CSC2420 Fall 2012: Algorithm Design, Analysis and Theory

Allan Borodin

April 7, 2016; Lecture 12

Annoucements and Todays Agenda

- Announcements
  1. The due date for Assignment 3 has been extended to noon, Monday, April 18. I am not here next week burt will be back in the office on Monday, the 18th. You can email the assignment if your wish. If submitting hard copy, please beign to my office SF 2303B and place under door if I am nort there.
  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.
  3. I must leave early today so we will lecture without a break today.
- Todays agenda
  1. Brief Introduction to spectral methods
  2. If time permits a very brief discussion of factor revealing LPs

A brief 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.
- One of the most active and influential researchers in this area is Dan Spielman. His Fall, 2015 course notes on spectral graoh theory can be found at http://www.cs.yale.edu/homes/spielman/561/. I have posted a tutorial by Dan Spielman on the course web page.
- I will just briefly introduce some terminology and give a glimpse of some applications of spectral graph theory. Spielman's course notes and tutorial will, of course, provide many further applications.

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".

Sparse cuts

- Recall that a cut in a graph $G = (V, E)$ is a partition of the vertices V into S and $V \setminus S$ (or equivalently the cut set of edges , that is, $cut(S) = \{e = (u, v) | u \in S, v \in V \setminus S\}$).
- We previously discussed min cuts in a graph and how they can be optimally computed using the the max flow-min cut theorem and (say) a Ford Fulkerson based algorithm. (For edge weighted graphs, Ford Fulkerson computes a cut of minimum weight.)
- Our goal now is to produce “balanced sparse cuts”. That is, we want to view the size of a cut relative to the sizes of S and $V \setminus S$. Such balanced cuts have applications to algorithms that work by decomposing a graph into roughly equal parts.
- The *conductance* $\phi(S)$ of a set S is defined as: $\frac{|cut(S)|}{\min\{vol(S), vol(V \setminus S)\}}$ where $vol(S) = \sum_{u \in S} degree(u)$.

Conductance

- Sometimes conductance is defined as $\frac{|cut(S)|}{|S|\cdot|V\setminus S|}$. These quantities are within a factor of 2.
- The conductance $\phi(G)$ of a graph is the $\min_{S:|S|\leq n/2} \phi(S)$. Computing the conductance of a graph is a well studied formulation of the *sparsest cut problem*. It is NP-hard and the best known approximation for about 15 years was the Leighton Rao $O(\log|V|)$ for about 15 years and then improved by Arora, Leighhton and Rao to $O(\sqrt{\log|V|})$.

Cheeger's Inequality

Cheeger's inequality has been called *the most important result in spectral graph theory*.

- To state this result it is useful to consider the following normalized adjacency and Laplacian matrices:
  $A' = D^{-1/2}AD^{-1/2}$ and $L' = D^{-1/2}LD^{-1/2}$
- Here $D^{-1/2}$ is the diagonal matrix with diagonal entries $d_{i,i} = degree(v_i)^{-1/2}$
- Letting $\{\alpha_i\}$ (resp. λ'_i) denote the eigenvalues of A' (resp. L'), it follows that
  $1 \geq \alpha_1 \geq \alpha_2 \ldots \geq \alpha_n \geq -1$ and $0 = \lambda'_1 \leq \lambda'_2 \ldots \leq \lambda'_n \leq 2$.

Cheeger's inequality continued

- The *spectral gap* is the difference bewteen α_1 and α_2 (or between λ'_1 and λ'_2).
- The spectral gap is closely related to conductance as well as the graph expansion properties and random walk properties.

Cheeger's inequality

$\lambda'_2/2 \leq \phi(G) \leq \sqrt{2\lambda'_2}$

- The spectral gap is also closely related to the important concept of *expander graphs*.
- Intuitively, expander graphs $G = (V, E)$ satisfy the property that for all (not too large) subsets $S \subset V$, the size of the neighbourhood of S is sufficiently larger than the size of S.

