Introduction to Machine Learning
Problem Set: Decision Trees, Ensembles, Support Vector Classifier

Summer 2021

1. For this question, you will use the Tennis dataset presented in the lecture:

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

(a) In the lecture, we showed how the first node in a decision tree classifier would be built. Continue this procedure, and build the rest of the tree, using information gain as the criterion. (Show your computations at each stage, as in the lecture handout; don't just use a `DecisionTreeClassifier` from `sklearn`.) Stop building your tree when it achieves zero classification error on the training set.

Solution: In the lecture, we said that the best split at the root of the tree is on Outlook. So now we have three branches coming out of Outlook:

i. **Sunny**: (2+, 3-). Entropy$(S_{\text{Sunny}}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.97$.
Temperature: Hot: (2-), Mild: (1+, 1-), Cool: (1+).
Humidity: High(3-), Normal (2+).
Wind: Weak: (1+, 2-), Strong: (1+, 1-).
Without any computation, we can see that splitting on Humidity will give us pure leaf nodes, with zero entropy. This will give us the best possible information gain. So we don't need any computation to decide on the split at this point.
However, for reference, here is an example of computing the gain for a split on Wind:

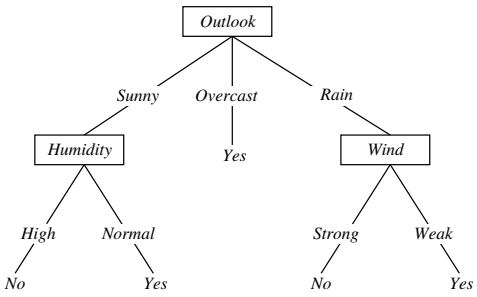
$$\text{Entropy}(S_{\text{Sunny}}|\text{Wind}) = \frac{3}{5}\left(-\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3})\right) + \frac{2}{5}\left(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})\right)$$

$$\text{Gain}(S_{\text{Sunny}}, \text{Wind}) = \text{Entropy}(S_{\text{Sunny}}) - \text{Entropy}(S_{\text{Sunny}}|\text{Wind}) = 0.97 - 0.95 = 0.02$$

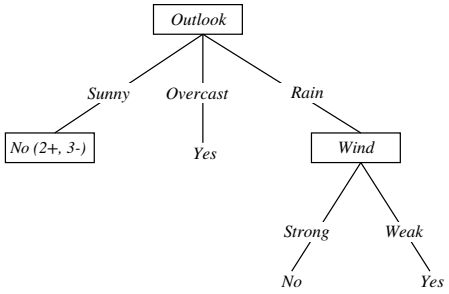
ii. **Overcast**: (4+, 0-). This node is already pure, so no further split is necessary.

iii. **Rain**: (3+, 2-). $\text{Entropy}(S) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.97$
Temperature: Mild: (2+, 1-), Cool: (1+, 1-).
Humidity: High(1+, 1-), Normal (2+, 1-).
Wind: Weak: (3+), Strong: (2-).
Again, without any computation, splitting on Wind will give us pure leaf nodes.

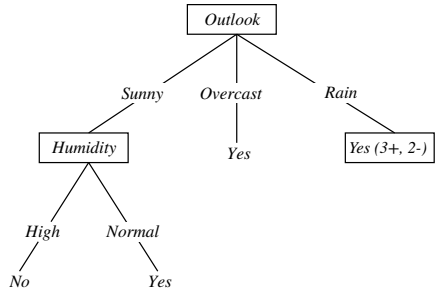
The completed tree is illustrated in Figure 1a.



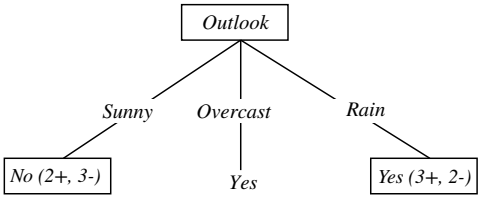
(a) Completed decision tree.



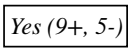
(b) Optimal subtree, stage 1.



(c) Alternative subtree, stage 1.



(d) Optimal subtree, stage 2.



(e) Optimal subtree, stage 3.

