Midterm (Take Home)

Out: Thursday, March 14, 2019
Due: Thursday, March 21, 2019 at 2:30pm

Exam instructions (please read carefully):

• **Exam structure.** This exam contains 4 problems. For 1000-level credit, choose any 3 of the 4 problems. For 2000-level credit, do all 4 problems. Each problem is worth 10 points. (If you take the exam for 1000-level credit and answer more than 3 questions, we will only grade your first 3 answers.)

• **Allowed resources.** You are allowed to consult the course web site, your notes, and the course textbook. You are **not** allowed to use other resources, such as other books, other web sites, or other people.

• **Non-collaboration.** No collaboration is permitted on this exam. It is trusted that you will not discuss this exam or related course material with any other person (classmate or otherwise). You must abide by the Brown University Academic Code concerning examinations, quizzes, and tests (see http://goo.gl/mQtfSa).

• **Questions.** There will be no TA hours during this week. For any clarification of possible textual ambiguities email cs1550tas@lists.brown.edu with your question. We will post clarifications (anonymously) to the Google Doc on the course website.

• **Handing in.** Print your solutions. Then print this cover sheet (the first two pages of this handout), sign and write your name below, and attach to the front of your packet. Bring to class by 2:30pm. Late handins will not be accepted without prior authorization from the CS 155 instructor.

**PLEASE SIGN (failure to sign voids the exam):** I solemnly state that I have abided by the exam instructions stated above, including the tenets of the Brown University Academic Code concerning examinations, quizzes and tests.

<table>
<thead>
<tr>
<th>Signature:</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Name (please print):</th>
<th></th>
</tr>
</thead>
</table>
Problem 1

Problem 2

Problem 3

Problem 4

Total
Problem 1
Consider the following model for a digital communication channel with noise. Alice wants to communicate a message \( m \) to Bob. First, Alice encodes \( m \) as a string \( s \) of length \( n \), using an error-correcting code. This means that it is enough for Bob to have \( n(1 - q) \) correct bits of \( s \) (where \( q \) is a constant between 0 and 1) in order to correctly decode it and obtain Alice’s original message \( m \). That is, if Bob has a string \( s' \) that matches \( s \) in at least \( n(1 - q) \) places, then Bob can recover \( m \).

Each bit that Alice sends over the channel is received correctly by Bob with probability \( p > \frac{1}{2} \), independently of other bits. Alice wants to send \( s \) over the channel, but to improve the probability that Bob can recover \( m \), Alice sends \( s \) several times and Bob takes the majority value for each bit to obtain \( s' \). Bob then attempts to recover \( m \) from \( s' \).

1. How many times does Alice need to send \( s \) so that the probability that a given bit of \( s \) has a correct majority is at least \( 1 - r \)?

2. What values of \( r \) does Alice need to use so that Bob can recover \( m \) correctly with probability at least \( 1 - \epsilon \)? You can give an inequality involving \( r \), instead of solving for \( r \) explicitly.

Problem 2
Consider the following algorithm for finding the median in linear time. (There are some differences between this algorithm and the algorithm presented in class/the textbook, so please read the problem carefully.)

Randomized Median Algorithm:
Input: A set \( S \) of \( n \) elements over a totally ordered universe.
Output: The median element of \( S \), denoted by \( m \).

1. Pick a (multi-)set \( R \) of \( \lceil \frac{n}{2} \rceil \) samples from \( S \), chosen independently and uniformly at random with replacement. (This means that \( R \) can contain duplicate elements. In the rest of the problem, when we refer to \( R \) as a set, we really mean a multiset.)
2. Sort the set \( R \).
3. Let \( d \) be the \( \lfloor \frac{1}{2} n^{1/2} - n^{1/4} \log n \rfloor \)-th smallest element in the sorted set \( R \).
4. Let \( u \) be the \( \lceil \frac{1}{2} n^{1/2} + n^{1/4} \log n \rceil \)-th smallest element in the sorted set \( R \).
5. By comparing every element in \( S \) to \( d \) and \( u \), compute the set \( C = \{ x \in S : d \leq x \leq u \} \), and the numbers \( \ell_d = \lvert \{ x \in S : x < d \} \rvert \) and \( \ell_u = \lvert \{ x \in S : x > u \} \rvert \).
6. If \( \ell_d > n/2 \) or \( \ell_u > n/2 \) then FAIL.
7. If \( |C| \leq 6n^{1/4} \log n \), then sort the set \( C \), otherwise FAIL.
8. Output the \( (\lfloor \frac{n}{2} \rfloor - \ell_d + 1) \)-st element in the sorted order of \( C \).

