

06 - AdaBoost

Ethan Graham

20th June 2023

General Idea

The idea is to combine linear classifiers on non-linearly separable data. We call these classifiers *weak* and write a *strong* classifier as a linear combination of the weak ones. We build this iteratively. $y(\mathbf{x}) = \alpha_1 y_1(\mathbf{x}) + \dots + \alpha_p y_p(\mathbf{x})$

AdaBoost Algorithm

For a training set $\chi = \{\mathbf{x}_n, t_n\}$ where $t_n \in \{-1, 1\}$ and for $1 \leq n \leq N$:

1. Initialize weights at $1/N$ for all points.
2. for $t = [1, \dots, T]$:
 - a. Find classifier $y_t : \chi \rightarrow \{-1, 1\}$ that minimizes weight error of misclassified points.
 - b. fuck it i'm too lazy to write the formula. Not super important at this stage tbh...
 - c. update weights.

The final classification of a point is sign of the sum of all classifications made by weak classifiers.

Changing the weak learners

We can change the weak learners (to make them squares/circles instead of lines for example, in a 2-D setting)

Training Adaboost

The **training** error goes down exponentially if the weighted errors of the component classifiers is always strictly inferior to 0.5. However the **testing** error may rise again over time due to overfitting. \rightarrow use a validation set!

Failure mode

Sometimes data is distributed in such a way that not all classifiers could possibly work.

Cascading

Reject large portions of (for example) images in the first stages can allow for a large potential speedup at run-time (inference)