11/30/2018

Implement Assignment 3

CS534-Decision Tree, Forest Tree, Adaboost

Yi-Jung Chiang

Chi Wen

Han-Yu Wu

Part 1: Decisoin Tree(DT).

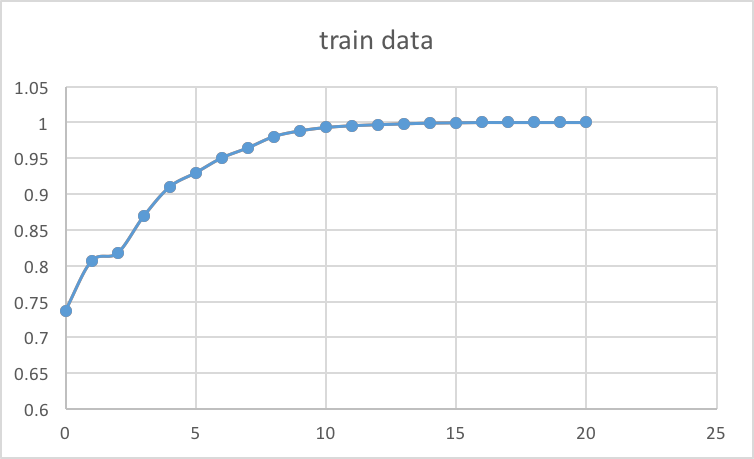
1. Create a decision tree with maximum depth of 20 on the train data.

**Run IA3.py**

1. Using the created decision tree, compute and plot the train and validation accuracy versus depth.

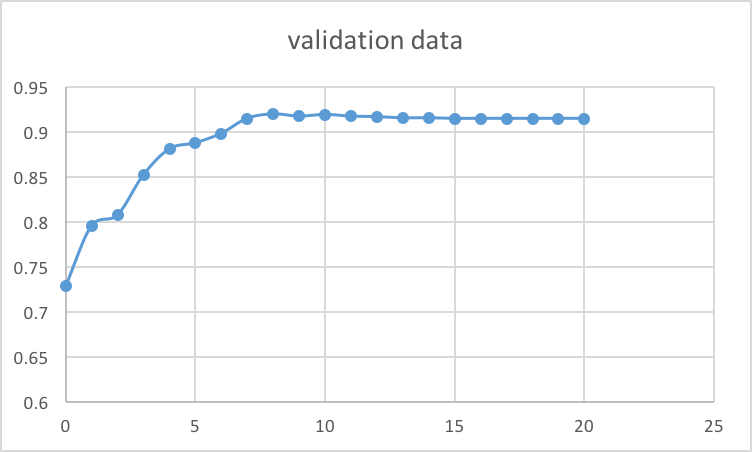
**The result of train data:**

|  |  |
| --- | --- |
| depth | accuracy |
| 1 | 0.73711293 |
| 2 | 0.806260229 |
| 3 | 0.817512275 |
| 4 | 0.86886252 |
| 5 | 0.909983633 |
| 6 | 0.929828151 |
| 7 | 0.950490998 |
| 8 | 0.964607201 |
| 9 | 0.980360065 |
| 10 | 0.988338789 |
| 11 | 0.99304419 |
| 12 | 0.995499182 |
| 13 | 0.99693126 |
| 14 | 0.998158756 |
| 15 | 0.999181669 |
| 16 | 0.999590835 |
| 17 | 1 |
| 18 | 1 |
| 19 | 1 |
| 20 | 1 |



**The result of validation data:**

|  |  |
| --- | --- |
| depth | accuracy |
| 1 | 0.728667894 |
| 2 | 0.79558011 |
| 3 | 0.807857581 |
| 4 | 0.852056476 |
| 5 | 0.880908533 |
| 6 | 0.887661142 |
| 7 | 0.898096992 |
| 8 | 0.914671578 |
| 9 | 0.919582566 |
| 10 | 0.917127072 |
| 11 | 0.918968692 |
| 12 | 0.917127072 |
| 13 | 0.916513198 |
| 14 | 0.915285451 |
| 15 | 0.915285451 |
| 16 | 0.914671578 |
| 17 | 0.914671578 |
| 18 | 0.914671578 |
| 19 | 0.914671578 |
| 20 | 0.914671578 |



1. Explain the behavior of train/validation performance against the depth. At which depth the train accuracy reaches to 100% accuracy? If your tree could not get to 100% before the depth of 20, keep on extending the tree in depth until it reaches 100% for the accuracy.

**For train data, according to the result, we can observe that when depth is 17, the accuracy will achieve 100%. However, for validation data, the result doesn’t reach to 100% accuracy, but we have to stop updating accuracy because the train data has already reaches 100% accuracy when depth is 17, there won’t be any threshold from the train data. Both train and validation, the accuracy keeps increasing when the depth increases.**

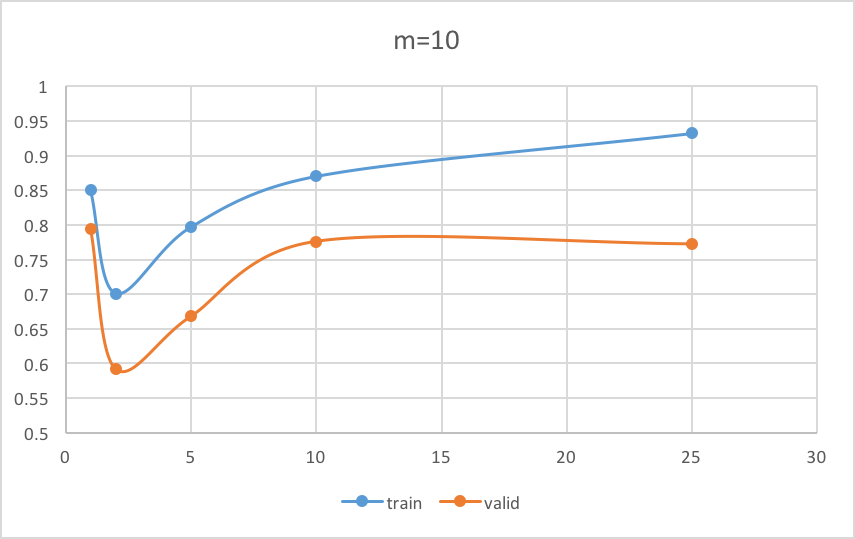
1. Report the depth that gives the best validation accuracy?

