10-701 Machine Learning, Fall 2012: Homework 1

Due 9/26 at the beginning of class.

1 Decision Trees, [25pt, Martin]

1. **[10 points]** As of September 2012, 800 extrasolar planets have been identified in our galaxy. Super-secret surveying spaceships sent to all these planets have established whether they are habitable for humans or not, but sending a spaceship to each planet is expensive. In this problem, you will come up with decision trees to predict if a planet is habitable based only on features observable using telescopes.

   (a) In Table 1 you are given the data from all 800 planets surveyed so far. The features observed by telescope are *Size* ("Big" or "Small"), and *Orbit* ("Near" or "Far"). Each row indicates the values of the features and habitability, and how many times that set of values was observed. So, for example, there were 20 "Big" planets "Near" their star that were habitable. Derive and draw the decision tree learned by ID3 on this data (use the maximum information gain criterion for splits, don't do any pruning). Make sure to clearly mark at each node what attribute you are splitting on, and which value corresponds to which branch. By each leaf node of the tree, write in the number of habitable and inhabitable planets in the training data (i.e. the data in Table 1) that belong to that node.

   (b) For just 9 of the planets, a third feature, *Temperature* (in Kelvin), has been measured, as shown in Table 2. Redo all the steps from part (a) on this data using all three features. For the *Temperature* feature, in each iteration you must maximize over all possible binary thresholding splits (such as $T \leq 250$ v.s. $T > 250$, for example). According to your decision tree, would a planet with the features (Big, Near, 280) be predicted to be habitable or not habitable?

Table 1: Planet size and orbit vs. habitability.

Size	Orbit	Habitable	Count
Big	Near	Yes	20
Big	Far	Yes	170
Small	Near	Yes	139
Small	Far	Yes	45
Big	Near	No	130
Big	Far	No	30
Small	Near	No	11
Small	Far	No	255

Table 2: Planet size, orbit, and temperature vs. habitability.

Size	Orbit	Temperature	Habitable
Big	Far	205	No
Big	Near	205	No
Big	Near	260	Yes
Big	Near	380	Yes
Small	Far	205	No
Small	Far	260	Yes
Small	Near	260	Yes
Small	Near	380	No
Small	Near	380	No

2. **[15 points]** In this problem you'll see why simple feature-wise (i.e. coordinate-wise) splitting of the data isn't always the best approach to classification. Throughout the problem, assume that each feature can be used for splitting the data multiple times in a decision tree. Suppose you are given n non-overlapping points in the unit square $[0,1] \times [0,1]$, each labeled either $+$ or $-$.

   (a) Prove that there exists a decision tree of depth at most $\log_2 n$ that correctly labels all n points. At each node the decision tree should only perform a binary threshold split on a single coordinate. (Note that a binary tree of depth $\log_2 n$ can have as many as $2^{\log_2 n} = n$ internal nodes, i.e. splits.)

   (b) Describe (either mathematically, or in a few concise sentences) a set of n points in $[0,1] \times [0,1]$, along with corresponding $+$ or $-$ labels, so that the smallest decision tree that correctly labels them all has at least $n-1$ splits. (Hint: if you can do it with $n = 3$, you can do it with arbitrary n.)

   (c) Describe n points and corresponding labels that, as in part (b), can only be correctly labeled by a tree with at least $n-1$ splits, with the additional condition that the points labeled $+$ and the points labeled $-$ must be separable by a straight line. In other words, there must exist a line segment splitting the unit square in two (not necessarily parallel to either axis), so that all points labeled $+$ are in one part, and all points labeled $-$ are in the other. (You will soon see classifiers that would have had a much easier time with this type of data.)

2 Maximum Likelihood Estimation, [25pt, Avi]

Figure 2 shows a system S which takes two inputs x_1, x_2 (which are deterministic) and outputs a linear combination of those two inputs, $c_1x_1 + c_2x_2$, introduces an additive error ϵ which is a random variable following some distribution. Thus the output y that you observe is given by equation 1. Assume that you have $n > 2$ instances $< x_{j1}, x_{j2}, y_j >_{j=1,\ldots,n}$ or equivalently $< x_j, y_j >$, where $x_j = [x_{j1}, x_{j2}]$.

$$y = c_1x_1 + c_2x_2 + \epsilon \tag{1}$$

In other words having n equations in your hand is equivalent to having n equations of the following form: $y_j = c_1x_{j1} + c_2x_{j2} + \epsilon_j,\ j = 1 \ldots n$. The goal is to estimate c_1, c_2 from those

