CSE 417T: Homework 7

Due: December 1, 2016 10am

Notes:

- There are 6 problems on 2 pages in this homework.
- You may **not** use late days on this homework. Solutions will be distributed in class on December 1.
- This homework has 50 points and a bonus problem worth 25 points.
- Submit all answers by committing a single pdf file named **`YOUR_WUSTL_KEY_hw7.pdf`** to the `hw7` folder in your SVN repository.

Problems:

1. (From Russell & Norvig) Suppose I pick some decision tree on functions going from 5 Boolean variables to a Boolean output. Now, I generate all possible inputs $\mathbf{x}_i$, and run them through the decision tree to produce the corresponding classes y_i. Finally, I take this generated dataset of all $(\mathbf{x}_i, y_i)$ pairs, and run a greedy decision tree learning algorithm using information gain as the splitting criterion. Am I guaranteed to get back the same tree that generated the data? If not, what kind of guarantee can I give about the tree that is returned?

2. Consider the following dataset: the class variable is whether or not a car gets good mileage, and the features are Cylinders (either 4 or 8), Displacement (High, Medium, or Low), and Horsepower (High, Medium, or Low).

Cylinders	Displacement	Horsepower	GoodMPG?
4	Low	Low	No
8	High	High	No
8	High	Medium	Yes
8	High	High	No
4	Low	Medium	Yes
4	Medium	Medium	No

   Give a decision tree that classifies this dataset perfectly. If you were to split this dataset using information gain, what would the first feature chosen to split on be?

3. Consider a training set of size n, where each example has two continuous features, no two examples have the exact same value for any of the two features, and the problem is a two-class problem. Suppose the training set is linearly separable. Can a decision tree correctly separate the data? What if the dataset is not linearly separable? In either case, if a decision tree can correctly separate the data, give the tightest bound that you can on the depth of that tree.

4. Suppose you apply bagging and boosting to a hypothesis space of linear separators. Will the hypothesis space of the ensemble still be linear for boosting? For bagging?

5. (From Russell & Norvig) Construct an SVM that computes the XOR function. Use values of +1 and -1 for the inputs and outputs. Map inputs (x_1, x_2) into a space consisting of x_1 and x_1x_2. Draw the four input points in this space and the maximal margin separator. What is the margin? Now draw the separating line back in the original input space.

6. The key point of the so-called "kernel trick" in SVMs is to learn a classifier that effectively separates the training data in a higher dimensional space without having to explicitly compute the representation $\Phi(\mathbf{x})$ of every point $\mathbf{x}$ in the original input space. Instead, all the work is done through the kernel function that computes dot products $K(\mathbf{x_i}, \mathbf{x_j}) = \Phi(\mathbf{x_i}) \cdot \Phi(\mathbf{x_j})$.

   Show how to compute the squared Euclidean distance in the projected space between any two points $\mathbf{x_i}, \mathbf{x_j}$ in the original space without explicitly computing the Φ mapping, instead using the kernel function K.

Bonus Problem (25 extra points):

Bagging reduces the variance of a classifier by averaging over several classifiers trained on subsets of the original training data. The subsets are obtained by uniform subsampling with replacement. I.e. if your data is $S = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\}$, at each iteration you create a new data set S' with n random picks, picking each example pair with probability $\frac{1}{n}$ each time. As a result you could end up with multiple identical pairs, or some not present at all.

Let $p_n(m, k)$ be the probability that you have drawn m unique examples after k picks with $|S| = n$. So clearly $p_n(m, k) = 0$ whenever $m > k$ (because you cannot end up with more unique elements m than you have drawn), and also $p_n(m, k) = 0$ whenever $m > n$.

(a) What are the base-case values of $p_n(1, 1), p_n(m, 1), p_n(1, k)$?

(b) Assume you are have already picked $k-1$ elements. What is the probability that the k^{th} pick will **not** increase your number of unique elements m? What is the probability that it will?

(c) Can you express $p_n(m, k)$ in terms of $p_n(m, k-1)$ and $p_n(m-1, k-1)$?

(d) Write out the formula for $E_{k=n}[\frac{m}{n}]$, the expected ratio of unique elements (m) and the total number of elements (n) after n picks (i.e. $k = n$) from set S with $|S| = n$.

(e) Write a little recursive function (in the programming language of your choice) that evaluates $E_{k=n}[\frac{m}{n}]$. Plot its value as n increases. What value does it converge to?

(f) If you average over M classifiers, trained on sub-sets $S'_1, \ldots, S'_M$ where $|S'_i| = n$, what is the probability that **at least** one input pair is never picked in any of the training data sets S'_i? Plot this function as M increases. (Assuming that n is large enough for the convergence as observed in (c).)