CSC2420 Fall 2012: Algorithm Design, Analysis and Theory

Allan Borodin

April 2, 2015; Lecture 12

Annoucements and Todays Agenda

- Announcements
  1. Assignment 3 is due Tuesday, April 7. I am not here next week so please send by email (if possible) and cc lalla@cs). If submitting hard copy, please contact Lalla to arrange where to drop off assignment.
  2. If you are an undergraduate planning to graduate this term, then please email me so that I can be sure that your assignments are graded first and a grade is calculated in time for you to graduate.
- Todays agenda
  1. The Miller-Rabin randomized primality test
  2. Monotone submodular maximization subject to matroid and independence constraints and the return of non-oblivious local search.
  3. The Lovasz Local lemma and the Moser-Tardos algorithm for finding a satisfying instance of an exact k-SAT formula in which every clause C shares a variable with at most $d < 2^k/e$ other clauses.
  4. Spectral methods

Some basic number theory we need for primality testing

- $Z_N^* = \{a \in Z_N : gcd(a, N) = 1\}$ is a (commutative) group under multiplication mod N.
- If N is prime, then
  1. For $a \neq 0(modN), a^{N-1} = 1(modN)$.
  2. Z_N^* is a cyclic group; that is there exists a generator g such that $\{g, g^2, g^3, \ldots, g^{N-1}\}$ (all mod N) is the set Z_N^*. This implies that $g^i \neq 1(modN)$ for any $1 \leq i < N - 1$.
  3. There are exactly two square roots of 1 in Z_N^*, namely 1 and -1.
- The Chinese Remainder Theorem: Whenever N_1 and N_2 are relatively prime (i.e. $gcd(N_1, N_2) = 1$), then for all $v_1 < N_1$ and $v_2 < N_2$, there exists a unique $w < N_1 \cdot N_2$ such that $v_1 = w \pmod{N_1}$ and $v_2 = w \pmod{N_2}$.

A simple but "not quite" correct algorithm

We also need two basic computational facts.

1. a^i mod N can be computed efficiently.
2. $gcd(a, b)$ can be efficiently computed.

The following is a simple algorithm that works except for an annoying set of nunbers called Carmichael numbers.

Simple algorithm ignoring Carmichael numbers

Choose $a \in Z_N$ uniformly at random.
If $gcd(a, N) \neq 1$, then Output Composite
If a^{N-1} mod $N \neq 1$, then Output Composite
Else Output Prime

When does the simple algorithm work?

- $S = \{a | gcd(a, N) = 1 \text{ and } a^{N-1} = 1\}$ is a subgroup of Z_N^*
- If there exists an $a \in Z_N^*$ such that $gcd(a, N) = 1$ but $a^{N-1} \neq 1$, then S is a proper subgroup of Z_N^*.
- By Lagrange's theorem, if S is a proper subgroup, $|S|$ must divide the order of the group so that if $|S| \leq \frac{N-1}{2}$
- Thus the simple algorithm would be a 1-sided error algorithm with probabiltiy $< \frac{1}{2}$ of saying Prime when N is Composite.
- The only numbers that give us trouble are the Carmichael numbers (also known as *false primes*) for which a^{N-1} for all a such that $gcd(a, N) = 1$.
- It was only recently (relatively speaking) that in 1994 it was proven that there are an infinite number of Carmichael numbers.
- The first three Carmicahel numbers are 561, 1105, 1729

Miller-Rabin 1-sided error algorithm

Let $N-1 = 2^t u$ with u odd %Since wlg. N is odd, $t \geq 1$
Randomly choose non zero $a \in Z_N$ %Hoping that a will be composite certificate
If $gcd(a, N) \neq 1$ then report Composite
$x_0 = a^u$ %All computation is done mod N
For $i = 1 \ldots t$
    $x_i := x_{i-1}^2$
    If $x_i = 1$ and $x^{i-1} \notin \{-1, 1\}$, then report Composite
End For
If $x_t \neq 1$, then report Composite $\%x^t = x^{N-1}$
Else report Prime

Analysis sketch of Miller-Rabin

- Let S be the set of $a \in N$ that pass (i.e. fool) the Rabin-Miller test.
- S is a subgroup of Z_N^*. We want to show that S is a proper subgroup and then as before by Langrange we will be done.
- It suffices then to find one element $w \in Z_N^*$ that will not pass the Miller-Rabin test.

