COMP6730 Homework 3

Chris Winsor

11/7/2021

Problem 1a:

Consider an instance of SAT with m clauses where every clause has exactly k literals. Prove there exists an assignment to the variables s.t. it satisfies at least $m(1-2^k)$ clauses.

Given a set of m clauses let k_i be the number of literals in the ith clause.

If we choose the value of each literal randomly the probability of a clause being satisfied is $1-2^{-k_i}$.

The expected number of satisfied clauses is $\sum_{i=1}^{m}(1-2^{-k_i})$

If $k = \min(k_i)$ we have

$$\sum_{i=1}^{m}(1-2^{-k_i}) > m(1-2^{-k})$$

Since the probability of satisfying $m(1-2^k)$ clauses is strictly greater than zero, we conclude there exists at least one such assignment.

Problem 1b:

Give a derandomization of a simple randomized algorithm using conditional expectations and show it achieves the lower bound.

Suppose we have established the first k clauses, with the remaining clauses established randomly. An algorithm that is locally greedy in choosing clause $k+1$ is expressed as:

$$E[C(\,) \mid x1, \ldots x_k] \leq E[C(\,) \mid x_1, \ldots x_k, x_{k+1}]$$

Problem 2a:

Prove that for every integer n there exists a coloring of the edges of the complete graph K_n by two colors s.t. the total monochromatic copies of K_4 is at most $\binom{n}{4}2^{-5}$.

Let K_n be a complete graph.

There are $\binom{n}{2}$ edges and $2^{\binom{n}{2}}$ colorings.

If we color each edge randomly the probability of a particular coloring is $2^{-\binom{n}{2}}$.

Consider all 4-vertex cliques of K_n. There are $\binom{n}{4}$ such cliques.

For each $i = 1..\binom{n}{4}$ let A_i be the event that clique i is monochromatic.

Given that the first edge can be either color but the remaining edges must follow, we have

$$\Pr(A_i) = 2^{-(\binom{4}{2}-1)} = 2^{-5}$$

The Union Bound says:

$$The\ probability\ of\ any\ event <= The\ sum\ of\ probabilities\ of\ each\ event$$

$$\Pr\left(\bigcup_{i=1}^{\binom{n}{4}} A_i\right) <= \sum_{i=1}^{\binom{n}{4}} \frac{1}{8} = \binom{n}{4} 2^{-5} < 1$$

and from that

$$\Pr\left(\bigcap_{i=1}^{\binom{n}{4}} A_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{\binom{n}{4}} A_i\right) > 0$$

Since the probability of choosing a 4-node clique that is monochromatic is strictly greater than zero, we conclude there exists at least one 4-node clique that is monochromatic.

Problem 2b:

Give a randomized algorithm ... that runs in time polynomial in n.

To achieve polynomial time construction, we first need a polynomial time sampling algorithm. In this case we simply randomly choose color for each node in $O(n)$.

We also need to establish the number of samples. If p is the probability of generating a sample that satisfies our 4-vertex clique requirement, the number of samples required is a geometric random variable the expectation of which is $\frac{1}{p}$. Thus we need $\frac{1}{p}$ in polynomial time. The number of 4-cliques is $\binom{n}{4}$ and probability that each is monochromatic is 2^{-5} thus:

$$p = \frac{\binom{n}{4} 2^{-5}}{\binom{n}{2}} = 12(n-4)(n-3)\ \ which\ is\ polynomial\ time\ O(n^2)$$

Problem 2c:

Show how to construct... using conditional expectations.

(Reference Section 6.3 in Mitzenmacher/Upfal)

Imagine we have $\binom{n}{k}$ cliques $A_1, \ldots A_{\binom{n}{k}}$. If we randomly assign vertices to cliques then the average clique size is $\frac{|V|}{\binom{n}{k}}$ and the expected clique size is denoted:

$$E[CS\left(A_1, \ldots A_{\binom{n}{k}}\right)$$

If instead we choose to deterministically assign the first m vertices, with the remaining assigned randomly then the expected clique size of such assignment is:

$$E[CS\left(A_1, \ldots A_{\binom{n}{k}} \mid x_1, \ldots x_m\right)$$

We wish to establish an algorithm such that our deterministic assignments result in expected clique size no larger than that of random assignment. In other words:

$$E[CS\left(A_1, \ldots A_{\binom{n}{k}} \mid x_1, \ldots x_m\right) \leq E[CS\left(A_1, \ldots A_{\binom{n}{k}}\right)$$

Consider a single assignment Y_{m+1} and place it randomly. We know:

$$E[CS\left(A_1, \ldots A_{\binom{n}{k}} \mid x_1, \ldots x_m\right) = \frac{1}{\binom{n}{k}} \sum_1^{\binom{n}{k}} E[CS\left(A_1, \ldots A_{\binom{n}{k}} \mid x_1, \ldots x_m, Y_{m+1}\right)$$

(above is just summing across the conditioned variable)

It follows that

$$E[CS\left(A_1, \ldots A_{\binom{n}{k}} \mid x_1, \ldots x_m\right) \leq \min_{Y \in A}(E[CS\left(A_1, \ldots A_{\binom{n}{k}} \mid x_1, \ldots x_m, Y_{m+1}\right))$$

In other words - if we assign $Y_{m_1}$to the minimum sized clique then the resulting expected clique size will be no greater than it was before.

