Machine Learning, 10-701 and 15-781, 2003

Homework 1: Decision Trees

Primary contact for questions and clarifications: Rong Zhang (rongz@cs.cmu.edu)

Due Date: 10.30am Thursday October 2nd

Reading: Chapter 3 of "Machine Learning" By Tom Mitchell

Overview: This assignment gives you an opportunity to review the knowledge you learned in the class and implement the Decision Tree algorithm on real world data sets. The assignment consists of two parts. In the first part, you are asked some questions that help your understanding of information theory and the Decision Tree algorithm. In the second part, you will implement the ID3 learning algorithm and evaluate your program with two data sets. There is also an extra credit section for the benefit of people who want to try out new ideas.

Groupwork: If you wish, you may work in a team of two people. If you'd like to work in a pair but do not have an obvious partner available, please email Andrew (awm@cs.cmu.edu) who will act as a lonely-hearts matching service. A pair should hand in one copy of the homework with both their names and Andrew IDs on the copy.

Cheating. Please do not copy answers from other people (or other pairs). Do not explicitly collaborate with other individuals or pairs. We will, of course, follow up aggressively if we detect such cheating or collaboration. You may discuss general issues relevant to the homework.

What you hand in. Please hand in your work in hard copy form at the start of class on the day that the homework is due. Please include your first and last names and your Andrew userids in very clear block capital letters (your Andrew id will be used as the primary key in our grades database).

Part 1 Problem Set

1 Entropy, Conditional Entropy and Information Gain

1.1 Imagine that we have a memoryless source W of English text, emitting words from an alphabet $A = \{w_1, w_2, ..., w_n\}$ according to a distribution P. The *entropy* of the source W is

$$H(W) = -\sum_{i=1}^{n} P(w_i)\log_2 P(w_i) .$$

This is the average number of bits required to specify each word in a sequence of words output by W, assuming P is known.

Please show that $H(W) \le \log_2 n$.

(A memoryless source generates symbols from a discrete alphabet $A = \{w_1, w_2, ..., w_n\}$ by drawing independently at random from some distribution P over A.)

1.2 Continue 1.1. Now suppose we don't know P. By observing the output of W for a long time, we can come up with a guess at P, which we'll call Q. The *cross-entropy* of P given Q is defined as

$$H_Q(W) = -\sum_{i=1}^{n} P(w_i) \log_2 Q(w_i),$$

which is the average number of bits required to specify each word in the output from W, under the assumption that the source emits according to the distribution Q.
Please show that $H(W) \le H_Q(W)$.

1.3 Assume we have a new memoryless source S that emits word pairs $< x, y >$, in which $x \in \{x_1, x_2, ..., x_m\}$ and $y \in \{y_1, y_2, ..., y_n\}$, according to a probability $P(x = x_i, y = y_j)$. We have learned the definitions of *Specific Conditional Entropy* $H(Y \mid x = x_i)$, *Average Conditional Entropy* $H(Y \mid X) = \sum_{i=1}^{m} P(x = x_i) H(Y \mid x = x_i)$, and *Information Gain* $IG(Y \mid X) = H(Y) - H(Y \mid X)$.
Please show that $IG(Y \mid X) = IG(X \mid Y)$.

1.4 Continue 1.3. Please show that $IG(Y \mid X) \ge 0$ and describe what situation can lead to $IG(Y \mid X) = 0$.

Hint: You may need to use some classical inequalities, e.g. $\ln x \le x - 1$ for $x > 0$. You don't need to worry about the extreme situation that $P(x = x_i) = 0$ when dealing with $1/P(x)$; Just assume this won't happen.

2 Decision Tree Questions

2.1 A student told me he could build a discrete data set with two classes, and I am allowed to choose ***part*** (not all) of the instances from it to train a Decision Tree. He claimed that no matter how I choose the data and train the tree, the classification error that my Decision Tree makes on the instances not contained in the training set will be no less than 50%. Do you think he is correct? Explain why or give an example.