1. Prove that with probability \( 1 - o(n^{-2}) \), the median is in the set \( C \). (Recall the meaning of the little-O notation: we say \( f \) is \( o(g) \) if \( f(n)/g(n) \to 0 \) as \( n \to \infty \).)
2. Prove that with probability \( 1 - o(n^{-2}) \), \( |C| \leq 6n^{3/4} \log n \).
3. Conclude that with probability $1 - o(n^{-2})$, the algorithm finds the median of the set $S$ in $O(n)$ time.

[Hint: Review section 3.5 in the textbook, but to obtain the above error bounds you’ll need to use large deviation bounds instead of the Chebyshev’s Inequality.]

**Problem 3**

Let $X = \mathbb{R}^2$, and consider the set of concepts $C$ where $c \in C$ if $c = \{(x, y) \mid x^2 + y^2 \leq r^2\}$ for some real number $r$ (i.e., a concept $c$ is defined by a unique value $r$, and it corresponds to all points in the disk of radius $r$ centered at $(0,0)$).

1. What is the VC dimension of $(X, C)$?

2. Show that $C$ is $(\epsilon, \delta)$-PAC-learnable from a training set of size $m = \frac{1}{\epsilon} \ln \frac{1}{\delta}$. (Note that this is the exact value of $m$, not using big-O notation.)

**Problem 4**

1. Given two range spaces $(X, \mathcal{R}_1)$ and $(X, \mathcal{R}_2)$, each with VC-dimension $d$, we define a “union” range space $(X, \mathcal{R}_{1 \cup 2})$:

   $$\mathcal{R}_{1 \cup 2} = \{r_1 \cup r_2 \mid r_1 \in \mathcal{R}_1 \text{ and } r_2 \in \mathcal{R}_2\}.$$ 

   Prove that the VC dimension of $\mathcal{R}_{1 \cup 2}$ is $O(d)$.

   You may use without proof the result of Exercise 14.9: for $n \geq 2d$ and $d \geq 1$, the growth function satisfies $G(d, n) = \sum_{i=0}^{d} \binom{n}{i} \leq 2 \left(\frac{en}{d}\right)^d$. (You may not use Theorem 14.5 or Corollary 14.6 directly, because those results are not proven in the book.)

2. Environmental scientists divide the country into polluted and not polluted areas. They want to identify the 5 axis-aligned squares that most accurately represent the polluted areas. We assume the map of the country is drawn on a 2D plane, and the axis-aligned squares are chosen from this map. Now, let $C = \{S_1, \ldots, S_5\}$ be a set of 5 axis-aligned squares (each $S_i$ is an axis-aligned square). For a uniformly randomly chosen location (point) $X$ on the map, we denote $X \in C$ if $X \in \bigcup_{i=1}^{5} S_i$. We also denote $P$ as the set of all polluted locations on the map. Define the error of $C$ as

   $$\text{err}(C) = \Pr(X \in P, X \notin C) + \Pr(X \notin P, X \in C),$$

   which measures how well $C$ minimizes the coverage of non-polluted areas and maximizes the coverage of polluted areas. Let $C^*$ be the set of the 5 squares with minimum error. The scientists can sample uniformly at random a location in the country and characterize it as polluted or not polluted. For a given $\epsilon > 0$, their goal is to find a set of 5 squares $C'$ such that $\text{err}(C') \leq \text{err}(C^*) + \epsilon$. How many samples do they need in order to find such a set of 5 axis-aligned squares with probability at least $1 - \delta$, for $\delta \in (0,1)$? You can give the sample complexity using big $O(\cdot)$ notation.