Expander graphs and applications

- Expander graphs have many applications (e.g. in coding theory, random walks, error probability amplification and derandomization).
- There are various combinatorial parameterized definitions and we will soon present two specific definitions.
- Expansion is (with high probability) a property of random graphs and in a sense expander graphs are often surrogates for random graphs.
- There is a considerable amount of research on the construction of *explicitly defined* expander graphs of small degree.
- We will see that algebraically, expander graphs can also be characterized as graphs having a suitable spectral gap, and also equivalently as graphs having rapid (i.e. $O(\log n)$) mixing time to equilibrium in a random walk.

Two specific conbinatorial expander definitions

Two expander definitions that occur are the following:

An $(n, d, c))$ node expander

A (n, d, c) node expander is an n node d-regular biparitite graph multi-graph $G = (X, Y, E)$ with $|X| = |Y| = n/2$ such that any subset $S \subseteq X$ satsifies
—Neighbourhood of $S| \ \geq (1 + c(1 - \frac{2|S|}{n})|S|$.

An $(n, d, c))$ edge expander

(N, d, c) edge expander is an n node, d-regular multi-graph $G = (V, E)$ such that any subset $S \subseteq V$ with $|S| \leq n/2$ has at least $cd|S|$ edges between S and $V \setminus S$.

- In general, one wants small degree d and a constant $c > 0$.
- Most random d-regular bipartite garphs are such expanders but we usually need explicitly constructed exapnders (which are known).

Expanders and the spectral gap

Given this type of edge expander, we have the following seminal relation with the spectral gap (due to Noga Alon). Here we let $\{\lambda_i\}$ be the eigenvalues of the adjacency matrix $A(G)$.

Relating expansion and spectral gap

If G is a (n, d, c) edge expander then $\lambda_1 = d$, then $\frac{1}{2}(1 - \lambda_2/d) \leq c \leq \sqrt{2(1 - \lambda_2/d)}$

Random walks and node expanders

- It is convenient again to normalize the adjacency matrix to form $P = A(G)/d$.
- Since G is bipartite there is no stationary distribution so to make the process aperiodic, define $Q = (I + P)/2$ meaning that with probability $1/2$, the process stays in the same state. Q now represents a doubly stochastic Markov process with a uniform stationary distribution $\{\pi_j\}$.
- The eigenvalues are now $\{\lambda'_i\}$ satisfying $\lambda'_i = \frac{1+\lambda_i/d}{2}$ so that $\lambda'_1 = 1$. Now suppose we have an expander with $\lambda'_2 \leq 1 - \frac{\epsilon}{2d}$.

Fast convergence

$\max_j \frac{|q_j^t - \pi_j|}{\pi_j} \leq n^{1.5}(\lambda'_2)^t$ where
q_j^t is the probability of being in state j at time t.

- This implies convergence in $O(\log n)$ steps. See Motwani and Raghavan, Section 6.7.2.

Application to probability correctness amplification

- Consider a polynomial time RP (1-sided error) set (e.g. the composite numbers, represented in binary or decimalor BPP (2-sided error) set.
- Suppose the algorithm has error probability c (e.g. $c = 1/4$) using n random bits.
- For an RP (resp. BPP) set we can amplifiy the error bound to c^k by doing k independent trial and hence using kn bits.
- Suppose we have an explicit expander with constant degree (say degree $d = 8$). Consider a random walk on an (exponential size) expander where the nodes correspond to n bit strings.
- Since the stationary distribution is the uniform distribution, the idea is to do such a random walk and sample some $O(k)$ nodes, sampling every b steps (for some appropriate b)
- Starting at a random node (using n bits), we will only need $n + O(k)$ bits to obtain enough trials and the desired c^k error.

$\#P$ counting problems

- Recall that an NP set L can be defined by $L = \{x|R(x,y)\}$ where R is a polynomial time verification algorithm and y is a polynomial length certificate. (Similarly, RP sets are those where the fraction of certificates is some constant $c > 0$.)
- A $\#P$ counting problem $\#L$ is one that can be defined as the number of certficates for an NP set L.
- For example, $\#SAT$ is the counting problem that outputs the number $\#x$ of satsifying formulas for an input formula x encoding a CNF formual F.
- Clearly if $\#L$ is polynomial time computable, then so is L so we certainly do not expect counting problems coresponding to NP complete sets to be computable in polynomial time.