2.2 Let's consider a "new" tree growing algorithm for training set with discrete attributes.
Algorithm:

- Create a *Root* node that owns all the training instances, and add it to the *tree*.
- For each *leaf* node *n* in the tree (*Leaf* node is the node that doesn't have children):
  - If node *n* has been labeled with class, move to the next available *leaf* node.
  - If there are attributes that haven't been used by *ancestors* of *n* for splitting, just ***randomly*** select one of them. Create children nodes for node *n*, and assign training instances owned by node *n* to the corresponding child

    - node, according to the values of the selected attribute and instance. Add the children nodes to the *tree*.
  - Otherwise, label node *n* with the dominant class in its training instances.
- Repeat until all the *leaf* nodes are labeled.

Assume there are sufficient training instances in our training set to ensure that every possible combination of attribute values will have a non-empty subset of instances. Now look at one of the *leaf* nodes (call it l). By construction, the input patterns for node l are exactly the same, say $(A_1 = a_1, A_2 = a_2, ..., A_k = a_k)$, for all the training instances owned by l, where A_i denotes one attribute, and a_j is a possible value for attribute A_j. It would be trivial to estimate the distributions (on training set) of $P(A_1 = a_1, A_2 = a_2, ..., A_k = a_k)$ and $P(class = c_i \mid A_1 = a_1, A_2 = a_2, ..., A_k = a_k)$ using knowledge we have learned.

Please give an expression for the training error rate of this new algorithm in terms of $P(A_1 = a_1, A_2 = a_2, ..., A_k = a_k)$ and $P(class = c_i \mid A_1 = a_1, A_2 = a_2, ..., A_k = a_k)$.

If we build a Decision Tree using the ID3 algorithm (no pruning) on the same training set, what do you expect for the training error rate of the Decision Tree compared with the error rate of the new algorithm? (e.g. ≥ , =, ≤, >, <, …) Why?

2.3 Now we have built two trees, one using the new algorithm, another using ID3 (no pruning). We are told that our estimates for $P(A_1 = a_1, A_2 = a_2, ..., A_k = a_k)$ and $P(class = c_i \mid A_1 = a_1, A_2 = a_2, ..., A_k = a_k)$ are not accurate, and we are given a test set generated by *true distributions* of attributes and classes. Which tree do you think will work better on the test set? We decide to use post-pruning technique for Decision Tree. We divide the training set into two parts: a pure training set and a hold-out set. We use the ID3 algorithm to grow a tree from the pure training set, and then prune it based on the hold-out set. Which tree do you think will work better this time? The tree built by our new algorithm, or the pruned ID3 tree? Briefly explain why.

3 Real-Valued Inputs

Assume we are investigating the relationship between the *weight* of a graduate student and the *degree* he or she is studying for (Masters or Ph.D.). The following table shows some data we collected in CMU. (The records have been sorted on the value of *weight* in pounds)

Weight	110.1	122.0	130.5	137.8	150.9	160.6	168.6	195.0	195.0
Degree	P	P	M	M	P	P	M	M	P

3.1 First we want to calculate Information Gain $IG(degree \mid weight)$. It has been shown that the value of splitting threshold (*cut-point*) that maximizes Information Gain must always lie at class a boundary. (See the textbook "Machine Learning" for reference.) Please show the candidate cut-points and the value of $IG(degree \mid weight)$.

3.2 Now we consider a "new" method, "three-way splitting", which has two thresholds, t_1 and t_2, that could split the input instances into three subsets, $weight \le t_1$,

$t_1 < weight \le t_2$ and $t_2 < weight$. Please show how to calculate the three-way Information Gain IG^3.

3.3 Please show that $IG(degree \mid weight) \le IG^3(degree \mid weight)$. (You don't need to calculate the explicit value. Instead, we would like a one or two sentence explanation or proof.)

3.4 Encouraged by the result of 3.3, we decide to try 4-way splitting, 5-way splitting, …, N-way splitting. What do you expect for the Information Gain, or will $IG^n \ge IG^{n-1}$? Is it a good idea to use N-way splitting for real-valued attribute? Why or why not?

