CPSC 340 Assignment 1 (due September 18)

Summary Statistics and Data Visualization, Decision Tress and Cross-Validation, Probability

- You can work in groups on the assignments. However, please hand in your own assignments and state the group members that you worked (as well as other sources of help like online material).
- It preferred that you type up the solution to your assignment. If you are going to submit hand-written parts, please write legibly. Marks will be taken off for unreadable or unclear solutions.
- Please organize your submission according to the sections used in this document.
- Place your name and student number on the first page, and (if submitting a paper company) then please staple your document together.
- All Sections (1-4) are equally weighted.
- There may be updates/clarifications to the assignment after the first version is put online. Any modifications will be marked in red.
- We may change from paper submission to an electronic submission as a PDF file. If that change takes place, instructions will be placed here and the assignment will still be due at the start of Friday's class.

1 Logistic Survey

Please fill out the survey located here:
`https://survey.ubc.ca/surveys/37-7d0090012c11ea5c07f0bca610f/cpsc-340-logistic-survey`

2 Summary Statistics and Data Visualization

Download and expand the file *a1.zip*, which contains data on the athletes from the last summer olympics.[1] You can load this data into Matlab from the directory containing the file using:

```
load london2012.csv
```

This creates a matrix 'london2012', where each row corresponds to an athlete and the columns correspond to:

1. Age.
2. Height.
3. Weight.
4. Gender (1 - female, 0 - male).

[1] Data obtained at: `http://www.theguardian.com/sport/datablog/2012/aug/07/olympics-2012-athletes-age-weight-height#data`, and I removed athletes with missing values.

5. Number of bronze medals.
6. Number of silver medals.
7. Number of gold medal.

2.1 Summary Statistics

Report the following statistics:

1. Range of age values (i.e., minimum and maximum).
2. Median of age value for each gender.
3. The 10%, 25%, 50%, 75%, and 90% age quantiles.

Answer:

1. Minimum age is 13, maximum age is 71.
2. Median height for females is 25, median height for males is 26.
3. The quantiles are 20, 23, 26, 29, 33.

Code:

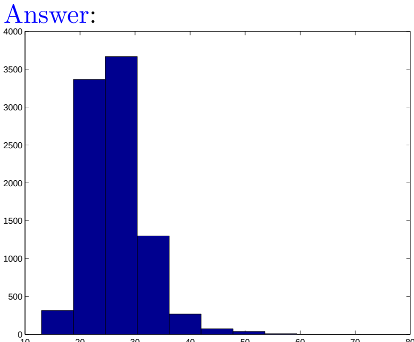
```
min(london2012(:,1))
max(london2012(:,1))

F = find(london2012(:,4)==1);
M = find(london2012(:,4)==0);
median(london2012(F,1))
median(london2012(M,1))

quantile(london2012(:,1),[.1 .25 .5 .75 .9])
```

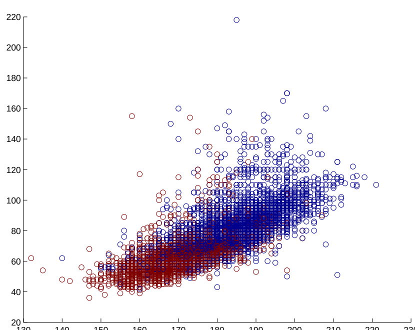
2.2 Data Visualization

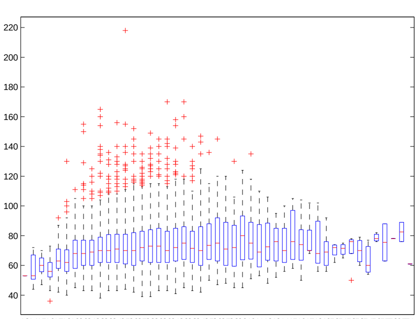
Show the following figures:

1. Histogram of age values.
2. Scatterplot of height and weight values, coloured by gender.
3. Boxplot of weight values for each age value.

Answer:







Code:
hist(london2012(:,1));
scatter(london2012(:,2),london2012(:,3),[],london2012(:,4));
boxplot(london2012(:,3),london2012(:,1));

