CS 486/686 Assignment 2:
Decision Trees and Variable Elimination (100 marks)

Jesse Hoey & Josh Jung

Due Date: March 5, 2024; Time: 11:59pm Waterloo Time (EST)

Instructions

- Submit written solutions in a file named `writeup.pdf` to the A2 Dropbox on `Learn` (https://learn.uwaterloo.ca). If your written submission on Learn does not contain **one** file named `writeup.pdf`, we will deduct 5 marks from your final assignment mark.
- No late assignment will be accepted.
- This assignment is to be done individually. All code you write **must be your own**
- The Due Date is March 5, 2024, 11:59pm Waterloo Time (EST)
- Lead TAs:
  - Max Ku (m3ku)
  - Mehdi Bolourian (mbolouri)

  The TAs' office hours will be scheduled on Piazza.

Question 1 [60pts]: Text Categorization with Decision Trees

Text categorization is an important task in natural language processing and information retrieval. For instance, news articles, emails or blogs are often classified by topics. In this assignment, you will implement (in the language of your choice) a decision tree algorithm to learn a classifier that can assign a discussion board topic to an article.

Train and test your algorithm with the provided datasets in trainData.txt and testData.txt. These datasets contain a subset of Reddit posts sourced from `https://files.pushshift.io/reddit/` and processed using Google BigQuery. The dataset includes the first 1500 comments of August 2019 of each of the r/books and r/atheism subreddits, cleaned by removing punctuation and some offensive language, and limiting the words to only those used more than 3 times among all posts. These 3000 comments are split evenly into training and testing sets (with 1500 documents in each). To simplify your implementation, these posts have been pre-processed and converted to the *bag of words* model. More precisely, each post is converted to a vector of binary values such that each entry indicates whether the document contains a specific word or not.

Each line of the files trainData.txt and testData.txt are formatted "docId wordId" which indicates that word wordId is present in document docId. The files trainLabel.txt and testLabel.txt indicate the label/category (1=`atheism` or 2=`books`) for each document (docId = line#). The file words.txt indicates which word corresponds to each wordId (denoted by the line#). Your code will need to read this information from the provided files.

Implement a decision tree learning algorithm. Here, each decision node corresponds to a word feature, and the leaf nodes correspond to a prediction of what subreddit the article belongs in. You will experiment with two feature selection mechanisms as follows:

1. average information gain (evenly weighted across the leaves)

$$I_G = I(E) - [\frac{1}{2} * I(E_1) + \frac{1}{2} * I(E_2)]$$

2. information gain weighted by the fraction of documents on each side of the split, as we discussed in class

$$I_G = I(E) - [\frac{N_1}{N} I(E_1) + \frac{N_2}{N} I(E_2)]$$

where $I(E)$ is the information content in the N examples before the split, and $I(E_i)$ is the information content of the N_i examples in the i^{th} leaf after the split. Method (1) does not deal well with splits that put a lot of examples on one side, and very few on the other. For example, suppose you have evenly distributed data across the target class (so $I(E) = 1$), N examples, and 2 features. Suppose the first feature splits one example off from the other $N - 1$, so Method (2) gives:

$$\begin{aligned} I(E_1) &= 0, \\ I(E_2) &\approx 1, \\ I_G &\approx 1 - [\frac{1}{N} * 0 + \frac{N-1}{N} * 1] = \frac{1}{N} \end{aligned}$$

Using Method (2), I_G is a small value. On the other hand, Method (1) will make I_G look better than it should be, since it will compute:

$$I_G \approx 1 - [\frac{1}{2} * 0 + \frac{1}{2} * 1] = \frac{1}{2}.$$

Use $I(\{\}) = 1$ (the entropy of the empty set is maximal).

As discussed in class, build each decision tree by maintaining a priority queue of leaves, sorted by the maximum value of the feature (the information gain of the best performing next feature to split on). At each iteration, split the leaf

where the split will give the largest information gain (according to each of the measures above), and do the split on the word feature that gives that highest gain. Grow the tree one node at a time, up to 100 internal nodes, by training with the training set only. The point estimate should be the majority class at the leaf (not a probability).

Plot the training and testing accuracy (i.e., percentage of correctly classified articles) after adding each new node. Generate two graphs (one for each method of feature selection) and plot two curves on each (one for training accuracy and one for test accuracy). These graphs should have accuracy as a percentage on the y-axis and number of nodes (1-100) on the x-axis. You may generate them however you wish (e.g. Matplotlib for Python).

