CSC2420: Lecture 11

- Todays agenda :
- The streaming model. Frequency moments
- The constructive Lovasz Local Lemma
- Spectral algorithms

The streaming model

- In the data stream model, the input is a sequence A of inputs $a_1, \ldots, a_n$ which is assume to be too large to store in memory. The space available S(n) is some sublinear function. The input streams by and what can only store information in the sublinear space allotted. (It is also often desirable that that each input is process efficiently, perhaps even in time O(1).)
- The goal is to approximately compute some function or statistic of the data or identify some particular elements of the data stream.
- Most results concern the space required for a one pass algorithm. But there are other results concerning the tradeoff between the space and number of passes.

Some well studied streaming problems

- Computing frequency moments. Let A = $a_1 \ldots a_m$ be a date stream with a_i in N = {1,2,…n}. Let m_i denote the number of occurences of value a_i in the stream A. The k th frequency moment $F_k = \text{sum}_{i \text{ in } N} (m_i)^k$. $F_1 = m$, the length of the sequence. F_0 is the number of distinct elements in the stream and F_2 is a special case of interest called the repeat index (aka Gini's homogeneity index).
- Finding k-heavy hitters; i.e. those elements appearing at least m/k times in stream A.
- Finding rare or unique elements in A.

What is known about computing F_*k*

- Given an error eps and confidence bound delta, the goal is to compute an estimate Y = F'_*k* such that |F'_*k*-F_*k*| > eps F_*k* with probability at most delta.
- For all *k >1*, there is a (space) lower bound of n^{*1-2/k*}. There is a nearly matching O~() bound.
- For *k = 0* and every *c > 2*, there is an O(log *n*) space alg : that (*1/c*) F_0 <=alg <= *c* F_0 with delta = *2/c*
- For *k =1, log m* is obvious but an estimate can be obtained with space O(log log *m* + 1/eps) space
- For *k =2,* $O(\frac{log(1/\delta}{\epsilon^2}(logn + logm))$ space

Computing F_*k*

- The basic idea behind these randomized approximation algorithms is to define a random variable Y whose expected value is close to F_*k*, variance is sufficiently small such that this r.v. can be calculated under the space constraint.

- The seminal paper on this topic was by Alon, Matias and Szegedy in STOC 1996 where they have a space O~(*n*^{1-1/*k*}) upper bound (and the improved result for F_0,F_1 and F_2). The improved bound of O~(*n*^{1-2/*k*}) is due to Indyk and Woodruff (STOC 05).
- For simplicity we assume the input stream length m is known but it can be estimated and updated as the stream unfolds.

The AMS F_*k* algorithm

- Let constants s1 and s2 be defined as follows:
$$s_1 = \frac{8}{\epsilon^2} n^{1-\frac{1}{k}} \qquad s_2 = 2log\frac{1}{\delta}$$
- The output Y of the algorithm is the median of s_2 random variables Y1 , Y2 , ..., Y*s_2* where Y_i is the average of *s_1* random variables X_*ij* , *1 ≤ j ≤ s_1* . All X_*ij* are independent identically distributed random variables. Each X = X_*ij* is calculated in the same way using only O(log *n* + log *m*) bits as follows: Choose randomly *p* in [*1,…, m*], then see the value of *a_p* . Maintain *r_p* = |{*q*|*q* ≥ *p* and a_q = a_p }|. Define X = *m*(*r^k* − (*r* − *1*)^*k*).
- Note that in order to calculate X , we only require to store *a_p* (i.e. log *n* bits) and *r* (i.e. at most log *m* bits). We need to show that E(X) = F_*k* and that the variance is small enough to use the Chebyshev inequality.

AMS analysis continued

- We first show E[X] = F_*k* . By telescoping we have :

$$
\begin{aligned}
E(X) &= \frac{m}{m}[(1^k + (2^k - 1^k) + \ldots + (m_1^k - (m_1 - 1)^k)) \\
&\quad +(1^k + (2^k - 1^k) + \ldots + (m_2^k - (m_2 - 1)^k)) + \ldots \\
&\quad +(1^k + (2^k - 1^k) + \ldots + (m_n^k - (m_n - 1)^k))] \\
&= \sum_{i=1}^{n} m_i^k = F_k
\end{aligned}
$$

AMS analysis continued.