Figure 1: Completed decision tree and possible subtrees for pruning.

(b) Next, you will perform weakest-link pruning on the tree you built in (a). Show your computations and the optimal subtree at each stage.

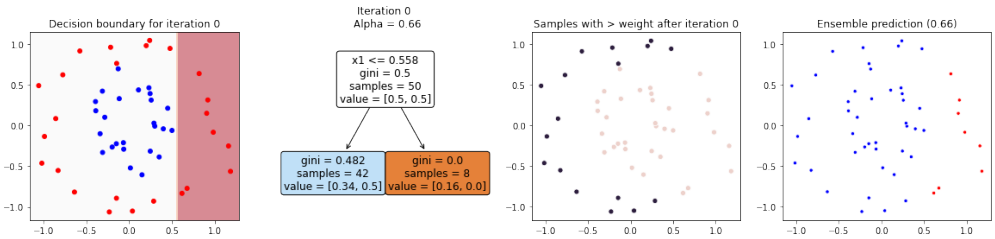
Solution: Since this is a classification tree, we will prune using classification error rate as a loss function.

i. First pruning stage. Two possible subtrees have the same classification error rate. I will choose the subtree in Figure fig:tree-wind, but either choice is equally valid.
   - Merge humidity split into parent (Figure 1c): $\frac{2}{14}$ samples are misclassified.
   - Merge wind split into parent (Figure 1c): $\frac{2}{14}$ samples are misclassified.

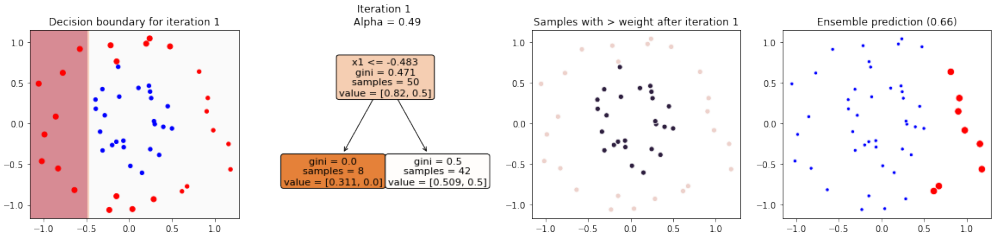
ii. Second pruning stage. It is possible at this stage to create a subtree by merging two out of the three branches out of Outlook, but I will follow a policy of only pruning branches where all children of a parent can be merged. Therefore, there is only one possible subtree in this round.
   - Merge wind split into parent (Figure 1d): $\frac{4}{14}$ samples are misclassified.

iii. Third pruning stage. Since I follow a policy of only merging *all* branches of a particular node at a time, there is only one possible subtree.
   - Merge wind split into parent (Figure 1e): $\frac{5}{14}$ samples are misclassified.

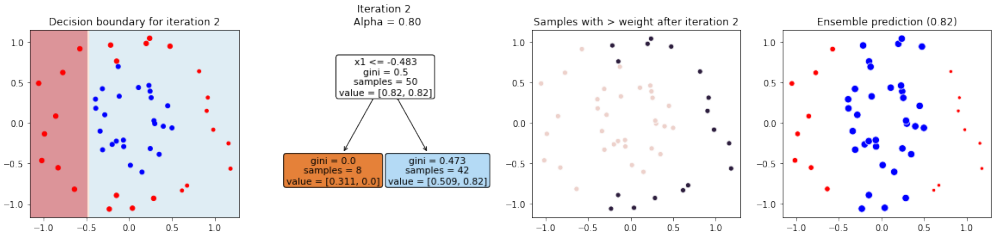
2. (*AdaBoost notebook.*) This question is about the *Demo: AdaBoost Classifier* notebook we used in class. Review that notebook, and try running it yourself, before answering the questions.



(a) Iteration 0



(b) Iteration 1



(c) Iteration 2

Figure 2: First three iterations of an AdaBoost classifier

(a) Figure 2a and Figure 2b show iteration 0 and 1 of an AdaBoost classifier. In both iterations, the predicted class labels of the ensemble are the same. However, in the second iteration, the ensemble output is more *confident* about eight samples in the red class. Why is the ensemble more confident about the class of those eight samples after two iterations? Why is the ensemble not as confident about any of the other samples after two iterations?