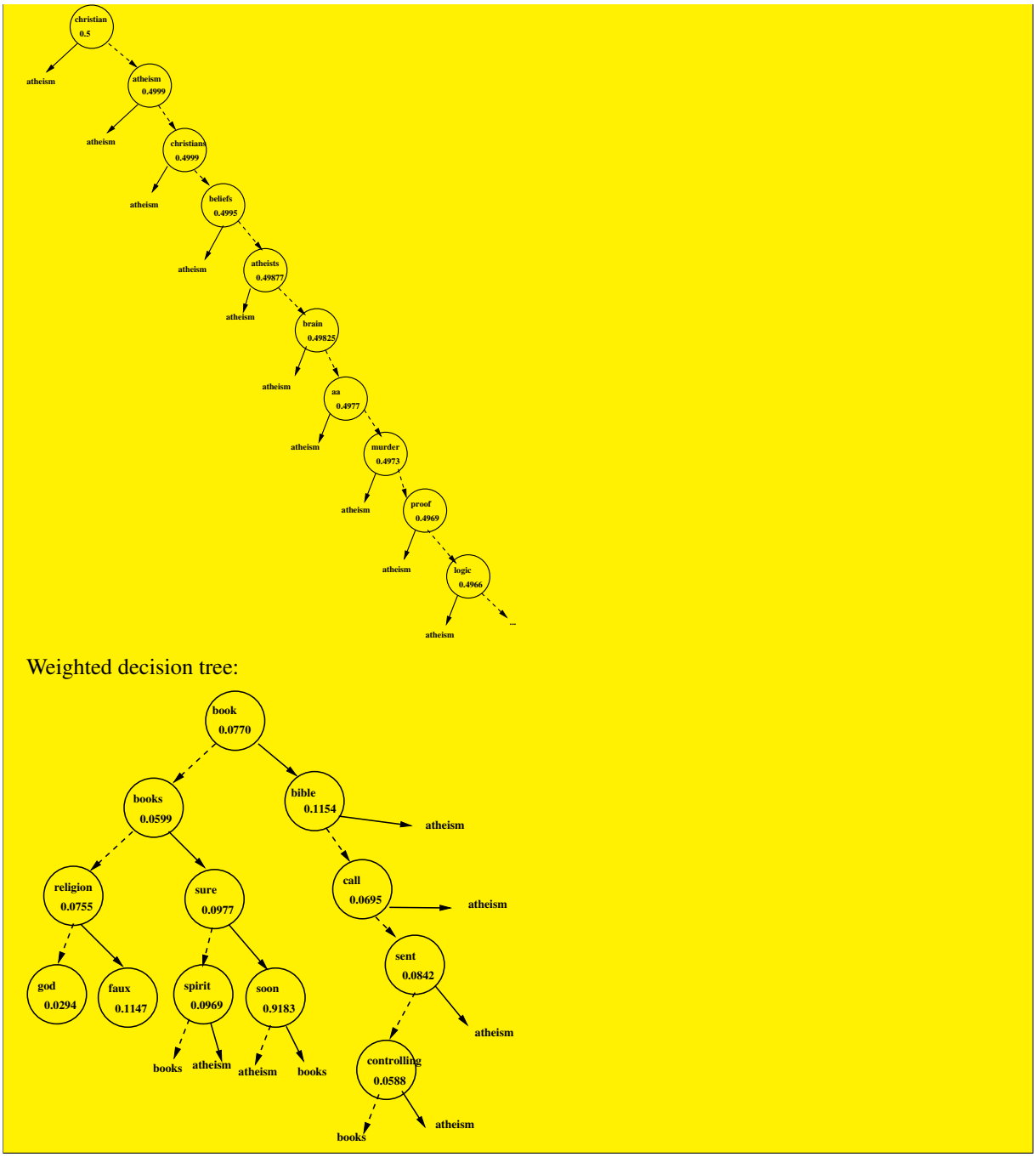
What to hand in:

a) **[20 pts]** A printout of your code. Your code must be your own, and you *must* use the iterative priority queue model. If your code is recursive, or doesn't expand nodes by order of information gain throughout the tree, you will not get these 20pts.

> **SOLUTION**: Here is what I told the students:
>
> The code should be implementing the PQ version of the decision tree. the markers will try to see if this is the case and award the 20pts if they see this is done. If you make the code, correct, efficient, properly commented and "styled", it will be easier for the markers to see what you are doing. If they cannot figure out what your code is doing, or if they see you are doing the recursive version, you will not get the 20pts.

b) **[20 pts]** A printout (or hand drawing) showing two decision trees (one tree for each feature selection method) after adding the first 10 word features. Show the (method-specific) information gain and word feature selected at each internal node. Show the prediction at each leaf.

