COMS 4771 HW4

Due: Sun Dec 03, 2017

You are allowed to work in groups of (at max) three students. These group members don't necessarily have to be the same from previous homeworks. Only one submission per group is required by the due date. Name and UNI of all group members must be clearly specified on the homework. You must cite all the references you used to do this homework. You must show your work to receive full credit.

1 **[Estimating parameters with complete and incomplete data]** Consider the data generation process for observation pair (a, b) as follows:

- a is the outcome of an independent six-faced (possibly loaded) dice-roll. That is, chance of rolling face '1' is p_1, rolling face '2' is p_2, etc., with a total of six distinct possibilities.
- Given the outcome a, b is drawn independently from a density distributed as $q_a e^{-q_a b}$ (where $q_a > 0$).

(i) List all the parameters of this process. We shall denote the collection of all the parameters as the variable θ (the parameter vector).

(ii) Suppose we run this process n times independently, and get the sequence:

$$(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n).$$

What is the likelihood that this sequence was generated by a specific setting of the parameter vector θ?

(iii) What is the most likely setting of the parameter vector θ given the complete observation sequence $(a_i, b_i)_{i=1}^n$? that is, find the Maximum Likelihood Estimate of θ given the observations.

(iv) What is the probability of the partial (incomplete) observation $b_1, b_2, \ldots, b_n$ given a specific setting of the parameter vector θ?

(v) Derive the Expectation Maximization (EM) algirthm to estimate of the parameters given the incomplete observation sequence $(b_i)_{i=1}^n$.

2 **[Cost-sensitive classification]** Suppose you have a binary classification problem with input space $\mathcal{X} = \mathbb{R}$ and output space $\mathcal{Y} = \{0, 1\}$, where it is c times as bad to commit a "false positive" as it is to commit a "false negative" (for some real number $c \geq 1$). To make this concrete, let's say that if your classifier predicts 1 but the correct label is 0, you incur a penalty of $\$c$; if your classifier predicts 0 but the correct label is 1, you incur a penalty of $\$1$. (And you incur no penalty if your classifier predicts the correct label.)

Assume the distribution you care about has a class prior with $\pi_0 = 2/3$ and $\pi_1 = 1/3$, and the class conditional are Gaussians with densities $N(0, 1)$ for class 0, and $N(2, 1/4)$ for class 1. Let $f^* : \mathbb{R} \to \{0, 1\}$ be the classifier with the smallest expected penalty.

(i) Assume $1 \leq c \leq 14$. Specify precisely the subset of $\mathbb{R}$ in which the classifier f^* predicts 1. (E.g., $[0, 5c] \cup [6c, +\infty)$.)

(ii) Now instead assume $c \geq 15$. Again, specify precisely the region in which the classifier f^* predicts 1.

3 **[PAC learning and VC dimension]**

(i) Compute the tightest possible VC dimension estimate of the following model classes:

(a) $(\mathcal{F}_1 \cup \mathcal{F}_2)$, where each $\mathcal{F}_i \subseteq \mathcal{F}$. Here $\mathcal{F} := \{f \mid f : X \to \{0, 1\}\}$, that is, the collection of all functions from a domain X to $\{0, 1\}$.

(b) $\mathcal{F} :=$ Convex polygons in $\mathbb{R}^2$.

(ii) Given a collection of models $\mathcal{F}$, suppose you were able to develop an algorithm $\mathcal{A} : (x_i, y_i)_{i=1}^n \mapsto f_n^{\mathcal{A}}$ (that is, given n labeled training samples, $\mathcal{A}$ returns a model $f_n^{\mathcal{A}} \in \mathcal{F}$) that has the following property: for all $\epsilon > 0$, with probability 0.55 (over the draw of $n = O(\frac{1}{\epsilon^2})$ samples,

$$\text{err}(f_n^{\mathcal{A}}) - \inf_{f \in \mathcal{F}} \text{err}(f) \leq \epsilon,$$

where $\text{err}(f) := P_{(x,y)}[f(x) \neq y]$.