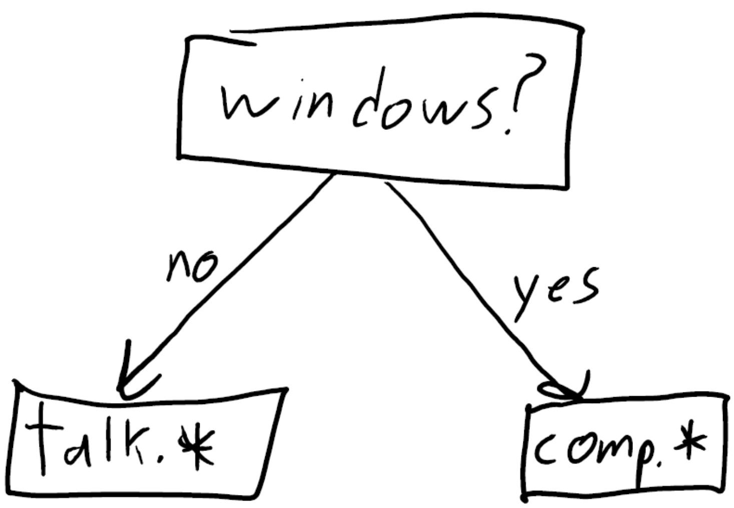
3 Decision Trees and Cross-Validation

The file *newsgroups.mat* is a Matlab file containing the following objects:

1. *groupnames*: The names of four newsgroups.
2. *wordlist*: A list of words that occur in posts to these newsgroups.
3. X: A sparse binary matrix. Each row corresponds to a post, and each column corresponds to a word from the word list. A value of 1 means that the word occured in the post.
4. y: A vector with values 1 through 4, with the value corresponding to the newsgroup that the post came from.
5. *Xtest* and *ytest*: the word lists and newsgroup labels for additional newsgroup posts.

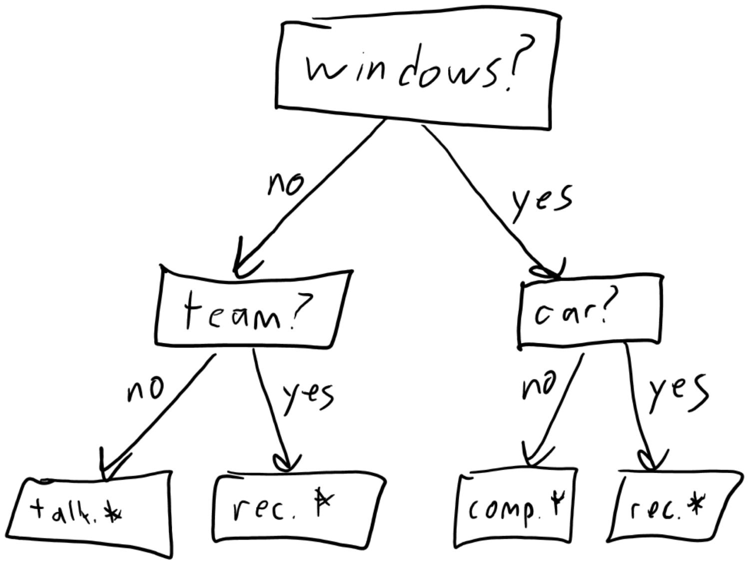
3.1 Decision Stumps and Decision Trees

The function *example_decisionStump* shows how to load the data, fit a decision stump to the training data, and then evaluate the error of the model on the training data. Running the demo shows that decision stumps have a classification error of 0.60, which is a bit better than just predicting the most common label (which obtains an error of 0.66). The image below gives an interpretation of the decision stump that is learned:



This is an interpretable but not very accurate model. Modify this demo so that it uses the provided *decisionTree* function rather than the *decisionStump* function. By looking through the decision tree code, draw a picture (similar to the one above) showing the learned decision tree when the maximum depth is 2, and report the accuracy when using a decision tree of depth 10.

Answer:



The error achieved using a tree of depth 10 is 0.38.

3.2 Cost of Fitting Decision Stumps and Decision Trees

The bottleneck in fitting a decision stump with binary features is the loop over the features. If we have D features, we pass through this loop D times. Inside this loop, the bottlenecks are computing operations that involve going through all N examples, and applying a simple $O(1)$ operation to each example. Thus, the total cost of fitting a decision stump is $O(ND)$, which is the size of the dataset. This indicates that fitting decision stumps is very fast.[2] What is the cost of fitting a decision tree of depth M in terms of N, D, and M? (Hint: even thought there could be 2^{M-1} decision stumps, keep in mind not every stump will need to go through every example. Note also that we stop growing the decision tree if a node has no examples, so we may not even need to do anything for many of the 2^{M-1} decision stumps.)