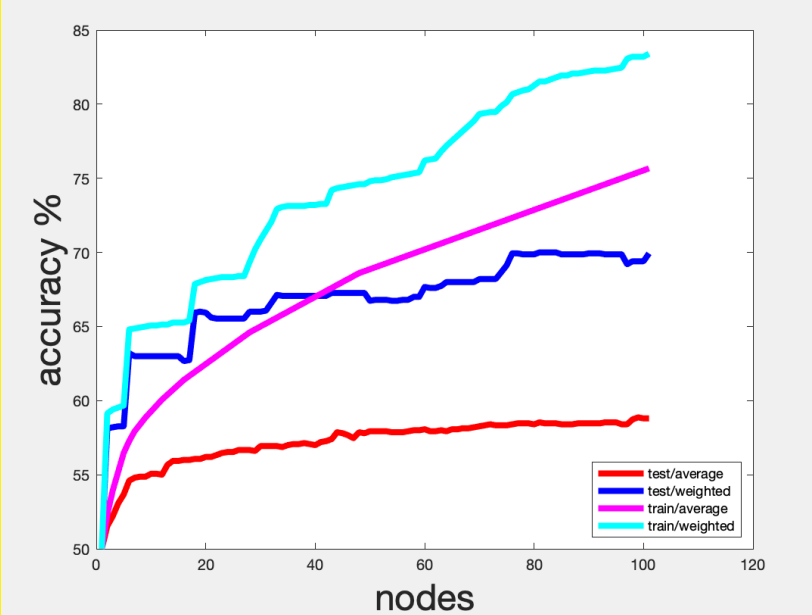
> **SOLUTION**:
>
> There may be different decision trees submitted as there are ties in the data, however, the weighted one should look a lot more balanced than the unweighted one. To mark this question, just ensure the overall structure of the trees look ok and the words selected and IG values are roughly right. You can deduct marks if the tree is hard to visualize (i.e. if they submit some text version)
>
> dashed arrows mean the word is not in the document
>
> solid arrows mean it is
>
> Average decision tree: (DONE IN LOG BASE 2)



Each printed or drawn tree should have 10 internal nodes and 11 leaves.

c) **[20 pts]** Two graphs (one for each feature selection method) showing the training and testing accuracy as the number of nodes increases from 1 to 100.

SOLUTION: Here is the graph I got:



I also tried with a different tie-breaking method (take the last one not the first one) and got about the same diagram. Again, here, just look for the overall structure - they should be getting close to 70%/60% for weighted and average methods.

Question 2 [40pts]: Bayesian Networks and Variable Elimination

Suppose you are working for a financial institution and you are asked to implement a fraud detection system. You plan to use the following information:

- When the card holder is travelling abroad, fraudulent transactions are more likely since tourists are prime targets for thieves. More precisely, 1% of transactions are fraudulent when the card holder is travelling, where as only 0.4% of the transactions are fraudulent when they are not travelling. On average, 5% of all transactions happen while the card holder is travelling. If a transaction is fraudulent, then the likelihood of a foreign purchase increases, unless the card holder happens to be travelling. More precisely, when the card holder is not travelling, 10% of the fraudulent transactions are foreign purchases where as only 1% of the legitimate transactions are foreign purchases. On the other hand, when the card holder is travelling, then 90% of the transactions are foreign purchases regardless of the legitimacy of the transactions.
- Purchases made over the internet are more likely to be fraudulent. This is especially true for card holders who don't own any computer. Currently, 60% of the population owns a computer and for those card holders, 1% of their legitimate transactions are done over the internet, however this percentage increases to 2% for fraudulent transactions. For those who don't own any computer, a mere 0.1% of their legitimate transactions is done over the internet, but that number increases to 1.1% for fraudulent transactions. Unfortunately, the credit card company doesn't know whether a card holder owns a computer, however it can usually guess by verifying whether any of the recent transactions involve the purchase of computer related accessories. In any given week, 10% of those who own a computer purchase with their credit card at least one computer related item as opposed to just 0.1% of those who don't own any computer.

