CS 5350/6350: Machine Learning Fall 2016

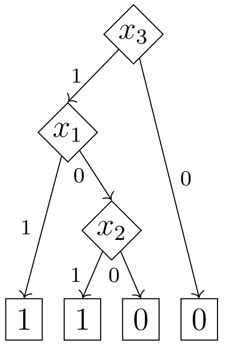
Homework 1

Gopal Menon

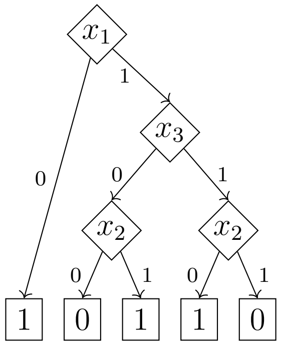
September 12, 2016

1 Decision trees (35 points)

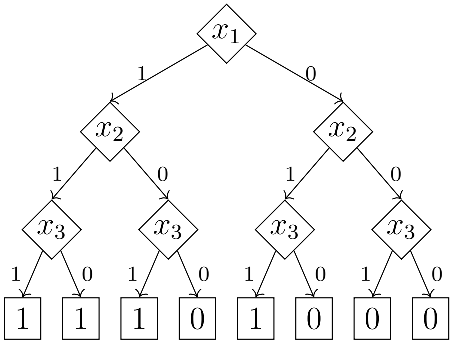
1. (a) $(x_1 \vee x_2) \wedge x_3$



   (b) $(x_1 \wedge x_2)$ xor $(\neg x_1 \vee x_3)$



   (c) The 2-of-3 function defined as follows: at least 2 of $\{x_1, x_2, x_3\}$ should be true for the output to be true.



2. (a) [2 points] How many possible functions are there to map these four features to a Boolean decision?

The number of possible rows in a truth table with all possible values of the four features will be

Number of types of Berries × Number of types of Balls × Number of colors × Number of Pokémon types
$= 2 \times 3 \times 3 \times 4 = 72$

Each row in this truth table can have a value of Yes or No as a label for whether the Pokémon can be caught or not. The Yes and No can be represented as bits 1 and 0. Each possible function will map to one combination of these labels and will be of the form of a 72-bit binary number. So the total possible number of functions will be 2^{72}, which is the count of all the possible 72-bit binary numbers.

(b) [2 points] What is the entropy of the labels in this data? (When calculating entropy, The base of the logarithm should be 2.)

The entropy of a collection S where the target label can take on c different values is defined as [1]

$$Entropy(S) = \sum_{i=1}^{c} -p_i log_2 p_i$$

where p_i is the proportion of S belonging to label i.

$$\begin{aligned} Entropy(Pok\acute{e}monData) &= -\frac{8}{16} log_2 \frac{8}{16} - \frac{8}{16} log_2 \frac{8}{16} \\ &= -\frac{1}{2} log_2 \frac{1}{2} - \frac{1}{2} log_2 \frac{1}{2} \\ &= -log_2 \frac{1}{2} \\ &= 1 \end{aligned}$$

(c) [8 points] Calculate information gain for four features respectively. Keep 3 significant digits.

$$\begin{aligned}
Values(Berry) &= Yes, No \\
S &= [8+, 8-] \\
S_{Yes} &= [6+, 1-] \\
S_{No} &= [2+, 7-] \\
Gain(S, Berry) &= Entropy(S) - \frac{7}{16} Entropy(S_{Yes}) - \frac{9}{16} Entropy(S_{No}) \\
&= 1 - \frac{7}{16} \times 0.5917 - \frac{9}{16} \times 0.7642 \\
&= 0.311
\end{aligned}$$

$$\begin{aligned}
Values(Ball) &= Poké, Great, Ultra \\
S_{Poké} &= [1+, 5-] \\
S_{Great} &= [4+, 3-] \\
S_{Ultra} &= [3+, 0-] \\
Gain(S, Ball) &= Entropy(S) - \frac{6}{16} Entropy(S_{Poké}) - \frac{7}{16} Entropy(S_{Great}) - \frac{3}{16} Entropy(S_{Ultra}) \\
&= 1 - \frac{6}{16} \times 0.65 - \frac{7}{16} \times 0.9852 - \frac{3}{16} \times 0 \\
&= 0.3252
\end{aligned}$$