Matroids and independence systems

- Let $M = (U, \mathcal{F})$, where U is a set of elements, $\mathcal{F} \subseteq 2^{|U|}$; $I \in \mathcal{F}$ is called an independent set.
  An (heriditary) independence system satisfies the following properties:
  1) $\varnothing \in \mathcal{F}$; often stated although not necessary if $\mathcal{F} \neq \varnothing$
  2) $S \subseteq T, T \in \mathcal{F} \Rightarrow S \in \mathcal{F}$
- A matroid is an independence system that also satisfies:
  3) $S, T \in \mathcal{F}, |S| < |T|$, then $\exists x \in T \setminus S$ such that $S \cup \{x\} \in \mathcal{F}$
- Sets having at most k elements constitute the independent sets in a uniform matroid
- Other common examples, include
  1. partition matroids where U is the disjoint union $U_1 \cup U_2 \ldots \cup U_r$ and there are individual cardiality constraints k_i for each block U_i of the partition.
  2. Graphic matroids where U is the set of edges E in a graph $G = (V, E)$ and $E' \subseteq E$ is independent if $G = (V, E')$ is acyclic.
  3. Linear matroids where U is a set of vectors in a vector space and I is independent in the usualy sense of linear independence.

Matroids, k-independence systems and the natural greedy algorithm

- Beautiful development starting in the 1950's with the work of Rado[1957], Gale[1968] and Edmonds[1970,71], (extended by Korte and Lovász[1980,1984], and others) as to contexts in which *"the natural"* greedy algorithm will produce an optimal solution.
- In particular, matroids characterize those hereditary set systems for which the *natural* greedy algorithm (determined by the order $c_1 \geq c_2 \ldots$ for maximization) will optimize any linear objective function $\sum_{x_i \in I} c_i x_i$ subject to the constraint that I is an independent set in a matroid $M = (U, \mathcal{I})$.
- Here the best known example is perhaps the minimum (or maximum) spanning tree problem where the edges of a graph are the elements and the indepedent sets are forests in the graph. Kruskal's greedy algorithm is the natural greedy MST algorithm.

More general independence systems

There are many equivalent ways to define matroids. In particular, the exchange property immediately implies that in a matroid M every maximal independent set (*base*) has the same cardinality, the *rank* of M. We can also define a base for any subset $S \subseteq U$. Matroids are those independence systems where all bases have the same cardinality.
A (Jenkyns) k-independence system satisfies the weaker property that for any set S and two bases B and B' of S, $\frac{|B|}{|B'|} \leq k$. Matroids are precisely the case of $k = 1$.
Examples:

- The intersection of k matroids
- Mestre's k-extendible systems where the matroid exchange property is replaced by : If $S \subseteq T$ and $S \cup \{u\}$ and T are independent, then $\exists Y \subseteq T - S : |Y| \leq k$ and $T - Y \cup \{u\}$ is independent.
- Independent sets in $k + 1$ claw free graphs. In such graphs, the neighbourhood of every node has at most k independent vertices.

The standard greedy algorithm for k-systems and $k+1$ claw free graphs

Jenkyns shows that the standard greedy algorithm is a k-approximation for maximizing a linear function subject to independence in a k-independence system. It follows that the standard greedy algorithm is a k-approximation for independence in a $k+1$ claw free graph.
This implies constant approximations for many classes of graphs, in particular for many types of graphs induced by intersections of geometric objects.

Monotone submodular function maximization

- As previously mentioned, the monotone problem is only interesting when the submodular maximization is subject to some constraint.
- Probably the simplest and most widely used constraint is a cardinality constraint; namely, to maximize $f(S)$ subject to $|S| \leq k$ for some k and since f is monotone this is the same as the constraint $f(S) = k$.
- Following Cornuéjols, Fisher and Nemhauser [1977] (who study a specific submodular function), Nemhauser, Wolsey and Fisher [1978] show that the standard greedy algorithm achieves a $1 - \frac{1}{e}$ approximation for the cardinality constrained monotone problem. More precisely, for all k, the standard greedy is a $1 - (1 - \frac{1}{k})^k$ approximation for a cardinality k constraint.

Standard greedy for submodular functions wrt cardinality constraint

$S := \varnothing$
While $|S| < k$
  Let u maximize $f(S \cup \{u\}) - f(S)$
  $S := S \cup \{u\}$
End While

Generalizing to a matroid constraint

- Nemhauser and Wolsey [1978] showed that the $1 - \frac{1}{e}$ approximation is optimal in the sense that an exponential number of value oracle queries would be needed to beat the bound for the cardinalily constraint.
- Furthermore, Feige [1998] shows it is NP hard to beat this bound even for the explicitly represented maximum k-coverage problem.
- Following their first paper, Fisher, Nemhauser and Wolsey [1978] extended the cardinality constraint to a matroid constaint. Matroids are an elegant abstraction of independence in a variety of settings.
- Fisher, Nemhauser and Wolsey show that both the standard greedy algorithm and the 1-exchange local search algorithm achieve a $\frac{1}{2}$ approximation for an arbitrary matroid constraint.
- They also showed that this bound was tight for greedy and for the 1-exchange local search.

