CS 174: Combinatorics and Discrete Probability Fall 2012

Homework 8

Due: Thursday, November 1, 2012 by **9:30am**

Instructions: *You should upload your homework solutions on bspace. You are strongly encouraged to type out your solutions using LaTeX. You may also want to consider using mathematical mode typing in some office suite if you are not familiar with LaTeX. If you must handwrite your homeworks, please write clearly and legibly. We will not grade homeworks that are unreadable. You are encouraged to work in groups of 2-4, but you* **must** *write solutions on your own. Please review the homework policy carefully on the class homepage.*

Note: You *must* justify all your answers. In particular, you will get no credit if you simply write the final answer without any explanation.

Problem 1. *(8 points)* We consider a problem motivated by recommendation systems used by online merchants such as Amazon and Netflix. Given two sets of integers A, B of size n, we would like to quickly determine if $A = B$, or if $|A \cap B|$ is very small, say $|A \cap B| < 0.01n$. (In the intermediate case, where $A \cap B$ is of moderate size, we do not care what the output is.) In the case of Amazon's recommendation system, A and B could be the list of books purchased by different consumers, and n could be very large.

(a) Sketch a simple deterministic algorithm that computes $|A \cap B|$ exactly using $O(n \log n)$ comparisons.

Solution: Sort both sets A and B and compare. Once A and B are sorted, we can easily compute $A \cap B$ in linear time.

Our aim is to beat this algorithm, using randomization and exploiting the fact that we only want to distinguish the case where $A = B$ from the case where they are very different. Specifically, we seek an algorithm with the following properties:

- if $A = B$, then the algorithm should output *yes* with probability at least 3/4.
- if $|A \cap B| \leq 0.01n$, then the algorithm should output *no* with probability at least 3/4.
- the algorithm uses $O(\sqrt{n} \log n)$ comparisons.

(The value 3/4 here is for convenience only; it can easily be boosted to value $1 - \delta$ for any desired δ using only $O(\log(1/\delta))$ repeated trials.)

Here is the proposed algorithm, where the constant c is to be determined:

(1) choose a subset X of A by picking each element of A independently with probability $c/\sqrt{n}$.

(2) choose a subset Y of B by picking each element of B independently with probability $c/\sqrt{n}$.

(3) if $|X| > 2c\sqrt{n}$ or $|Y| > 2c\sqrt{n}$, output *yes*.

(4) compute $|X \cap Y|$; if $|X \cap Y| \geq 0.1c^2$, output *yes*, else output *no*.

In the rest of this problem, we will show that the algorithm achieves the required properties for a sufficiently large constant c.

(b) Show that the algorithm does indeed use only $O(\sqrt{n}\log n)$ comparisons, assuming that c is constant.

Solution: We only have to compute $|X \cap Y|$ when $|X|$, $|Y| \leq 2c\sqrt{n}$, in which case we only need $O(\sqrt{n}\log n)$ comparisons.

(c) Suppose $A = B$. Show that the algorithm outputs *yes* with probability at least $1 - e^{-0.81c^2/2}$.

Solution: Suppose $A = B$. Fix an element s in $A \cap B$. Then, $\Pr[s \in X \wedge s \in Y] = c^2/n$. We may then write $|X \cap Y|$ as the sum of independent 0-1 r.v.'s, one for each s in $A \cap B$. Hence, by linearity of expectation, $E[|X \cap Y|] = c^2$. Applying a Chernoff bound with $\delta = 0.9$ and $\mu = c^2$, we obtain $\Pr[|X \cap Y| \leq 0.1c^2] \leq e^{-0.81c^2/2}$.

(d) Suppose $|A \cap B| \leq 0.01n$. Show that the algorithm outputs *yes* with probability at most $e^{-0.81c^2/11} + 2e^{-\Omega(\sqrt{n})}$.

Solution: Suppose $|A \cap B| \leq 0.01n$. Then, $E[|X \cap Y|] \leq 0.01c^2$. Again by a Chernoff bound with $\delta = 9$ and $\mu = 0.01c^2$, we obtain $Pr[|X \cap Y| \geq 0.1c^2] \leq e^{-}0.81c^2/11$. Also, writing X as the sum of n independent 0-1 r.v.'s and applying a Chernoff bound with $\delta = 1$ and $\mu = c\sqrt{n}$, we have $Pr[|X| > 2c\sqrt{n}] = e^{-\Omega(\sqrt{n})}$. Similarly, we have $Pr[|Y| > 2c\sqrt{n}] = e^{-\Omega(\sqrt{n})}$. The sum of these three probabilities is an upper bound on the error probability.

(e) Indicate briefly how to choose the constant c so as to achieve the 1/4 error probabilities specified earlier. (You do not need to actually perform the calculation.)

