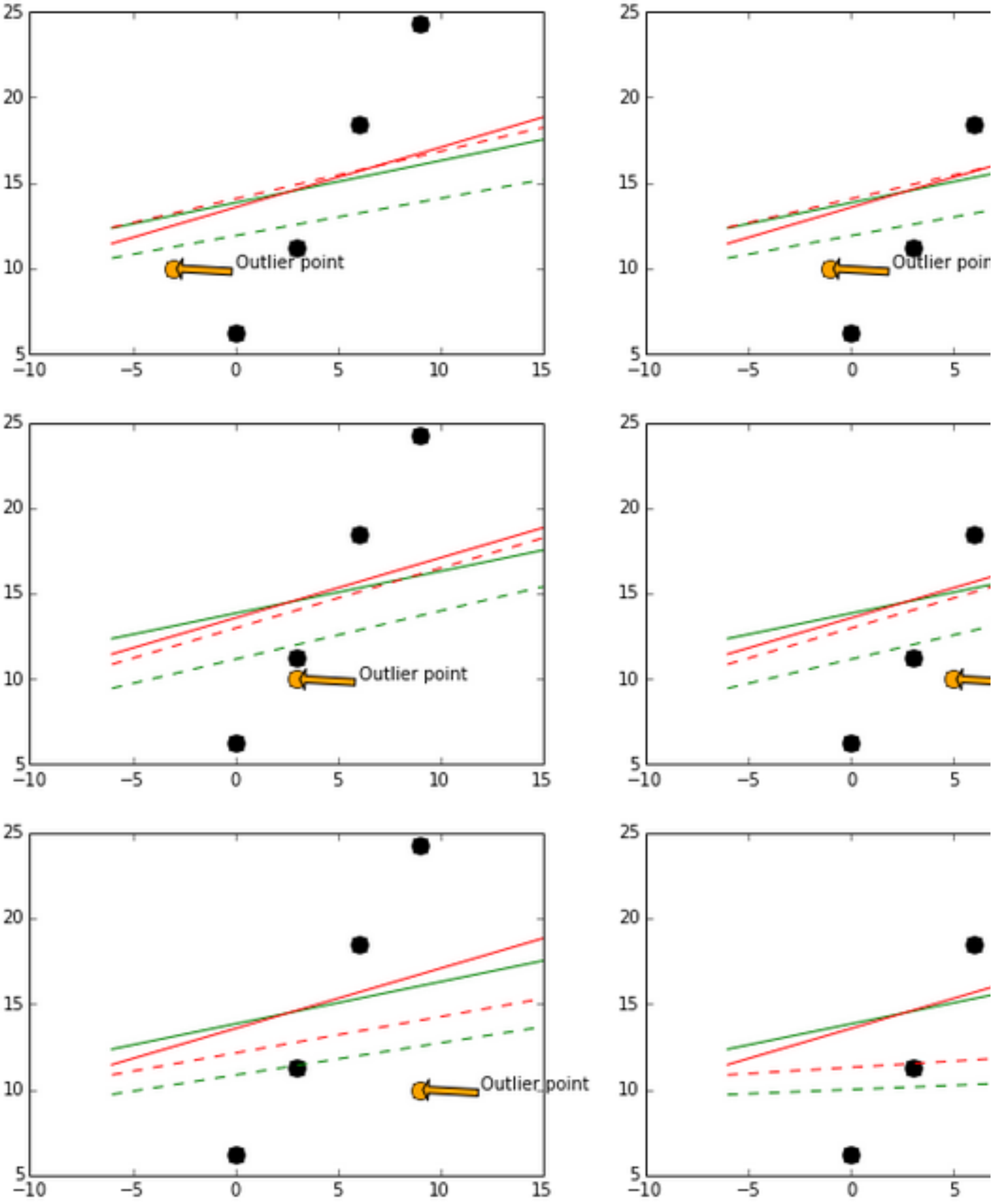
The method of least absolute deviations finds applications in many areas, due to its robustness compared to the least squares method. Least absolute deviations is robust in that it is resistant to outliers in the data. This may be helpful in studies where outliers may be safely and effectively ignored. If it is important to pay attention to any and all outliers, the method of least squares is a better choice.

Intuitively speaking, since a L2-norm squares the error (increasing by a lot if error > 1), the model will see a much larger error (e vs e^2) than the L1-norm, so the model is much more sensitive to this example, and adjusts the model to minimize this error. If this example is an outlier, the model will be adjusted to minimize this single outlier case, at the expense of many other common examples, since the errors of these common examples are small compared to that single outlier case.

Stability, per wikipedia, is explained as:

The instability property of the method of least absolute deviations means that, for a small horizontal adjustment of a datum, the regression line may jump a large amount. The method has continuous solutions for some data configurations; however, by moving a datum a small amount, one could "jump past" a configuration which has multiple solutions that span a region. After passing this region of solutions, the least absolute deviations line has a slope that may differ greatly from that of the previous line. In contrast, the least squares solutions is stable in that, for any small adjustment of a data point, the regression line will always move only slightly; that is, the regression parameters are continuous functions of the data.

Below is a diagram generated using a real data and a real fitted model:


Outlier point
Outlier poir
Outlier point
Outlier point

The base model here used is a GradientBoostingRegressor, which can take in L1-norm and L2-norm loss functions. The green and red lines represent a model using L1-norm and L2-norm loss function respectively. A solid line represents the fitted model trained also with the outlier point (orange), and the dotted line represents the fitted model trained without the outlier point (orange).

