CSCI 567 Spring 2024 Practice Problems for Part II of class

Instructor: Vatsal Sharan

1 Multiple choice questions

(1) Which of the following about neural networks is correct?

(A) A neural network with a fixed architecture can represent any continuous function.
(B) Dropout is useful for preventing overfitting when training a nerual net.
(C) A neural network with one hidden layer and ReLU activations is a convex model, since ReLU is a convex function.
(D) A convolution layer is a special case of a fully connected layer.

(2) Suppose we are training a neural networkwork with mini-batch SGD with batch size 50 and 50000 training samples. How many gradient updates would there be in total over during 5 training epochs?

(A) 50000
(B) 1000
(C) 5000
(D) 250000

(3) Suppose a convolution layer takes a 4×6 image with 3 channels as input and outputs a $3 \times 4 \times 8$ volume. Which of the following is a possible configuration of this layer? (You can refer to slide 50 of Lecture 7.)

(A) One 2×3 filter with depth 8, stride 1, and no zero-padding.
(B) Eight 2×3 filters with depth 3, stride 1, and no zero-padding.
(C) Eight 2×2 filters with depth 3, stride 2, and 1 pixel of zero-padding.
(D) Eight 2×2 filters with depth 3, stride 2, and 2 pixels of zero-padding.

(4) How many parameters do we need to learn for the following network structure? An $8 \times 8 \times 3$ image input, followed by a convolution layer with 4 filters of size 3×3 (stride 1 and 1 pixel of zero-padding), then another convolution layer with 3 filters of size 2×2 (stride 2 and no zero-padding), and finally a max-pooling layer with a 2×2 filter (stride 2 and no zero-padding). (Note: the depth of all filters are not explicitly spelled out, and we assume no bias/intercept terms used here.)

(A) 48 (B) 144 (C) 156 (D) 168

(5) What is the final output dimension of the last question? (You can refer to slide 50 of Lecture 7.)

(A) $2 \times 2 \times 3$ (B) $4 \times 4 \times 3$ (C) $2 \times 2 \times 1$ (D) $4 \times 4 \times 1$

(6) Which of the following about decision trees is correct?

(A) Good interpretability is a key advantage of decision trees.
(B) Decision tree algorithms are usually implemented using recursion.
(C) Entropy is the only way to measure the uncertainty of a node when building a decision tree.

(D) Regularization is not applicable to decision trees since they do not minimize a certain loss function.

(7) Consider a binary dataset with 50 positive examples 50 negative examples. Decision stump $\mathcal{T}_1$ splits this dataset into two children where the left one has 20 positive examples and 40 negative examples, while another decision stump $\mathcal{T}_2$ results in a left child with 25 positive examples and 25 negative examples. Which of the following is correct?

(A) The entropy of the left child of $\mathcal{T}_1$ is $\frac{1}{3}\ln 3 + \frac{2}{3}\ln\frac{3}{2}$.
(B) The entropy of the right child of $\mathcal{T}_1$ is $\frac{1}{4}\ln 4 + \frac{3}{4}\ln\frac{4}{3}$.
(C) The entropy of either child of $\mathcal{T}_2$ is the maximum possible entropy for two classes.
(D) Based on conditional entropy, $\mathcal{T}_2$ is a better split than $\mathcal{T}_1$.

(8) Which of the following about Principal Component Analysis (PCA) is correct?

(A) PCA can be used to compress a dataset.
(B) PCA is useful for visualizing a dataset.
(C) PCA requires finding all the eigenvectors of the covariance matrix.
(D) To decide how many principal components to use, we can plot some of the eigenvalues and see if they seem to be becoming small.

(9) Which of the following about k-means is correct?

(A) k-means always finds the global minimum of the k-means objective, but it might take a very long time to converge.
(B) k-means always finds a local minimum of the k-means objective.
(C) There are initialization algorithms (such as k-means++) for the k-means objective, such that the algorithm is always guaranteed to find the global minimum of the objective in polynomial time.
(D) k-means may not find the best possible cluster assignment for the cluster centers it finds.

(10) Which of the following about Gaussian Mixture Model (GMM) are correct?

