CSC2420: Algorithm Design, Analysis and Theory **Fall 2017**

Lecture 7: Randomized Algorithms

Lecturer: Allan Borodin *Scribe: Lily Li*

7.1 Continuation of Randomized Algorithms

Recall exact max $-k$ SAT. Johnson's algorithm gives a deterministic algorithm for weighted max SAT. Later on Yannakakis gives the naive randomized algorithm and the analysis of Johnson's algorithm to be a 1/2-approximation with an upper bound of 2/3. Poloczek and Schnithger gets a 3/4-approximation using a randomized algorithm called SLACK. The key is to make sure the unit clauses are not causing you too much trouble by weighing them accordingly.

There are some limitations of deterministic simply max-SAT algorithms. Namely we want to ask what sort of model the input are coming in as. Ideally we want to be able to use all the information about a literal (what clause it is in, what it is called etc) to get a better approximation.

Another issue regarding this randomized to deterministic algorithm, is the way we choose to define (or really not define) derandomization. Namely there is no formal method to say that a deterministic algorithm is a derandomized version of a randomized algorithm. One possible approach is to take the probability tree in the randomized case, prune the tree to polynomial size, and take the best value on the leaves of the resulting tree. Consider the following IP/LP formulation of max-SAT.

Example 7.1 *Sub-modular maximization problems. A set function $f : 2^U \to \Re$ is* ***sub-modular*** *if*

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$$

Note, if the function is linear, then equality holds. Equivalently, f is sub-modular if it satisfies decreasing marginal gains. Assume that f is ***normalized*** *($f(\emptyset) = 0$), non-negative, and monotone.*

The goal is to compute S to maximize $f(S)$ subject to some constraint (you don't want to just take everything).

The case is much more interesting if your function is not monotone. Here you don't even need constraints. The canonical problem is max-*CUT (also* max-*DI-CUT with directed graphs). Given a graph, partition the vertices to obtain the largest cut.*

There is a 2007 result by Feige, Mirrokni, and Vondrak which give several approximations for the max-CUT problem. In practical they upper bound any algorithm with only value oracle calls to be $(1/2 + \epsilon)$ for any $\epsilon > 0$. Suffices to say, none of their algorithms were that good and they were pretty complicated.

Then Buchbinder in 2012 came up with a deterministic 1/3-approximation which was then extended to a 1/2-approximation. The "two sided" idea is quite nice. You start with *both* the empty set X and the entire universe Y. When you look at each element you decide to keep it or get rid of it depending on both removing it from Y and adding it to X. This algorithm can be extended to the max-SAT problem to achieve an 3/4-approximation with randomization.

There is this thing called the *method of conditional expectations* that is used in the second pass of a deterministic on-line algorithm.

7.2 De-randomization

Buchbinder and Feldman had a neat derandomization idea which only uses one pass and solves max-SAT.

7.3 Vertex and Edge Weighted On-line Bipartite Matching

In the on-line bipartite matching problem, bipartite graph G with nodes $U \cup V$. U is the on-line vertices and V is the off-line vertices. One way to solve this problem is via the Randing algorithm. The algorithm is as follows: choose a random permutation of the off-line nodes. When an on-line node $u \in U$ appears, math u to the highest ranked unmatched $v \in V$ such that $uv \in E$. The simpler technique of randomly matching u obtains only a 1/2-approximation. The described algorithm is instead a $1 - 1/e$-approximation in the adversarial model.

There is a correspondence between on-line algorithms using t bits of advices and algorithms with 2^t on-line streams (then taking the best output).

Karande et al showed: any ROM approximation result implies the same result for an unknown i.i.d. model which implies the known i.i.d. model (can you show that this is the case)?

Mehta had a survey paper in 2013 with a nice table summarizing all known and conjectured results.

The secretary problem is actually an instance of the on-line bipartite matching problem where there is only one off-line node (the position) and the set of n secretaries coming in on-line.