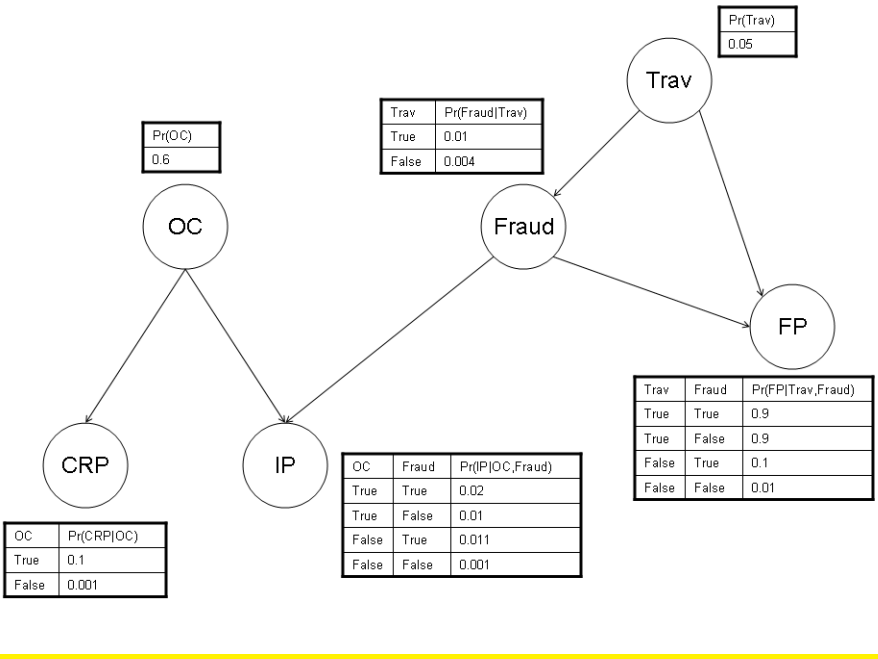
a) **[10 pts]** Construct a Bayes Network that your fraud detection system can use to indentify fraudulent transactions.

What to hand in: Show the graph defining the network and the Conditional Probability Tables associated with each node in the graph. This network should encode the information stated above. Your network should contain exactly six nodes, corresponding to the following binary random variables:

- OC – card holder owns a computer.
- $Fraud$ – current transaction is fraudulent.
- $Trav$ – card holder is currently travelling.
- FP – current transaction is a foreign purchase.
- IP – current purchase is an internet purchase.
- CRP – a computer related purchase was made in the past week.

The arcs defining your Bayes Network should accurately capture the probabilistic dependencies between these variables.

SOLUTION: The tables need to contain only a value for variable=true (or false) - as this is how I did it in class



b) **[15 pts]** What is the prior probability (i.e., before we search for previous computer related purchases and before we verify whether it is a foreign and/or an internet purchase) that the current transaction is a fraud?

What is the probability that the current transaction is a fraud once we have verified the following 3 facts: (1) it is a foreign transaction; (2) it is not an internet purchase; and (3) the card holder purchased computer related accessories in the past week? (Note: we are describing one query, not three separate queries.)

What to hand in: Indicate the two queries (i.e., $Pr(variables|evidence)$) you used to compute those probabilities. Show the factors computed at each step of variable elimination. For each step, show a formula giving the new factor to be created, e.g.

$$f_7(B) = \sum_A [f_2(B,A) * f_3(A) * f_6(A,B)]$$

B	$f_7(B)$
t	0.007
f	0.640

And show the resulting factor (you don't need to show all steps of the calculation). Use the following variable ordering when summing out variables in variable elimination: $Trav$, FP, $Fraud$, IP, OC, CRP. No marks are awarded for answering the questions using external software.

SOLUTION: **Query:** $Pr(fraud)$

Here students can sum over just the relevant variable, Trav.

$$\begin{aligned} Pr(fraud) &= Pr(fraud|trav)Pr(trav) + Pr(fraud|\neg trav)Pr(\neg trav) \\ &= 0.01 * 0.05 + 0.004 * 0.95 \\ &= 0.0043 \end{aligned}$$

With variable elimination, we have:

$Trav$	$f_1(Trav) = Pr(fraud\|Trav)$
t	0.01
f	0.004

$Trav$	$f_2(Trav) = Pr(Trav)$
t	0.05
f	0.95

$$\begin{aligned} Pr(fraud) &= sumout_{Trav}[f1(Trav)f2(Trav)] \\ &= 0.01 * 0.05 + 0.004 * 0.95 \\ &= 0.0043 \end{aligned}$$

Query: $Pr(fraud|fp, \neg ip, crp)$

OC	$f_1(OC) = Pr(OC)$
t	0.6
f	0.4

$Fraud$	$Trav$	$f_2(Fraud, Trav) = Pr(Fraud\|Trav)$
t	t	0.01
t	f	0.004
f	t	0.99
f	f	0.996

$Trav$	$f_3(Trav) = Pr(Trav)$
t	0.05
f	0.95

OC	$f_4(OC) = Pr(crp\|OC)$
t	0.1
f	0.001

OC	$Fraud$	$f_5(OC, Fraud) = Pr(\neg ip\|OC, Fraud)$
t	t	0.98
t	f	0.99
f	t	0.989
f	f	0.999