$\#P$ complete counting problems

- Clearly $\#SAT$ is a $\#P$ complete problem in the sense that that any NP counting problem can be reduced to this problems.
- But even if L is polynomial time computable, it does not show that $\#L$ is polynomial time.
- For example, given a proper DNF formula F, it is immediately clear that F is satisfiable and given a bipartite graph G, we can efficiently determine if G has a perfect matching. However, both #DNF-SAT and #bipartite-matching are $\#P$ complete counting problems.

Approximating $\#P$ complete counting problems

- Given the hardness of $\#P$ complete problems, what we can hope for is to compute an estimate of $\#x$ for an input instance x. We want an estimate to fall in the range $[(1-\epsilon)\#x, (1+\epsilon)\#x]$ for every input instance x.
- For a randomized algorithm we would want such an estimate to be obtained with probability error some $\delta < 1$.
- Given that we can encode an ϵ in $\log 1/\epsilon$ bits, we might hope for such an algorithm to have time bounded by a polynomial is $n, \log 1/\epsilon$ and for randomized algorithms also in time $\log 1/\delta$. But it turns out that this would imply $P = \#P$ (or BPP $= \#P$ for randomized algorithms).
- Instead we will be happy to get algorithms with run time polynomial in $n, (1/\epsilon$, and $\log 1/\delta$. Such a randomized algorithm are called an FPRAS algorithm.

Approximate counting

- Unlike the BPP$=$ P question, it turns out that there are some counting problems (e.g. volume estimnation of a convex body in n dimensions) for which randomization provably helps.
- When the underlying decision problem L is in P, there is a natural randomized approach, which we can call "basic Monte Carlo sampling".
- Namely, sample from the space of all possible inputs and let the fraction of good inputs (i.e. those in the set L) be an estimate of the fraction of all inputs that are good
- Here then is the "abstract estimation problem": Let f be a Boolean function over a universe U such that $f(U)$ can be efficiently computed for any $u \in U$. Assume that U can be sampled uniformly at random. We want to estimate the size of $G = \{u|f(u) = 1\}$.

The natural estmation approach and its limitation

- Let $Y_i = 1$ iff $f(u_i) = 1$ where u_i is the i^{th} sampled input. Choose m random samples and then estimate $|G|$ by $Z = |U| \sum_i Y_i/m$.
- Let $\rho = |G|/|U|$. Then the basic Monte Carlo estimation is an FRPAS if $m \geq \frac{4}{\epsilon^2 \rho} \ln \frac{2}{\delta}$.
- So what is the limitation?

What happens when ρ is small or when it is hard to sample uniformly

- Whe ρ is small (as it can be for say DNF-SAT or bipartite matchings) or when it is not clear how to sample uniformly (e.g. trying to sample from the set M_k of all size k matchings and then using this sampling to recursively estimate the number m_k of size k matchings.
- When the minimum degree is at least $n/2$ here is the approach. The idea is to sample uniformly from $M_k \cup M_{k-1}$. This will give estimates of $r_k = m_k/m_{k-1}$
- Noting that $m_1 = |E|$, the desired estimate is $m_n = m_1 \Pi_{i=2}^{n} r_i$.
- To do the sampling, one needs to construct a doubly stochastic matrix (as in the amplification analysis) such that the resulting Markov process is rapidly mixing.

Applications of the smallest eigenvalue

- We return to the normalized adjacency matrix $A'(G)$ with eigenvalues $\{\alpha_i\}$.
- It can be shown that G is bipartitie iff $\alpha_1 = -\alpha_n$.
- Recalling that the eigenvalues of A' are in [1,-1], the matrix $I + A'$ has eigenvalues in [0,2].
- A graph $G = (V, E)$ is "close to bipartite" if the smallest value of $I + A'$ is close to 0.
- Another way to think about being close to bipartite is to have a large maximum $cut(S)$ relative to $|E|$.
- The best approximation for this NP hard problem is the same $\approx .878$ achieved by the same kind of SDP we saw for Max-2-Sat. This ratio is the best possible assuming the UGC.
- The obvious greedy algorithm for max cut (or the naive random algorithm) gives a $1/2$ approximation and it remained an open problem to beat $1/2$. Trevisan uses a spectral based algorithm that acheievs ratio .531 which was then improved by Sato to ,614. Can we achieve the $3/4$ for Max-Cut by a combinational algorithm as we have for Max-Sat?