$$\begin{aligned}
Values(Color) &= Green, Yellow, Red \\
S_{Green} &= [2+, 1-] \\
S_{Yellow} &= [3+, 4-] \\
S_{Red} &= [3+, 3-] \\
Gain(S, Color) &= Entropy(S) - \frac{3}{16} Entropy(S_{Green}) - \frac{7}{16} Entropy(S_{Yellow}) \\
&\quad - \frac{6}{16} Entropy(S_{Red}) \\
&= 1 - \frac{3}{16} \times 0.9183 - \frac{7}{16} \times 0.9852 - \frac{6}{16} \times 1 \\
&= 0.0218
\end{aligned}$$

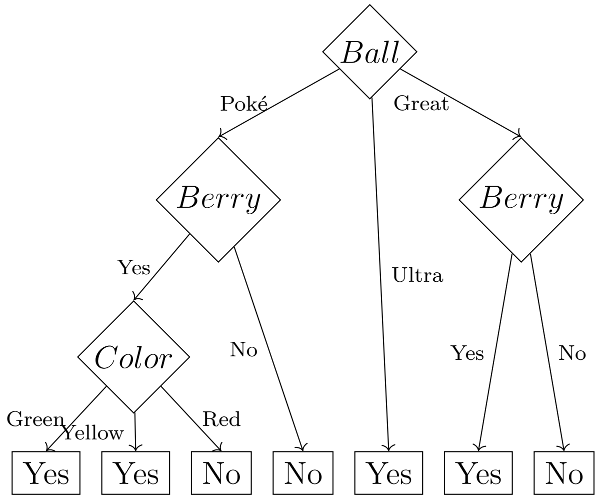
$$\begin{aligned}
Values(Type) &= Normal, Water, Flying, Psychic \\
S_{Normal} &= [3+, 3-] \\
S_{Water} &= [2+, 2-] \\
S_{Flying} &= [3+, 1-] \\
S_{Psychic} &= [0+, 2-]
\end{aligned}$$

$$\begin{aligned}
\mathit{Gain}(S, \mathit{Type}) &= \mathit{Entropy}(S) - \frac{6}{16}\mathit{Entropy}(S_{\mathit{Normal}}) - \frac{4}{16}\mathit{Entropy}(S_{\mathit{Water}}) \\
&\quad - \frac{4}{16}\mathit{Entropy}(S_{\mathit{Flying}}) - \frac{2}{16}\mathit{Entropy}(S_{\mathit{Psychic}}) \\
&= 1 - \frac{6}{16} \times 1 - \frac{4}{16} \times 1 - \frac{4}{16} \times 0.8113 - \frac{2}{16} \times 0 \\
&= 0.172
\end{aligned}$$

(d) [3 points] According to your results, using ID3 algorithm which attribute should be root for the decision tree?

The root should be feature *Ball*, since it has the largest entropy gain.

(e) [4 points] Construct a decision with the root you selected in the previous question. You do not have to use the ID3 algorithm here, you can show any tree with the chosen root.



(f) [2 points] Using your decision tree to predict label in the test set in the table below, what is your label for the each example? What is your accuracy?

Only one out of three was classified correctly. So accuracy is low.

Berry	Ball	Color	Type	Caught	Prediction
Yes	Great	Yellow	Psychic	Yes	Yes
Yes	Poké	Green	Flying	No	Yes
No	Ultra	Red	Water	No	Yes

(g) [1 points] Do you think it is a good idea to use decision tree in this Pokémon Go problem?

It is not a good idea to use a decision tree for this particular Pokémon Go problem. The training data looks to be not enough for a learning algorithm and the test data seems to be adversarial. A decision tree may perform better with a larger test data set.