Achieving the $1 - \frac{1}{e}$ approximation for arbitrary matroids

- An open problem for 30 years was to see if the $1 - \frac{1}{e}$ approximation for the cardinality constraint could be obtained for arbitrary matroids.
- Calinsecu et al [2007, 2011] positively answer this open problem using a very different (than anything in our course) algorithm consiting of a continuous greedy algorithm phase followed by a pipage rounding phase.
- Following Calinsecu et al, Filmus and Ward [2012A, 2012B] develop (using LP analysis to guide the development) a sophisticated non-oblivious local search algorithm that is also able to match the $1 - \frac{1}{e}$ bound, first for the maximum coverage problem and then for arbitrary monotone submodular functions.

Another application of non-oblivious local search: weighted max coverage

The weighted max coverage problem

Given: A universe E, a weight function $w : E \to \Re^{\geq 0}$ and a collection of of subsets $\mathcal{F} = \{F_1, \ldots, F_n\}$ of E. The goal is to find a subset of indices S (subject to a matroid constraint) so as to maximize $f(S) = w(\cup_{i \in S} F_i)$ subject to some constraint (often a cardinality or matroid constraint).
Note: f is a monotone submodular function.

- In a matroid, all maximal independent sets have the same size; the rank of a matroid is the size of the largest maximal independent set. Conversely, if all maximal independent sets in an independence system M have the same size, then M is a matroid.
- For $\ell < r = \mathit{rank}(M)$, the ℓ-flip oblivious local search for max coverage has locality gap $\frac{r-1}{2r-\ell-1} \to \frac{1}{2}$ as r increases. (Recall that greedy achieves $\frac{1}{2}$.)

The non-oblivious local search for max coverage

- Given two solutions S_1 and S_2 with the same value for the objective, we again ask (as we did for Max-k-Sat), when is one solution better than the other?
- Similar to the motivation used in Max-k-Sat, solutions where various elements are covered by many sets is intuitively better so we are led to a potential function of the form $g(S) = \sum \alpha_{\kappa(u,S)} w(u)$ where $\kappa(u, S)$ is the number of sets F_i ($i \in S$) such that $u \in F_i$ and $\alpha : \{0, 1, \ldots, r\} \rightarrow \Re^{\geq 0}$.
- The interesting and non-trivial development is in defining the appropriate scaling functions $\{\alpha_i\}$ for $i = 0, 1, \ldots r$
- Filmus and Ward derive the following recurrence for the choice of the $\{\alpha_i\} : \alpha_0 = 0, \alpha_1 = 1 - -\frac{1}{e}$, and $\alpha_{i+1} = (i+1)\alpha_i - i\alpha_{i-1} - \frac{1}{e}$.

The very high level idea and the locality gap

- The high-level idea behind the derivation is like the factor revealing LP used by Jain et al [2003]; namely, they formulate an LP for an instance of rank r that determines the best obtainable ratio (by this approach) and the $\{\alpha_i\}$ obtaining this ratio.

The Filmus-Ward locality gap for the non oblivious local search

The 1-flip non oblivious local search has locality gap $O(1 - \frac{1}{e} - \epsilon)$ and runs in time $O(\epsilon^{-1} r^2 |\mathcal{F}||U| \log r)$
The ϵ in the ratio can be removed using partial enumeration resulting in time $O(r^3 |\mathcal{F}|^2 |U|^2 \log r)$.

A non oblivious local search for an arbitrary monotone submodular function

- The previous development and the analysis needed to obtain the bounds is technically involved but is aided by having the explicit weight values for each F_i. For a general monotone submodular function we no longer have these weights.
- Instead, Filmus and Ward define a potential function g that gives extra weight to solutions that contain a large number of good sub-solutions, or equivalently, remain good solutions on average even when elements are randomly removed.
- A weight is given to the average value of all solutions obtained from a solution S by deleting i elements and this corresponds roughly to the extra weight given to elements covered $i+1$ times in the max coverage case.
- The potential function is :

$$g(S) = \sum_{k=0}^{|S|} \sum_{T:T\subseteq S,|T|=k} \frac{\beta_k^{(|S|)}}{\binom{|S|}{k}} f(T) = \sum_{k=0}^{|S|} \beta_k^{(|S|)} \mathbf{E}_T[f(T)]$$

