Adaboost Project Presentation

By Moses Mbabaali

Adaboost in detail.

Why was Adaboost discovered?

- To take care of the problem of bias for weak binary classifiers.
- In the pursuit to eliminate bias, Adaboost uses an ensemble $(h_i(x))$ of weak binary classifiers to solve a classification problem.
- Each individual binary classifier $h_i(x)$ produces a result and they are all put together to for the final computation.

Setting the stage for Adaboost

- Given N training samples where $X=\{(x_i, y_i)\}$ and $y_i \in (-1,1)$
- To initialize the parameters of the adaboost classifier, the number of iterations *T* of the classifier are known, the weak classifier to be used and the weights are initialized. In my case the weak learner is the Decision Stump and the Support Vector Classifier.
- The weights ${}^{i}_{j}w$ for each training sample, ${}^{i}w$ represents weights for each training iteration where by ${}^{i}w = [\, {}^{i}_{1}w \ldots.. {}^{i}_{N}w \,]$
- Where ${}^{i}_{j}w \in (0,1)$ and $\sum_{i=1}^{N} {}^{i}_{j}w = 1$ all total weights per iteration have to equal j to 1.
- Initially all weights are set where by ${}^{1}_{}w_j = \frac{1}{N}, J = 1 \ldots. N$
- The error total rate for each weak classifier must be lower than *0.5*

Algorithm major operational steps (Sampling)

- The three major algorithmic operational steps are

  1. **Sampling**, that is where samples are *D* are drawn from the training set.

  2. **Training step**, this is where all classifiers are trained and error rate *E* for each classifier is noted.

  3. **Combination**, this is the last step where by all classifiers are put together after training.

- Sampling generally is done in 2 ways, with replacement and without replacement. Adaboost operates with the the first option that is with replacement.

- As mentioned earlier the weights at first are set at the same value ${}^{1}w = [{}^{1}_{1}w \ldots\ldots {}^{1}_{N}w]$, ${}^{i}_{j}w \in (0,1)$ $\sum_{i=1}^{N} {}^{i}_{j}w = 1$, where ${}^{i}_{j}w$ is the weight of the *jth* sample at the *ith* iteration.

Algorithm major operational steps (Sampling)

- After an iteration happens weights are updated.

$$ {}_{i}^{t+1}w = \frac{{}_{i}^{t}w}{Z_t}(e^{\alpha_t} if\ y_i \neq h_t(x_i), e^{-\alpha_t} otherwise. $$ *Eqn 1*

- If an element x_i is predicted correctly, $y_i = h_t(x_i)$, this implies that $y_i h_t(x_i) = 1$, if $y_i \neq h_t(x_i)$ its negative.
- So the earlier equation we saw for the weight updates can be rewritten as

$$ {}_{i}^{t+1}w = \frac{{}_{i}^{t}w}{Z_t} \mathrm{e}^{-\alpha_t y_i h_t} $$

- The terms in the equation above represent the following;
- ${}_{i}^{t+1}w$ Is the weight for an observation I in the next iteration.
- α_t Is the classifier weight h_t and $\alpha_t = \frac{1}{2}\log\frac{1 - Error}{Error}$, the *Error* in this case is the one for the classifier h_t during training.

Algorithm major operational steps (Sampling)

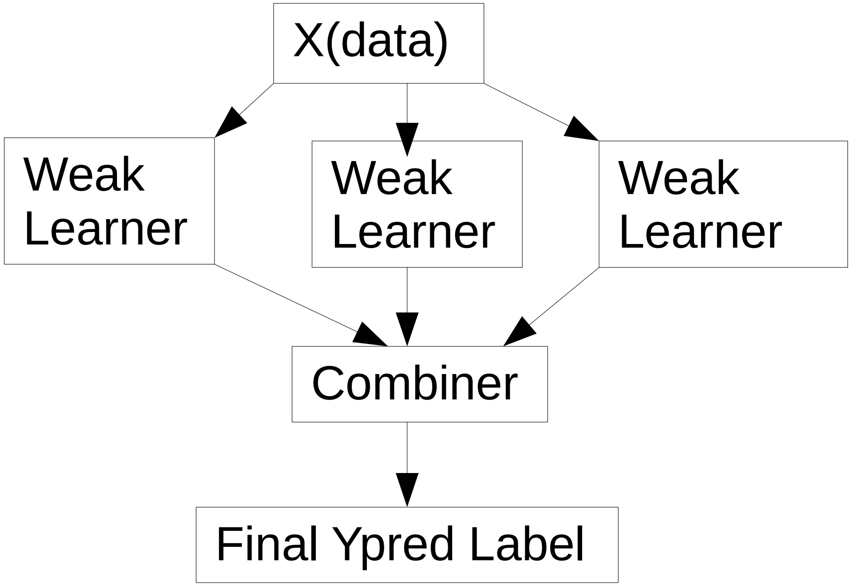
- Z_t is defined as the normalization factor $Z_t = \sum_i^t w e^{-\alpha_t y_i h_t}$

Algorithm major operational steps (Training)

- During the training phase of Adaboost, one weak trainer is used per iteration using the training set.
- The training error is computed denoted with $\epsilon_t = \sum_{i=1}^{N} w_j^t l_j^t$ where each of the terms represent the following.
- The values for l_j^t are *1* if h_t classifies x_i wrongly, on the other hand its 0.
- For a weak learner alpha is calculated as $\alpha_t = \frac{1}{2} \log \frac{1 - \epsilon_t}{\epsilon_t}$
-

Algorithm major operational steps (Combination)

The whole purpose of this step is to put together everything that has happened before in the algorithm, that is sampling and training for all the learners.



- All the results from the individual learners are put together as shown above. Using the equation below

$$Ypred = sign\left(\sum_t \alpha_t h_t(x_{test})\right)$$

The Algorithm. (Adaboost) Pseudocode

- The whole algorithm can be summarized in the following lines.
- Initialize the weights $w_i = 1/N$
- For j = 1 to J:

  Fit a classifier to the training set Xtest.

  Compute the Error

  Compute alpha.

  Update the weights
- Return the final predictions with sign()

References

- Tharwat, Alaa. (2018). AdaBoost classifier: an overview. 10.13140/RG.2.2.19929.01122.
- Kevin M. (2012). *Machine Learning a Probabilistic Perspective.* The MIT Press Cambridge, Massachusetts.
- Weinberger, Kilian "Boosting." https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote19.html
  Accessed 22 June 2020.
- Yoav, Freund and Rob, Schapire "*A tutorial on Boosting*." http://groups.di.unipi.it/~cardillo/AA0304/fabio/boosting.pdf
  Accessed 22 June 2020.