Berry	Ball	Color	Type	Caught
Yes	Great	Yellow	Psychic	Yes
Yes	Poké	Green	Flying	No
No	Ultra	Red	Water	No

Table 1: Test set for Pokémon Go

3. Recall that in the ID3 algorithm, we want to identify the best attribute that splits the examples that are relatively pure in one label. Apart from entropy, which you used in the previous question, there are other methods to measure impurity. One such impurity measure is the Gini measure, that is used in the CART family of algorithms. If there are k possible outcomes $1, \cdots, i, \cdots, k$, each with a probability $p_1, \cdots, p_i, \cdots, p_k$ of occurring, the Gini measure is defined as:

$$Gini(p_1, \cdots, p_k) = 1 - \sum_{i=1}^{k} p_i^2$$

The Gini measure can be used to replace entropy in the definition of information gain to pick the best attribute.

(a) [4 points] Using the Gini measure, calculate the information gain for the four features respectively. Use 3 significant digits.

$$\begin{aligned}
Gini(Pok\acute{e}monData) &= 1 - \left(\frac{8}{16}\right)^2 - \left(\frac{8}{16}\right)^2 \\
&= 1 - 0.25 - 0.25 \\
&= 0.5 \\
Gain(S, Berry) &= Gini(S) - \frac{7}{16}Gini(S_{Yes}) - \frac{9}{16}Gini(S_{No}) \\
&= 0.5 - \frac{7}{16}\left(1 - \left(\frac{6}{7}\right)^2 - \left(\frac{1}{7}\right)^2\right) - \frac{9}{16}\left(1 - \left(\frac{2}{9}\right)^2 - \left(\frac{7}{9}\right)^2\right) \\
&= 0.5 - \frac{7}{16} \times \frac{12}{49} - \frac{9}{16} \times \frac{28}{81} \\
&= 0.5 - 0.1071 - 0.1944 \\
&= 0.198
\end{aligned}$$

$$\begin{aligned}
Gain(S, Ball) &= Gini(S) - \frac{6}{16}Gini(S_{Poké}) - \frac{7}{16}Gini(S_{Great}) - \frac{3}{16}Gini(S_{Ultra}) \\
&= 0.5 - \frac{6}{16}\left(1 - \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2\right) - \frac{7}{16}\left(1 - \left(\frac{4}{7}\right)^2 - \left(\frac{3}{7}\right)^2\right) \\
&- \frac{3}{16}\left(1 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2\right) \\
&= 0.5 - \frac{6}{16} \times \frac{10}{36} - \frac{7}{16} \times \frac{24}{49} - \frac{3}{16} \times 0 \\
&= 0.5 - 0.1042 - 0.2143 \\
&= 0.181
\end{aligned}$$

$$\begin{aligned}
Gain(S, Color) &= Gini(S) - \frac{3}{16}Gini(S_{Green}) - \frac{7}{16}Gini(S_{Yellow}) - \frac{6}{16}Gini(S_{Red}) \\
&= 0.5 - \frac{3}{16}\left(1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2\right) - \frac{7}{16}\left(1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2\right) \\
&- \frac{6}{16}\left(1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2\right) \\
&= 0.5 - \frac{3}{16} \times \frac{4}{9} - \frac{7}{16} \times \frac{24}{49} - \frac{6}{16} \times \frac{1}{2} \\
&= 0.5 - 0.0833 - 0.2143 - 0.1875 \\
&= 0.015
\end{aligned}$$

$$\begin{aligned}
Gain(S, Type) &= Gini(S) - \frac{6}{16}Gini(S_{Normal}) - \frac{4}{16}Gini(S_{Water}) - \frac{4}{16}Gini(S_{Flying}) \\
&- \frac{2}{16}Gini(S_{Psychic}) \\
&= 0.5 - \frac{6}{16}\left(1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2\right) - \frac{4}{16}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) \\
&- \frac{4}{16}\left(1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2\right) - \frac{2}{16}\left(1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2\right) \\
&= 0.5 - \frac{6}{16} \times \frac{1}{2} - \frac{4}{16} \times \frac{1}{2} - \frac{4}{16} \times \frac{6}{16} - \frac{2}{16} \times 0 \\
&= 0.5 - 0.1875 - 0.125 - 0.09375 \\
&= 0.094
\end{aligned}$$

(b) [3 points] According to your results in the last question, which attribute should be the root for the decision tree? Do these two measures (entropy and Gini) lead to the same tree?

As per the results in the last question based on Gini calculation, the attribute *Berry* must be the root for the decision tree as it has the maximum gain. In the case of Entropy calculation, the attribute *Ball* was the root of the decision tree. So the two measures will lead to different trees.

