NATIONAL UNIVERSITY OF SINGAPORE

CS5339: Theory and Algorithms for Machine Learning

(Semester 2: AY2019/20)

Time Allowed: 120 + 60 Minutes

INSTRUCTIONS TO CANDIDATES

a) This paper consists of **THREE (3)** questions and **NINE (9)** printed pages including this page. Answer all questions.

b) You have 120 + 60 minutes to earn 40 marks. Do not spend too much time on any problem. Read them all through first and attack them in the order that allows you to make the most progress.

c) You may quote results that are stated in the lecture notes or homework solutions without re-deriving them if you need the results as part of your answer.

d) Show your work, as *credit will be allocated for derivations and explanations*. You will be graded on the **correctness** and **optimality** of your answers, and also on your *clarity*. Be clear and always explain your reasoning.

e) This is an **OPEN BOOK** assessment.

f) Please write your Student Number only. Do not write your name.

Student number:

For Examiner's Use Only		
	Max Marks	Earned Marks
Problem 1	15	
Problem 2	10	
Problem 3	15	
TOTAL:	40	

Problem 1. Short Questions (15 Marks)

a) Consider doing MAP estimation for linear regression where we are assuming additive zero mean Gaussian noise around the target function. For the prior distribution on the weights of the linear function, assume that we have a mixture of *k equally probable* Gaussians with the *same* covariance matrix Σ and centers at $w_1, \ldots, w_k$. Write down the corresponding optimization problem in the form of a weighted sum of the empirical risk and a regularizer.

b) True or False. Justify your answer.
Consider representing the PARITY function using decision trees in the case where the distribution of instances is known to be uniform over all instances with at most two non-zero inputs. The size of a decision tree that can obtain zero expected error under this distribution of instances must be exponential in the number of inputs *d*.

c) Consider a finite Boolean function class H, a target function $f \in H$ and a distribution of instances D. Assume the expected error of any hypothesis $h \in H, h \neq f$ is at least τ, i.e. $L_{(D,f)}(h) \geq \tau$. Show that the expected error of the empirical risk minimizer decreases at least exponentially with the i.i.d. training sample size m.

d) True or False. Justify your answer.
Let the VC dimension of the function class F be d and let S be *any* set of d points from the domain of functions in H. Then the Rademacher complexity $R(F \circ S)$ is equal to 1.

e) **True or False.** Justify your answer.
Consider running the Perceptron algorithm on inputs with norm bounded by 1. Assume that there exists a halfspace with margin 0.1 that correctly classifies all the inputs. Then the Perceptron algorithm will never make more than 100 mistakes.

Problem 2. Graph Convolution Network (10 Marks)

We will examine the VC dimension of a simple version of graph convolution networks (GCN). We will consider doing binary classification of graphs using GCN. We will assume that all graphs to be classified have a fixed structure with n vertices and the same $n \times n$ adjacency matrix A. Each *instance of a graph* to be classified will be described by a $n \times d$ feature matrix X where row i of the matrix is a feature vector of length d, associated with vertex v_i.

Each layer of the GCN takes a matrix H_l as input where $H_0 = X$, and outputs a matrix H_{l+1} using the function $H^{l+1} = \sigma(AH^lW^l)$, where W^l is a $d^l \times d^{l+1}$ matrix of weights and σ applies the threshold function to each entry of the matrix AH^lW^l. For simplicity, assume $d^i = d$ for each i. After L layers, the matrix H^L is converted into a vector of length nd, composed with a linear function, then thresholded for classification (for simplicity, assume that all threshold functions are thresholded at zero for this question).

a) Assume that the GCN is run on a computer where each weight (parameter) in the network is discretized to 64 bits. Derive a good asymptotic upper bound for the VC dimension of the GCN in terms of n, d, and L.

b) Now assume that all the weights in the network are real-valued. Derive a good asymptotic upper bound for the VC dimension of the GCN in terms of n, d, and L.

