Saagar Sanghavi, Jin Seok Won, Kevin Zhu

Project T Final

Multiplicative Weights and Ensemble Methods - Boosting Quiz

Question 1: Which of the following is NOT a similarity between Orthogonal Matching Pursuit and Adaboost?

a. Both employ a greedy heuristic to guide the direction of the next iteration.
b. Both start each following iteration from the residual of the previous.
c. Both employ a stopping condition when the error passes a certain threshold.
d. Both solve an optimization problem at the final iteration, yielding the result.

Answer: c. This is true for OMP - one stopping condition is once the error is below a certain threshold. However, for Adaboost, you stop training after M models, a hyperparameter.

Question 2: When applying bagging, would we expect to receive better results when the models have higher or lower correlation with each other? Explain.

Answer: Lower correlation. In boosting, we bootstrap in order to simulate the notion of having multiple datasets. Intuitively, having lower correlation better suits this purpose since the data is more independent from each other. Indeed, when each of the models are perfectly correlated, there is no benefit from bagging at all.

Question 3: Which of the following is the correct general use case of bagging and boosting?

a. Bagging: High bias, low variance. Boosting: Low bias, high variance.
b. Bagging: High bias, high variance. Boosting: Low bias, low variance.
c. Bagging: Low bias, low variance. Boosting: High bias, high variance.

d. Bagging: Low bias, high variance. Boosting: High bias, low variance.

Answer: d. Bagging allows our model to have lower variance since we are simulating the effect of training on more data, and intuitively getting a more diverse range of perspectives from which we can characterize the underlying true function. Boosting allows us to focus effectively on more difficult data points, allowing us to reduce the bias in our model.

Question 4: In multiplicative weights, what is an advantage of using probabilistic decision making rather than a deterministic strategy?

Answer: Choosing the highest weighted expert all the time is vulnerable to the adversary. By making probabilistic choices, even if the adversary knows our strategy, they are prevented from fully countering our strategy.

Question 5: Suppose we had n experts and losses on each day in [0, 1]. What is the tightest bound the multiplicative weights algorithm can guarantee about your total loss over T days?

Answer: T. In the worst case, we will incur the full loss at each iteration. Suppose each expert has the max loss at each iteration - then no matter our strategy, we will be stuck with T total loss. This is true even in a more usual case, albeit with very low probability.

Question 6: In multiplicative weights, how can an adversary take advantage of someone using the halving algorithm from the notebook?

Answer: An adversary can make a dataset where one expert does extremely well in the first days, until the halving algorithm chooses that expert to be its best expert. Then after the decision is *made, the adversary can make that expert start making very bad predictions, leading to high loss by the halving algorithm.*

Question 7: How do decision boundaries of boosting with linear weak classifiers differ from boundaries of decision trees?

Answer: Boosting with linear classifiers can continue iterating even when all points are identified correctly. Decision trees cannot continue when there are no more points since it will hit its base case, which is reflected in the decision boundary.

Question 8: What is the advantage of using random forests over simple decision trees?

Answer: Random forests are more stable than decision trees, having more smooth decision boundaries and more consistent output. This lowers the variance of the model and generally increases the overall performance.

Question 9: The classifiers that are part of the Adaboost must be weak learners which have accuracy of over 50%. Why are decision stumps good weak learners for Adaboost?

Answer: Decision stumps are very simple classifiers that can almost reliably get accuracy of over 50%. This simplicity gives them nice adaptability, having consistent performance for different datasets.

Question 10: Gradient boosting generalizes the idea of boosting to arbitrary differentiable loss functions. Which of these applications can gradient boosting to be used in?

a. Binary Classification

b. Multi-class Classification

c. Regression

d. All of the above

Answer: d. All of the above. By allowing optimization of arbitrary loss functions, gradient boosting can combine weak learners to a more general use.