Ensemble Learning and Weak Learners

FADML Scribe
Urja Kumari
24BM6JP58

Goal of Learning

Our main goal is to minimize the error. Since we cannot directly control the out-of-sample error, we aim to minimize the **in-sample error**, as it tracks the out-of-sample error.

Two-Class Classification Problem

- If the error lies in the range of 0.5 to 1, then the learning algorithm is performing worse than the random assignment of classes.
- For error close to 0, it is a **strong learner**.
- For error close to 0, it's a **weak learner**.

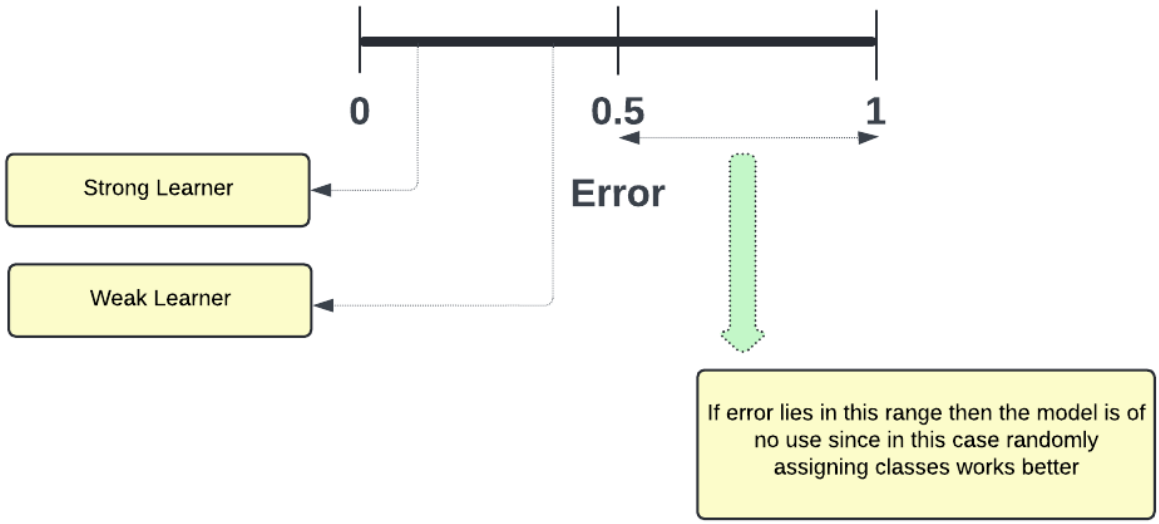

Figure 1: Leaners classification based on generalisation

Weakness in Learning

Weakness in learning could be attributed to many factors:

- It could be due to overfitting.
- It could be due to using a simple model which fails to capture the complex patterns properly (fails to capture deterministic noise).

A weak learner is one that does not generalize well. However, a weak learner is not bad — if we can combine multiple weak learners properly, it can even perform better than a strong learner.

How Do We Get Different Learners?

- Different learning algorithms acting over the same dataset result in different learners. (Neural Networks & backpropagation, SVM & Quadratic Programming, Perceptron Learning Algorithm, etc.)
- Different subsets of the same training set used on the same algorithm produce different learners.
- Different hyperparameters (e.g., λ in regularization, η for learning rate, the order of the data points in which we visit and train due to the rough surface where we are trying to minimize, etc.).
- Different activation functions (Sigmoid, Tanh, ReLU, Leaky ReLU, etc.)

Having variation in any of the above results in a different learner.

Ensemble Learning with Weak Learners

Let h_1, h_2, h_3, h_4, h_5 be a set of weak learners. Since they are weak learners, there are regions (shaded) over which they do not perform well and produce errors.