Show that one can construct an algorithm $\mathcal{B} : (x_i, y_i)_{i=1}^{n'} \mapsto f_{n'}^{\mathcal{B}}$ with the property: for all $\epsilon > 0$ and all $\delta > 0$, with probability at least $1 - \delta$ over a draw of n' samples:

$$\text{err}(f_{n'}^{\mathcal{B}}) - \inf_{f \in \mathcal{F}} \text{err}(f) \leq \epsilon.$$

Moreover show that $n' = \text{poly}(\frac{1}{\epsilon}, \frac{1}{\delta})$ samples are enough for the algorithm $\mathcal{B}$ to return such a model $f_{n'}^{\mathcal{B}}$. Hence, the model class $\mathcal{F}$ is *efficiently* PAC-learnable.
(Hint: Algorithm $\mathcal{B}$ can make multiple calls to the algorithm $\mathcal{A}$.)

4 **[From distances to embeddings]** Your friend from overseas is visiting you and asks you the geographical locations of popular US cities on a map. Not having access to a US map, you realize that you cannot provide your friend accurate information. You recall that you have access to the relative distances between nine popular US cities, given by the following distance matrix D:

Distances (D)	BOS	NYC	DC	MIA	CHI	SEA	SF	LA	DEN
BOS	0	206	429	1504	963	2976	3095	2979	1949
NYC	206	0	233	1308	802	2815	2934	2786	1771
DC	429	233	0	1075	671	2684	2799	2631	1616
MIA	1504	1308	1075	0	1329	3273	3053	2687	2037
CHI	963	802	671	1329	0	2013	2142	2054	996
SEA	2976	2815	2684	3273	2013	0	808	1131	1307
SF	3095	2934	2799	3053	2142	808	0	379	1235
LA	2979	2786	2631	2687	2054	1131	379	0	1059
DEN	1949	1771	1616	2037	996	1307	1235	1059	0

Being a machine learning student, you believe that it may be possible to infer the locations of these cities from the distance data. To find an embedding of these nine cities on a two dimensional map, you decide to solve it as an optimization problem as follows.

You associate a two-dimensional variable x_i as the unknown latitude and the longitude value for each of the nine cities (that is, x_1 is the lat/lon value for BOS, x_2 is the lat/lon value for NYC, etc.). You write down the an (unconstrained) optimization problem

$$\text{minimize}_{x_1,\ldots,x_9} \quad \sum_{i,j} \big(\|x_i - x_j\| - D_{ij}\big)^2,$$

where $\sum_{i,j}(\|x_i - x_j\| - D_{ij})^2$ denotes the embedding discrepancy function.

(i) What is the derivative of the discrepancy function with respect to a location x_i?

(ii) Write a program in your preferred language to find an optimal setting of locations $x_1, \ldots, x_9$. You must submit your code to Courseworks to receive full credit.

(iii) Plot the result of the optimization showing the estimated locations of the nine cities. (here is a sample code to plot the city locations in Matlab)

```
>> cities={'BOS','NYC','DC','MIA','CHI','SEA','SF','LA','DEN'};
>> locs = [x1;x2;x3;x4;x5;x6;x7;x8;x9];
>> figure; text(locs(:,1), locs(:,2), cities);
```

What can you say about your result of the estimated locations compared to the actual geographical locations of these cities?

5 **[Regression on large-scale dataset]** We shall use Kaggle platform to evaluate a regressor you design for a large scale regression dataset.

(i) If you haven't already, signup on http://www.kaggle.com with your columbia email address.

(ii) Visit the COMS 4771 regression task at: https://www.kaggle.com/t/d8e1a8bc2ec3424bb9240c5a1046e35f and develop a regressor for the large-scale dataset available there. You can use any resource publicly available to develop your regressor. (You don't need to submit your code for this question.)

(iii) Your pdf writeup should describe your design for your regressor: What preprocessing techniques and regressor you used? Why you made these choices? What resources you used and were helpful? What worked and what didn't work?