COMS 4995 (Randomized Algorithms): Problem Set #4

Due by 11:59 PM on Monday, November 18, 2019

Instructions:

(1) Form a group of 1-3 students. You should turn in only one write-up for your entire group.

(2) Submission instructions: We are using Gradescope for the homework submissions. Go to www.gradescope.com to either login or create a new account. Use the course code 9D6V5E to register for this class. Only one group member needs to submit the assignment. When submitting, please remember to add all group member names in Gradescope. See the course Web site for detailed instructions.

(3) Please type your solutions if possible and we encourage you to use the LaTeX template provided on the course home page.

(4) Write convincingly but not excessively.

(5) Some of these problems are difficult, so your group may not solve them all to completion. In this case, you can write up what you've got (subject to (3), above): partial proofs, lemmas, high-level ideas, counterexamples, and so on.

(6) Except where otherwise noted, you may refer to your lecture notes and the specific supplementary readings listed on the course Web page *only*. You can also review any relevant materials from your undergraduate algorithms course. If you do use any approved sources, make you sure you cite them appropriately, and make sure that all your words are your own.

(7) You can discuss the problems verbally at a high level with other groups. And of course, you are encouraged to contact the course staff (via Piazza or office hours) for additional help.

(8) If you discuss solution approaches with anyone outside of your group, you must list their names on the front page of your write-up.

(9) Refer to the course Web page for the late day policy and the School of Engineering honor code.

Problem 19

(12 points) This problem relaxes the independence assumption in the Chernoff bound to a negative correlation assumption.

Let $X_1, \ldots, X_n$ be a set of binary random variables satisfying the condition

$$\mathbf{Pr}[X_i = 1 \text{ for all } i \in S] \leq \prod_{i \in S} \mathbf{Pr}[X_i = 1] \tag{1}$$

for all non-empty subsets $S \subseteq \{1, 2, \ldots, n\}$. Prove that the usual Chernoff bounds apply to the sum $\sum_{i=1}^{n} X_i$. (You don't need to go through all the different Chernoff bounds, just the first one listed in Exercise 24.)

Problem 20

(25 points) Recall that a given set of keys (from a totally ordered set) can be organized as a heap in many different ways, and also as a binary search tree in many different ways. A *treap* is a binary tree whose nodes contain two values, a *key* x and a *priority* p_x, such that the tree is a valid heap with respect to nodes' priorities and a valid search tree with respect to nodes' keys. (We will always assume that all keys are distinct, and similarly for priorities.)

(a) (4 points) Prove that for a totally ordered set X of elements and a function p assigning unique priorities to the elements of X, there is a unique treap with keys X and priorities p.

(b) (4 points) The running times of most search tree operations are governed by the tree's height, so it's interesting to understand the typical height of a treap with random priorities.

Precisely, assume that $X = \{1, 2, \ldots, n\}$ and consider a uniformly random permutation π (a bijection from X to itself), with $\pi(1)$ indicating the element of X with the highest priority (call it priority 1), $\pi(2)$ the element with the second-highest priority (priority 2), and so on.

For a fixed permutation π, call the element $\pi(j)$ in position $j \in \{1, 2, \ldots, n\}$ a *peak* if all smaller positions $1, 2, \ldots, j-1$ contain elements $\pi(1), \ldots, \pi(j-1)$ that are smaller than $\pi(j)$. (For example, the highest-priority element is always a peak.) Analogously, j is a *valley* if all smaller positions contain elements that are larger than $\pi(j)$. (Again, the highest-priority element is always a valley.)

Let Y_j be the indicator random variable indicating whether or not the element $\pi(j)$ in position j is a peak. Prove that for every choice of $y_{j+1}, \ldots, y_n \in \{0, 1\}$,

$$\mathbf{Pr}[Y_j = 1 \mid Y_{j+1} = y_{j+1}, \ldots, Y_n = y_n] = \frac{1}{j},$$

where the probability is over the uniformly random choice of π; and similarly for valleys.