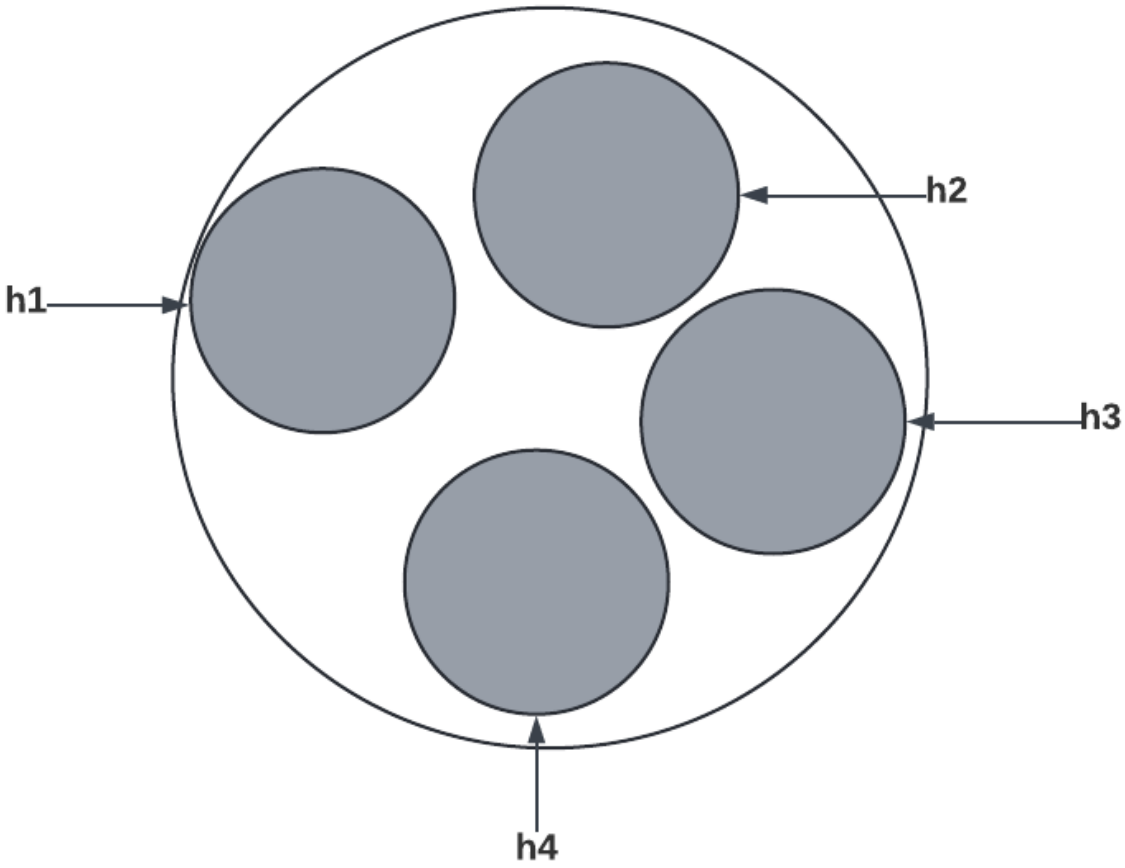

Figure 2: Independent weak learners

If the hypotheses perform poorly on disjoint portions of the data, then ensemble learning will work well since, where a particular hypothesis (e.g., h_1) is performing poorly, other hypotheses (e.g., h_2 and h_3) are not performing poorly, so taking a majority vote gives the correct answer.
For a 2-class classification problem, let's assume we have 3 hypotheses or 3 weak learners:

$$H = \text{sign}(h_1 + h_2 + h_3)$$

Here we are taking a majority voting by the three learners to determine the label of a data point. In this case, the majority voting gives the correct answer, as the region where h_1 is performing poorly (giving a wrong answer), h_2 and h_3 are giving the correct answer.

Ensembling in Different Algorithms

Ensembling in Bayes Classifier

$$P(C_i \mid \mathbf{x}) = \sum_{M_j} P(C_i \mid \mathbf{x}, M_j)\, P(M_j)$$

$$y = \sum_{C_i}^{\text{classes}} P(C_i \mid h_1(x))\, P(TE \mid h_1(x))\, P(h_1)$$

$$P(C_i \mid \mathbf{x}) = \sum_{M_j} P(C_i \mid \mathbf{x}, M_j) \cdot P(M_j)$$

Where:

- $P(C_i \mid \mathbf{x})$ — Posterior probability of class C_i given data $\mathbf{x}$
- M_j — The j-th model or hypothesis
- $P(C_i \mid \mathbf{x}, M_j)$ — Posterior probability of C_i under model M_j
- $P(M_j)$ — Prior probability of model M_j

Ensembling in Linear Regression We can take different subsets of data , train on them and ensemble the result in the case of linear regression as well. However this is not usually done in practice, because the loss function in linear regression is mean squared error which comes within the allowable range of the value obtained through ensembling, hence there is no need to do ensembling.

When weak Learners are not independent

There can be situations in which the portions of error are not disjoint. That is, there are common points where both learners make errors. In this case, ensembling and majority voting may not give the correct result. For example, if h_1 and h_2 make errors and assigns 0 as the class label in a particular region, and h_3 gives the correct result (assigning 1 as the class label), assuming that the correct label is 1, then the majority voting would suggest 0 as the correct answer, which is wrong.
The decision-making will be wrong only in the shaded portions. Therefore, if the area of the unshaded portion is greater than the shaded portions, then there is a higher probability of getting the right answer. Hence, the more independent the learners are, the better the result we can get through ensembling.