Problem 3. Virus Again (15 Marks)

Consider the problem covered in lecture, where we would like to check whether a file contains a virus. In the problem, a file is assumed to contain a virus if it contains a signature (fixed substring) that identifies the virus. We assume that the signature is not known but is known to be of length **exactly** *d* (this is a simplification of the case covered in lecture, where we assume that it is of length no more than *d*). In lecture, we discussed using kernel method for solving the problem. In this question, we will consider using gradient descent and boosting.

As in lecture, we will use the set of indicator features $\psi_v(x)$ which takes the value 1 if the file *x* contains the string *v* and takes the value 0 otherwise, and *v* is of length exactly *d*. The indicator features $\psi_v(x)$ will be used as features for a **linear function** to be learned using gradient descent and stochastic gradient descent. Assume a training set of size *m*, where each training example is a file of length *n*. Assume that the logistic loss is used with the squared norm of the weights as a regularizer, i.e. the objective function to be minimized is $\frac{1}{m}\sum_{i=1}^{m}(l(w^T\psi(x_i), y_i) + \beta\|w\|_2^2)$ where $\psi(x)$ is the vector of features, w is the weight vector, $l(a, y) = \ln(1 + e^{-ya})$ is the logistic loss, β is a constant, and $y \in \{-1,1\}$ is the label.

The number of features is exponential in *d*, but we would like to obtain algorithms that do not run in time exponential in *d*. After preprocessing of the training set, given any substring *v* of length *d*, it is possible to find the component of w that correspond to a feature ψ_v in time linear in *d*. We will assume that *d* is constant, and so the time required to do that is constant.

a) Argue that the weight for any feature $\psi_v(x)$ where *v* is not a substring in any training example is zero in an optimal solution of $\frac{1}{m}\sum_{i=1}^{m}(l(w^T\psi(x_i), y_i) + \beta\|w\|_2^2)$.

b) We will use gradient descent to optimize the regularized loss function. From part (a), any feature associated with substrings that are not in the training set will have zero weight in the optimal solution, so we will not include any such feature in our computation. Give an *efficient* algorithm for doing *one* iteration of gradient descent on the objective function. What is the asymptotic runtime of your algorithm in terms of *m* and *n*?

c) Now consider minimizing the objective function $\frac{1}{m}\sum_{i=1}^{m}(l(h(x_i), y_i) + \beta\|w\|_2^2)$ using stochastic gradient descent, where at each iteration, an index i is randomly selected with equal probability from 1 to *m* and the stochastic gradient update is performed on $l(h(x_i), y_i) + \beta\|w\|_2^2$. Give an *efficient* algorithm for doing *k* iterations of stochastic gradient descent on the objective function, assuming that sampling an index *i* takes constant time. (You should handle the regularizer correctly.) What is the asymptotic runtime of your algorithm in terms of *k*, *m* and *n*?

We now consider constructing functions from the substrings for use in boosting. Let the function $\psi'_v(x)$ output the class 1 if the file *x* contains the string *v* and -1 otherwise, where *v* is of length exactly *d*. As before, we would like to avoid having to consider all possible substrings of length *d*, which would take time exponential in *d*.

d) Let $H_1 = \{\psi'_v(x): v\ consist\ of\ all\ strings\ of\ length\ d\}$ and $H_2 = \{\psi'_v(x): v\ consist\ of\ all\ substrings\ of\ length\ d\ in\ the\ training\ set\} \cup \{\psi''(x)\}$ where $\psi''(x)$ is the function that outputs -1 for all *x*. Assume that the training set is reweighted using some distribution *D*. Argue that minimum expected error achievable by functions in H_1 and H_2 under the distribution *D* is the same.

e) Assume that the training set is reweighted using some distribution *D*. Give an efficient algorithm for finding a function in H_2 that minimizes the expected error under distribution *D*. What is the asymptotic runtime of your algorithm in terms of *m* and *n*?

END OF PAPER