CPTS 570 Machine Learning, Fall 2016
Homework #3

Due Date: Oct 18

1. (**15 points**) Suppose $x = (x_1, x_2, \cdots, x_d)$ and $z = (z_1, z_2, \cdots, z_d)$ be any two points in a high-dimensional space (i.e., d is very large).

   a. (**10 points**) Try to prove the following, where the right-hand side quantity represent the standard Euclidean distance.

   $$\left(\frac{1}{\sqrt{d}}\sum_{i=1}^{d} x_i - \frac{1}{\sqrt{d}}\sum_{i=1}^{d} z_i\right)^2 \leq \sum_{i=1}^{d} (x_i - z_i)^2 \tag{1}$$

   Hint: Use Jensen's inequality – If X is a random variable and f is a convex function, then $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$.

   b. (**5 points**) We know that the computation of nearest neighbors is very expensive in the high-dimensional space. Discuss how we can make use of the above property to make the nearest neighbors computation efficient?

2. (**10 points**) We briefly discussed in the class about Locality Sensitive Hashing (LSH) algorithm to make the nearest neighbor classifier efficient. Please read the following paper and briefly summarize the key ideas as you understood:

   Alexandr Andoni, Piotr Indyk: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Communications of ACM 51(1): 117-122 (2008) `http://people.csail.mit.edu/indyk/p117-andoni.pdf`

3. (**15 points**) We know that we can convert any decision tree into a set of if-then rules, where there is one rule per leaf node. Suppose you are given a set of rules $R = \{r_1, r_2, \cdots, r_k\}$, where r_i corresponds to the i^{th} rule. Is it possible to convert the rule set R into an equivalent decision tree? Explain your construction or give a counterexample.

4. (**10 points**) You are provided with a training set of examples (see Figure 1). Which feature will you pick first to split the data as per the ID3 decision tree learning algorithm? Show all your work: compute the information gain for all the four attributes and pick the best one.

5. (**50 points**) Programming and empirical analysis question.

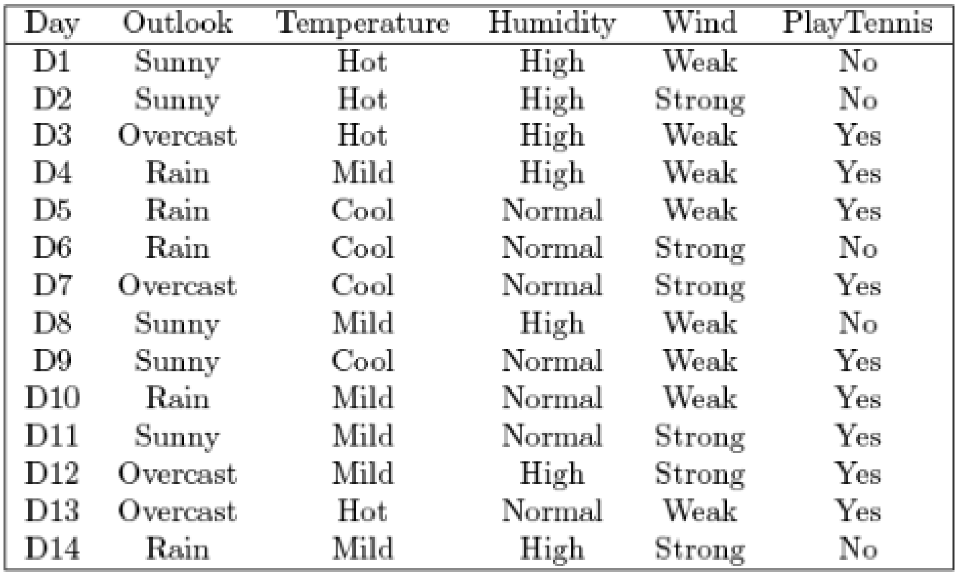
   You are given two datasets. Each dataset has three parts: training set, validation set, and testing set. Datasets are in CSV (Comma Separated Values). The first line in the file gives attribute (or feature) names. Each line after that is an example: input features followed by a class label (last value). You can assume that all features take only binary (0 or 1) values.

   a. Implement the ID3 decision tree learning algorithm that we discussed in the class. The key step in the decision tree learning is choosing the next feature to split on. Implement the information gain heuristic for selecting the next feature. Please see lecture notes or `https://en.wikipedia.org/wiki/ID3_algorithm` for more details.

   b. Run the decision tree construction algorithm on the training examples. Compute the accuracy on validation examples and testing examples. Compute the confusion matrix on the testing data. Perform this experiment on the given two datasets.

   c. Implement the decision tree pruning algorithm discussed in the class (via validation data).

   d. Run the pruning algorithm on the decision tree constructed using training examples. Compute the accuracy on validation examples and testing examples. Compute the confusion



Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Figure 1: Table with training examples. Each row corresponds to a single training example. There are four features, namely, outlook, temperature, humidity, and wind. "PlayTennis" is the class label.

matrix on the testing data. Perform this experiment on the given two datasets. List your observations by comparing the performance of decision tree with and without pruning.

To debug and test your implementation, you can employ Weka (weka.classifiers.trees.J48): `http://www.cs.waikato.ac.nz/ml/weka/downloading.html`