Requirements

We need to do two things:

- Create different learning algorithms that are as independent as possible.

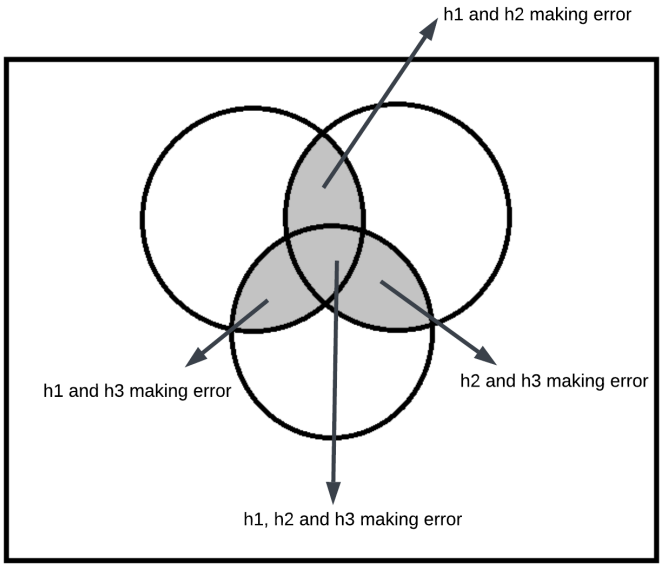

Figure 3: Dependent Weak Learners

- Produce a learning algorithm that combines these weak learners and gives a good result.

Each hypothesis h_i gives an outcome d_i, and the variance that the hypothesis is making over the outcome is $\text{Var}(d_i)$ for each hypothesis. Then the variance of the outcome y is:

$$\text{Var}(y) = \frac{1}{k^2}\left[\sum_{i=1}^{k}\text{Var}(d_i) + \sum_{i=1}^{k}\sum_{j=1}^{k}\text{Cov}(d_i, d_j)\right]$$

When considering independence, the covariance term reduces to 0 (lesser the dependence, smaller the covariance value), and hence the variance of the outcome reduces and becomes smaller than the variance of each weak learner.
Assuming the variances to be equal, we get:

$$\text{Var}(y) = \frac{k \cdot \text{Var}(d_i)}{k^2} = \frac{\text{Var}(d_i)}{k}$$

This shows that ensembling reduces the variance.

Example: Probability of Correct Classification

If we consider 3 learners, each with:

- Probability of predicting correctly = 0.7

- Probability of making an error = 0.3

What is the probability of predicting correctly by taking majority voting from the 3 learners? The probability of predicting wrongly is:

$$P_{\text{error}} = \binom{3}{2}(0.3)^2(0.7) + \binom{3}{3}(0.3)^3$$

$$P_{\text{error}} = 3 \cdot 0.09 \cdot 0.7 + 1 \cdot 0.027 = 0.189 + 0.027 = 0.216$$

Therefore, the probability of predicting correctly is:

$$P_{\text{correct}} = 1 - 0.216 = 0.784$$

General Formula

$$P_{\text{correct}} = 1 - \sum B(n, n_w, \varepsilon)$$

Where:

- n: Total number of learners
- n_w: Number of learners predicting wrongly (majority)
- ε: Probability of error

Bias-Variance Tradeoff and Ensembling

Usually, we know that when the bias goes up, variance comes down.
If we have multiple learners, the hypothesis set grows and thickens, which means that the chances of finding the best hypothesis from the set with respect to the actual one improves, thus reducing bias. However, the variance increases since the more hypotheses we have, the more difficult it becomes to find the best hypothesis. However in ensembling combining multiple independent weak learners, the variance decreases as has been proved earlier.

