### CSC2420: Algorithm Design, Analysis and Theory Spring 2019

Allan Borodin

February 28, 2019

Announcements:

- Some comments about the grading scheme.
  - One more standard type of assignment.
  - 2 A critical review of a topic.
- A question from last class re a natural problem in BPP-RP

### Todays ambitious agenda

Todays agenda Some proofs on the board

- Randomized algorithms with applications to max-sat, unconstrained non-monotone submoodular maximization, bipartite matching.
- The Yannakakis IP/LP ramdomized rounding for max-sat.
- The Buchbinder et al deterministic and randomized "double-sided" online algorithms for maximizing an unconstrained non-monotone submodular function.
- De-randomizing the Buchbinder et al randomized algorithm into a parallel stream of deterministic online algorithms.
- Extension to submodular max-sat
- The Poloczek et al presentation of the Buchbinder et al, and van Zuylen randomized max-sat algorithm and its derandomization to a two pass "online" algorithm.
- A quick overview of online bipartite matching and extensions
- The KVV Ranking algorithm for unweighted online bipartite matching.
- Looking ahead to other online models

# Yannakakis' IP/LP randomized rounding algorithm for Max-Sat

- We will formulate the weighted Max-Sat problem as a  $\{0,1\}$  IP.
- Relaxing the variables to be in [0, 1], we will treat some of these variables as probabilities and then round these variables to 1 with that probability.
- Let F be a CNF formula with n variables  $\{x_i\}$  and m clauses  $\{C_j\}$ . The Max-Sat formulation is : maximize  $\sum_j w_j z_j$ subject to  $\sum_{\{x_i \text{ is in } C_j\}} y_i + \sum_{\{\bar{x}_i \text{ is in } C_j\}} (1 - y_i) \ge z_j$  $y_i \in \{0, 1\}; z_i \in \{0, 1\}$
- The y<sub>i</sub> variables correspond to the propositional variables and the z<sub>j</sub> correspond to clauses.
- The relaxation to an LP is  $y_i \ge 0$ ;  $z_j \in [0, 1]$ . Note that here we cannot simply say  $z_j \ge 0$ .

#### Randomized rounding of the y<sub>i</sub> variables

- Let  $\{y_i^*\}, \{z_i^*\}$  be the optimal LP solution,
- Set  $\tilde{y}_i = 1$  with probability  $y_i^*$ .

#### Theorem

Let  $C_j$  be a clause with k literals and let  $b_k = 1 - (1 - \frac{1}{k})^k$ . Then  $Prob[C_j \text{ is satisifed }]$  is at least  $b_k z_j^*$ .

- The theorem shows that the contribution of the *j*<sup>th</sup> clause *C<sub>j</sub>* to the expected value of the rounded solution is at least *b<sub>k</sub>w<sub>j</sub>*.
- Note that  $b_k$  converges to (and is always greater than)  $1 \frac{1}{e}$  as k increases. It follows that the expected value of the rounded solution is at least  $(1 \frac{1}{e})$  LP-OPT  $\approx .632$  LP-OPT.
- Taking the max of this IP/LP and the naive randomized algorithm results in a  $\frac{3}{4}$  approximation algorithm that can be derandomized. (The derandomized algorithm will still be solving LPs.)

# Unconstrained (non monotone) submodular maximization

- Feige, Mirrokni and Vondrak [2007] began the study of approximation algorithms for the unconstrained non monotone submodular maximization (USM) problem establishing several results:
  - **(**) Choosing S uniformly at random provides a 1/4 approximation.
  - 2 An oblivious local search algorithm results in a 1/3 approximation.
  - **a** A non-oblivious local search algorithm results in a 2/5 approximation.
  - Any algorithm using only value oracle calls, must use an exponential number of calls to achieve an approximation (1/2 + ϵ) for any ϵ > 0.
- The Feige et al paper was followed up by improved local search algorithms by Gharan and Vondrak [2011] and Feldman et al [2012] yielding (respectively) approximation ratios of .41 and .42.
- The  $(1/2 + \epsilon)$  inapproximation (assuming an exponental number of value oracle calls), was augmented by Dobzinski and Vondrak showing the same bound for an explicitly given instance under the assumption that  $RP \neq NP$ .

