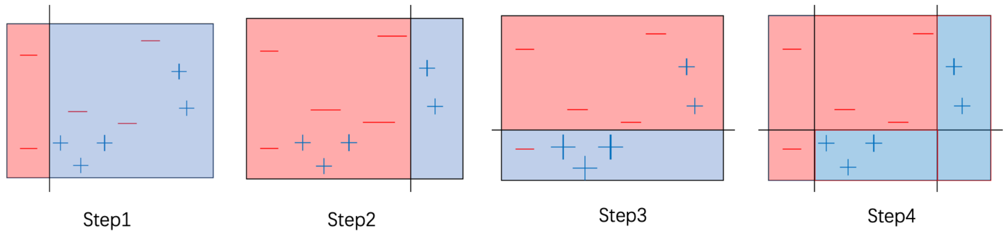
Homework 6

Problem1:

(1) **Adaboost algorithm:** The first three images are steps of the Adaboost algorithm, and the last image is the final classification of the ensemble.



(2) A strong learner is PAC learnable. Given sufficient number of training examples, the error $L_D(H_s) \leq \epsilon$ with probability of at least $1 - \delta$ for arbitrary ϵ and δ.

A weak learner is $\gamma - learnable$. The error $L_D(H_s) \leq \frac{1}{2} - \gamma$, for fixed $\gamma \leq \frac{1}{2}$.

In other words, for a weak learner, the error doesn't have to be smaller than a very small number ϵ. The upper bound of the error can be a number smaller than 0.5, just a little better than pure guess.

(3) When $T = 1$, the only learner is a weak learner. Therefore, the bound of error is $\frac{1}{2} - \gamma$, for fixed $\gamma \leq \frac{1}{2}$.

When $T = inf$, we can combine all the weak learner into a strong learner. In this case, the bound of error is ϵ, which is an arbitrary small number.

This is how the Adaboost algorithm works. It can help us to generate a strong learner out of multiple weak learners. As a result, the boosting increase the complexity of the hypothesis class, and make the model less underfitting.

Project Report 6

Question1:
The training and test error on chess dataset is shown in table1.

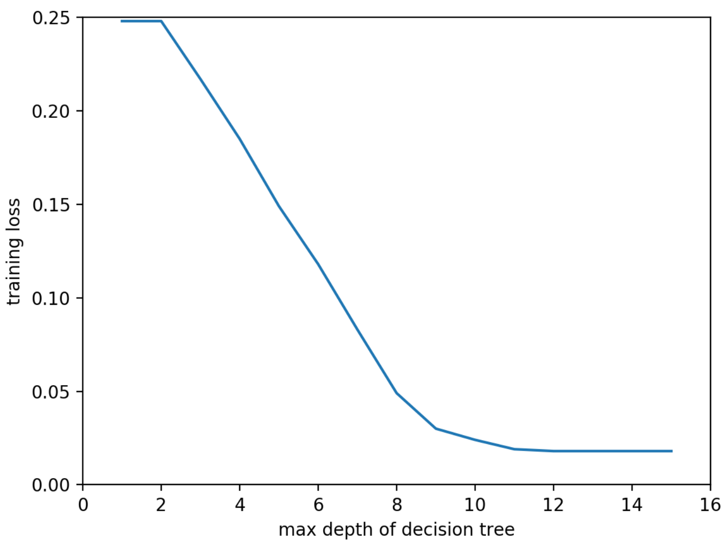
	Training Error	Entropy	Gini
Training Loss(not-pruned)	0	0	0
Test Loss(not-pruned)	0.22241	0.08695	0.23076
Training Loss(pruned)	0.048	0.023	0.059
Test Loss(pruned)	0.20484	0.07357	0.2107

The training and test error on spam dataset is shown in table2.

	Training Error	Entropy	Gini
Training Loss(not-pruned)	0.014	0.013	0.013
Test Loss(not-pruned)	0.17723	0.17262	0.18569
Training Loss(pruned)	0.05	0.052	0.052
Test Loss(pruned)	0.16608	0.16032	0.17454

From the tables above, we can find that the measurement of entropy has the minimum loss. Meanwhile, pruning can increase the training error slightly, and decrease the test error, which means pruning can make the model less overfitting.

Question2:
The figure below shows the loss of the decision tree on the training set for trees with maximum depth set to each value between 1 to 15.



From the figure, we can find that at first the training error decreases with the max depth of the tree. Then the training loss reaches its limit and doesn't change as the max depth increases.

Therefore, if a decision tree is limited to a small depth, the decision tree becomes very simple, and is supposed to be underfitting. However, if the tree is deep enough, it can include enough nodes that learned from the training set. In this case, increase the max depth won't decrease the training loss. Also, the model can be overfitting if the decision tree is too complicated.