2 Linear Classifiers (15 points)

In the questions in this section, we have four features x_1, x_2, x_3 and x_4 and the label is represented by o.

1. [3 points] Write a linear classifier that correctly classifies the given dataset. You don't need to run any learning algorithm here. Try to find the weights and the bias of the classifier using the definition of linear separators.

x_1	x_2	x_3	x_4	o
0	0	0	1	1
0	0	1	1	1
0	0	0	0	-1

I obtained the weights and bias by trial and error.

$$w = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$b = -1$$

2. [5 points] Suppose the dataset below is an extension of the above dataset. Check if your classifier from the previous question correctly classifies the dataset. Report its accuracy.

x_1	x_2	x_3	x_4	o
1	0	1	1	1
0	1	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1
1	1	1	0	1
0	0	1	0	-1

Using the above classifier, the output is

Classifier Output	*Expected Output*
1	1
1	1
1	1
-1	1
1	1
1	1
1	-1

The classifier correctly classified 5 out of 7 outputs, giving an accuracy of 71.43%.

3. [7 points] Given the remaining missing data points of the above dataset in the table below, find a linear classifier that correctly classifies the whole dataset (all three tables together).

x_1	x_2	x_3	x_4	o
0	1	0	0	-1
0	1	1	0	-1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	1	0	1	1

Upon writing down the entire dataset, it becomes evident that the output depends only on the value of x_1. When $x_1 = 0$, the output is -1 and when $x_1 = 1$, the output is 1. When that is the case, the weight and bias will be

$$w = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$b = -1$$

3 Experiments

In this question you will be implementing a decision tree learner. You will experiment with the decision tree hyperparameters using cross-validation.

This problem uses the Mushroom Data set from the UCI machine learning repository. Each data point has 22 features indicating different characteristics of a mushroom. You can find definitions of each feature in the `mushroom.names` file. Your goal is to use the ID3 algorithm on the provided training datasets to train a predictor and see how well it does on the test data. You may use Java, Python, Matlab, C/C++ for this assignment. If you want to use a different language, you must contact the instructor first. Any other language you may want to use **MUST** run on the CADE machines.

Cross-Validation

The depth of the tree is a hyper-parameter to the decision tree algorithm that helps reduce overfitting. You will see later in the semester that many machine learning algorithm (SVM, logistic-regression etc) have some hyper-parameters as their input. One way to determine a proper value for the hyper-parameter is to use a technique called *cross-validation*.

As usual we have a training set and a test set. Our goal is to discover good hyperparameters using the training set. To do so, you can put aside some of the training data aside, and when training is finished, you can test the resulting classifier on the held out data. This allows you to get an idea of how well the particular choice of hyper-parameters does. However, since you did not train on your whole dataset you may have introduced a statistical bias in the classifier. To correct for this, you will need to train many classifiers with different subsets of the training data removed and average out the accuracy across these trials.

For problems with small data sets, a popular method is the leave-one-out approach. For each example, a classifier is trained on the rest of the data and the chosen example is then evaluated. The performance of the classifier is the average accuracy on all the examples. The downside to this method is for a data set with n examples you must train n different classifiers. Of course, this is not practical for the data set you will use in this problem, so you will hold out subsets of the data many times instead.

Specifically, for this problem, you should implement k-fold cross validation. The general approach for k-fold cross validation is the following: Suppose you want to evaluate how good a particular hyper-parameter is. You split the training data into k parts. Now, you will train your model on $k-1$ parts with the chosen hyper-parameter and evaluate the trained model on the remaining part. You should repeat this k times, choosing a different part for evaluation each time. This will give you k values of accuracy. Their *average cross-validation accuracy* gives you an idea of how good this choice of the hyper-parameter is. To find the best value of the hyper-parameter, you will need to repeat this procedure for different choices of the hyper-parameter. Once you find the best value of the hyper-parameter, use the value to retrain you classifier using the entire training set.

Setting A [25 points]

1. [10 points] **Implementation**

   For this problem, you will be using the data found in the `SettingA` folder. This folder contains two files, `SettingA/training.data` and `SettingA/test.data`. In this setting you will be training your algorithm on the training file (`SettingA/training.data`). Remember that you should not look at or use your testing file until your algorithm is complete.