# The Buchbinder et al (1/3) and (1/2) approximations for USM

In the FOCS [2012] conference, Buchbinder et al gave an elegant linear time deterministic 1/3 approximation and then extend that to a randomized 1/2 approximization. The conceptually simple form of the algorithm is (to me) as interesting as the optimality (subject to the proven inapproximation results) of the result. Let  $U = u_1, \ldots u_n$  be the elements of U in any order.

#### The deterministic 1/3 approximation for USM

$$\begin{array}{l} X_{0} := \varnothing; Y_{0} := U \\ \text{For } i := 1 \dots n \\ a_{i} := f(X_{i-1} \cup \{u_{i}\}) - f(X_{i-1}); \ b_{i} := f(Y_{i-1} \setminus \{u_{i}\}) - f(Y_{i-1}) \\ \text{If } a_{i} \ge b_{i} \\ \text{then } X_{i} := X_{i-1} \cup \{u_{i}\}; Y_{i} := Y_{i-1} \\ \text{else } X_{i} := X_{i-1}; Y_{i} := Y_{i-1} \setminus \{u_{i}\} \\ \text{End If} \\ \text{End For} \end{array}$$

#### The randomized 1/2 approximation for USM

- Buchbinder et al show that the "natural randomization" of the previous deterministic algorithm achieves approximation ratio 1/2.
- That is, the algorithm chooses to either add  $\{u_i\}$  to  $X_{i-1}$  with probability  $\frac{a'_i}{a'_i+b'_i}$  or to delete  $\{u_i\}$  from  $Y_{i-1}$  with probability  $\frac{b'_i}{a'_i+b'_i}$  where  $a'_i = \max\{a_i, 0\}$  and  $b'_i = \max\{b_i, 0\}$ .
- If  $a_i = b_i = 0$  then add  $\{u_i\}$  to  $X_{i-1}$ .
- Note: Part of the proof for both the deterministic and randomized algorithms is the fact that a<sub>i</sub> + b<sub>i</sub> ≥ 0.
- This fact leads to the main lemma for the deterministic case:

 $f(OPT_{i-1} - f(OPT_i) \le [f(X_i - f(X_{i-1}] + [f(Y_i) - f(Y_{i-1}]])]$ Here  $OPT_i = (OPT \cup \{X_i\}) \cap Y_i$  so that  $OPT_i$  coincides with  $X_i$  and  $Y_i$  for elements  $1, \ldots, i$  and coincides with OPT on elements  $i + 1, \ldots, n$ . Note that  $OPT_0 = OPT$  and  $OPT_n = X_n = Y_n$ . That is, the loss in OPTs value is bounded by the total value increase in the algorithm's solutions.

### Applying the algorithmic idea to Max-Sat

Buchbinder et al are able to adapt their randomized algorithm to the Max-Sat problem (and even to the Submodular Max-Sat problem). So assume we have a monotone normalized submodular function f (or just a linear function as in the usual Max-Sat). The adaption to Submodular Max-Sat is as follows:

- Let φ : X → {0} ∪ {1} ∪ Ø be a standard partial truth assignment. That is, each variable is assigned exactly one of two truth values or not assigned.
- Let C be the set of clauses in formula Ψ. Then the goal is to maximize f(C(φ)) where C(φ) is the sat of formulas satisfied by φ.
- An extended assignment is a function φ': X → 2<sup>{0,1}</sup>. That is, each variable can be given one, two or no values. (Equivalently φ' ⊆ X × {0,1} is a relation.) A clause can then be satisfied if it contains a positive literal (resp. negative literal) and the corresponding variable has value {1} or {0,1} (resp. has value {0} or {0,1}.
- $g(\phi') = f(\mathcal{C}(\phi'))$  is a monotone normalized submodular function. '

#### Buchbinder et al Submodular Max-Sat

Now starting with  $X_0 = X \times \emptyset$  and  $Y_0 = Y \times \{0, 1\}$ , each variable is considered and set to either 0 or to 1 (i.e. a standard assignment of precisely one truth value) depending on the marginals as in USM problem.

