CS264: Homework #9

Due by midnight on Wednesday, December 3, 2014

Instructions:

(1) Students taking the course pass-fail should complete the exercises. Students taking the course for a letter grade should also complete some of the problems — we'll grade your problem solutions out of a total of 40 points (with any additional points counting as extra credit).

(2) All other instructions are the same as in previous problem sets.

Lecture 17 Exercises

Exercise 52

Prove that every distribution p with support size s — i.e., there are only s points $x \in X$ with $p_x > 0$ — has entropy at most $\log_2 s$.

[Hint: prove and use that the entropy function $\sum_{x \in X} p_x \log_2 \frac{1}{p_x}$ is convex.]

Exercise 53

This exercise considers a minor variant of the optimal search tree problem mentioned in lecture. Recall that a binary search tree on a totally ordered set X is a binary tree with nodes in correspondence with X, with the property that every node in the left (right) subtree of a node x must be less than (greater than) x. Recall that there are many different binary search trees on the same set X (from long chains to balanced trees). The search time $s_T(x)$ for a node x in a search tree T is one plus its depth in the tree (1 for the root, 2 for the root's immediate children, and so on.). Given a positive probability p_x for every $x \in X$, the *optimal search tree* is the search tree T that minimizes the expected search time for a node $x \in X$, $\sum_{x \in X} p_x s_T(x)$.

Give a dynamic programming algorithm that, given p_x's for all $x \in X$, computes an optimal search tree.

[Hints: shoot for a dynamic programming algorithm that solves $O(n^2)$ subproblems in time $O(n)$ each. (Here $n = |X|$.) To get started, if you happened to know the root node of the optimal search tree, what could you say about its subtrees?]

Exercise 54

The point of this exercise is to explain why, when constructing a near-optimal search tree, it is enough to consider only the elements with large (at least $1/|X|^{\epsilon}$) probability, handling the rest via binary search. This justifies the implementation of the approximate search tree construction outlined at the end of lecture.

The precise exercise is the following. Consider a totally ordered set X with n elements. Let $D = \{p_x\}_{x \in X}$ be a probability distribution on X and $S \subseteq X$ the elements with $p_x \geq n^{-\epsilon}$, where $\epsilon > 0$ is an arbitrary constant. Prove that

$$\sum_{x \in S} p_x \log_2 \frac{1}{p_x} + \sum_{x \notin S} p_x \log_2 n = O(H(D)),$$

where the constant hidden in the big-oh notation can depend on ϵ. Explain the relevance of this statement to the algorithm described in lecture.

Exercise 55

Name at least two places in the proof of this lecture's main result where we used the assumption that the x_i's are independent.

[Hint: one is in one of the Problems, below.]

Lecture 18 Exercises

Exercise 56

This exercise and the next concern the *Vickrey auction*. As in lecture, assume there is a seller with a single item to sell. But assume now that there are *two* potential buyers.[1] An auction must decide which, if any, of the buyers to sell the item to. A natural approach is to ask each buyer i to submit a bid b_i and then award the good to the higher bidder. In the Vickrey auction, the winning bidder pays a price equal to the *lower* bid.[2]

Define the utility of a bidder i with value v_i as 0 if it loses (and pays nothing) and as $v_i - p$ if it wins and pays the price p. Prove that for each bidder $i = 1, 2$, no matter i's value v_i is and no matter the other bidder's bid is, i maximizes its utility by setting its bid b_i equal to its value v_i.

[Hint: consider two cases.]
[Remark: such auctions are called "truthful."]

Exercise 57

The Vickrey auction is "prior-independent," in the sense that its description does not depend on any prior distribution (in contrast to the monopoly price, for example). The point of this exercise is to use results from lecture to prove that, when bidders' values happen to be i.i.d. draws from a distribution, the Vickrey auction has pretty good expected revenue.

Specifically, suppose there are two buyers and that their values v_1, v_2 are drawn i.i.d. from a regular distribution F (i.e., a distribution with a concave revenue curve). Prove that the expected (over v_1, v_2) revenue of the Vickrey auction is at least $\max_p p(1 - F(p))$, the optimal expected revenue that can be obtained from a single bidder (via the monopoly price).

[Hint: show that each bidder effectively faces a random price drawn from F, and apply a result from lecture.]

Problems

Problem 28

This problem fills in a gap from Lecture #17, that with high probability over the set V of n bucket boundaries, the expected (over a new random input $x_1, \ldots, x_n$) squared size of every bucket is at most a constant. Recall how V is constructed: for c a sufficiently large constant and $\lambda = c \ln n$, a sorted list of all the elements in the first λ inputs (λn inputs in all) is constructed, and V is defined as every λth element of this sorted list (n elements in all). You can assume throughout this problem that n is sufficiently large.

(a) (5 points) Consider the λn elements in the first λ inputs, in their original unsorted order. For $k, \ell \in \{1, 2, \ldots, \lambda n\}$, suppose that the kth and ℓth elements take on values a and b with the property that the expected number of the other elements (out of the other $\lambda n - 2$) that lie between a and b is at least 4λ. Prove that, for c sufficiently large, the probability (over the other elements) that less than λ of the other elements lie in this interval is at most $1/n^2(\lambda n)^2$.

[Hint: Chernoff bounds.]

[1] We restrict attention to two buyers for simplicity; everything in this and the next exercise extends naturally to any number of bidders.

[2] This seems weird the first time you see it, but think about an English auction (like at Christie's or Soethby's) or eBay — the winning bidder does not generally pay its maximum willingness-to-pay, but rather that of its nearest competitor.

