CS 5350/6350: Machine Learining Spring 2015

Homework 1

Handed out: Jan 21, 2015
Due date: Feb 4, 2015

General Instructions

- You are welcome to talk to other members of the class about the homework. I am more concerned that you understand the underlying concepts. However, you should write down (and program) your own solution. Please keep the class collaboration policy in mind.
- Feel free ask questions about the homework with the instructor or the TAs.
- Your written solutions should be brief and clear.
- Handwritten solutions will not be accepted.
- The homework is due by midnight of the due date. Please submit the homework on Canvas.
- Some questions are marked **For 6350 students**. Students who are registered for CS 6350 should do these questions. Of course, if you are registered for CS 5350, you are welcome to do the question too, but you will not get any credit for it.

1 Decision Trees

Several questions below ask you to write down decision trees. Report decision trees as a series of `if-then` statements. For example, you could write a tree as follows:

```
if feature 0 = x:
    if feature 1 = y:
        if feature 2 = z:
            class = +
        if feature 2 != z:
            class = -
    if feature 1 != y:
        class = +
if feature 0 != x:
    if feature 1 = r:
        class = +
    if feature 1 != r:
        class = -
```

1. [Boolean Functions, 9 points] Recall from class that Boolean functions can be represented as decision trees. Represent the following boolean functions as decision trees:

(a) $(A \vee B) \wedge C$ [3 points]

(b) $A \oplus B$ [3 points]

(c) $A \wedge \neg B \wedge \neg C \wedge D$ [3 points]

2. [Inducing Decision Trees, 21 points] For this question, you will use the Balloons dataset from the UCI Machine Learning repository[1]. The data consists of four attributes (Color, Size, Action and Age) and a binary label.

   The entire dataset is shown in Table 1 below.

Label	Color	Size	Action	Age
T	YELLOW	SMALL	STRETCH	ADULT
T	YELLOW	SMALL	STRETCH	CHILD
T	YELLOW	SMALL	DIP	ADULT
F	YELLOW	SMALL	DIP	CHILD
F	YELLOW	SMALL	DIP	CHILD
T	YELLOW	LARGE	STRETCH	ADULT
T	YELLOW	LARGE	STRETCH	CHILD
T	YELLOW	LARGE	DIP	ADULT
F	YELLOW	LARGE	DIP	CHILD
F	YELLOW	LARGE	DIP	CHILD
T	PURPLE	SMALL	STRETCH	ADULT
T	PURPLE	SMALL	STRETCH	CHILD
T	PURPLE	SMALL	DIP	ADULT
F	PURPLE	SMALL	DIP	CHILD
F	PURPLE	SMALL	DIP	CHILD
T	PURPLE	LARGE	STRETCH	ADULT
T	PURPLE	LARGE	STRETCH	CHILD
T	PURPLE	LARGE	DIP	ADULT
F	PURPLE	LARGE	DIP	CHILD
F	PURPLE	LARGE	DIP	CHILD

Table 1: Balloon Data Set

(a) Find the entropy of the balloon dataset. [2 points]

(b) Find the information gain of the Action feature. [2 points]

(c) Use the ID3 heuristic we have seen in the class to manually construct a decision tree that correctly classifies the data. [7 points]

(d) We will now develop another heuristic for learning decision trees instead of ID3. If, at some node, we stopped growing the tree and assign the majority label of the remaining examples at that node, then the empirical error on the training set at that node will be

$$MajorityError = 1 - \max(p, 1 - p)$$

[1]You can learn more about the UCI repository at `http://archive.ics.uci.edu/ml/`. Look for information about the balloons data set at `http://archive.ics.uci.edu/ml/datasets/Balloons`.

where, p is the fraction of examples that are labeled T and, hence, $1 - p$ is the fraction labeled F. Notice that $MajorityError$ can be thought of as a measure of impurity just like entropy.

Redefine information gain using $MajorityError$ as the measure of impurity. What is the redefined information gain of the Action feature? [3 points]

(e) Using the $MajorityError$-based impurity measure, construct a decision tree for the balloons data. [7 points]

3. (**For CS 6350 students**, 15 points) Use the first twelve examples to create decision trees using both the heuristics. How well do the trees perform on the remaining examples? Report the error rates (i.e the ratio of the errors to the number of examples) of the two algorithms.

2 Nearest Neighbors

The nearest neighbors algorithm partitions the space of examples into regions corresponding to different labels. In two dimensions, the decision boundaries can be represented as a Voronoi diagram, which shows regions of the plane associated with each label.

