SL05: Ensemble Learning - Boosting

Introduction:

- Ensemble Learning is in general the process of combining some simple rules into a single more complicated rule that can generalize well.
- It works by dividing the data into smaller subsets, learn over each individual subset to come up with a rule, then combine all these rules in a single more complex rule.

Bagging (Bootstrap Aggregation):

- Choosing subsets: Uniformly randomly pick a subset of data. Apply any learning algorithm on it.
- Combining learners: Assuming all the learners are equal, we can take the mean of all of them.

Boosting:

- Choosing subsets: Rather than selecting random subset, we'll learn as we go, creating subsets containing the hardest examples, those ones that don't perform well given the current rule.
- Combining learners: Apply weighted mean.
- Error:
    - In regression, the error is the squared difference between correct and predicted values.
    - In classification, the error is the ratio of the total number of mismatches over the total number of examples. This implies implicitly that all the examples are equally important.
    - In the case of classification, instead of considering only the number of mismatches, we should also consider the probability that the learner will disagree with the true concept on a particular instance.
    - This shifts the perspective from the notion of only being wrong, to the notion of what is the effect of being wrong in a case by case investigation.
- Weak learner: A Weak Learner is a learner that no matter what the distribution is, always get an error rate that is better than chance (or better that 50%).

Boosting in Code:

- Given training data $\{(x_i, y_i)\}$, $y_i \in \{-1, +1\}$
- For $t = 1$ to T:
    - Construct the distribution $\mathbb{D}_t$
    - Find weak classifier $h_t(x)$ with small error $\varepsilon_t = P_{D_t}[h_t(x_i) \neq y_i]$
    - Output H_{final}

Distribution:

- The distribution at the beginning will be uniform over all the examples.

$$\mathbb{D}_1 = \frac{1}{n}$$

- For each distribution afterwards:

$$\mathbb{D}_{t+1} = \frac{\mathbb{D}_t . e^{-\alpha_t y_i h_t(x_i)}}{z_t}$$

$$where \qquad \alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t}$$

- We take the old distribution of a particular example, and we make it bigger or smaller based upon how well the current hypothesis does on that example.
- This equation guarantees that If the prediction was correct its probability in the distribution generally decreases.
- If the learner gets the prediction wrong, the probability in the distribution increases, but it ultimately depends on the rest of the distribution and how the learner performed.
- Final hypothesis:

$$H_{final}(x) = sgn\left(\sum_t \propto_t h_t(x)\right)$$

- Alpha is a measure of how well the hypothesis actually performed. The final hypothesis is the weighted sum of the hypotheses to multiplied to $\propto_t$.

Why Boosting works?

- The reason why boosting works and produces progressively is that the overall error has to necessarily stay go down with each iteration of boosting, or at least stay the same, due to how the distribution changes in response to getting cases wrong and constantly having to find a weak learner.
- Rule of thumb: Boosting doesn't overfit.