CSE 573 Spring 2014

Assignment 4: Learning and Decision Theory

Turn in a zip file that contains your code, README and report, via Catalyst Dropbox by 11pm, June 4.

1. Decision Theory

1.1 Economists often make use of an exponential utility function for money: $U(x) = -e^{-x/R}$, where R is a positive constant representing an individual's risk tolerance. Risk tolerance reflects how likely an individual is to accept a lottery with a particular expected monetary value (EMV) versus some certain payoff. As R (which is measured in the same units as x) becomes larger, the individual becomes less risk-averse.

(a) (8 points) Assume Mary has an exponential utility function with R=400. Mary is given the choice between receiving $400 with certainty (probability 1) or participating in a lottery that has a 60% probability of winning $5000 and a 40% probability of winning nothing. Assuming Mary acts rationally, which option would she choose? Show how you derived your answer.

(b) (7 points) Consider the choice between receiving $100 with certainty (probability 1) or participating in a lottery that has a 50% probability of winning $500 and a 50% probability of winning nothing. Approximate the value of R (to 3 significant digits) in an exponential utility function that would cause an individual to be indifferent to these two alternatives.

1.2 Consider a student who has the choice to buy or not buy a textbook for a course. We'll model this as a decision problem with one Boolean decision node, B, indicating whether the agent chooses to buy the honk, and two Boolean chance nodes, M, indicating whether the student has mastered the material in the book, and P, indicating whether the student passes the course. Of course, there is also a utility node, U. A certain student, Sam, has an additive utility function: 0 for not buying the book and -$100 for buying it; and $2000 for passing the course and 0 for not passing. Sam's conditional probability estimates are as follows:

P(p|b,m) = 0.9	P(p|b, ¬m) = 0.5	P(p|¬b,m) = 0.8	P(p|¬b, ¬m) = 0.3
~~P(p|¬b) = 0.3~~	P(m|b) = 0.9	P(m|¬b) = 0.7	

You might think that P would be independent of B given M. But this course has an open-book final, so having the book helps.

(a) (5 points) Draw the decision network for this problem.

(b) (10 points) Compute the expected utility of buying the book and of not buying it. What should Sam do?

2. Decision Tree Induction

2.1 Description

The project is based on a task posed in KDD Cup 2000. It involves mining clickstream data collected from Gazelle.com. Please browse the KDD Cup website to understand the domain, and read the organizer's report. Your task is Question 1 from the KDD Cup: Given a set of page views, will the visitor view another page on the site or will he leave?

The data set given to you are in a different form from the original. We have discretized numerical values ~~by partitioning them into 5 intervals of equal frequency~~. This way, we get a data set where, for each feature, we have a finite set of values. We mapped these values to integers, so that the data is easier for you to handle.

2.2 Datasets

Download the compressed file containing the data (HW4Source/Data) from the course website. There you will find 5 files in .csv format:

- trainfeat.csv: Contains 40,000 examples, each with 274 features in the form of a 40,000 × 274 matrix.
- trainlabs.csv: Contains the labels (class) for each training example (did the visitor view another page?)
- testfeat.csv: Contains 25,000 examples, each with 274 features in the form of a 25,000 × 274 matrix.
- testlabs.csv: Contains the labels (class) for each testing example.
- featnames.csv: Contains the "names" of the features. These might be useful if you need to know what the features represent.

Remember that in a "real" application, you can synthesize new features from the original ones if you think that they will be useful (if you want you can try that for yourself, it is not required). The format of the files is not complicated, just rows of integers separated by empty spaces.

2.3 Chi-squared criterion

What about split stopping? Suppose that the attribute that we want to split on is irrelevant. Then we would expect the positive and negative examples (labels 1 and 0) to be distributed according to a specific distribution. Suppose that splitting on attribute T, will produce sets $\{T_i\}_{i=1}^{m}$. Let p and n denote the number of positive and negative examples that we have in our dataset (not the whole set, the remaining one that we work on at this node). Let (N is the total number of examples in the current dataset):

$$p_i' = p\frac{|T_i|}{N}$$
$$n_i' = n\frac{|T_i|}{N}$$

be the expected number of positives and negatives in each partition, if the attribute is irrelevant to the class. Then the statistic of interest is:

$$S = \sum_{i=1}^{m} \left(\frac{(p_i' - p_i)^2}{p_i'} + \frac{(n_i' - n_i)^2}{n_i'} \right)$$

where p_i and n_i are the positives and negatives in partition T_i. The main idea is that S is distributed (if the class is irrelevant) accoring to a chi-squared distribution with m−1 degrees of freedom (if you want you can work out why, not required).

Now we must compute the p-value. This is the probability of observing a value X at least as extreme as S coming from the distribution. To find that, we compute P(X ≥ S). The test is passed if the p-value is smaller than some threshold. Remember, the smaller that probability is, the more unlikely it is that S comes from a chi-squared distribution and consequently, that T is irrelevant to the class. You may use the provided code *Chi.java*.

2.4 Implementation

(40 points) Your task is to implement the ID3 decision tree learner. You may program in Java or Python. If you are not proficient in either of these, please contact the TA. Your program should take the given .csv files as input. For initial debugging, it is recommended that you construct a very simple data set (e.g., based on a boolean formula) and test your program on it. Your program should use the chi-squared split-stopping criterion with the p-value threshold given as a parameter.

2.5 Questions

Use your implementation with the threshold for the criterion set to 0.05, 0.01 and 1. Remember, 1 corresponds to growing the full tree. Answer the following questions:

(a) (12 points) For each value of the threshold, what is your tree's accuracy and size (size equals number of internal nodes and leaves)? What do you observe? If all your accuracies are low, tell us what you have tried to improve the accuracies and what you suspect is failing.
(b) (5 points) Explain which options work well and why.
(c) (5 points) For your best-performing tree, choose a path that you think explains a large fraction of the data. Pick the first few nodes and argue whether they make sense or not (as in do they intuitively provide a lot of information).
(d) (8 points) Suppose you have 10 − 20 different values for the p-value threshold. How would you decide on the best model and why?
    - Would you pick the model with the lowest training error? Explain why.
    - Would you pick the model with the lowest testing error? Explain why.
    - Would you do something else? Explain why.
(e) Extra Credit (5 points): Try to improve your predictive accuracy. There are many approaches you could try. You could construct new attributes from the existing ones and augment the examples with them. You could also try alternative classification techniques, or modify one that you used above. Use the unfiltered data provided if necessary. Explain method and results.

Your report should also contain a good high-level description of how your code works. Document well any complicated parts you might have. You are not required to provide code that is extremely well optimized, but you should look for any obvious, gross optimizations that are possible.