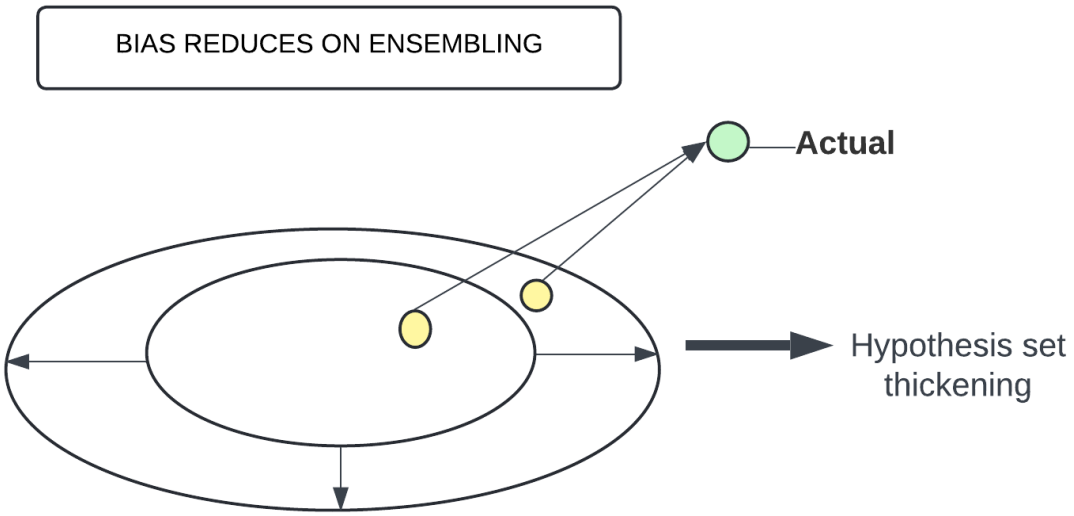

Figure 4: Bias Reduces on Ensembling

Hence, ensembling is a very powerful technique as it reduces both bias and variance (by a factor of k).

Bagging

$$H = \text{sign}(h_1 + h_2 + \cdots + h_m)$$

A dataset D consisting of N data points is subsampled randomly with replacement, leading to data portions $D_1, D_2, \ldots, D_m$ and hypotheses $h_1, h_2, \ldots, h_m$.

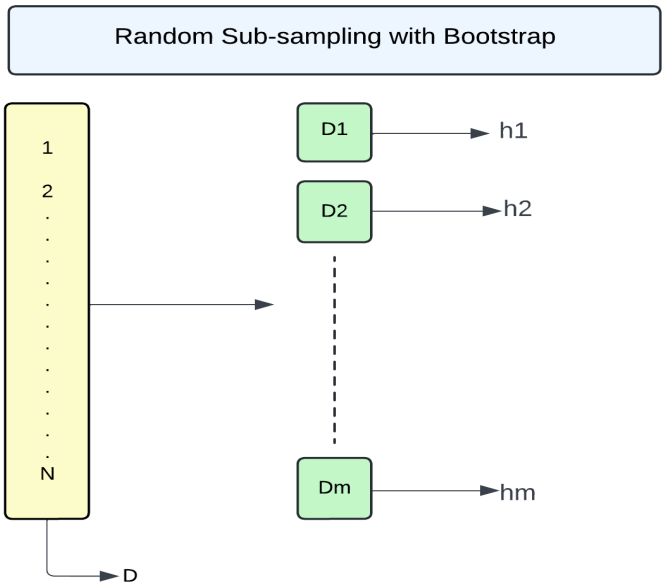

Figure 5: Random Forest Conceptual Diagram

$$\text{Probability of picking a point} = \frac{1}{N}$$

$$\text{Probability of not picking a point} = 1 - \frac{1}{N}$$

$$\text{Probability of not picking a point in any of the } M \text{ subsamples} = \left(1 - \frac{1}{N}\right)^M$$

$$\text{Probability of picking a point in any of the } M \text{ subsamples} = 1 - \left(1 - \frac{1}{N}\right)^M$$

This value is large when N is large. Thus, the probability of picking a point in any subsample chunk increases as N becomes large and the pobability of a point to belong in any two of the subsamples is small, and hence we can treat each subsample portion as independent. So the weak learners formed through this way are considered to be independent. This technique is known as **Bagging**.

Random Forest

For Random Forests, we start with the entire set of attributes, then create subsets of attributes either randomly or through some technique. Based on these subsets of attributes, we construct decision trees to full length (somewhat overfitiing and hence considered to be a weak learner). For each data point, predictions are made through each decision tree, and the final prediction is determined by majority voting of all predictions.