(c) (5 points) For $i \in \{1, 2, \ldots, n\}$, let A_i and B_i denote the sets $\{1, 2, \ldots, i\}$ and $\{i, i+1, \ldots, n\}$ of elements before and after i, respectively (including i itself). A permutation π on $\{1, 2, \ldots, n\}$ induces permutations on A_i and B_i (with the relative order of the priorities of elements in A_i or B_i the same as in π).[1]

Prove that, in the (unique) treap corresponding to the permutation π, an element j is on the path from the root to the node containing i if and only if j is a peak with respect to the permutation induced by π on A_i or a valley with respect to the permutation induced by π on B_i.

(d) (3 points) Conclude that for every $i \in X$, the expected length (over π) of the path from the root to the node containing i in the treap corresponding to π is $O(\log n)$.

[You can assume without proof that $\sum_{j=1}^{n} \frac{1}{j} = O(\log n)$; this is easy to prove by approximating the sum with an integral.]

(e) (5 points) Prove that the random variables $Y_1, \ldots, Y_n$ from part (b) satisfy the condition (1) in the previous problem.

(f) (4 points) Conclude that with high probability (approaching 1 as $n \to \infty$) over the choice of π, the height of the treap corresponding to π is $O(\log n)$.

[1] For example, suppose $\pi(1) = 4$, $\pi(2) = 2$, $\pi(3) = 1$, $\pi(4) = 3$. In A_2, the highest-priority element (according to π) is 2, the second-highest is 1. (So 2 is both a peak and a valley in the induced ordering, while 1 is a valley only.) In B_2, the highest-priority element is 4 (so both a peak and a valley), followed by 2 (a valley but not a peak) and then 3 (neither a peak nor a valley).

Problem 21

(16 points) Consider two parties, the all-powerful and omniscient Alice, and the computationally bounded Bob. The complexity class $\mathcal{NP}$ can be characterized as the set of languages decidable using a particular type of Alice-Bob interaction. Specifically, a language L (e.g., 3SAT) is in $\mathcal{NP}$ if and only if there is a communication protocol of the following type:

1. A third party writes down an input $x \in \{0,1\}^*$ on a billboard visible to Alice and Bob. (x may or may not belong to L, e.g., may or may not be satisfiable.)
2. Alice sends Bob a message m with length bounded by a polynomial function of the length $|x|$ of the input.
3. Bob spends a polynomial amount of time processing m and proposes whether x belongs to L. More formally, Bob uses a polynomial-time Turing machine M that takes as input x and m, and outputs its answer $M(x,m)$ (which is either "yes" or "no").

with correctness defined by two properties:

- (Completeness) If $x \in L$, Alice can choose the message m so that Bob accepts (i.e., $M(x,m)$ is "yes").
- (Soundness) If $x \notin L$, then no matter what message m Alice sends to Bob, Bob rejects (i.e., $M(x,m)$ is "no").

For example, suppose L is 3SAT. Bob (i.e., M) will propose "yes" if and only if Alice sends him a message m encoding a satisfying assignment to the input x. (Bob can check that m is well formed and encodes a satisfying assignment to x in polynomial time.) Completeness holds because when $x \in L$ Alice can send a satisfying assignment and Bob will accept. (Recall that Alice is all-powerful and so can compute such an assignment.) Soundness holds because if $x \notin L$, then any message Alice could send fails to encode a satisfying assignment and Bob will reject.[2]

It is interesting to explore the generalization of such protocols in which Bob is permitted to use a *randomized* polynomial-time Turing machine for verification. In other words, M is now a polynomial-time Turing machine that takes as input x, Alice's message m, and a random string r. (The behavior of M is deterministic for a fixed r, but randomized when r is random.) The running time of M (and hence the lengths of m and r) is bounded by a polynomial function $p(|x|)$ of the input length $|x|$. The correctness properties become:

- (Completeness) If $x \in L$, Alice can choose the message m so that Bob accepts with probability at least $\frac{2}{3}$. (I.e., so that $\mathbf{Pr}[M(x,m,r) = \text{"yes"}] \geq \frac{2}{3}$, where the probability is over a uniformly random choice of r.)
- (Soundness) If $x \notin L$, then no matter what message m Alice sends to Bob, Bob rejects with probability at least $\frac{2}{3}$.