_	
Algorithm 3: RandomizedSSAT $(f, \Psi)$	
1	$X_0 \leftarrow \emptyset, Y_0 \leftarrow \mathcal{N} \times \{0, 1\}.$
2	for $i = 1$ to $n$ do
3	$a_{i,0} \leftarrow g(X_{i-1} \cup \{u_i, 0\}) - g(X_{i-1}).$
4	$a_{i,1} \leftarrow g(X_{i-1} \cup \{u_i, 1\}) - g(X_{i-1}).$
5	$b_{i,0} \leftarrow g(Y_{i-1} \setminus \{u_i, 0\}) - g(Y_{i-1}).$
6	$b_{i,1} \leftarrow g(Y_{i-1} \setminus \{u_i, 1\}) - g(Y_{i-1}).$
7	$s_{i,0} \leftarrow \max\{a_{i,0} + b_{i,1}, 0\}.$
8	$s_{i,1} \leftarrow \max\{a_{i,1} + b_{i,0}, 0\}.$
9	with probability $s_{i,0}/(s_{i,0}+s_{i,1})^*$ do:
	$X_i \leftarrow X_{i-1} \cup \{u_i, 0\}, Y_i \leftarrow Y_{i-1} \setminus \{u_i, 1\}.$
10	else (with the compliment probability
	$s_{i,1}/(s_{i,0}+s_{i,1}))$ do:
11	$ X_i \leftarrow X_{i-1} \cup \{u_i, 1\}, Y_i \leftarrow Y_{i-1} \setminus \{u_i, 0\}. $
12	<b>return</b> $X_n$ (or equivalently $Y_n$ ).
	* If $s_{i,0} = s_{i,1} = 0$ , we assume $s_{i,0}/(s_{i,0} + s_{i,1}) = 1$ .

### Further discussion of the Unconstrained Submodular Maximization and Submodular Max-Sat algorithms

- The Buchbinder et al [2012] online randomized 1/2 approximation algorithm for Unconstrained Submodular Maximization (USM) cannot be derandomized into a "similar" deterministic online or priority style algorithm by a result of Huang and Borodin [2014]. Like the Poloczek result, we claimed that this was "in some sense" evidence that this algorithm cannot be derandomized.
- Their algorithm is shown to have a  $\frac{3}{4}$  approximation ratio for Monotone Submodular Max-Sat.
- Poloczek et al [2017] show that the Buchbinder et al algorithm turns out to be equivalent to a previous Max-Sat algorithm by van Zuylen.

# The randomized (weighted) max-sat $\frac{3}{4}$ approximation algorithm

The idea of the algorithm is that in setting the variables, we want to balance the weight of clauses satisfied with that of the weight of clauses that are no longer satisfiable.

Let  $S_i$  be the assignment to the first *i* variables and let  $SAT_i$  (resp.  $UNSAT_i$ ) be the weight of satisfied clauses (resp., clauses no longer satisfiable) with respect to  $S_i$ . Let  $B_i = \frac{1}{2}(SAT_i + W - UNSAT_i)$  where W is the total weight of all clauses.

# The randomized (weighted) max-sat $\frac{3}{4}$ approximation algorithm

The idea of the algorithm is that in setting the variables, we want to balance the weight of clauses satisfied with that of the weight of clauses that are no longer satisfiable.

Let  $S_i$  be the assignment to the first *i* variables and let  $SAT_i$  (resp.  $UNSAT_i$ ) be the weight of satisfied clauses (resp., clauses no longer satisfiable) with respect to  $S_i$ . Let  $B_i = \frac{1}{2}(SAT_i + W - UNSAT_i)$  where W is the total weight of all clauses.

The algorithm's plan is to randomly set variable  $x_i$  so as to increase  $\mathbb{E}[B_i - B_{i-1}]$ .

# The randomized (weighted) max-sat $\frac{3}{4}$ approximation algorithm

The idea of the algorithm is that in setting the variables, we want to balance the weight of clauses satisfied with that of the weight of clauses that are no longer satisfiable.

Let  $S_i$  be the assignment to the first *i* variables and let  $SAT_i$  (resp.  $UNSAT_i$ ) be the weight of satisfied clauses (resp., clauses no longer satisfiable) with respect to  $S_i$ . Let  $B_i = \frac{1}{2}(SAT_i + W - UNSAT_i)$  where W is the total weight of all clauses.