Solution: The eight samples for which the ensemble becomes more confident are the samples for which both decision stumps that make up the ensemble predict the same class. The magnitude of the ensemble output, which is a weighted sum of the individual decision stumps' outputs, is therefore larger, indicating greater confidence in the classification of those samples.

For all other samples, the two decision stumps disagree, so their opposite-signed outputs "cancel out" to some extent, leading to smaller magnitude of the ensemble output which indicates less confidence in the classification.

(b) Figure 2b and Figure 2c show iteration 1 and iteration 2 of an AdaBoost classifier. In

both iterations, the decision stump splits on the same feature, at the same cutpoint. In iteration 1, the red class is predicted in both regions. In iteration 2, however, the blue class is predicted in the region on the right. Why are the predictions different, even though the boundary is the same? Explain what changed between iteration 1 and iteration 2, and why.

Solution: The predictions are different because of the sample weights. In Iteration 2, the blue (+1) samples in the middle have increased weight relative to Iteration 1. As a result, the weighted sum of samples belonging to each class in the region with $x_1 > -0.483$ is different.

For Iteration 1, the weighted sum is $[0.509, 0.5]$ with the majority class being the -1 class, so the prediction for the region is -1. For iteration 2, the weighted sum for the same region with the same samples (but different weights) is $[0.509, 0.82]$ with the majority class being the $+1$ class, so the prediction for the region is $+1$.

3. (Based on a problem by CMU/M. Kolar.) This question should be completed by hand; no programming is involved.

   Suppose we are given a dataset of feature-label pairs in $\mathbb{R}^1$:

   $$(-1,-1),(0,-1),(1,-1),(-3,+1),(-2,+1),(3,+1)$$

   (a) Plot the data in $\mathbb{R}^1$. Is the data linearly separable in $\mathbb{R}^1$?

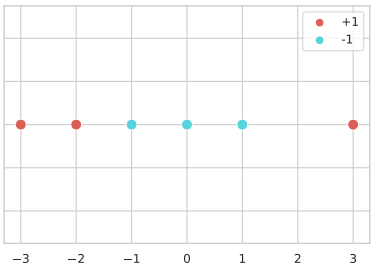


Figure 3: Data in R1

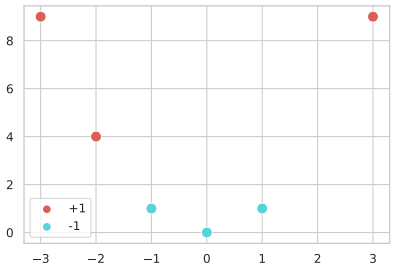


Figure 4: Data in R2

   Solution: The data is not linearly separable in $\mathbb{R}^1$.

   (b) Using the basis function $\phi(x) = (x, x^2)$, the points in $\mathbb{R}^1$ can be transformed to points in $\mathbb{R}^2$. Apply $\phi(x)$ to the data and plot the transformed data in $\mathbb{R}^2$. Is the data linearly separable in $\mathbb{R}^2$? For the rest of this problem, use the transformed data.

   Solution: The data is linearly separable in $\mathbb{R}^2$.

   (c) Using geometric intuition (i.e. looking at the plot), identify two points x_+ and x_- (from the positive class and negative class, respectively) that should be support vectors of a maximal margin classifier for this data in $\mathbb{R}^2$. Circle them, and explain why you selected these points.

   Solution: The support vectors are $x_+ = (-2, 4)$ and $x_- = (-1, 1)$ - these should be support vectors because they are the closest points belonging to different classes, so the separating hyperplane will be equidistant between them, and they will be on the margin.

   (d) Construct the maximal margin classifier using the following geometric procedure: Draw a line segment l connecting the two support vectors. Mark the midpoint of this line; the separating hyperplane should pass through this point (why?). Then, draw the separating hyperplane as a solid line perpendicular to l. Finally, draw two dashed lines parallel to the separating hyperplane and passing through the support vectors, showing the margin.