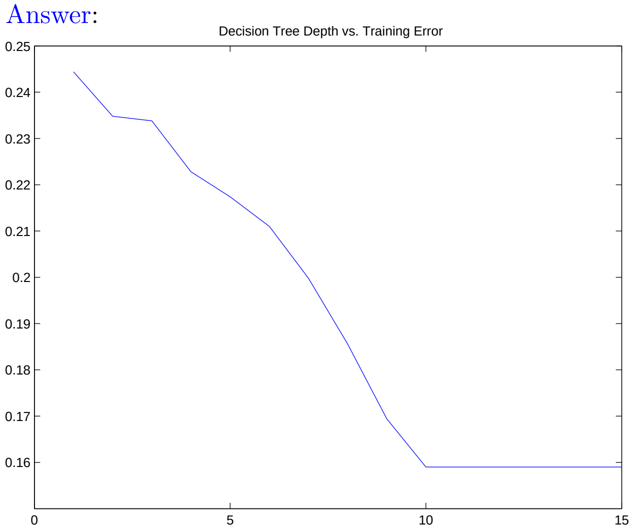
Answer:
The number of stumps that we need to fit for a decision tree of depth M will be 2^{M-1}, so a naive analysis would indicate that we need $O(ND2^{M-1})$ time. However, at each depth we have split the N examples across the decision stumps. This means that each depth will only need to look at N examples, and the cost for each layer is still $O(ND)$. This gives a total cost $O(NDM)$ to go through all M layers.

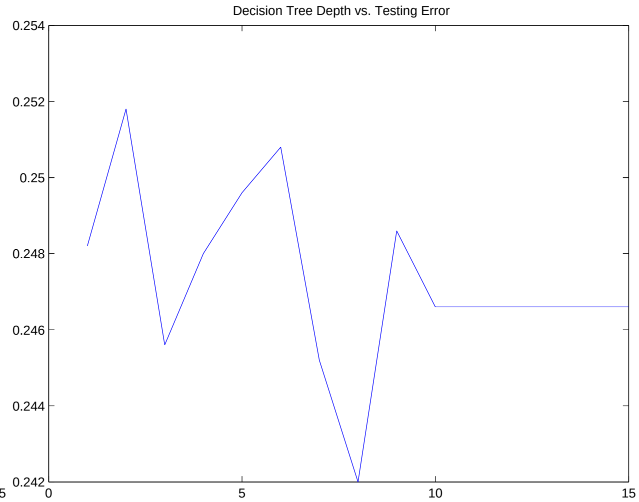
3.3 Training Error vs. Testing Error

The function *example_trainTest* shows how to evaluate the training and testing error of a decision tree fit with the information-gain criterion on a dataset with 10 features. Modify this demo to make a plot with the depth of the decision tree on the x-axis (varying it from 1 through 15) and the error on the training data $\{X, y\}$ on the y-axis. Make the same plot, but instead of the training error (on $\{X, y\}$ plot the testing error on $\{Xtest, ytest\}$).

[2] If you aren't familiar with big-O notation for analyzing algorithms, some useful links are: `https://rob-bell.net/2009/06/a-beginners-guide-to-big-o-notation` and `https://www.interviewcake.com/article/big-o-notation-time-and-space-complexity`

Answer:





Here is the code to produce these plots:

```
load DTdata.mat

[N,D] = size(X);
T = length(ytest);
depthVals = 1:15
for d = 1:length(depthVals)
    depth = depthVals(d)
    model = decisionTree_InfoGain(X,y,depth);
    yhat = model.predictFunc(model,X);
    errorTrain(d) = sum(yhat ~= y)/N;
    yhat = model.predictFunc(model,Xtest);
    errorTest(d) = sum(yhat ~= ytest)/T;
end
figure(1);
plot(depthVals,errorTrain);
title('Decision Tree Depth vs. Training Error');
figure(2);
plot(depthVals,errorTest);
title('Decision Tree Depth vs. Testing Error');
```

3.4 Cross-Validation

On the 10-feature dataset from Question 3.3 (DTdata.mat), compute the 2-fold cross-validation scores on the training data alone (using decisionTree_infoGain). To split the data, use examples 1 to 2500 as the first fold and examples 2501 through 5000 as the second fold. Report the cross-validation error for all depths 1 through 15 (averaging over the error for the two folds), and report the depth that would be chosen by cross-validation.

Answer:
Here are the values I got:
Depth 1: 0.2542
Depth 2: 0.2566
Depth 3: 0.2540
Depth 4: 0.2610
Depth 5: 0.2446
Depth 6: 0.2494
Depth 7: 0.2446
Depth 8: 0.2416
Depth 9: 0.2460

Depth 10: 0.2474
Depth 11: 0.2474
Depth 12: 0.2474
Depth 13: 0.2474
Depth 14: 0.2474
Depth 15: 0.2474
Here is the code to produce these numbers:

```
X1 = X(1:N/2,:);
y1 = y(1:N/2);
X2 = X(N/2+1:end,:);
y2 = y(N/2+1:end,:);
depthVals = 1:15;
for d = 1:length(depthVals)
    depth = depthVals(d)

    % Fold 1
    model = decisionTree_InfoGain(X1,y1,depth);
    yhat = model.predictFunc(model,X2);
    errorFold1(d) = sum(yhat ~= y2)/(N/2);

    % Fold 2
    model = decisionTree_InfoGain(X2,y2,depth);
    yhat = model.predictFunc(model,X1);
    errorFold2(d) = sum(yhat ~= y1)/(N/2);
end
(errorFold1'+errorFold2')/2
```

The minimum value occurs at a depth of 8. This data was actually generated from a depth-3 decision tree, so a much larger depth was needed because of the greedy way that the decision tree is constructed.