Solution: It suffices to pick c such that $\max\{e^{-0.81c^2}, e^{-0.81c^2/11} + 2e^{-\Omega(\sqrt{n})}\} < 1/4$.

Problem 2. *(Exercise 7.2 from MU – 5 points)* Consider the two-state Markov chain with the following transition matrix.

$$\mathbf{P} = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}.$$

Find a simple expression for $P^t_{0,0}$.

Solution: We can observe that $P^{t+1}_{0,0} = pP^t_{0,0} + (1-p)P^t_{0,1}$ and $P^t_{0,1} = 1 - P^t_{0,0}$. From this, we can derive the recursion

$$P^t_{0,0} = (2p-1)P^{t-1}_{0,0} + (1-p)$$

whose solution is

$$P^t_{0,0} = (2p-1)^t + (1-p)\sum_{s=0}^{t-1}(2p-1)^s = \frac{1+(2p-1)^t}{2}$$

This can be verified by plugging the solution back into the recursion.

There is a second way to do this problem. To be in state 0 at time t, either we never moved from state 0, or we took a number of trips to state 1 and came back. Hence, the number of steps of transition between the two states has to be even. Note that no matter what state we are in, $(1-p)$ is the probability of changing to the other state, and p is the probability of staying in the same state. Hence, we need only the odd terms in $(p + (1-p))^t$, (i.e., all the terms where $(1-p)$ is raised to an even power). This allows us to derive the following equation:

$$P^t_{0,0} = \sum_{i=0}^{\lfloor (t+1)/2 \rfloor} B_{2i+1}(p, 1-p, t)$$

where $B_k(a,b,t) = \binom{t}{k-1} a^{t-k+1} b^{k-1}$ is the k^{th} term in the binomial expansion of $(a+b)^t$. This formula can be verified by calculating the $(0,0)^{th}$ element of the matrix P^t.

Problem 3. *(Exercise 7.3 from MU – 5 points)* Prove that the communicating relation defines an equivalence relation.

Solution:

1. Reflexive: by definition, $P^0 i, i = 1 > 0$ so $i \leftrightarrow i$.
2. Symmetric: if $i \leftrightarrow j$, then i and j are both accessible from each other, so $j \leftrightarrow i$.
3. Transitive: $i \leftrightarrow j$ and $j \leftrightarrow k$ implies that for some n, $P^n_{i,j} > 0$ and for some m, $P^m_{j,k} > 0$. Thus $P^{m+n}_{i,k} \geq P^n_{i,j} P^m_{j,k} > 0$ and so $i \leftrightarrow k$.

Problem 4. *(Exercise 7.6 from MU – 5 points)* In studying the 2-SAT algorithm, we considered a 1-dimensional random walk with a completely reflecting boundary at 0. That is, whenever position 0 is reached, with probability 1 the walk moves to position 1 at the next step. Consider now a random walk with a partially reflecting boundary at 0. Whenever position 0 is reached, with probability 1/2 the walk moves to position 1 and with probability 1/2 the walk stays at 0. Everywhere else the random walk moves either up or down 1, each with probability 1/2. Find the expected number of moves to reach n, starting from position i and using a random walk with a partially reflecting boundary.

Solution: Let h_i denote the expected hitting time to n from position i. We can write down the following recurrence equations:

$$\begin{aligned}
h_0 &= \frac{1}{2}h_0 + \frac{1}{2}h_1 + 1 \\
h_1 &= \frac{1}{2}h_0 + \frac{1}{2}h_2 + 1 \\
h_2 &= \frac{1}{2}h_1 + \frac{1}{2}h_3 + 1 \\
&\cdots \\
h_{n-1} &= \frac{1}{2}h_{n-2} + \frac{1}{2}h_n + 1 \\
h_n &= 0
\end{aligned}$$

From these equations, we can derive that $h_i = h_{i+1} + 2(i+1)$. Plugging in the boundary condition of $h_n = 0$, we get

$$h_i = (n+i+1)(n-i)$$

Problem 5. *(7 points)* A property of states in a Markov chain is called a *class property* if, whenever states i and j communicate, (*i.e.* each is reachable from the other), either both states have the property or neither do. Show that being periodic is a class property.

Solution:

Suppose i has period Δ. Since i and j communicate, there must be a path from i to j (call this P) and from j to i (call this Q) such that the length of the loop PQ is a multiple of Δ. Given any loop R from j back to j, we know the length of PRQ is also a multiple of Δ so R must have length a multiple of Δ too. Thus j must be periodic with period Δ' which is a multiple of Δ.

The argument works the same with i and j exchanged, so Δ must also be a multiple of Δ'. Thus $\Delta = \Delta'$.