(b) (5 points) Prove that, with high probability over the choice of V (at least $1 - 1/n^2$), the expected number of elements (out of a new random input $x_1, \ldots, x_n$) landing in each bucket is at most 4.

(c) (5 points) Prove that, with high probability over the choice of V (at least $1 - 1/n^2$), the expected squared number of elements landing in each bucket is at most 20.

[Hint: write the bucket size as a sum of indicator random variables and expand.]

Problem 29

The point of this problem is to prove that, for interesting positive results for self-improving sorting algorithms, some type of restriction on the distribution over inputs (like independence of the x_i's) is necessary. Throughout this problem, you can restrict attention to deterministic sorting algorithms (for simplicity), and you can assume that the array length n is sufficiently large.

(a) (2 points) For a set $\mathcal{S}$ of 2^n permutations of $\{1, 2, \ldots, n\}$, let $D_{\mathcal{S}}$ denote the distribution that is uniform on $\mathcal{S}$ (and 0 for permutations outside $\mathcal{S}$). Explain why the entropy of $D_{\mathcal{S}}$ is n.

[Remark: this means that we are aspiring toward a self-improving sorter that uses $O(n)$ expected comparisons whenever the inputs are i.i.d. draws from a distribution of the form $D_{\mathcal{S}}$.]

(b) (2 points) Prove that there are at least $(n!/2^n)^{2^n}$ distinct choices of $\mathcal{S}$ and hence of $D_{\mathcal{S}}$.

(c) (2 points) Suppose that sorting algorithm A uses at most cn expected comparisons to sort inputs drawn from $D_{\mathcal{S}}$. Prove that A correctly sorts at least half of the permutations of $\mathcal{S}$ using at most $2cn$ comparisons.

(d) (2 points) How many different permutations can a sorting algorithm correctly sort using at most k comparisons?

(e) (8 points) Let $c > 0$ be an arbitrary constant (independent of n). Prove that if $\mathcal{A}$ is a collection of sorting algorithms such that, for every distribution of the form $D_{\mathcal{S}}$, there is an algorithm $A \in \mathcal{A}$ that requires at most cn expected comparisons to correctly sort inputs drawn from $D_{\mathcal{S}}$, then $\mathcal{A}$ has size doubly exponential in n.

[Hint: use (c) and (d) to upper bound the number of distinct distributions of the form $D_{\mathcal{S}}$ that a single sorting algorithm can simultaneously have good performance for. Then use (b).]

(f) (4 points) Explain why (e) implies that every self-improving sorter that works for arbitrary input distributions requires space and a number of samples that is exponential in n.

Problem 30

This problem fills in a gap from Lecture #18, that with high probability over the samples $v_1, \ldots, v_m \sim F$, the hypotheses of the relaxed version of Claim 1 hold. Throughout this problem, assume that the number m of samples is at least $c\frac{H}{\epsilon^2} \ln \frac{H}{\epsilon}$, where c is a sufficiently large constant, H is such that the support of the distribution F lies in $[1, H]$, and $\epsilon > 0$ is an approximation parameter. Recall that $q(p) = 1 - F(p)$ denotes the probability (over $v \sim F$) of a sale at the price p.

(a) (5 points) Prove that with probability at least $1 - \frac{\epsilon}{H}$, for every $t \in [\frac{1}{H}, 1]$, there exists a sample v_i such that $q(v_i) \in [\frac{t}{1+\epsilon}, t]$.

[Hint: this is an elementary calculation, plus a Union Bound.]

(b) (6 points) Prove that with probability at least $1 - \frac{\epsilon}{H}$, for every sample v_i with $q(v_i) \geq \frac{1}{H(1+\epsilon)}$,

$$\hat{q}(v_i) \in [(1-\epsilon)q(v_i), (1+\epsilon)q(v_i)]$$

and hence

$$v_i \cdot \hat{q}(v_i) \in [(1-\epsilon)v_i \cdot q(v_i), (1+\epsilon)v_i \cdot q(v_i)].$$

(Recall from lecture that $\hat{q}(v_i) = n_i/m$, where n_i is the number of v_j's that are at least v_i (including v_i itself).)

[Hint: Chernoff bounds.]

(c) (4 points) Prove that with probability at least $1 - \frac{\epsilon}{H}$, the Algorithm from lecture will not return a sample v_i with $q(v_i) < \frac{1}{H(1+\epsilon)}$.

[Hint: use (b) to deduce that every such sample v_i satisfies $\hat{q}(v_i) < \frac{1}{H}$.]

Problem 31

(15 points) In Lecture #18, in the course of proving a sample complexity bound of $O(\frac{H}{\epsilon^2} \ln \frac{H}{\epsilon})$ to achieve an expected revenue-approximation of $(1 - \epsilon)$, we noted that the selling probability $q^* = 1 - F(p^*)$ of the monopoly price p^* is at least $\frac{1}{H}$.

Explain the changes to the proof (in Lecture #18 and Problem #30) required to prove the following more general statement. (Or, if you prefer, simply reprove it from scratch.) Suppose F is an unknown distribution, with arbitrary support, but it is known a priori that the selling probability q^* at the monopoly price p^* is at least $\delta > 0$. Then, provided $m \geq c\frac{1}{\delta\epsilon^2} \ln \frac{1}{\delta\epsilon}$ for a sufficiently large constant c, with probability at least $1 - \epsilon\delta$, the Algorithm from lecture returns a price p such that $p \cdot (1 - F(p)) \geq (1 - \epsilon)p^* \cdot 1 - F(p^*)$.