(A) GMM assumes that the data are generated probabilistically in a particular way, and thus can only be applied if we know the training data is indeed generated from this model.
(B) The global maximizer of the maximum likelihood objective of a GMM model can be found using the EM algorithm.
(C) Learning a GMM via EM gives not only the cluster (soft) assignments and centers, but also other parameters such as mixture weights and covariance matrices.
(D) GMM for clustering always performs better than k-means since it learns more parameters.

(11) When applying a GMM with k components to a dataset of n points, if we representing γ_{ij} (the soft assignment/posterior probability of datapoint i belonging to cluster j) as a matrix of n rows and k columns, what is the sum of all the entries of this matrix?

(A) 1
(B) k
(C) n
(D) nk

(12) In a multi-armed bandit problem with two arms and binary rewards, suppose that we have selected the first arm 6 times, 3 of which yield reward 1, and the second arms 3 times, 2 of which yield reward 1. Which of the following about the behavior of the UCB algorithm in the next round is correct (you can refer to the lecture notes for the UCB algorithm)?

(A) UCB selects the second arm because $\frac{1}{2} + 2\sqrt{\frac{\ln 10}{6}} < \frac{2}{3} + 2\sqrt{\frac{\ln 10}{3}}$.

(B) UCB selects the second arm because $\frac{1}{2} + 2\sqrt{\frac{\ln 10}{3}} < \frac{2}{3} + 2\sqrt{\frac{\ln 10}{2}}$.

(C) UCB selects the second arm because $\frac{1}{2} + 2\sqrt{\frac{\ln 5}{3}} < \frac{2}{3} + 2\sqrt{\frac{\ln 5}{2}}$.

(D) UCB selects the second arm with probability $\frac{2}{3}$.

2 EM

Consider the following probabilistic model to generate a non-negative integer x. First, draw a hidden binary variable z from a Bernoulli distribution with mean $\pi \in [0, 1]$, that is, $p(z = 1; \pi) = \pi$ and $p(z = 0; \pi) = 1 - \pi$. If $z = 0$, set $x = 0$; otherwise, draw x from a Poisson distribution with parameter λ so that

$$p(x|z = 1; \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}.$$

Given N samples $x_1, \ldots, x_n$ generated independently in this way, follow the steps below to derive the EM algorithm for this model. (This is also a good example to see why finding the exact MLE is difficult; you can try it yourself.)

2.1

Fixing the model parameters π and λ, for each sample n, find the posterior distribution of the corresponding hidden variable z_i: $p(z_i|x_i; \pi, \lambda)$. You will find it useful to consider the cases $x_i > 0$ and $x_i = 0$ separately. Given that z_i is binary, this means that you have to find the value of the following four quantities:

- $\gamma_0' = p(z_i = 0|x_i > 0; \pi, \lambda)$,
- $\gamma_1' = p(z_i = 1|x_i > 0; \pi, \lambda)$,
- $\gamma_0 = p(z_i = 0|x_i = 0; \pi, \lambda)$,
- $\gamma_1 = p(z_i = 1|x_i = 0; \pi, \lambda)$.

2.2

Suppose that we have computed $\gamma_0, \gamma_1, \gamma_0', \gamma_1'$ from the previous value of π and λ. Now, write down the expected complete likelihood function $Q(\pi, \lambda)$ in terms of the data $x_1, \ldots, x_i$, the posteriors $\gamma_0, \gamma_1, \gamma_0', \gamma_1'$, and the parameters π and λ (show intermediate steps). This completes the E-step.

2.3

Find the maximizer π^* and λ^* for the function $Q(\pi, \lambda)$ from the previous question (show your derivation). You might find it convenient to use the notation $N_0 = |\{n : x_i = 0\}|$ (that is, the number of examples with value 0) in your solution. This completes the M-step.

2.4

Combining all the results, write down the EM update formulas for π^{new} and λ^{new}, using only the data $x_1, \ldots, x_i$ and the previous parameter values π^{old} and λ^{old} (do not use $\gamma_0, \gamma_1, \gamma_0', \gamma_1'$).