I gradually move the outlier point from left to right, which it will be less “outlier” in the middle and more “outlier” at the left and right side. When the outlier point isless “outlier” (in the middle), L2-norm has less changes while the fitted line using L1-norm has more changes.

In the case of a more “outlier” point (upper left, lower right, where points are to the far left and far right), both norms still have big change, but again the L1-norm has more changes in general.

By visualizing data, we can get a better idea what stability is with respective to these two loss functions.

As Regularization

Regularization is a very important technique in machine learning to prevent overfitting. Mathematically speaking, it adds a *regularization term* in order to prevent the coefficients to fit so perfectly to overfit. The difference between the L1 and L2 is just that L2 is the sum of the square of the weights, while L1 is just the sum of the weights. As follows:

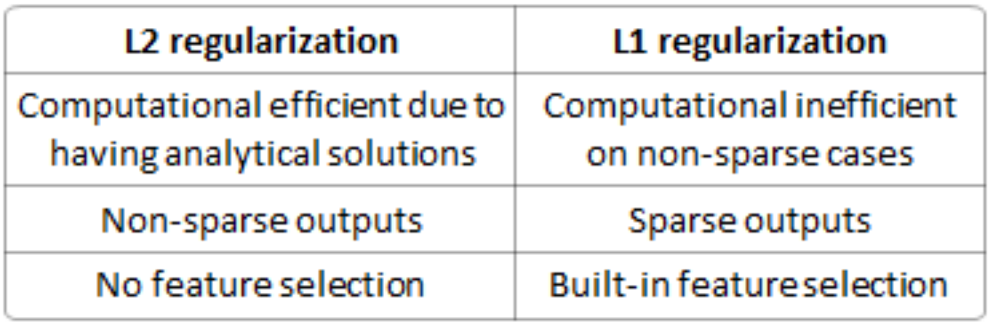
L1 regularization on least squares:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2 + \boxed{\lambda \sum_{i=1}^{k} |w_i|}$$

L2 regularization on least squares:

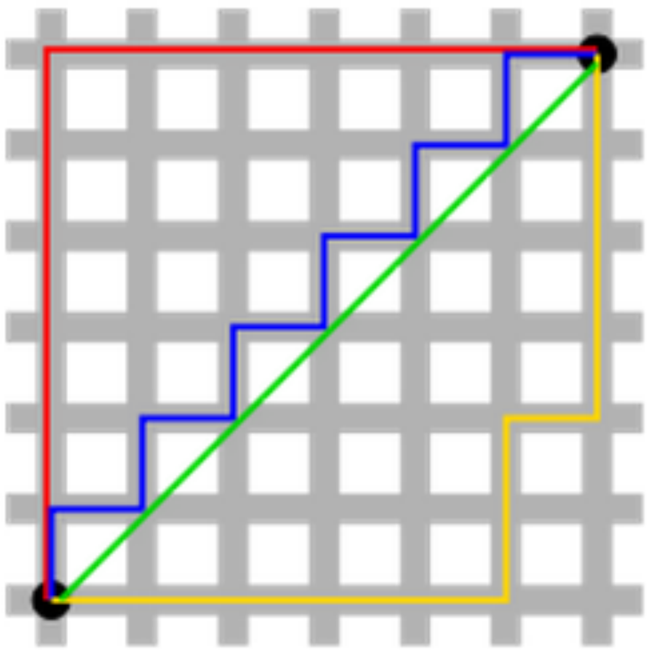
$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2 + \boxed{\lambda \sum_{i=1}^{k} w_i^2}$$

The difference between their properties can be promptly summarized as follows:



L2 regularization	L1 regularization
Computational efficient due to having analytical solutions	Computational inefficient on non-sparse cases
Non-sparse outputs	Sparse outputs
No feature selection	Built-in feature selection

Solution uniqueness is a simpler case but requires a bit of imagination. First, this picture below:



The green line (L2-norm) is the unique shortest path, while the red, blue, yellow (L1-norm) are all same length (=12) for the same route. Generalizing this to n-dimensions. This is why L2-norm has unique solutions while L1-norm does not.

Built-in feature selection is frequently mentioned as a useful property of the L1-norm, which the L2-norm does not. This is actually a result of the L1-norm, which tends to produces sparse coefficients (explained below). Suppose the model have 100 coefficients but only 10 of them have non-zero coefficients, this is effectively saying that "the other 90 predictors are useless in predicting the target values". L2-norm produces non-sparse coefficients, so does not have this property.

Sparsity refers to that only very few entries in a matrix (or vector) is non-zero. L1-norm has the property of producing many coefficients with zero values or very small values with few large coefficients.

Computational efficiency. L1-norm does not have an analytical solution, but L2-norm does. This allows the L2-norm solutions to be calculated computationally efficiently. However, L1-norm solutions does have the sparsity properties which allows it to be used along with sparse algorithms, which makes the calculation more computationally efficient.

References:

[*[2014/11/30: Updated the L1-norm vs L2-norm loss function via a programmatic validated diagram. Thanks readers for the pointing out the confusing diagram. Next time I will not draw mspaint but actually plot it out.]*

While practicing machine learning, you may have come upon a choice of the mysterious L1 vs L2. Usually the two decisions are : 1) L1-norm vs L2-norm loss function; and 2) L1-regularization vs L2-regularization.