4 Probability Exercises

Please read the *Notes on Probability* on the course webpage, if you need a refresher on probabilities. Use probabilistic arguments to address the following problems. Show your calculations in addition to giving the final result. The last problem comes from Chapter 2 of Murphy's Machine Learning book.

4.1 Bayes rule for drug testing

Suppose a drug test produces a positive result with probability 0.99 for drug users, $P(T = 1|D = 1) = 0.99$. It also produces a negative result with probability 0.99 for non-drug users, $P(T = 0|D = 0) = 0.99$. The probability that a random person uses the drug is 0.001, so $P(D = 1) = 0.001$.

What is the probability that a random person who tests positive is a user, $P(D = 1|T = 1)$?

Answer:

$$
\begin{aligned}
p(D=1|T=1) &= \frac{p(T=1|D=1)p(D=1)}{p(T=1|D=1)p(D=1)+p(T=1|D=0)p(D=0)} \\
&= \frac{(0.99)(0.001)}{(0.99)(0.001)+(1-0.99)(1-0.001)} \\
&= \frac{0.0099}{0.0099+0.0999} = \frac{0.0099}{0.1098} \\
&\approx 0.09
\end{aligned}
$$

4.2 Two sons problem

I independently toss two fair coins (each having a 0.5 probability of landing 'heads' and 0.5 probability of landing 'tails'). If I tell you that the first coin landed 'heads', what is the probability that the second coin landed 'heads'. If I instead tell you that at least one coin landed 'heads', what is the probability that both coins land heads?

Answer:
We'll use the notation $C_1 = H$ for coin 1 landing heads, $C_1 = T$ for coin 1 landing tails, and similarly for coin 2. Because they are fair coins we know that

$$p(C_1 = H) = 0.5, \quad p(C_1 = T) = 0.5, \quad p(C_2 = H) = 0.5, \quad p(C_2 = T) = 0.5,$$

and because they are tossed independently we know that

$$p(C_1, C_2) = p(C_1)p(C_2).$$

To answer the first part of the question, we have from the definition of conditional probability that

$$p(C_2 = H|C_1 = H) = \frac{p(C_1 = H, C_2 = H)}{p(C_1 = H)} = \frac{P(p_1 = H)p(C_2 = H)}{p(C_1 = H)} = \frac{(0.5)(0.5)}{0.5} = 0.5.$$

One way to answer the second part of the question is to use that the probability of at least one 'head' is

$$p(C_1 = H \cup C_2 = H) = p(C_1 = H) + p(C_2 = H) - p(C_1 = H, C_2 = H) = 0.5 + 0.5 - 0.25 = 0.75.$$

and use Bayes rule to get

$$\begin{aligned} p(C_1 = H, C_2 = H|C_1 = H \cup C_2 = H) &= \frac{p(C_1 = H \cup C_2 = H|C_1 = H, C_2 = H)p(C_1 = H, C_2 = H)}{p(C_1 = H \cup C_2 = H)} \\ &= \frac{(1)(0.5)(0.5)}{0.75} = 1/3 \end{aligned}$$

This seems counter-intuitive (especially when you replace 'heads or tails' with 'son or daughter') because the information 'at least one head' implies that one coin lands heads (which seems like it's the same information as telling you coin 1 landed heads), but this is weaker information than knowing which coin landed heads. For the related paradox, see `http://en.wikipedia.org/wiki/Boy_or_Girl_paradox`

4.3 Prosecutor's fallacy

A crime has been committed in a large city and footprints are found at the scene of the crime. The guilty person matches the footprints. Out of the innocent people, 1% match the footprints by chance. A person is interviewed at random and his/her footprints are found to match those at the crime scene. Determine the probability that the person is guilty, or explain why this is not possible.

Answer:
Let F be the event that the footprints match. Let G be the event that the person is guilty, and $\neg G$ be the event that the person is innocent. We are told that $p(F|G) = 1$ and $p(F|\neg G) = 0.01$. We are asked to compute $p(G|F)$. By Bayes rule, this is

$$p(G|F) = \frac{p(F|G)p(G)}{p(F)} = \frac{p(F|G)p(G)}{p(F|G)p(G) + p(F|\neg G)p(\neg G)}$$

We cannot determine the probability of guilt without knowing the prior probability of any one person being guilty, $p(G)$.