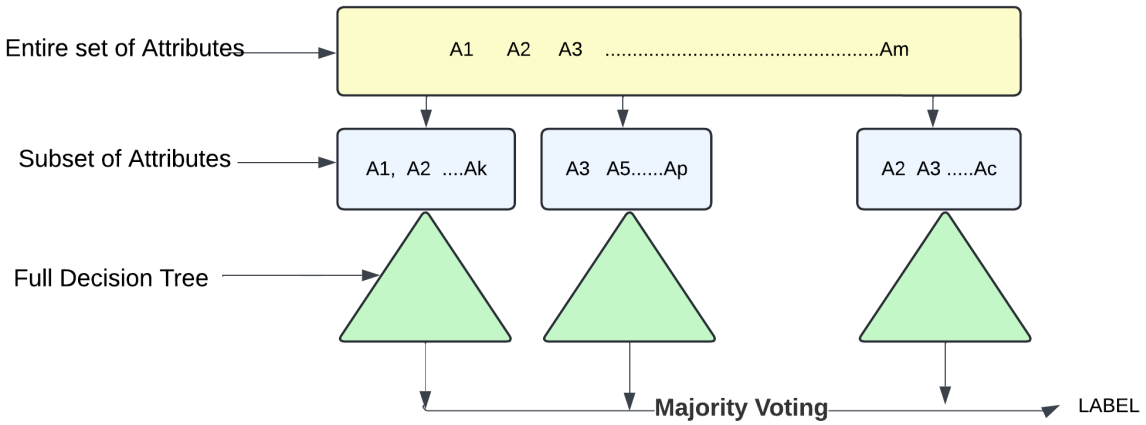

Figure 6: Random Forest Conceptual Diagram

Boosting Technique

- Instead of randomly selecting a new dataset, we begin by creating a subsample dataset D_1 and a corresponding weak learner hypothesis h_1. When tested on the entire dataset, this hypothesis will provide correct predictions for some data points, while making errors on others.
- To improve performance, instead of choosing another dataset randomly, we adjust the selection process by increasing the likelihood of selecting previously misclassified data points, and decreasing the chances of selecting correctly classified ones. This process is known as data aggregation.
- Additionally, another form of data aggregation is applied when creating subsample D_3. Here, we increase the probability of selecting points that were assigned different class labels by hypotheses h_1 and h_2, because the true label for these points can be determined through majority voting from h_1, h_2, and h_3.
- This process is repeated iteratively, and the technique is referred to as Boosting, since it effectively "boosts" the probability of selecting the points where the hypotheses disagree.
- In some cases, rather than taking a simple majority vote, a **weighted majority vote** is used, where more importance is given to the hypotheses that make more accurate predictions. This approach is often called **expert's weighted wisdom**.

The final decision rule is represented by:

$$H = \text{sign}\left(\sum_{i=1}^{k} w_i h_i\right)$$

where $\sum_{i=1}^{k} w_i = 1$.

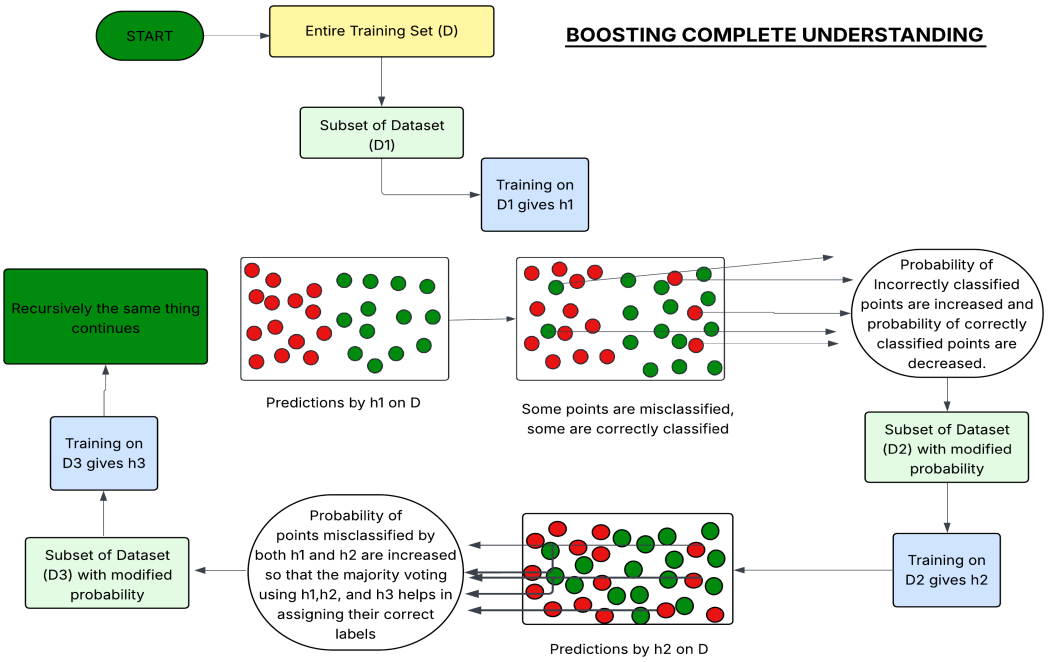

Figure 7: Explaining Boosting Ensembling