Boosting

Boosting is a process of converting multiple weak learners into a strong learner.

A weak learner is defined as a learning algorithm that consistently performs slightly better than random guessing.

A strong learner is a learning algorithm that can consistently give an arbitrarily high accuracy.

There are various boosting techniques.

Any learning algorithm can just be plugged into these techniques without any modification to give a much better accuracy.

One such widely used technique is **AdaBoost**

ADABoost

AdaBoost is short for Adaptive Boosting is a way to combine many weak learners to make a strong learner.

Weak learner is a learning algorithm that performs consistently better than random guessing.

AdaBoost combines many such learners to form a hypothesis that gives arbitrarily high accuracy.

The AdaBoost algorithm is adaptive, which means that it learns from its mistakes

In short,more weak classifiers combine and form a single strong classifier.

How ADABoost works

Lets see how AdaBoost works for binary classification tasks.

Given a training set, AdaBoost learns a simple and weak classifier in its first iteration.

It notes the examples that this weak classifier incorrectly classifies and increases their weight for the next iteration.

In the next iteration, it agains learns a new weak classifier that considers the increased weights of the previously incorrectly classified examples.

Now we have 2 weak classifiers.

The algorithm again notes the incorrectly classified examples by the second classifier and increases their weights for the next iteration and so on till it gets multiple classifiers.

A weighted sum of all these classifiers gives the final boosted classifier. The weights are a function of the accuracy of each classifier.

Modifying weights

AdaBoost requires a learning algorithm that can account for weighted input examples

i.e. the loss function should give more importance to examples with higher weights.

In case we have an algorithm that does not support weighted inputs, we can use sampling.

At every iteration, a classifier is learnt on a sampled training data.

The training data is sampled according to a probability distribution that mimics the weight distribution.

Samples with higher weight are repeated in the sampled set.

If the weights are normalized i.e if they sum to 1, the weight distribution can be used as the probability distribution for sampling.

Pseudo Algorithm

Consider the input training set –

$$X = \{x_0, x_1, x_2, \ldots, x_n\}$$

and corresponding labels which are either -1 or +1

For t in 1 to T

1. Set weights of all training examples
2. Train a weak classifier c_t that minimizes
3. Set
4. Update weights of training samples

The final output classifier, which classifies a new sample x, is given by

$$C(x) = \text{sign}\left(\sum_{i=1}^{T} \alpha_i c_i(x)\right)$$