The algorithm's plan is to randomly set variable  $x_i$  so as to increase  $\mathbb{E}[B_i - B_{i-1}]$ .

To that end, let  $t_i$  (resp.  $f_i$ ) be the value of  $w(B_i) - w(B_{i-1})$  when  $x_i$  is set to true (resp. false).

### The randomized max-sat approximation algorithm continued

```
For i = 1 \dots n

If f_i \leq 0, then set x_i = \text{true}

Else if t_i \leq 0,

then set x_i = \text{false}

Else set x_i true with probability \frac{t_i}{t_i + f_i}.

End For
```

## The randomized max-sat approximation algorithm continued

```
For i = 1 \dots n

If f_i \le 0, then set x_i = \text{true}

Else if t_i \le 0,

then set x_i = \text{false}

Else set x_i true with probability \frac{t_i}{t_i + f_i}.

End For
```

Consider an optimal solution (even an LP optimal)  $\mathbf{x}^*$  and let  $OPT_i$  be the assignment in which the first *i* variables are as in  $S_i$  and the remiaing n - i variables are set as in  $\mathbf{x}^*$ . (Note:  $x^*$  is not calculated.)

The analysis follows as in Poloczek and Schnitger, Poloczek, and explicitly in Buchbinder et al. One shows the following:

• 
$$t_i + f_i \geq 0$$

•  $\mathbb{E}[w(OPT_{i-1}) - w(OPT_i)] \le \mathbb{E}[w(B_i) - w(B_{i-1})]$ 

We ended the lecture for week 7 at slide 13. I am including the rest of my slides that I had intended to discuss. Chris will talk about some of this material regarding online bipartite matching. I have also posted preliminary versions of some chapters of the new online text (with Denis Pankratov) on the web page where more details can be found.

## The Buchbinder and Feldman derandomization of the USM algorithm

- Contrary to the Poloczek, (resp. Huang and B.) priority inapproximations for Max-Sat (resp. USM), there is another sense in which these algorithms can be derandomized.
- In fact the derandomization becomes an "online algorithm" in the sense that an adversary is choosing the order of the input variables. However rather than creating a single solution, the algorithm is creating a tree of solutions and then taking the best of these.
- The idea is as follows. The analysis of the randomized USM approximation bound shows that a certain linear inequality holds at each iteration of the algorithm. Namely,

$$E[f(OPT_{i-1} - f(OPT_i)] \le \frac{1}{2}E[f(X_i) - f(X_{i-1}) + f(Y_i) - f(Y_{i-1})]$$

That is, the expected change in restricting OPT in an iteration (by setting the  $i^{th}$  variable) is bounded by the average change in the two values being maintained by the algorithm.

# Continuing the Buchbinder and Feldman derandomization idea

- These inequalities induce two additional inequalties per iteration on the distributions of solutions that can exist at each iteration.
- This then gets used to describe an LP corresponding to these 2*i* constraints we have for the distributions that hold at each iteration of the algorithm.
- But then using LP theory again (i.e. the number of non-zero variables in a basic solution). It follows that we only need distributions with support 2*i* at each iteration rather than the naive 2<sup>*i*</sup> that would follow from just considering the randomized tree.
- Finally, since there must be at least one distribution (amongst the final 2*n* distributions) for which the corresponding solution is at least as good as the expected value. Thus if suffices to take the max over a "small" number of solutions.

### Randomized online bipartite matching and the adwords problem.

- We return to online algorithms and algorithms in the random order model (ROM). We have already seen evidence of the power of randomization in the context of the USM and MaxSat problems.
- Another nice sequence of results begins with a randomized online algorithm for bipartite matching due to Karp, Vazirani and Vazirani [1990]. We quickly overview some results in this area as it represents a topic of continuing interest. (The FOCS 2012 conference had a session of three papers related to this topic.)
- In the online bipartite matching problem, we have a bipartite graph G with nodes U ∪ V. Nodes in U enter online revealing all their edges. A deterministic greedy matching produces a maximal matching and hence a <sup>1</sup>/<sub>2</sub> approximation.
- It is easy to see that any deterministic online algorithm cannot be better than a <sup>1</sup>/<sub>2</sub> approximation even when the degree of every u ∈ U is at most (equal) 2