Let $\mathcal{A}$ denote the class of languages decided by such "Alice-Bob" protocols (which includes all of $\mathcal{NP}$, and possibly other languages as well).

What happens if we force Bob to go first, before receiving any message from Alice, thereby allowing Alice to respond as a function of Bob's random string r? Precisely, consider communication protocols of the following form:

1. A third party writes down x on a publicly visible billboard.
2. Bob sends Alice a description of M and a random string r of the appropriate length (at most $p(|x|)$).
3. Alice chooses a message m with length at most $p(|x|)$ and proposes the answer computed by $M(x,m,r)$.

[2] The same argument works for any problem in $\mathcal{NP}$, not just 3SAT. Conversely, if a language is decided by such a protocol, then it must be in $\mathcal{NP}$: a nondeterministic Turing machine can guess the message m that Alice would send to make Bob accept. Put differently, such a message m acts as a witness for membership in L, with Bob's machine M acting as the polynomial-time verifier.

Correctness is now:

- (Completeness) If $x \in L$, with probability at least $\frac{2}{3}$ over the choice of r, Alice can choose a message m so that $M(x, m, r) =$ "yes".
- (Soundness) If $x \notin L$, with probability at least $\frac{2}{3}$ over the choice of r, there is no message m for which $M(x, m, r) =$ "yes".

Let $\mathcal{B}$ denote the class of languages decided by such "Bob-Alice" protocols.[3] Prove that $\mathcal{A} \subseteq \mathcal{B}$—that every language decided by an Alice-Bob protocol can also be decided by a Bob-Alice protocol.

[Hints: Look to the proofs of Newman's and Adleman's Theorems (Lecture #14) for inspiration. Begin by amplifying the correctness probability of Alice-Bob protocols. Look to eventually take a Union Bound over all of Alice's options for m.]

Problem 22

(15 points) Recall that a *d-dimensional linear classifier* is specified by $d+1$ points $a_0, \ldots, a_d \in \mathbb{R}$ and assigns each point $\mathbf{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ a label of 1 if $a_0 + \sum_{i=1}^{d} a_i x_i \geq 0$ and a label of 0 otherwise. This problem concerns the VC dimension of linear classifiers. (Recall that the VC dimension of a class $\mathcal{H}$ is the largest m such that $G_{\mathcal{H}}(m) = 2^m$. (And see Exercise 37 for the definition of $G_{\mathcal{H}}$.))

(a) (4 points) Prove that the VC dimension of the set of d-dimensional linear classifiers is at least $d + 1$.

(b) (4 points) Prove that, for every set $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_{d+2}\}$ of $d + 2$ points in $\mathbb{R}^d$, there exist coefficients $\alpha_1, \ldots, \alpha_{d+2}$, not all zero, satisfying the scalar equation

$$\sum_{i=1}^{d+2} \alpha_i = 0$$

and also the vector equation

$$\sum_{i=1}^{d+2} \alpha_i \mathbf{x}_i = 0.$$

[Hint: Consider the points $\{(\mathbf{x}_i; 1)\}_{i=1}^{d+2}$ in $\mathbb{R}^{d+1}$.]

(c) (4 points) Prove that, for every point set S as in (b), there exists a partition of S into non-empty sets A and B such that the convex hulls of A and B intersect.[4]

[Hint: Group points according to the signs of the α_i's obtained from (b).]

(d) (3 points) Prove that the VC dimension of d-dimensional linear classifiers is at most $d + 1$.