Above gives us a deterministic way to assign vertices such that expected clique size no larger than that of random assignment. Specifically, our algorithm is to assign the node to the clique having fewest vertices.

Problem 3a

Show that each $S(\sigma)$ is an independent set in G.

We use induction: Given a permutation σ of the vertices, and subset $S(\sigma)$ of k vertices that are an independent set we shall call "CurrentSet"

$$CurrentSet = \{v_1, \ldots v_k\}$$

If $v_{k+1}$is not a neighbor of any node in $CurentSet$ then

$$NewSet = \{CurrentSet, v_{k+1}\}$$

is an independent set.

The basis: $\{v_1\}$, the set consisting of just the first node in the permutation, is an independent set.

Problem 3b

Suggest a natural randomized algorithm to product σ for which ... the expected cardinality of $S(\sigma) = \sum_{i=1}^{n} \frac{1}{d_i+1}$ where d_i is the degree of vertex i.

Our goal is an algorithm to produce σ (permutation on S) s.t. $|S(\sigma)|$ meets the criterion above.

We use Sample and Modify (Mitzenmacher pg. 142)

The first step is to select with probability p elements from S to form a permutation σ. This permutation does not meet our criterion but has the characteristic:

$$E[X] = |S(\sigma)| = \frac{1}{p}$$

Above is considering the selection process as a geometric series w/ probability p and thus the expected number of elements $E[X] = 1/p$.

The second step is to sequentially consider each element v_i in σ and remove it if there is an element j where $j < i$. Another way to view this is by creating a new σ' and only including elements from σ if $Neighbors(v_i) \notin \sigma'$. At each step the probability of including v_i is the degree of the node d_i over the number of nodes n. Thus, for a particular node being considered the increase in the expected number nodes is:

$$E[Y_i] = \frac{d_i}{n}$$

and over all nodes in σ'

$$E[Y] = \frac{1}{n} \sum_{i \in \sigma} d_i$$

When the algorithm terminates, we are left with $S[\sigma']$ having cardinality

$$| S[\sigma'] | = E[Y] - E[X]$$

We have thus shown a randomized algorithm to generate a permutation on S having an expected cardinality that is dependent only on number of nodes and degree of each node.

Problem 4a

Give a winning strategy for chooser when $k \geq 2^n$

On round 1 the chooser sets $A = 2^{n-1}$and $B = 2^{n-1}$. Tokens not in A or B are left behind.

For round 2 and beyond the chooser is left with 2^{n-r} tokens which they will evenly split. Thus, at round $r = n$ there are $2^0 = 1$ token (a winning strategy).

Problem 4b

Use probabilistic method to show winning strategy exists for remover when $k < 2^n$.

If tokens are randomly chosen for removal, the probability of a token remaining at round r is 2^{-r}, so the expected number of tokens at round r is

$$E(t_n) = k\ 2^{-r}$$

$$E(t_n) < 2^n\ 2^{-r}$$

and where r = n:

$$E(t_n) < 2^n\ 2^{-n} = 1$$

Since the expected value is strictly less than one there must exist a scenario where the remover wins.

Problem 4c:

Explain how to use conditional expectations to derandomize the winning strategy for remover when $k < 2^n$.

If $D_x$is the number of tokens removed on round x, the expected number of rounds is denoted

$$E[R(D_1, \dots D_n)]$$

If we consider as given an initial set of m rounds, with the remaining rounds random then we have

$$E[R(D_1, \dots D_n \mid r_1, \dots r_m)\ ]$$

Given the constraint on the chooser that $A + B + C = \ number\ of\ tokens$, where C is to be left behind and the remover may remove A or B, then we conclude at least one of A or B must be less than $\frac{numberOfTokens}{2}$. If the policy of the remover is to remove the larger of A or B then

$$E[R(D1, \dots\ D_n|r_1, \dots r_m, r_{m+1})] \leq E[R(D_1, \dots D_n \mid r_1, \dots r_m)]$$

In other words - if the remover removes the larger of A or B then the expected number of rounds does not increase.

Problem 5a

Prove that in any network of $|E|$ *edges there is a way to assign the signs of the edges s.t. the number of "+" edges is no more than* $\frac{|E|}{2}$.

Pick an arbitrary node v_1 and perform BFS. The depth of the resulting tree cannot exceed the diameter of the network, and $diameter \leq \frac{|nodes|}{2}$. Thus, if we assign (+) to edges represented transitions in the BFS tree that go from even to odd levels, and (-) to edges represented as odd-to-even level transitions of the BFS tree then there will be no more than $\frac{n}{2}$ positive edges.

Problem 5b

An algorithm to assign (+) and (-) to edges follows naturally from above. Starting with arbitrary node v_1 assign (+) to each edge. Performing BFS from that point we alternate between (-) and (+) to edges discovered at each level of the BFS. The algorithm achieves (+) edges no more than $\frac{|E|}{2}$.