3 Naive Bayes and Boosting

Consider the dataset of 5 examples shown in Table 1. Each example corresponds to a mushroom and has two features "Color" and "Aroma", and one label "Label" which denotes whether or not that mushroom was poisonous. They all have two possible values: red or gray for Color, chemical or earthy for Aroma, and poisonous or edible for Label.

i	Color	Aroma	Label
1	red	earthy	edible
2	gray	chemical	edible
3	red	earthy	poisonous
4	red	chemical	poisonous
5	gray	earthy	poisonous

Table 1: Training set

$p(\text{L} = \text{poisonous})$	
$p(\text{L} = \text{edible})$	

	C = red	C = gray	A = chemical	A = earthy
L = poisonous				
L = edible				

Table 2: Naive Bayes parameters

(a) Suppose that we run Naive Bayes on this dataset. Write down all the 10 parameters learned by this algorithm in Table 2. Specifically, for the table on the top, fill in the unconditional probability of the label, and for the table on the bottom, fill in the probability of each feature conditioning on the label: for example, the first entry represents $p(\text{C} = \text{red}|\text{L} = \text{poisonous})$ (C, A and L are shorthands for Color, Aroma, and Label respectively). No reasoning needed.

(b) For each of the 4 possible feature combinations $\boldsymbol{x}$, write down the joint probability $p(\boldsymbol{x}, L = \text{poisonous/edible})$ in Table 3 (still under the Naive Bayes assumption), as well as the prediction of the algorithm. No reasoning needed.

$\boldsymbol{x}$	$p(\boldsymbol{x}, L = \text{poisonous})$	$p(\boldsymbol{x}, L = \text{edible})$	Prediction (poisonous or edible)
(red, chemical)			
(red, earthy)			
(gray, chemical)			
(gray, earthy)			

Table 3: Naive Bayes predictions

(c) Suppose that we run AdaBoost with Naive Bayes as the base algorithm. What you have figured out in the last two questions shows you what this base algorithm will do in the first round of AdaBoost. Calculate the importance of this classifier β_1 and the distribution D_2 over the 5 training examples for the next round of AdaBoost (refer to the lecture notes for the expressions for β_1 and D_2). Show your derivation.

4 Boosting for Regression

We know that AdaBoost can be derived by greedily minimizing the exponential loss. In this problem, you will follow the same idea to derive a boosting algorithm for regression by greedily minimizing the square loss.

Specifically, our goal is to output a function $f(\boldsymbol{x}) = \sum_{\tau=1}^{T} \beta_\tau h_\tau(\boldsymbol{x})$ where $\beta_\tau \in \mathbb{R}$ and h_τ is from a fixed class $\mathcal{H}$ which consists of mappings from $\mathbb{R}^d$ to $[-1, 1]$. Denote $f_t = \sum_{\tau=1}^{t} \beta_\tau h_\tau$ (so that $f = f_T$), and suppose we have found f_{t-1}. Now, we want to find β_t and h_t to minimize the square loss on a set of examples $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n) \in \mathbb{R}^d \times [-1, 1]$:

$$L(\beta_t, h_t) = \sum_{i=1}^{n} (f_t(\boldsymbol{x}_i) - y_i)^2 = \sum_{i=1}^{n} (\beta_t h_t(\boldsymbol{x}_i) - r_i)^2,$$

where $r_i = y_i - f_{t-1}(\boldsymbol{x}_i)$.

(a) Given h_t provided by a base algorithm, find the value of β_t that minimizes $L(\beta_t, h_t)$.

(b) Suppose that $\mathcal{H}$ is symmetric in the sense that if $h \in \mathcal{H}$, then $-h \in \mathcal{H}$. Plug the optimal value of β_t from the last question into $L(\beta_t, h_t)$, and show that if the base algorithm wants to minimize $L(\beta_t, h_t)$, it needs to find

$$h_t^* = \operatorname*{argmin}_{h_t \in \mathcal{H}} \sum_{i=1}^{n} \left(\frac{h_t(\boldsymbol{x}_i)}{\|h_t\|} - r_i \right)^2,$$

where $\|h_t\| = \sqrt{\sum_{i=1}^{n} h_t(\boldsymbol{x}_i)^2}$.