Problem 23

(20 points) In the $\mathcal{NP}$-hard *Hitting Set* problem, the input is a universe U of n elements and a list of m subsets $\mathcal{C} = \{S_1, \ldots, S_m\}$ of U. The goal is to find a subset $H \subseteq U$ that intersects ("hits") all of the sets of $\mathcal{C}$: $H \cap S_i \neq \emptyset$ for each $i = 1, 2, \ldots, m$.

In this problem, we'll consider the special case of the Hitting Set problem in which the given set system has bounded VC dimension. (It turns out that this special case is still $\mathcal{NP}$-hard, even when the VC dimension is 2.) The VC dimension of a set system is defined as the VC dimension of the corresponding family of

[3] $\mathcal{B}$ includes all of $\mathcal{NP}$—do you see why?

[4] Recall that the *convex hull* of a point set $P = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is the set of all possible convex combinations of points in P, where a *convex combination* is a linear combination $\sum_{i=1}^{n} \alpha_i \mathbf{x}_i$ in which the α_i's are nonnegative and sum to 1. Equivalently, the convex hull of P is the intersection of all convex sets that include P.

(binary) characteristic functions. In other words, the VC dimension of the set system $\mathcal{C} = \{S_1, \ldots, S_m\}$ is the largest size of a subset $X \subseteq U$ such that, for every $Y \subseteq X$, there exists a set $S \in \mathcal{C}$ with $S \cap X = Y$.

We will need one fact about set systems with bounded VC dimension.[5] Let D be a distribution on U—that is, a nonnegative mass $p(x)$ for each $x \in U$ with $\sum_{x \in U} p(x) = 1$. An *ϵ-net* (with respect to $\mathcal{C}$ and D) is a subset $N \subseteq U$ that hits every set in $\mathcal{C}$ that has probability at least ϵ:

$$N \cap S \neq \emptyset \text{ for all } S \in \mathcal{C} \text{ with } \sum_{x \in S} p(x) \geq \epsilon.$$

We will use the following fact without proof:[6] there is a constant $c > 0$ (independent of d and D) such that, if $\mathcal{C}$ has VC dimension d and

$$m \geq c \cdot \frac{d}{\epsilon} \ln \frac{1}{\epsilon}, \tag{2}$$

then with probability at least $\frac{1}{2}$, a random size-m subset R of U (with each element of R drawn i.i.d. according to the distribution D) is an ϵ-net with respect to $\mathcal{C}$ and D.

Returning to the Hitting Set problem, we assume from now on that the VC dimension d of the given set system is bounded by a constant (i.e., $O(1)$). The plan is to reduce the problem to the computation of an ϵ-net with respect to a suitable distribution D over U. (ϵ-nets are like hitting sets except that they might miss some low-probability sets.) The idea is to evolve the distribution D until an ϵ-net has to be a bona fide hitting set. (This is very different from the setup in our learning applications, where the distribution D was given and not under the control of the algorithm designer.)[7] The exact algorithm is:

1. Initialize a weight $w(x) = 1$ for every element $x \in U$, and D as the uniform distribution over U.
2. Repeat:
   (a) $N := \emptyset$.
   (b) While N is not an ϵ-net with respect to $\mathcal{C}$ and D:
      i. Redefine N as a random sample of m elements drawn i.i.d. from D, where m is defined as in (2).
   (c) If N is a hitting set for $\mathcal{C}$, halt and return N.
   (d) Otherwise, choose an arbitrary set $S \in \mathcal{C}$ with $S \cap N = \emptyset$ and double the weight of every element $x \in S$.
   (e) Redefine D to be proportional to elements' current weights.

This algorithm is underspecified in that we have not yet committed to a setting of the parameter ϵ that appears in (2).

(a) (4 points) Let $W := \sum_{x \in U} w(x)$ denote the sum of elements' current weights. Prove that, in every iteration of the algorithm's outer loop, W increases by at most a factor of $1 + \epsilon$.