Continued spectral applications

- Steuer gives some evidence for annd against the UGC. The evidence against is an improved UGC algorithm that exploitts the entire spectrum (of eigenvalues).
- More classical results go back to Hoffman who related the independence number $\alpha(G)$ and chromatic number $\chi(G)$ of graph to the spectrum.
- Namely, for $\{\lambda_i\}$ again being the eigenvalues for the adjcency matrix, $\alpha(G) \leq \frac{-\lambda_n}{d_{max}-\lambda_n}$ and
  $\chi(G) \geq 1 - \frac{\lambda_1}{\lambda_n}$.

Factor revealing LPs

- In the dual fitting method (that we illustrated with the natural greedy algorithm for set cover problem), the dual solution is not a feasible dual. But the dual solution appropriately scaled down is a a feasible dual. For the set cover problem, if d is the maximum size of any set, then H_d is a sufficient scaling factor. (This is dual complementary slackness.)
- Is there a principled way to think about deriving appropriate scaling factors so that dual solutions become feasible? This will be the goal of factor revealing LPs.
- The greedy algorithm can be recast as a primal dual algorithm where the $price(e_i)$ becomes the dual variable y_i associated with element e_i. These dual $\{y_i\}$ variables are raised simultaneously and whenever a dual constraint becomes tight for a set S_j, all the dual variables in S_j are frozen (i.e. no longer raised) and withdraw their contribution from all other sets in which they occur. Then S_j is added to the cover.

Factor revealing LPs continued

- This then has the nice interpretation of the dual variables paying for the sets in the cover.
- By renaming, let the order in which the dual variables are covererd be $e_1, e_2, \ldots, e_m$. By the uniform raising of the dual variables we then have $y_1 \leq y_2 \ldots \leq y_m$.
- Let us say that a k element set S_j is selected when $i-1$ of its elements have already been covered (and hence frozen). Then $(k-i+1)y_i \geq w_j$.
- The goal then is to see what is the least scaling factor that can be used to insure dual feasibility.
- For a fixed size problem (i.e. fixing n and m, the number of sets and elements), we want to maximize over all sets and all instances of that size to reveal a satisfactory scaling factor.

Factor revealing LPs continued

For the set cover greedy algorithm recast as a primal dual algorithm, we have the following factor revealing LP problem (for instances of a given size):

Factor revealing LP for set cover greedy algorithm

Maximize $\frac{\sum_{i=1}^{k} y_i}{w_S}$ over $\{y_i\}$ and all sets S (noting that w_S is now considered a variable)
subject to

- $y_i \leq y_{i+1} \quad 1 \leq i \leq k-1$
- $(k-i+1)y_i \leq w_S \quad 1 \leq i \leq k-1$
- $y_i \geq 0 \quad 1 \leq i \leq k$
- $w_S \geq 1$

Factor revealing LP conclusion

- For any fixed size, the factor revealing LP provides an appropriate scaling factor.
- One then needs to consider the supremum of these values as the instancd size grows.
- The hope is that by inspection of some small cases that one can see determine an appropriate scaling factor for all instance sizes. That is, the approach provides guidance for an eventual human derived proof.
- Factor revealing LPs have been used in a number of algorithmic analyses. It was first explicitly presented by Jain et al [2003] for greedy algorithms for the facility location problem.
- It has been extended by Mahdian and Yan to [2011] to the KVV Ranking algorithm for bipartite matching in the ROM model. Their extension to strongly factor revealing LPs is such that any member of the family of factor revealing LPs can be used to establish an appropriate scaling factor.
- Another variant called tradeoff revealing LPs was used by Mehta et al [2015] to analyze a greedy algorithm for the the adwords problem.