CS B551 –Elements of Artificial Intelligence– Fall 2011

Homework #5
(Machine Learning)
Due: 10/27/11 (5:15 pm)

How to complete this HW: Either 1) type your answers in the empty spaces below each problem and print out this document, or 2) print this document and write your answers in the empty spaces on the printout. Return the homework in class, during office hours, or slip it under the door of Info E 257 before **5:15 on Thursday, 10/27/11**.

Your name: Enrique Areyan

Your email address:

Note on Honor Code: You must NOT look at previously published solutions of any of these problems in preparing your answers. You may discuss these problems with other students in the class (in fact, you are encouraged to do so) and/or look into other documents (books, web sites), with the exception of published solutions, without taking any written or electronic notes. If you have discussed any of the problems with other students, indicate their name(s) here:
N/A
Any intentional transgression of these rules will be considered an honor code violation.

General information: Justify your answers, but keep explanations short and to the point. Excessive verbosity will be penalized. If you have any doubt on how to interpret a question, tell us in advance, so that we can help you understand the question, or tell us how you understand it in your returned solution.

Grading:

Problem#	Max. grade	Your grade
I	25	
II	25	
III	20	
IV	30	
Total	100	

I. Time series data (25 points)

You are an IU basketball fan, and you want to learn a predictive model of the results of each IU possession. The n'th possession is denoted as a random variable X_n, and X_n can be equal to either a basket (B), turnover (T) or foul (F). As you are watching the game you make the following observations on 25 plays (grouped into groups of 5 just for readability):

B, B, B, F, T,
B, B, B, B, T,
T, F, B, T, T,
T, B, F, B, B,
B, B, B, F, F

1. Model this time series as a set of independent and identically distributed events (i.e., a 0-th order Markov Chain). That is $P(X_n=x) = P(X_m=x)$ for all m and n. What are the maximum likelihood parameters of the probability distribution over X_t?

$$\theta_{m1} = \begin{cases} \frac{14}{25} \; if \; X = B \\ \frac{5}{25} \; if \; X = F \\ \frac{6}{25} \; if \; X = T \end{cases}$$

   Note: θ_{m1} stands for the parameters of model 1.

2. Now suppose the 1st order Markov model $P(X_n=x|X_{n-1}=y)$ which depends on the outcome y at the prior time step. For the first time step, assume $P(X_1=x)$ is identical to the distribution that you estimated in question 1. What are the maximum likelihood parameters of the transition distribution?
   This are better formatted as a transition matrix:
   $\theta_{m2}(i|j) =$

$X_t \setminus X_{t-1}$	B	F	T
B	9/24	2/24	2/24
F	3/24	1/24	1/24
T	2/24	1/24	3/24

3. For each of the probabilistic models from Q1 and Q2, write an expression involving numerical values that gives the likelihood of the data. Using a computer or calculate, evaluate the log of this expression. How do the likelihoods compare, and why?

$For\ Q1: P(d|\theta) = \Pi_j P(d_j|\theta) = (\frac{14}{25})^{14}.(\frac{5}{25})^{5}.(\frac{6}{25})^{6}$

$$\log\big(P(d|\theta)\big) = \log\left(\left(\frac{14}{25}\right)^{14}.\left(\frac{5}{25}\right)^{5}.\left(\frac{6}{25}\right)^{6}\right) = 14\log(14) + 5\log(5) + 6\log(6) - 25\log(25)$$
$$= 14 * 2.63 + 5 * 1.60 + 6 * 1.79 - 25 * 3.21 = -24.72$$

For Q2:

$$P(d|\theta) = [\Pi_j P\big(d_j\big|\theta_{m2}, d_{j-1}\big)]\; P(d1|\theta_{m2}) =$$
$$= \left[\left(\frac{9}{24}\right)^{9}.\left(\frac{2}{24}\right)^{2}.\left(\frac{2}{24}\right)^{2}.\left(\frac{3}{24}\right)^{3}.\left(\frac{1}{24}\right).\left(\frac{1}{24}\right).\left(\frac{2}{24}\right)^{2}.\left(\frac{1}{24}\right).\left(\frac{3}{24}\right)^{3}\right]\frac{9}{24}$$
$$\log\big(P(d|\theta)\big) = \log\left(\left(\frac{9}{24}\right)^{10}.\left(\frac{2}{24}\right)^{2}.\left(\frac{2}{24}\right)^{2}.\left(\frac{3}{24}\right)^{3}.\left(\frac{1}{24}\right).\left(\frac{1}{24}\right).\left(\frac{2}{24}\right)^{2}.\left(\frac{1}{24}\right).\left(\frac{3}{24}\right)^{3}\right)$$
$$= 10\log(9) + 6\log(2) + 6\log(3) + 3\log(1) - 25\log(24) = -46.72$$

4. Now use your probabilistic models from Q1 and Q2 to predict the value of X_n given the *true* value of X_{n-1} observed in the data (for X_1 just use the distribution $P(X_1)$). Repeat this for all n. What accuracy do you obtain using the classifier from Q1? From Q2?