   (e) What is an equation of the line (in the form $w_0 + w_1 x_1 + w_2 x_2 = 0$) that defines the separating hyperplane you drew?

   Solution: The hyperplane is defined by $-x_1 + 3x_2 - 9 = 0$ (or any scaled version of this expression).

   (f) If we add another training sample at $x = 5, y = +1$, would the hyperplane change? Would the margin change? Explain.

   Solution: No change - a new training sample at $x = 5, x^2 = 25$ with a $+1$ label is classified correctly and is outside the margin. Samples that are outside the margin and are classified correctly won't affect the hyperplane or margin.

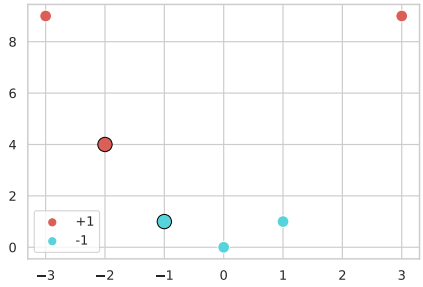


Figure 5: Support vectors

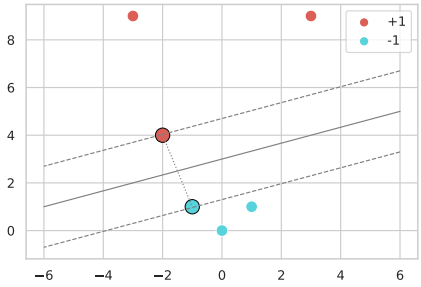


Figure 6: Separating hyperplane

(g) To find $\alpha_i, i = 1, 2$ for the maximal margin classifier, we will solve the dual problem

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \\ \text{s.t.} \quad & \sum_i \alpha_i y_i = 0 \\ & \alpha_i \geq 0 \end{aligned}$$

(note: this maximal margin classifier problem is slightly different from the support vector classifier problem! There is no slack variable for the maximal margin classifier.)

We know that $\alpha_i = 0$ for any point that is not a support vector. So, write out the expression

$$\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j)$$

in terms of α_1, α_2, the support vectors x_+ and x_-, and their labels y_+ and y_-. (Note that x_+ and x_- are vectors in the transformed data. The others are scalars.) Then substitute the values of x_+ and x_-, y_+ and y_-, and simplify, so that the only unknowns are α_1, α_2.

Solution:

$$\alpha_1 + \alpha_2 - \frac{1}{2} \left(a_1 a_1 y_+ y_+ \phi(x_+)^T \phi(x_+) + 2 a_1 a_2 y_+ y_- \phi(x_+)^T \phi(x_-) + a_2 a_2 y_- y_- \phi(x_-)^T \phi(x_-) \right)$$

$$\alpha_1 + \alpha_2 - \frac{1}{2} \left(a_1^2 \phi(x_+)^T \phi(x_+) - 2 a_1 a_2 \phi(x_+)^T \phi(x_-) + a_2^2 \phi(x_-)^T \phi(x_-) \right)$$

$$\alpha_1 + \alpha_2 - \frac{1}{2} \left(a_1^2 [-2, 4]^T [-2, 4] - 2 a_1 a_2 [-2, 4]^T [-1, 1] + a_2^2 [-1, 1]^T [-1, 1] \right)$$

$$\alpha_1 + \alpha_2 - \frac{1}{2} \left(20 a_1^2 - 12 a_1 a_2 + 2 a_2^2 \right)$$

$$\alpha_1 + \alpha_2 - \left(10 a_1^2 - 6 a_1 a_2 + a_2^2 \right)$$

(h) Note that for our example, since there is one support vector in the positive class and one support vector in the negative class, the first constraint of the dual problem described above implies that $\alpha_1 = \alpha_2$. In your expression for the previous part, substitute $\alpha_1 = \alpha$ and $\alpha_2 = \alpha$, and simplify.

Solution:

$$2\alpha - 5\alpha^2$$

(i) Take the derivative of the expression from part (h) with respect to α, and set it equal to zero. Solve to find the value of $\alpha_1 = \alpha_2 = \alpha$. Also find the values of the coefficients $w_j = \sum_{i \in S} \alpha_i y_i x_{ij}, j = 1, 2$.