4 Splitting Criteria

Entropy is a natural measure to quantify the impurity of a data set. The Decision Tree learning algorithm uses entropy as a splitting criterion by calculating the information gain to decide the next attribute to partition the current node. However, there are other impurity measures that could be used as the splitting criteria too. Let's investigate two of them. Assume the current node n has k classes $c_1, c_2, ..., c_k$.

Gini Impurity: $i(n) = 1 - \sum_{i=1}^{k} P(c_i)^2$

Misclassification Impurity: $i(n) = 1 - \max_{i=1}^{k} P(c_i)$

4.1 Assume node n has two classes, c_1 and c_2. Please draw a figure in which the three impurity measures (Entropy, Gini and Misclassification) are represented as the function of $P(c_1)$.

4.2 Now we can define new splitting criteria based on the Gini and Misclassification impurities, which is called *Drop-of-Impurity* in some literatures. That is the difference between the impurity of the current node and the impurities of children. For the binary category splits, *Drop-of-Impurity* is defined as

$$\Delta i(n) = i(n) - P(n_l)i(n_l) - P(n_r)i(n_r)$$

where n_l and n_r are left and right child of node n after splitting. Please calculate the *Drop-of-Impurity* (using both Gini and Misclassification Impurity) for the following example data set in which y is the class variable to be predicted.

A	a_1	a_1	a_1	a_2	a_2	a_2
y	y_1	y_1	y_2	y_2	y_2	y_2

4.3 We choose the attribute that can maximize the *Drop-of-Impurity* to split a node. Please create a data set and show that on this data set, Misclassification Impurity based $\Delta i(n)$ couldn't determine which attribute should be used for splitting (e.g. $\Delta i(n) = 0$ for all the attributes), but Information Gain and Gini Impurity based $\Delta i(n)$ can.

5 Optimal Growing

The Decision Tree growing algorithm we have looked at so far is a greedy, heuristic search with one-step look-ahead and no backtracking. Let's consider how this would improve if we were to look-ahead two steps.

Originally, we determined which attribute to select for splitting a node by comparing the information gain of each of the attributes from A_1 to A_D (assume we have *D* discrete attributes, each of which has two possible values). Now, for each attribute we will also consider the best way to split the resulting children. We will try each remaining attribute for both the left and right child separately, and then pick the pair (A_l, A_r) which maximizes

$$IG(S \mid A) + \frac{N_l}{N_l + N_r} IG(S_l \mid A_l) + \frac{N_r}{N_l + N_r} IG(S_r \mid A_r)$$

where N_l and N_r are the number of data points in the left and right children of the node which is considered how to split, and S_l and S_r are those two sets of data points themselves. We will call this $IG_2(S \mid A)$, the two-step look-ahead information gain for attribute A on data set S.

5.1 Briefly explain why we need the factors $\frac{N_l}{N_l + N_r}$ and $\frac{N_r}{N_l + N_r}$.

5.2 Assuming there are *D* discrete attributes with two possible values apiece, how many information gain computations must be carried out to determine how to split a single node?

5.3 Is this a richer model class (hypothesis language)? That is, does this change allow us to represent new functions from input variables to classifications that were not previously representable with Decision Trees? Briefly explain why or why not.

Part 2 Experiments

In this part, you are asked to implement the Decision Tree learning algorithm, and experiment your program with the simple *PlayTennis* learning data described in Chapter 3 of the textbook, and a larger set of data, the UCI *Adult* census database that aims to predict whether a person makes over 50k a year. You may program in any computer language, such as C, C++, Lisp, Matlab, Perl, …

1 The *PlayTennis* data

The training data for *PlayTennis* from Table 3.2 in the textbook is available on the file **play-tennis.ssv**.

1.1 Implement the Decision Tree learning algorithm (for discrete attribute only).

1.2 Take a look at the training data on **play-tennis.ssv**, and then run your decision tree learning program on all the training examples to be sure it produces the correct decision tree for this data (it should produce the decision tree shown in Figure 3.1 of the textbook). Now try running it on a randomly chosen subset containing half of the examples for training, and using half for test. What are the training and test accuracies?