### The randomized ranking algorithm

- The algorithm chooses a random permutation of the nodes in V and then when a node u ∈ U appears, it matches u to the highest ranked unmatched v ∈ V such that (u, v) is an edge (if such a v exists).
- Aside: making a random choice for each u is still only a  $\frac{1}{2}$  approx.
- Equivalently, this algorithm can be viewed as a deterministic greedy (i.e. always matching when possible and breaking ties consistently) algorithm in the ROM model.
- That is, let  $\{v_1, \ldots, v_n\}$  be any fixed ordering of the vertices and let the nodes in U enter randomly, then match each u to the first unmatched  $v \in V$  according to the fixed order.
- To argue this, consider fixed orderings of U and V; the claim is that the matching will be the same whether U or V is entering online.

### The KVV result and recent progress

#### **KVV** Theorem

Ranking provides a (1 - 1/e) approximation.

- Original analysis is not rigorous. There is an alternative proof (and extension) by Goel and Mehta [2008], and then another proof in Birnbaum and Mathieu [2008]. Other alternative proofs have followed.
- Recall that this positive result can be stated either as the bound for a particular deterministic algorithm in the stochastic ROM model, or as the randomized Ranking algorithm in the (adversarial) online model.
- KVV show that the (1 1/e) bound is essentially tight for any randomized online (i.e. adversarial input) algorithm. In the ROM model, Goel and Mehta state inapproximation bounds of  $\frac{3}{4}$  (for deterministic) and  $\frac{5}{6}$  (for randomized) algorithms.
- In the ROM model, Karande, Mehta, Tripathi [2011] show that Ranking achieves approximation at least .653 (beating 1 - 1/e) and no better than .727. This ratio was improved to .696 by Mahdian and Yan [2011]].

#### And some more recent progress

- Karande et al show that any ROM approximation result implies the same result for the unknown i.i.d. model.
- Manshadi et al give a .823 inapproximation for biparitie matching in the known i.i.d. distribution model. This implies the same inapproximation in the unknown i.i.d. and ROM models improving the <sup>5</sup>/<sub>6</sub> inapproximation of Goel and Mehta.
- There is a large landscape (and continuing research) of weighted versions of online bipartite matching such as the *adwords* problem and the *display ads* problem that are motivated by applications to online advertising.
- Although slightly out of data, the survey by Mehta [2013] is an excellent reference. Note: The table in the survey identifies the ROM and unknown i.i.d. model and although there are no *provable* separations of these models, there is evidence that the ROM model is a more general model (i.e. an i.i.d. result does not necessarily imply a ROM result).

### Getting past the (1-1/e) bound

- The ROM model can be considered as an example of what is called stochastic optimization in the OR literature. As we have discussed early in the term, there are other stochastic optimization models that are perhaps more natural, namely i.i.d sampling from known and unknown distributionsi and Markov distributions.
- Feldman et al [2009] study the known distribution case and show a randomized algorithm that first computes an optimal offline solution (in terms of expectation) and uses that to guide an online allocation.
- They achieve a .67 approximation (improved to .699 by Bahmani and Kapralov [2010] and also show that no online algorithm can achieve better than  $26/27 \approx .99$  (improved to .902).
- Karande, et al [2011] show that an approximation in the ROM model implies thre same approximation in the unknown distribution model. They show that the KVV Ranking algorithm achieves approximation .653 in the ROM model and is no better than .727.

#### Weighted extensions of online bipartite matching

As mentioned, there are various weighted versions of online bipartite matching motivated by online auction advertising.

- Weighted online matching
- Adwords with small and large (compared to the budget) bids.
- The display ads problem with and without free disposal.
- The adwords problem with small bids is equivalent to the display ads problem with lage capcities. Both of these problems are generalized by the submodular welfare maximization problem.
- Online algorithms with Reassignments

# The adwords problem: an extension of bipartite matching

- In the (single slot) adwords problem, the nodes in U are queries and the nodes in V are advertisers. For each query q and advertiser i, there is a bid b<sub>q,i</sub> representing the value of this query to the advertiser.
- Each advertiser also usually has a hard budget  $B_i$  which cannot be exceeded. The goal is to match the nodes in U to V so as to maximize the sum of the accepted bids without exceeding any budgets. Without budgets and when each advertiser will pay for at most one query, the problem then is edge weighted bipartite matching.
- In the online case, when a query arrives, all the relevant bids are revealed.