   (a) [4 points] Implement the decision tree data structure and the ID3 algorithm for your decision tree. (Remember that the decision tree need not be a binary tree!). For debugging your implementation, you can use the previous toy examples like the Pokémon data from Table 1. Discuss what approaches or choices you had to make during this implementation.

   (b) [2 points] Report the error of your decision tree on the `SettingA/training.data` file.

   (c) [5 points] Report the error of your decision tree on the `SettingA/test.data` file.

   (d) [1 points] Report the maximum depth of your decision tree.

2. [15 points] **Limiting Depth**

   In this section you will be using 6-fold cross-validation in order to limit the depth of your decision tree, effectively pruning the tree to avoid overfitting. We have split the data into 6 parts for you: you will be using the 6 cross-validation files for this section, titled SettingA/CVSplits/training_0X.data where X is a number between 0 and 5 (inclusive).

(a) [10 points] Run 6-fold cross-validation using the specified files. Experiment with depths in the set $\{1, 2, 3, 4, 5, 10, 15, 20\}$, reporting the average cross-validation accuracy and standard deviation for each depth.

(b) [5 points] Using the depth with the greatest average cross-validation accuracy from your experiments: train your decision tree on the SettingA/training.data file. Report the accuracy of your decision tree on the SettingA/test.data file.

Setting B [25 points]

1. [10 points] **Experiments**

   For this problem, you will be using the data found in the SettingB folder. This folder contains the two files, SettingB/training.data and SettingB/test.data. In this setting you will be training your algorithm on the training file (SettingB/training.data). Remember that you should not look at or use your testing file until your algorithm is complete. You are not limiting the depth of your tree in this section.

   (a) [2 points] Report the error of your decision tree on the SettingB/training.data file.

   (b) [2 points] Report the error of your decision tree on the SettingB/test.data file.

   (c) [2 points] Report the error of your decision tree on the SettingA/training.data file.

   (d) [2 points] Report the error of your decision tree on the SettingA/test.data file.

   (e) [1 points] Report the maximum depth of your decision tree.

2. [15 points] **Limiting Depth**

   In this section you will be using 6-fold cross-validation in order to limit the depth of your decision tree, effectively pruning the tree to avoid overfitting. You will be using the 6 cross-validation files for this section, titled SettingB/CVSplits/training_0X.data where X is a number between 0 and 5 (inclusive).

   (a) [10 points] Run 6-fold cross-validation using the specified files. Experiment with depths in the set $\{1, 2, 3, 4, 5, 10, 15, 20\}$, reporting the cross-validation accuracy and standard deviation for each depth. Explicity specify which depth should be chosen as the best, and explain why.

   (b) [5 points] Using the depth with the greatest cross-validation accuracy from your experiments: train your decision tree on the SettingB/training.data file. Report the accuracy of your decision tree on the SettingB/test.data file.

Setting C (CS 6350 Students) [20 points]

In this setting, you are investigating what effect missing features have on a decision tree, and exploring which approach is most effective in dealing with missing features. More specifically, you will be trying:

- **Method 1:** Setting the missing feature as the majority feature value.
- **Method 2:** Setting the missing feature as the majority value of that label.
- **Method 3:** Treating the missing feature as a *special* feature.

The missing feature is represented by a `?` character. In order to determine which method is the best, you will be using 6-fold cross-validation, with the files being titled `SettingC/CVSplits/training_0X.data` where `X` is a number between 0 and 5 (inclusive). These files, along with `SettingC/training.data` and `SettingC/test.data` can be found in the `SettingC` folder.

1. [5 points] Update your decision tree implementation to have functionality to deal with missing features. Describe your approach/any choices you had to make in this implementation.
2. [10 points] Perform 6-fold cross-validation on each of the 3 methods described above. Report the accuracy for each method and the standard deviation.
3. [5 points] Using the best method selected from your experiments, train your decision tree on `SettingC/training.data`, and report the accuracy of your tree on `SettingC/test.data`.

References

[1] Mitchell, Tom M. “Decision Tree Learning.” *Machine Learning.* New York: McGraw-Hill, 1997. N. pag. Print.