Solution:

$$\frac{d}{d\alpha} 2\alpha - 5\alpha^2 = 0 \rightarrow \alpha = \frac{1}{5}$$

Then,

$$w_1 = -\frac{2}{5} + \frac{1}{5} = -\frac{1}{5}, w_2 = \frac{4}{5} - \frac{1}{5} = \frac{3}{5}$$

(j) To find w_0, use the fact that for points on the margin, $y_i = w_0 + \sum_{j=1}^{p} w_j x_{ij}$. Plug in values from either support vector to solve for w_0.

Solution: Use either

$$1 = w_0 + \frac{-1}{5}(-2) + \frac{3}{5}(4) \rightarrow \beta_0 = -\frac{9}{5}$$

or

$$-1 = w_0 + \frac{-1}{5}(-1) + \frac{3}{5}(1) \rightarrow \beta_0 = -\frac{9}{5}$$

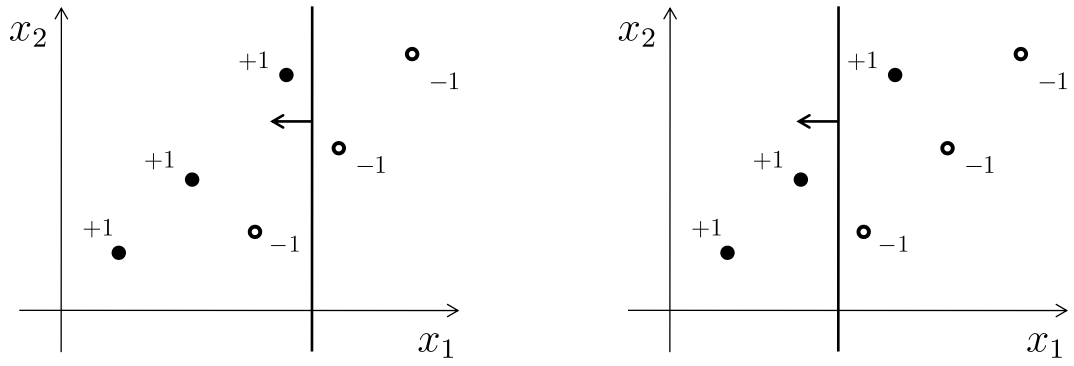
(k) Show that for the values of w_0, w_1, and w_2 you computed above, the separating hyperplane $w_0 + w_1 x_1 + w_2 x_2 = 0$ is the same as the hyperplane you found using the geometric approach in part (e).

Solution: The hyperplane

$$-\frac{1}{5} x_1 + \frac{3}{5} x_2 - \frac{9}{5} = 0$$

is equivalent because it is a scaled version of $-x_1 + 3x_2 - 9 = 0$.

4. (*AdaBoost.* By Eric Xing, Ziv Bar-Joseph at CMU.) We are interested in building an ensemble of decision stumps (trees with only one split) with the AdaBoost algorithm. The labeled training data and the first decision stump are illustrated above.



(a) First decision stump.

(b) The second decision stump will be a split on x_1 between the second and third samples along the x_1 axis.

Figure 7: Labeled training data and decision boundary of firstand second decision stumps for AdaBoost. The arrow points toward the positive decision region.

For the following questions, you should be able to answer without any computations. Explain each answer.

(a) Which training point(s) would have their weight increase for the next boosting iteration?

Solution: the negative sample that was misclassified (is on the positive side of the decision boundary).

(b) Draw the decision boundary of a stump that could be learned in the next boosting iteration.

Solution: See Figure 7b. The second decision stump will also have one misclassified sample. However, the negative sample that was misclassified in the previous iteration now has higher weight than the other samples, so it will prefer to misclassify any *other* point.

(c) Which of these two stumps will have a higher coefficient in the ensemble? In other words, will $\alpha_2 > \alpha_1$, or vice versa?

Solution: $\alpha_2 > \alpha_1$. Both stumps classify five points correctly and one point incorrectly. But the second stump correctly classifies the point with increased weight, so it has a higher weighted classification accuracy.