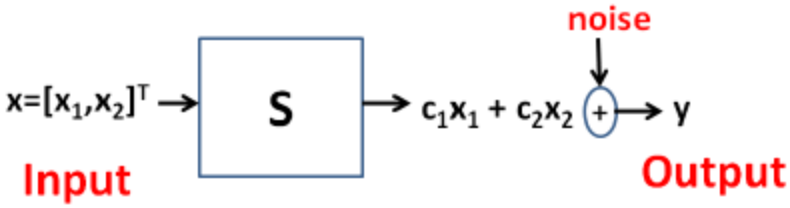

Figure 1: Exercise 2

measurements by maximizing conditional log-likelihood given the input, under different assumptions for the noise. Specifically:

1. **[10 points]** Assume that the ϵ_i for $i = 1 \dots n$ are iid Gaussian random variables with zero mean and variance σ^2.
   (a) Find the conditional distribution of each y_i given the inputs
   (b) Compute the loglikelihood of y given the inputs.
   (c) Maximixe the likelihood above to get c_{ls}
2. **[10 points]** Assume that the ϵ_i for $i = 1 \dots n$ are independent Gaussian random variable with zero mean and variance $Var(\epsilon_i) = \sigma_i$.
   (a) Find the conditional distribution of each y_i given the inputs
   (b) Compute the loglikelihood of y given the inputs.
   (c) Maximixe the likelihood above to get c_{wls}
3. **[5 points]** Assume that ϵ_i for $i = 1 \dots n$ has density $f_{\epsilon_i}(x) = f(x) = \frac{1}{2b} \exp(-\frac{|x|}{b})$. In other words our noise is iid following Laplace distribution with location parameter $\mu = 0$ and scale parameter b.
   (a) Find the conditional distribution of each y_i given the inputs
   (b) Compute the loglikelihood of y given the inputs.
   (c) Comment on why this model leads to more robust solution.

3 Naive Bayes vs Logistic Regression, [25pt, Derry]

In this problem you will implement Naive Bayes and Logistic Regression, and compare their performance on a classification task. The data for this task is given (http://www.cs.cmu.edu/ epxing/Class/10701/hw1-data.txt). The data is comma-separated (no header), with the first column being the class name. There are 2 classes: A and B, and 16 features. Each feature can take a value: 1, 2, or 3.

1. **[3 points]** Provide descriptions of Naive Bayes and Logistic Regression algorithms for the dataset above, deriving

(a) $P(Y = A|X_1...X_{16})$ and $P(Y = B|X_1...X_{16})$

(b) how to classify a new example (i.e. the classification rule)

(c) how to estimate the model parameters

Note: you only need to derive the equation, no need to plug in the actual values.

2. **[5 points]** In class, we showed that logistic regression is the discriminative counterpart to a Gaussian Naive Bayes classifier for continuous data. Consider the case where each X_i is boolean. Prove also for this case that $P(Y|X)$ in logistic regression follows the same form (and hence that Logistic Regression is also the discriminative counterpart to a Naive Bayes with Boolean features). Hint: represent $P(X_i|Y = A) = \theta_{iA}^{X_i}(1 - \theta_{iA})^{1-X_i}$, where $\theta_{iA} = P(X_i = 1|Y = A)$ and hence $1 - \theta_{iA} = P(X_i = 0|Y = A)$.

3. **[4 points]** Since Logistic Regression is the discriminative counterpart to a Gaussian Naive Bayes (we showed in class that the parameters w_i in Logistic Regression can be expressed in terms of the Gaussian Naive Bayes parameters), then

   (a) asymptotically (as the number of training examples grows toward infinity), do you think Logistic Regression and the Gaussian Naive Bayes will converge toward identical classifiers? Comment on why.

   (b) Naive Bayes has the assumption of conditional independence and may not work well when the data violates this assumption. Do you think Logistic Regression also faces this problem? If not, why?

4. **[10 points]** Implement Logistic Regression and Naive Bayes for the dataset above. Use add-one smoothing when estimating the parameters of your Naive Bayes classifier. For logistic regression, use a step size around .0001. To train and test, follow these steps:

   (a) Randomly split dataset into 2/3 training set, 1/3 testing set.

   (b) Choose a random subset of the training data to train, with training sizes m from 2 to 200 (with an increment of 1 or close to 1).

   (c) After training each subset, test against the held-out testing set. Calculate the classification error as the ratio of incorrectly classified to the total testing set size.

   (d) Repeat 100 times from beginning, averaging the classification error over the 100 runs.

   (e) Plot the average error vs. the training sizes m, comparing Logistic Regression and Naive Bayes.

   Submit your code online. Submit your printed plot along with your homework.

5. **[3 points]** Which model performs better:

   (a) at the beginning when there is little training data?

   (b) as there are more data?

   (c) which model would you prefer when there is little training data and which do you prefer when there is more training data and why? (Hint: Naive Bayes and Logistic Regression converge toward their asymptotic accuracies at different rates. Naive Bayes converge toward their asymptotic values in order $n = log\ d$ examples, where d is the dimension of X. Logistic regression converges more slowly, in order $n = d$ examples)