One more non oblivious local search

- We consider the weighted max (independent) vertex set in a $k+1$ claw free graph. Note that this is the standard graph theoretic notion of an indepedent set of vertices and this is not independence in a matroid. The problem is that of finding an independent set S of vertices so as to maximize a linear function $f(S)$ (i.e. weights given to vertices).
- The concept of an independent set in a $k+1$ claw free graph has been abstracted by Feldman et al [2011] to an independence system called k-exchange systems which are a proper subcase of Mestre's [2006] k-extendible systems which are a subcase of Jenkyn's [1976] k systems.
- The work of Jenkyns and Nemhauser et al show that the standard greedy algorithm is a $\frac{1}{k}$ approximation for weighted max independent set in a Jenkyn's k system.
- It remains an open problem to improve upon the greedy approximation for Mestre's k extendible systems and Jenkyn's k systems.

Oblivious and non-oblivious local search for $k+1$ claw free graphs

- The standard greedy algorithm and the 1-swap oblivious local search both achieve a $\frac{1}{k}$ approximation for the WMIS in $k+1$ claw free graphs. Here we define an "ℓ-swap" oblivous local search by using neighbrourhoods defined by bringing in a set S of up to ℓ vertices and removing all vertices adjacent to S.
- The standard greedy and 1-swap oblivious local search can be extended to the case of submodular (rather than linear) functions on the vertex sets. This results in a $\frac{1}{k+1}$ approximation (locality gap). The idea is to use marginal gain of an element (relative to the current solution).
- For the unweighted MIS, Halldórsson shows that a a 2-swap oblivious local search will yield a $\frac{2}{k+1}$ approximation.

Berman's [2000] non-oblivious local search

- For the weighted MIS, the "ℓ-swap" oblivous local search results (essentially) in an $\frac{1}{k}$ locality gap for any constant ℓ.
- Chandra and Halldórson [1999] show that by first using a standard greedy algorithm to initialize a solution and then using a "greedy" k-swap oblivious local search improves the approximation ratio to $\frac{3}{2k}$.
- Can we use non-oblivious local search to improve the locality gap? Once again given two solutions V_1 and V_2 having the same weight, when is one better than the other?
- Intuitively, if one vertex set V_1 is small but vertices in V_1 have large weights that is better than a large set V_2 whose vertices have small weights.
- Berman chooses the potential function $g(S) = \sum_{v \in S} w(v)^2$. Ignoring some small ϵ's, his k-swap non-oblivious local search achieves a locality gap of $\frac{2}{k+1}$ for WMIS on $k+1$ claw-free graphs.
- Linear function (resp. monotone submodular) maximization is extended to k exchangeable systems in Feldman et al [2011] (resp. Ward [2012]). Note: For the submodular case, the potential function introduces some obstacles in using the marginal weight.

The Lovász Local Lemma (LLL)

- Suppose we have a set of "bad" random events $E_1, \ldots, E_m$ with $Prob[E_i] \leq p < 1$ for each i. Then if these events are independent we can easily bound the probability that none of the events has occurred; namely, it is $(1-p)^m > 0$.
- Suppose now that these events are not independent but rather just have limited dependence. Namely suppose that each E_i is dependent on at most r other events. Then the Lovász local Lemma (LLL) states that if $e \cdot p \cdot (r+1)$ is at most 1, then there is a non zero probability that none of the bad events E_i occurred.
- As stated this is a non-constructive result in that it does not provide a joint event in which none of the bad events occured.
- There are a number of applications of LLL including (Leighton, Maggs, Rao) routing, thje restricted machines version of the Maxmin "Santa Claus" problem and as we shall now see, solving exact k-SAT under suitabl conditions on the clauses.

A somewhat canonical application of the LLL

- Let $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ be a an exact k CNF formula. From our previous discussion of the exact Max-k-Sat problem and the naive randomized algorithm, it is easy to see that if $m < 2^k$, then F must be satisfiable. ($E[\text{clauases satisfied}] = \frac{2^k-1}{2^k} m > m - 1$ when $m < 2^k$.)
- Suppose instead that we have an arbitrary number of clauses but now for each clause C, at most r other clauses share a variable with C.
- If we let E_i denote the event that C_i is not satisfied for a random uniform assignment and hence having probability $1/(2^k)$, then we are interested in having a non zero probability that none of the E_i occurred (i.e. that F is satisfiable).
- The LLL tells us that if $r + 1 \leq \frac{2^k}{e}$, then F is satisfiable.
- As nicely stated in Gebauer et al [2009]: "In an unsatisable CNF formula, clauses have to interleave; the larger the clauses, the more interleaving is required."