For Q1: all predictions are B because is the value with the highest probability. Therefore, the accuracy obtained is 14/25 (the number of B over all possibilities).

For Q2: At this point I'm still not sure how to do this. Could you please explain me what are we suppose to do here? Thanks :)

5. Discuss the pros and cons of using maximum a posteriori (MAP) to estimate the model parameters in Q2. What would you gain? What would you sacrifice? What decisions might be challenging?

To use MAP we would have to get an estimate a better initial estimate than ML. We would have to estimate a prior distribution over the nine parameters in $\theta_{m2}(i|j)$, which would be challenging. We would also have to optimize the posterior distribution, for wich we would need to use some sort of local search techniques.

II. Statistical Learning (25 points)

Consider building a probabilistic model of a dataset $D=(x_1,\ldots,x_n)$ in which each x_i is a continuous nonnegative value. Consider the following hypothesis class:

$$P(x|\theta) = \theta\, e^{-\theta x}$$

which is parameterized by the single hypothesis parameter θ. Note that this is a proper probability distribution over the set of nonnegative x because it integrates to 1. This problem will have you derive the maximum likelihood estimate θ_{ML}.

1. Give the mathematical expression for the likelihood of the data

$L(D;\theta) = P(D|\theta) = \Pi_j P(d_j|\theta) = \Pi_j \theta e^{-\theta X_j} = \left(\theta e^{-\theta X_1}\right)\left(\theta e^{-\theta X_2}\right) \dots \left(\theta e^{-\theta X_n}\right) = \theta^n\left(e^{-\theta X_1} e^{-\theta X_1} \dots e^{-\theta X_n}\right) = \theta^n\left(e^{-\theta X_1 - \theta X_1 - \cdots - \theta X_n}\right) = \theta^n e^{-\theta(X_1 + X_1 + \cdots + X_n)} = \theta^n e^{-\theta \sum_j X_j}$

2. Give the mathematical expression for the log-likelihood of the data

$l(D;\theta) = \log P(D|\theta) = \log\left(\theta^n e^{-\theta \sum_j X_j}\right) = \log(\theta^n) + \log\left(e^{-\theta \sum_j X_j}\right) = nlog(\theta) - \theta(\sum_j X_j)\log(e) = nlog(\theta) - \theta \Sigma_j X_j$

3. Give the mathematical expression for the derivative of the log-likelihood of the data

$dl/d\theta(D;\theta) = \frac{d(nlog(\theta) - \vartheta \Sigma_j X_j)}{d\theta} = \frac{d(nlog(\theta))}{d\theta} + \frac{d(-\theta \Sigma_j X_j)}{d\theta} = \frac{n}{\theta} - \Sigma_j X_j$

4. Solve for θ_{ML}, which is a value of θ that satisfies $dl/d\theta\,(D;\theta) = 0$. How is the ML parameter value related to the average value of the dataset?

$\frac{dl}{d\theta} = \frac{n}{\theta} - \Sigma_j X_j = 0 \iff \frac{n}{\theta} = \Sigma_j X_j \Rightarrow \theta = 1/\frac{1}{n}\Sigma_j X_j$

Therefore, the ML parameter value is the result of 1 divided by the average value of the data set. (The "inverse" of the average)

III. Decision Tree Learning (20 points)

1. Suppose we generate a training set from a decision tree and then apply decision-tree learning to that training set. Is it the case that the learning algorithm will eventually learn a tree that is consistent with the training set as the training-set size goes to infinity? Is it the case that the learning algorithm will eventually return the correct tree as the training-set size goes to infinity? Why or why not?

