Assignment - 2 Part A

Nandini Nair

2023-11-10

PART A

QA1

Bagging, also known as bootstrap aggregation, involves training multiple classifiers on different subsets of random data samples with replacement from the dataset. This is done because a single algorithm may not provide a perfect prediction for a given dataset.

Bagging helps address the issue of high variance (overfitting) but is not effective for dealing with high bias (underfitting).

Since base learners are trained on different data samples, there is diversity in the training process. These base learners are considered weak models, and they are less likely to overfit the training data compared to more complex models. Even if one base model overfits, they are likely to overfit in a different way compared to other base learners because they are trained on different data samples. Therefore, when we combine these base learners in an ensemble, the risk of the model overfitting is minimized.

Bagging cannot address high bias effectively. Each base learner is a weak model, and there is a possibility that these models may not capture the underlying patterns in the data. So, even when we combine these models in an ensemble, the resulting model may not perform well if the underlying base learners suffer from underfitting. Therefore, bagging is more suitable for mitigating overfitting rather than underfitting.

QA2

Bagging is the process of training different models on different subset of training data independently and combining them to make the final prediction whereas in boosting different models are trained sequentially.

Additionally, bagging does not need a lot of data as the models are trained in parallel which gives a better computational efficiency compared to boosting. Each model in boosting is trained on the output or errors made by its predecessor so that the errors made by the first model or patterns unrecognized by the first model is rectified or captured by the second model.

In boosting, models are fit iteratively so that the training of a given model at a given stage depends on the models fit at prior steps. When it comes to multiple models,it will become too computationally complex to fit sequentially when compared to bagging where models are trained and fit parallelly. Therefore we can say that bagging is more computationally efficient than boosting

QA3

An ensemble model requires two essential conditions to be met: firstly, the base learners should demonstrate predictive capability and be better than random guessing. Secondly, these base learners should be independent of each other. In the case discussed by James, the base learners are quite similar to one another, which leads to a lack of diversity among them. As a result, all the models tend to produce similar outputs. Consequently, even when these models are combined in an ensemble, it doesn't significantly enhance overall performance because the base learners neither outperform random models nor exhibit diversity amongst themselves.

QA4

The measure of disorder or entropy for two classes can be calculated using the following equation:

{[} Entropy =

$$Entropy = \sum_{i=1}^{n} \left(-P(x = i) \cdot \log_2\left(P(x = i)\right)\right.$$

Here, $P(x = i)$ represents the probability of class 'i'. Information gain is a concept used to identify the best split for a decision tree, which ultimately reduces the entropy.

Information gain can be computed as:]

Information Gain

To illustrate the calculation of information gain, let's break it down for a specific case:

Information Gain can be computed as:

$$\text{Information Gain} = \text{Entropy(parent)} - [\text{average entropy(children)}]$$

Where:

$$\text{Entropy(parent)} = \left(-\frac{9}{16} \cdot \log_2\left(\frac{9}{16}\right)\right) - \left(\frac{7}{16} \cdot \log_2\left(\frac{7}{16}\right)\right) = 0.988699$$

$$\text{Entropy(children(large))} = \left(-\frac{3}{8} \cdot \log_2\left(\frac{3}{8}\right)\right) - \left(\frac{5}{8} \cdot \log_2\left(\frac{5}{8}\right)\right) = 0.954433$$

$$\text{Entropy(children(small))} = \left(-\frac{6}{8} \cdot \log_2\left(\frac{6}{8}\right)\right) - \left(\frac{2}{8} \cdot \log_2\left(\frac{2}{8}\right)\right) = 0.811277$$

The weighted average entropy of children is obtained as:

$$\text{Weighted Average Entropy(children)} = \left(\frac{8}{16}\right) \cdot 0.954433 + \left(\frac{8}{16}\right) \cdot 0.811277 = 0.882854$$

Finally, the information gain is computed as the difference between the entropy of the parent and the average entropy of the children:

$$\text{Information Gain} = 0.988699 - 0.882854 = 0.105845$$

QA5

In a random forest model, we want each tree to be a bit different to add diversity and improve predictions. So, instead of considering all available attributes for each tree, we randomly pick a subset of them. This subset is called 'm,' and it determines how the trees make their decisions.

Here's a simple breakdown of 'm':

If 'm' is the same as the total number of attributes ('p'), it's like using all the attributes for every decision in every tree. It's like everyone doing the same thing.

If 'm' is very small, we only use a few attributes. This can make the trees less powerful and accurate because they're missing important information.

If 'm' is very large, we use almost all the attributes, which can make the trees too similar, reducing diversity.

The right 'm' value depends on the problem you're solving, so it's something you might need to adjust to get the best results