Answer whether or not the actions in the next three questions are possible or not. If possible, demonstrate it by running the decision tree learner, and turn in the parameter settings you used and your data set. If impossible, explain why. When answering questions 1.3, 1.4, and 1.5, assume that (a) the target concept is the concept described by the decision tree of Figure 3.1 of the textbook; (b) all training examples you add must be consistent with this target concept (meaning that the classification you include for the example must be the one with which the tree in Figure 3.1 would label it), and (c) you are not allowed to include post pruning when learning decision trees.

1.3 Is it possible to get ID3 to further elaborate the tree below the rightmost leaf in Figure 3.1 (and make no other changes to the tree), by adding a single new *correct* training example to the original fourteen examples?

1.4 Is it possible to get ID3 to learn an incorrect tree (i.e., a tree that is not equivalent to the target concept of Figure 3.1) by adding new *correct* training examples to the original fourteen?

1.5 Is it possible to produce some set of *correct* training examples that will get ID3 to include the attribute Temperature in the learned tree, even though the true target concept is independent of Temperature?

2 The *Adult* data

The data is extracted from the UCI ***Adult*** database that aims to predict whether a person makes over 50k a year. Originally, each record had 14 attributes (6 continuous and 8 discrete) besides the class label. To make the programming easier, we dropped the 6 continuous attributes from the database. In addition, we also removed the examples with unknown values. These resulted us a cleaned dataset, in which each example has 8 discrete attributes only.

The dataset used for this experiment consists of two files: training file ***adult.train.10k.discrete*** and test file ***adult.test.10k.discrete***, each of which has 10,000 examples. The first attribute of each example denotes the class it belongs to, either "**>50K**" or "**<=50K**". The next 8 are the input features:

(1) **workclass** (8 values): *Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked*.

(2) **education** (16 values): *Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.*

(3) **marital-status** (7 values): *Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse*.

(4) **occupation** (14 values): *Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces*.

(5) **relationship** (6 values): *Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried*.

(6) **race** (5 values): *White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black*.

(7) **sex** (2 values): *Female, Male*.

(8) **native-country** (41 values): *United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands*.

You task is to learn a decision tree that predicts whether a person's income exceeds $50k/yr, based on his/her census data.

2.1 Use the *Adult* training file and your program to train a decision tree. Use 25% of the training data, and all the test data (no pruning). Rerun this experiment several times and notice the impact of different random selection of the training data. Report the sizes and accuracies of these trees over half a dozen distinct runs.

2.2 Measure the impact of training set size on the accuracy and the size of the learned tree (no pruning; use all of the test data for testing). Consider training set sizes in the range 0-60% (include at least the values .02, .1, .2, .3, .4, and .6 for training fractions). Because of the high variance due to random splits, repeat the experiment with ten different random seeds for each training set size, then report the mean, maximum, and minimum accuracies at each training set size. Also measure the mean, max, and min tree size. Turn in two plots, showing how accuracy varies with training set size, and how the number of nodes in the final tree varies with training set size. Indicate maximum, minimum, and mean values for each training set size.

2.3 Select one of the post-pruning strategies and implement it in your program. Measure the impact of pruning on accuracy and size of the learned trees, under the same conditions as in question 2.2. Use the same training set sizes (.02, .1, .2, .3, .4, and .6), but this time use 40% of the training data (4000 examples) for pruning, and all of the test data for testing. Again plot the mean, maximum, and minimum accuracy for each training set size, and plot the mean, maximum, and minimum tree size for each training set size. In addition, use all of the training data for tree growing (no pruning), and all the test data for testing. Report the accuracy and tree size. What is the impact, if any, of post-pruning the tree?

3 Extra Credit

Investigate one or more improvements to the basic Decision Tree algorithm. Your investigation might include new splitting criteria or new search methods or novel ways to prune. You should summarize your idea and results in a clearly written two-page summary.

What to hand in for experiment

- Your source code.
- A brief description of your program that shows me: What is your programming language? What type of OS does your program need (e.g. Windows NT, XP, linux, unix, etc.)? How to compile and run your program? And anything special that you think deserves explanation.
- Answers or reports for each question.