#### Some results for the adwords problem

- Here we are just considering the combinatorial problem and ignoring game theoretic aspects of the problem.
- The problem has been studied for the special (but well motivated case) that all bids are small relative to the budgets. As such this problem is incomparable to the matching problem where all bids are in {0,1} and all budgets are 1.
- For this small bid case, Mehta et al [2005) provide a deterministic online algorithm achieving the 1 1/e bound and show that this is optimal for all randomized online algorithms (i.e. adversarial input).

#### Greedy for a class of adwords problems

- Goel and Mehta [2008] define a class of adwords problems which include the case of small budgets, bipartite matching and *b*-matching (i.e. when all budgets are equal to some *b* and all bids are equal to 1).
- For this class of problems, they show that a deterministic greedy algorithm achieves the familiar 1 1/e bound in the ROM model. Namely, the algorithm assigns each query (.e. node in U) to the advertiser who values it most (truncating bids to keep them within budget and consistently breaking ties). Recall that Ranking can be viewed as greedy (with consistent tie breaking) in the ROM model.

#### Vertex weighted bipartite matching

- Aggarwal et al [2011] consider a vertex weighted version of the online bipartite matching problem. Namely, the vertices  $v \in V$  all have a known weight  $w_v$  and the goal is now to maximize the weighted sum of matched vertices in V when again vertices in U arrive online.
- This problem can be shown to subsume the adwords problem when all bids b<sub>q,i</sub> = b<sub>i</sub> from an advertiser are the same.
- It is easy to see that Ranking can be arbitrarily bad when there are arbitrary differences in the weight. Greedy (taking the maximum weight match) can be good in such cases. Can two such algorithms be somehow combined? Surprisingly, Aggarwal et al are able to achieve the same 1-1/e bound for this class of vertex weighted bipartite matching.

#### The vertex weighted online algorithm

#### The perturbed greedy algorithm

For each  $v \in V$ , pick  $r_v$  randomly in [0,1]Let  $f(x) = 1 - e^{1-x}$ When  $u \in U$  arrives, match u to the unmatched v (if any) having the highest value of  $w_v * f(x_v)$ . Break ties consistently.

In the unweighted case when all  $w_v$  are identical this is the Ranking algorithm.

# Some concluding remarks on max-sat and bipartite matching

- The ROM model subsumes the stochastic model where inputs are chosen i.i.d. from an unknown distribution (which in turn subsumes i.i.d. inputs from a known distribution). Why? Hence a positive result in the ROM model implies a positive result in the i.i.d. unknown distribution model.
- A research problem of interest is to see to what extent some form of an extended online or priority framework can yield a deterministic online bipartite matching algorithm with approximation ratio better than 1/2.
- As mentioned before, Pena can show that a 3/4 max-sat approximation can be obtained by a deterministic "poly width" online algorithm.
- One can formulate the Buchbinder and Feldman method in the framework of the priority BT model of Alekhnovich et al. Can we show that a bounded width online (or priority) BT algorithm cannot obtain a 3/4 ratio?

# Online and priority width bounds for max-sat and bipartite matching

We have the following width inapproximation results.

- To improve upon the <sup>3</sup>/<sub>4</sub> approximation (using online width 2n) result, we need exponential width. More precisely, For any ε > 0 there exists δ > 0 such that, for k < e<sup>δn</sup>, no online width-cut-k algorithm can achieve an asymptotic approximation ratio of 3/4 + ε for unweighted exact max-2-sat with input model 2.
- For any ε > 0 there exists δ > 0 such that, for k < e<sup>δn</sup>, no PBR width-cut-k algorithm can achieve an asymptotic approximation ratio of 21/22 + ε for unweighted max-2-sat with input model 3.
- For any ε > 0, no bounded width online algorithm can achieve a <sup>1</sup>/<sub>2</sub> + ε approximation for bipartite matching.
- For any ε > 0, no priority algorithm can achieve a <sup>1</sup>/<sub>2</sub> + ε approximation for bipartite matching.