Improving Classification with the AdaBoost meta-algorithm

Sahel Mohammad Iqbal

January 21, 2022

National Institute of Science Education and Research

Problem Statement

Problem Statement

- For a classification problem (assume binary), we are given a "weak classifier".
- Weak classifier - Classifier that performs just slightly better than random guessing ($>$ 50% accuracy).
- Can we combine multiple instances of the weak classifier to obtain a strong classifier?

Meta-algorithms

- Methods that combine multiple classifiers are called ensemble methods or meta-algorithms.
- Bagging and boosting are two common types.

Bagging and Boosting

Bagging

- Given a dataset X, we randomly sample X (with replacement) S times to make S new datasets of equal size as X.
- The weak classifier is applied to each dataset individually.
- To classify a new data point, we apply our S classifiers to the new data points and take a majority vote.

Boosting

- Sequential use of classifiers over T rounds.
- In each subsequent round, the data points that were misclassified in the previous round are given higher priority.
- AdaBoost is the most popular boosting algorithm.

AdaBoost

AdaBoost

- To demonstrate the algorithms, we'll use decision stumps as the weak classifier.
- Decision stumps are decision trees of depth one which classify data points based on just one feature and one threshold.

AdaBoost

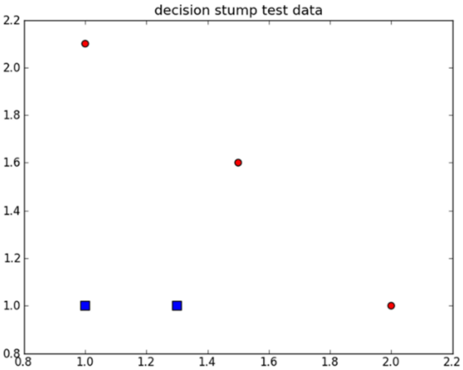

Figure 7.2 Simple data used to check the AdaBoost building functions. It's not possible to choose one threshold on one axis that separates the squares from the circles. AdaBoost will need to combine multiple decision stumps to classify this set without error.

Figure 1: Sample data for decision stumps.

AdaBoost Pseudocode

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right)$.
- Update:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

Figure 2: AdaBoost pseudocode [1].

AdaBoost Schematic

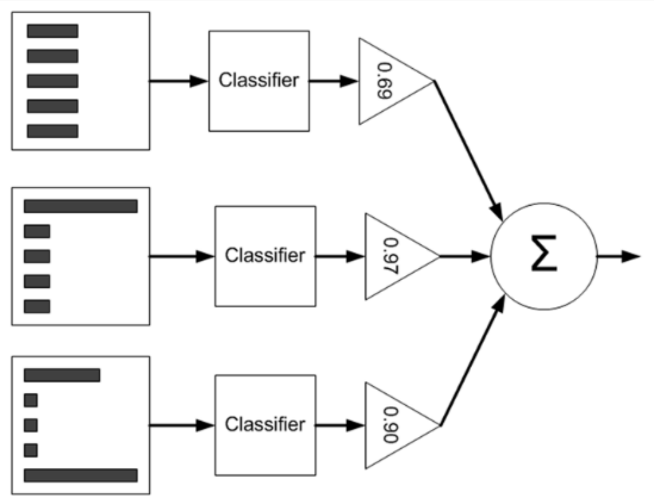

Figure 3: Schematic representation of AdaBoost.

Why this formula for α?

- If α takes the given form and $\alpha > 0$, it can be shown that the classification error exponentially decreases over multiple rounds [2].
- $\alpha_t \geq 0$ if $\epsilon_t \leq 1/2$, which is why we require the weak classifier to have greater than 50% classification accuracy.

Class Imbabalance

What is it?

- Let's say we're building a classifier to detect a rare brain tumor from MRI scans.
- In the dataset, for every positive sample there are 100,000 negative samples.
- A model that seeks to minimize classification error will perform poorly at detecting cancer patients.

How do we detect it?

- Classification error doesn't cut it, we need alternative performance metrics.
- Confusion matrix is useful here.

		redicted	
		+1	-1
Actual	+1	True Positive (TP)	False Negative (FN)
	-1	False Positive (FP)	True Negative (TN)

Figure 4: Confusion matrix for a binary classification problem.

How do we detect it?

- Precision $= \frac{TP}{TP+FP}$ = fraction of records that were positive from the group that the classifier predicted to be positive.
- Recall $= \frac{TP}{TP+FN}$ = fraction of positive examples the classifier got right.
- Very useful when used together.

How do we address it?

1. Manipulate the cost matrix.
2. Resample during training.

		Predicted	
		+1	-1
Actual	+1	0	1
	-1	1	0

		Predicted	
		+1	-1
Actual	+1	-5	1
	-1	50	0

Figure 5: Typical (top) and modified (bottom) cost matrices.

1. Freund, Y., Schapire, R. & Abe, N. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence* **14,** 1612 (1999).
2. Freund, Y. & Schapire, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences* **55,** 119–139. ISSN: 0022-0000. https://www.sciencedirect.com/science/article/pii/S002200009791504X (1997).

Why the name?

- Let the training error ϵ_t of h_t be given by $\frac{1}{2} - \gamma_t$.
- Previous learning algorithms required that γ_t be known a priori before boosting begins.
- AdaBoost adapts to the error rates of the individual weak hypotheses, thus the name 'adaptive'.