(b) (4 points) Let OPT denote the size of an optimal hitting set, and fix such a set H^*. Prove that after k iterations of the algorithm's outer loop, $\sum_{x \in H^*} w(x) \geq OPT \cdot 2^{k/OPT}$.

(c) (4 points) Prove that, if ϵ happens to be $\frac{1}{2OPT}$, then the algorithm terminates (necessarily with a hitting set) after $O(OPT \log \frac{n}{OPT})$ iterations of the outer loop (where $n = |U|$).

[5] This fact actually holds more generally for set systems with polynomially bounded growth functions.

[6] Though actually, the proof is almost the same as the one in Lecture #15, which established sample complexity bounds for uniform convergence with a hypothesis class with a polynomially bounded growth function. In fact it's a bit easier, with a variation of Exercise 36 substituting for the Chernoff-based argument in lecture (which explains why the dependence on $\frac{1}{\epsilon}$ is linear rather than quadratic).

[7] The ideal choice of D would be, for some optimal hitting set H^*, to set $p(x) = \frac{1}{|H^*|}$ for $x \in H^*$ and $p(x) = 0$ otherwise. In this case, any ϵ-net with $\epsilon = \frac{1}{|H^*|}$ must include all of H^* and hence be a hitting set. By (2), a random sample of $O(|H^*| \log |H^*|)$ elements from this distribution D would be a hitting set with constant probability.

(d) (4 points) Continuing part (c), prove that the expected running time of the algorithm (with $\epsilon = \frac{1}{2OPT}$) is polynomial and that it returns a hitting set of size at most $O(\log OPT)$ times the minimum possible.

[Note: your running time analysis should include the time needed to check the condition in step 2(b), as well as a proof that elements' weights remain small enough to be described with a polynomial number of bits.]

(e) (4 points) Add an additional outer loop to the algorithm that searches for the "right" value of the parameter ϵ. Prove that your algorithm is a Las Vegas randomized polynomial-time $O(\log OPT)$-approximation algorithm for the Hitting Set problem for set systems with constant VC dimension.

Problem 24

(12 points) The goal of this problem is to show that, for every set system $\mathcal{C} \subseteq 2^U$ (equivalently, every family of binary functions on U), there exists a distribution D on the universe U such that the sample complexity bound in (2) for ϵ-nets is tight (up to the logarithmic factor).[8]

Fix a universe U and a collection $\mathcal{C}$ of subsets of U. Fix $\epsilon > 0$, which you can assume is sufficiently small. Suppose (the family of binary functions corresponding to) $\mathcal{C}$ has VC dimension d.

(a) (7 points) Define a probability distribution D on U such that: for a sufficiently small constant $c > 0$ (independent of d), if $m \leq c\frac{d}{\epsilon}$ elements are drawn i.i.d. from U according to the distribution D, then with probability at least $\frac{1}{2}$ these elements fail to be an ϵ-net with respect to $\mathcal{C}$ and D.

[Hint: concentrate the probability mass on a shattered set, with the most of the mass on a single element.]

(b) (5 points) Explain why the result in part (a) implies a sample complexity lower bound for the ERM algorithm, even in the realizable special case. That is, in order for the ERM algorithm to successfully PAC learn with respect to a hypothesis class $\mathcal{H}$ with VC dimension d (i.e., with probability at least $\frac{1}{2}$ ERM outputs a hypothesis of $\mathcal{H}$ with generalization error at most ϵ more than the minimum possible), $m = \Omega(\frac{d}{\epsilon})$ samples are necessary (for a worst-case distribution), even when the ground truth function f belongs to the hypothesis class $\mathcal{H}$.

[8] The sample complexity bound for uniform convergence (with quadratic dependence on $\frac{1}{\epsilon}$) is similarly tight up to logarithmic factors, although the proof requires a bit more work.