Answer: The learning algorithm will eventually learn a tree that represent a function that is equivalent to the function embodied in the original tree, but in general it might not be exactly the same tree. This is because the space of possible functions is huge and thus intractable, from which we can reason that the probability of finding the exact same tree goes to zero pretty quickly. However, because several logical functions may be implemented differently, we have a chance of finding a logically equivalent function. For this process to work, it is important to have all (or at least) most combination of attributes in the training set.

2. Consider the following data set comprised of three binary input attributes A1, A2, and A3 and one binary output:

Example	A_1	A_2	A_3	Output y

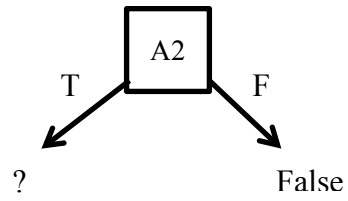
x_1	1	0	0	0
x_2	1	0	1	0
x_3	0	1	0	0
x_4	1	1	1	1
x_5	1	1	0	1

Use the DTL algorithm in the slides of lecture 14 to learn a decision tree for these data. Show the computations made to determine the attribute to split at each node.

Start:

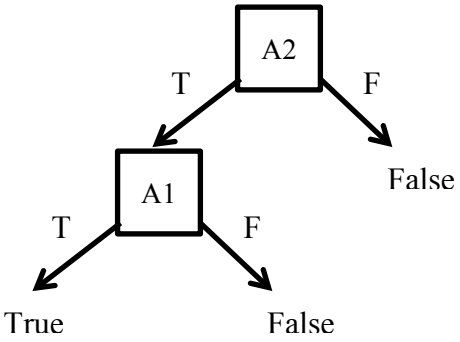
Predicate	# of misclassified examples
A1	2 (x1,x2)
A2	1 (x3)
A3	2 (x2,x5)

Best Predicate = A2



Predicate	# of misclassified examples
A2=T ∧ A1	0
A2=T ∧ A3	1 (x5)

Best Predicate = A1



CONCEPT ⇔ A2 ∧ A1

This is the final tree. It classifies all examples correctly. Note that we didn't need to include A3 to correctly classify all examples.

IV. Function Learning (30 points)

1. You have a dataset D of examples $(x_i,f(x_i))$ for i=1,…,n. Consider fitting the parameter θ of the constant model g(x,θ) = θ to this dataset (i.e., the model ignores the x coordinate).

   Write down the equation that expresses the sum of squared errors E as a function of θ.

   $E(\theta) = \Sigma_i (f(x_i) - g(x_i,\theta))^2 = \Sigma_i (f(x_i) - \theta)^2$

2. Find the value of θ that minimizes E(θ) for D={(2,1), (4,7), (5,6), (6,8), (7,8)}. You can do this either by hand, or using the fact that E'(θ)=0 at the minimum. What is another name for this value?

   $E(\theta) = \Sigma_i (f(x_i) - \theta)^2 = (1-\theta)^2 + (7-\theta)^2 + (6-\theta)^2 + 2(8-\theta)^2$
   $= 1 - 2\theta + \theta^2 + 49 - 14\theta + \theta^2 + 36 - 12\theta + \theta^2 + 128 - 32\theta + 2\theta^2)$
   $= 5\theta^2 - 60\theta + 214, \quad E'(\theta) = 10\theta - 60$. The minimum satisfy $E'(\theta) = 0$
   Thus, $10\theta - 60 = 0 \Rightarrow \theta = 6$. Another name for this value is the average.

3. Now consider the nonlinear model g(x,θ) = log(x*θ). Write a computer program (in your language of choice) that uses gradient descent to find a θ that minimizes E(θ) for the dataset given above. A template in Python code is given at http://www.cs.indiana.edu/classes/b551/gradient_template.py

   If you are using your own code, you may want to use the following expression for the derivative of E:

   $E'(\theta) = \Sigma_i (2 / \theta) (\log(x_i*\theta) - f(x_i))$

   In your implementation, use the starting value θ=50, step size 1.0, and run gradient descent for 1000 iterations. Report the final value of θ and E(θ), and the y values for each of the datapoints in D.

Optimized θ: 83.8982030544
E(θ): 24.2032002811
Predictions:
[(2, 5.122751376097381), (4, 5.815898556657326), (5, 6.039042107971536),
(6, 6.221363664765491), (7, 6.375514344592749)]

Code use for this part (only the relevant part)

```
def gradientDescent(theta0,alpha,iters):
        x_old = 0
        x_new = theta0
        for i in range(iters):
                x_old = x_new
                x_new = x_old - alpha * dE(x_new)
        return x_new
```