**When depth is 17, the validation accuracy is the best.**

Part 2: Random Forest(Bagging)

1. **Run IA3\_part2.py**
2. For d = 9, m = 10 and n∈[1, 2, 5, 10, 25], plot the train and validation accuracy of the forest versus the number of trees in the forest n.

|  |  |  |
| --- | --- | --- |
| n | train | valid |
| 1 | 0.850040917 | 0.794352363 |
| 2 | 0.700695558 | 0.592387968 |
| 5 | 0.796440262 | 0.667894414 |
| 10 | 0.869885434 | 0.775936157 |
| 25 | 0.93207856 | 0.772252916 |



1. What effect adding more tree into a forest has on the train/validation performance?

**When we add more trees into the forest, the accuracy of both train and validation data increases. In the graph, we have a drop on n=2, which we believe that is because the chance of having an overlap area when n=2 is really low, so as the number of tree increases, the accuracy increases.**

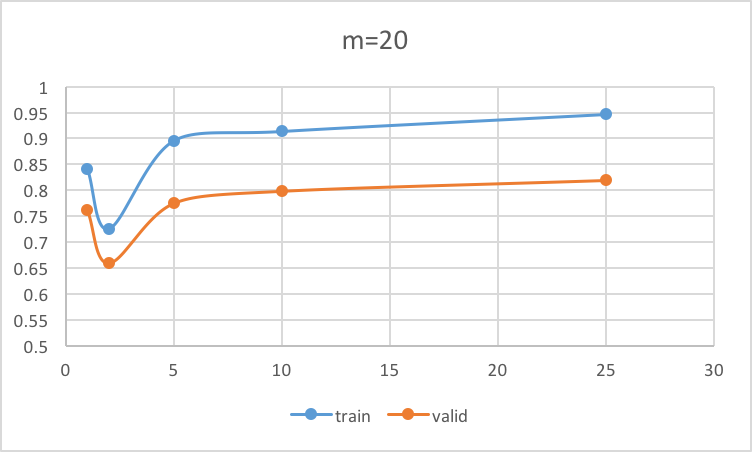
**Another reason would be us using a random predict value when having opposite voted result from two tree.**

1. Repeat above experiments for d = 9 and m∈[20, 50]. How greater m changes the train/validation accuracy? Why?

**When we change to larger m, we get better performance of the predication from both train and validation data. We think there are two reasons. First, increasing m, it means we might have better chance to cover some influential feature to build the decision tree. Second, having larger m could allow various data to make influence when building the tree.**

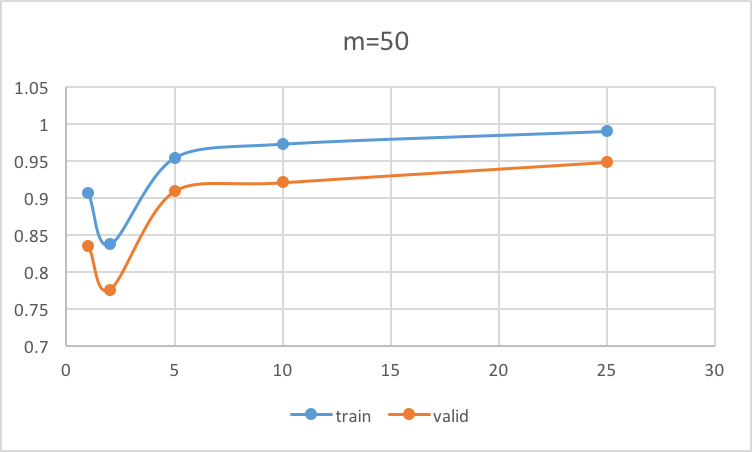
M=20

|  |  |  |
| --- | --- | --- |
| n | train | valid |
| 1 | 0.840425532 | 0.761203192 |
| 2 | 0.725245499 | 0.659914058 |
| 5 | 0.895662848 | 0.775322284 |
| 10 | 0.913666121 | 0.798649478 |
| 25 | 0.946603928 | 0.819521179 |



m=50

|  |  |  |
| --- | --- | --- |
| n | train | valid |
| 1 | 0.906301146 | 0.834868017 |
| 2 | 0.837356792 | 0.775936157 |
| 5 | 0.954378069 | 0.909146716 |
| 10 | 0.97299509 | 0.920810313 |
| 25 | 0.990180033 | 0.948434622 |



Part 3: AdaBoost(Boosting)

1. Let the weak learner be a decision tree with depth of 9. The decision tree should get a weight parameter D which is a vector of size 4888. Implement the decision tree with parameter D such that it considers D in its functionality.
2. Using the decision tree with parameter D implemented above, develop the AdaBoost algorithm as described in the slide with parameter L.
3. Report the train and validation accuracy for L∈[1, 5, 10, 20]
4. Explain the behavior of AdaBoost against the parameter L.