A constructive algorithm for the previous proof of satisfiability

- Here we will follow a somewhat weaker version (for $r \leq 2^k/8$) proven by Moser [2009] and then improved by Moser and G. Tardos [2010] to give the tight LLL bound. This proof was succinctly explained in a blog by Lance Fortnow
- This is a constructive proof in that there is a randomized algorithm (which can be de-randomized) that with high probability (given the limited dependence) will terminate and produce a satisfying assignment in $O(mlogm)$ evaluations of the formula.
- Both the algorithm and the analysis are very elegant. In essence, the algorithm can be thought of as a local search search algorithm and it seems that this kind of analysis (an information theoretic argument using Kolmogorov complexity to bound convergence) should be more widely applicable.

The Moser algorithm

We are given an exact k-CNF formula F with m variables such that for every clause C, at most $r \leq 2^k/8$ other clasues share a variable with C.

Algorithm for finding a satisfying truth assignment

Let τ be a random assignment
Procedure SOLVE
  While there is a clause C not satisfied
    Call FIX(C)
  End While

Procedure FIX(C)
  Randomly set all the variables occuring in C
  While there is a neighbouring unsatisfied clause D
    Call FIX(D)
  End While

Sketch of Moser algorithm

- Suppose the algorithm makes at least s recursive calls to FIX. Then $n + s * k$ random bits describes the algorithm computation up to the s^{th} call at which time we have some true assignment τ'.
- That is, te computation (if it halts in s calls is described by the n bits to describe the initial τ and the k bits for each of the s calls to FIX.
- Using Kolmogorov complexity, we state the fact that most random strings cannot be compressed.
- Now we say that r is sufficiently small if $k - \log v - c > 0$ for some constant c, Then the main idea is to describe these $n + s * k$ bits in a compressed way if s is large enough and r is small enough.

Moser proof continued

- Claim: Any C that is satisfied before Fix(C) is called in SOLVE remains satisfied.
- Claim: Working backwards from τ' we can recover the original $n + s * k$ bits using $n + m \log m + s(\log r + c)$ bits; that is n for τ', $m \log m$ for calls to FIX in SOLVE and $\log r + c$ for each recursive call.
- Note: Here it is not stated, but the algorithm does not always terminate

The briefest introduction to spectral methods

- Like other topics in the course, spectral methods and in particular spectral graph theory and spectral graph algorithms is really a topic in itself.
- Spectral methods are becoming more and more important with applications to many areas of research.
- When we say *spectral method*, we mean algorithmic methods relying on the eigenvalues and eigenvectors of a matrix. In particular, we will just highlight some results relating to matrices coming from undirected graphs.
- An excellent set of (hand-written) lecture notes are by Lap Chi Lau. These notea in turn s follow those of Dan Spielman who has been central to the recent activity in this area.
- I will just briefly introduce some terminology and give a glimpse of one application of spectral graph theory.

Spectral graph theory

- For undirected graphs, the adjacency matrix $A(G)$ of a graph G is a real symmetric matrix.
- A non-zero (column) vector x is an eigenvector of A with eigenvalue λ if $Ax = \lambda x$.
- (The spectrum of A or a graph G refers to the set of eigenvalues of A (resp $A(G)$).
- When A is a real symmetric matrix, then all the eigenvalues are real and there is an orthonormal basis of R^n consisting the eigenvectors of A. That is, the eigenvectors are orthogonal to each other and each normalized to length $= 1$.
- The question is what useful information about a graph can the spectrum provide?

The Laplacian

- In spectral graph theory, it is often better to consider the Laplacian of a graph which is defined as $L(G) = D(G) - A(G)$ where $D(G)$ is the diagonal matrix whose entries are the degrees of the vertices.
- In particular if G were d regular, then any eigenvector of $A(G)$ with eigenvalue λ is an eigenvector of $L(G)$ with eigenvalue $d - \lambda$ and vice versa.
- The nice property of the Laplacian $L(G)$ is that it is a positive semi-definite matrix which means that all its eigenvalues are non-negative.
- Furthermore, G is connected if and only if $\lambda = 0$ is an eigenvalue of $L(G)$ with multiplicity 1. More generally, G has k connected components iff 0 is an eigenvalue of multiplicity k.
- Why is this interesting? Ordering so that $\lambda_1 \leq \lambda_2 \ldots \leq \lambda_n$, we can think of the two smallest eigenvalues being close iff the graph is "close" to being disconnected iff there is a "sparse cut".