$Trav$	$Fraud$	$f_6(Trav, Fraud) = Pr(fp\|Trav, Fraud)$
t	t	0.9
t	f	0.9
f	t	0.1
f	f	0.01

We have:

$$\begin{aligned}Pr(fraud|fp,\neg ip,crp) &\propto sumout_{OC,Trav}[f_1(OC)*f_2(Fraud,Trav)*f_3(Trav)*f_4(OC)*f_5(OC,Fraud)*f_6(Trav,Fraud)]\\ &= sumout_{OC}[f_1(OC)*f_4(OC)*f_5(OC,Fraud)*sumout_{Trav}[f_2(Fraud,Trav)*f_3(Trav)*f_6(Trav,Fraud)]]\\ &= sumout_{OC}[f_1(OC)*f_4(OC)*f_5(OC,Fraud)*f_7(Fraud)]\\ &= f_7(Fraud)*sumout_{OC}[f_1(OC)*f_4(OC)*f_5(OC,Fraud)]\\ &= f_7(Fraud)*f_8(Fraud)\end{aligned}$$

Eliminate variable $Trav$:

$$f_7(Fraud) = sumout_{Trav}[f_2(Fraud,Trav)*f_3(trav)*f_6(Trav,Fraud)]$$

$Fraud$	$f_7(Fraud)$
t	0.0008
f	0.0540

Eliminate variable OC:

$$f_8(Fraud) = sumout_{OC}[f_1(OC)*f_4(OC)*f_5(OC,Fraud)]$$

$Fraud$	$f_8(Fraud)$
t	0.0592
f	0.0598

$$Pr(fraud|fp,\neg ip,crp) = k*f_7(fraud)*f_8(fraud) = 0.0150$$

where k is a normalizing constant:

$$k = \frac{1}{f_7(fraud)*f_8(fraud) + f_7(\neg fraud)*f_8(\neg fraud)}$$

c) **[10 pts]** After computing those probabilities, the fraud detection system raises a flag and recommends that the card holder be called to confirm the transaction. An agent calls at the domicile of the card holder but she is not home. Her spouse confirms that she is currently out of town on a business trip. How does the probability of fraud change based on this new piece of information? Use all evidence you have from Question 2b and this new evidence.

What to hand in: Same as for Question 2b.

SOLUTION: The restriction operation can be done first - so they can define factors after restricting.

OC	$f_1(OC) = Pr(OC)$
t	0.6
f	0.4

$Fraud$	$f_2(Fraud) = Pr(Fraud\|trav)$
t	0.01
f	0.99

$Trav$	$f_3() = Pr(trav)$
	0.05

OC	$f_4(OC) = Pr(crp\|OC)$
t	0.1
f	0.001

OC	$Fraud$	$f_5(OC, Fraud) = Pr(\neg ip\|OC, Fraud)$
t	t	0.98
t	f	0.99
f	t	0.989
f	f	0.999

$Fraud$	$f_6(Fraud) = Pr(fp\|trav, Fraud)$
t	0.9
f	0.9

Eliminate variable OC:

$$f_7(Fraud) = sumout_{OC}[f_1(OC) * f_4(OC) * f_5(OC, Fraud)]$$

$Fraud$	$f_7(Fraud)$
t	0.0592
f	0.0598

$$Pr(fraud|fp, \neg ip, crp, trav) = k * f_2(fraud) * f_3() * f_6(fraud) * f_7(fraud) = 0.0099$$

where k is a normalizing constant:

$$k = \frac{1}{\sum_{Fraud} f_2(Fraud) * f_3() * f_6(Fraud) * f_7(Fraud)}$$

d) **[5 pts]** Suppose you are not a very honest employee and you just stole a credit card. You know that the fraud detection system uses the Bayes net designed earlier but you still want to make an important purchase over the internet. What can you do prior to your internet purchase to reduce the risk that the transaction will be rejected as a possible fraud? This question is independent of question 2b and 2c.

What to hand in: State the action taken and indicate by how much the probability of a fraud gets reduced. Follow the same instructions as for Question 2b.

SOLUTION: When an internet purchase is made, the fraud detection system is likely to believe that the transaction is fraudulent unless it has reasons to believe that the card holder owns a computer. Therefore, an ingenious thief could simply make a computer related purchase first to fool the fraud detection system into believing the card holder owns a computer. After that, the thief can make the intended internet purchase with

a reduced risk of being rejected.

One can verify that the probability of a fraudulent transaction decreases when a computer related purchase is observed since $Pr(fraud|ip) = 0.0109$ whereas $Pr(fraud|ip, crp) = 0.0086$.