For this part, you will be drawing the decision boundaries for simple datasets.

1. [5 points] Using the Euclidean distance measure between points, show a Voronoi map corresponding to the nearest neighbor classification of the following four points. (That is, draw a diagram that shows how the nearest neighbor classification of the following four points partitions the two dimensional plane.)

Label	x	y
A	1	1
A	1	-1
B	-1	-1
C	2	-2

2. [5 points] Using the city-block distance measure, show a Voronoi map corresponding to the nearest neighbor classification of the following three points.

(Recall that the city-block measure/Manhattan distance/L_1 distance/taxicab metric between two points $\mathbf{x}$, $\mathbf{y}$ in the n dimensional space $\Re^n$ is defined as $\sum_{i=1}^{n} |x_i - y_i|$.)

Label	x	y
A	1	1
B	-1	-1
C	2	-2

3. [5 points] In the simple 1-nearest neighbor algorithm, the label for a point is the same as the label of its nearest neighbor in the training set. We will define a variant of this method, which we will call the *weighted nearest neighbor algorithm.* If there are M training points, *all* of them contribute to the score of final label.

   Formally, say the training points are represented by $\mathbf{x}_i$ with their corresponding label y_i. A test point is $\mathbf{x}$, then a label, say A, is associated with a score that is a function W of the distance of $\mathbf{x}$ from all the training points labeled as A.

$$Score(A) = \sum_{i=1;y_i=A}^{M} \frac{1}{d(\mathbf{x}_i, \mathbf{x})}.$$

   Here, d denotes the distance between points. For the purpose of this question, we will assume that d is the Euclidean distance. The final classification is the label with the highest score.

   Using the weighted nearest neighbor algorithm with all four points from question 1 above, what would the classification of a test point at $(4, 0)$ be? Show how you get the label.

3 The Badges Game (again)

In this question, you will implementing decision tree learners and the K-nearest neighbors algorithms.

This problem uses a variant on the badges data shown in the first lecture. You can find the data on the assignments page of the class website in a file called `badges.tar.gz`. It consists of two files – `badges-train.txt` and `badges-test.txt`, which you will use for training and testing respectively. Each file consists of a label (+/-) followed by a name. All the names in the files have been lower cased and new labels were generated using a new hidden function that is a composition of mathematical functions and **categorical** features extracted from the names.

Your goal is to use various learning algorithms on the training data to train a predictor and see how well it does on the test data.

You may use any programming language for your implementation. However, the graders should be able to execute your code on the CADE machines.

1. An important step in creating machine learning applications is to device useful features. For example, a feature function that selects the second letter would have been a great feature to use on the old badges data. eg: If this function was called ϕ, then ϕ("George Washington") ='e'. (And is unfortunately not so great for this data!)

   Devise four feature functions and use them to extract features from the new badges data. Describe each feature function in no more than one line.

   At the end, you should have transformed each name in the training and test set into a set of four feature values.

2. Implement a decision tree data structure. (Remember that the decision tree need not be a binary tree!)

3. Implement the ID3 learning algorithm for your decision tree implementation. For debugging your implementation, you can use the previous toy examples from the homework like the balloons data from Table 1.

4. (**For 6350 students**) Implement the $MajorityError$ based learning algorithm that you saw in the first question.

5. Implement a K Nearest Neighbor classifier for $K \in \{1 \ldots 5\}$. Note that your features are only categorical. So you have to make choices about how to measure distances between them. For example, you could consider using the Hamming distance between the feature representations.

6. For each algorithm you implemented show the accuracy on the testing data. Rank your algorithms in decreasing order of performance.

What to hand in for this problem:

1. The report should detail your experiments. For each step, explain in no more than a paragraph or so how your implementation works. You may provide the results for the final step as a table or a graph.

2. *Your code should run on the CADE machines.* You should include a shell script, `run.sh`, that will execute your code in the CADE environment. Your code should produce similar output to what you include in your report.

   You are responsible for ensuring that the grader can execute the code using only the included script. If you are using an esoteric programming language, you should make sure that its runtime is available on CADE.

3. Please do not hand in binary files!

Grading:

- Features: 8 points
- Decision tree implementation: 2 points
- ID3: 10 points
- (For 6350 students) $MajorityError$: 10 points
- K-nearest neighbors: 15 points
- Evaluation and report: 20 points

Not graded: Take a guess at how the labels were generated.