- Recall Y is the median of the Y_i. It is a standard probabilistic idea that the median Y of identical r.v.s Y_i (having constant probability of small deviation from their mean F_k) tranforms Y into a high probability of small deviation. The result needed is to show that $\mathrm{Prob}[|Y_i - F_k| > \mathrm{eps}\, F_k]$ is $\le 1/8$
- The Y_i values are an average of independent $X = X_{ij}$ variables but they can take on large values so Chernoff bounds are not that helpful here. Instead as indicated we AMS use the Chebyshev inequality: $\mathrm{Prob}[|X-E[X]| \ge t\,\mathrm{sqrt}\{\mathrm{Var}[X]\}| \le (1/t^2)$; that is, $\mathrm{Prob}[|X-E[X]| \ge T] \le \mathrm{Var}[X]/T^2$

Bounding the variance

- $Var[X] = E[X^2]-(E[X])^2 <= E[X^2]$
- Using $a^k-(a-1)^k <= (a-b)ka^{k-1}$, it is not difficult to show $E[X^2] <= k F_1 F_{2k-1}$
- It is shown that $F_1 F_{2k-1} <= n^{1-1/k} (F_k)^2$
- $Var[Y_i] = Var[(1/s_1) \text{ sum } X_{i,j}] = (1/s_1) Var[X]$ which is at most 1/8 by the def of s_1.
- For $k = 2$, this is an $O(\sqrt{n})$ space bound and AMS show how to improve this to $O(\log n + \log m)$.

Sketch of F_2 improvement

- We again take the median of $s_2 = 2 \log(1/\delta)$ random variables Y_i but now each Y_i be the sum of only a constant number $s_1=16/(\epsilon^2)$ of identically distributed variables $X = X_{ij}$.
- The key idea is that X will not maintain a count for a particular value separately but rather will count an appropriate sum $Z = \sum b_t m_t$ and then $X = Z^2$.
- Here is how the vector $(b_1,\ldots,b_n)$ is randomly chosen. $V =\{v_1,\ldots,v_h\}$ is a set of $O(n^2)$ vectors where the vectors v_p are in $\{1,-1\}^n$ and are 4-wise independent. (That is, for every b in $\{1,-1\}^4$, and every choice $j_1<j_2<j_3<j_4$ of 4 components, the probability that (over the $\{v_p\}$) is 1/16 that these 4 components will be set to b.

F_2 sketch continued

- Using the 4-wise independence (which also implies r-wise indep. for $r<4$), it then follows for $X = Z^2$:
- $E[X] = \text{sum } (m_t)^2 = F_2$
- $E[X^2] = \text{sum } (m_t)^4 + 6 \text{ sum}_{j<k}(m_j)^2(m_k)^2$
- $Var[X] = 4 \text{ sum}_{j<k} (m_j)^2 (m_k)^2 <= 2 (F_2)^2$
- $Var[Y_i] = (1/s_1) Var[X] <= (2 (F_2)^2)/s_1$
- $Prob[Y_i-F_2]>eps F_2] <= Var[Y_i]/(eps^2 (F_2)^2)$ which gives the desired 1/8 probanbility.

Before moving on

- Streaming is an important field of recent algorithmic research. We have just considered one problem in one particular input model, the so-called time series model. There are two more general input models in which this and other similar problems can be studied.
- Cash register model: the input stream is a sequence $(I_1, I_2, \ldots, I_m)$ where $I_t = (j, c_t)$ and $c_t \geq 1$ representing how much to increase the count for item a_j. The current count state $(m_1(t), \ldots, m_n(t))$ at time t is then $m_i(t) = m_i(t-1) + c_t$ if $i = j$ and otherwise $m_i(t) = m_i(t-1)$.
- The turnstile model: this is the same model but now $|c_t| \geq 1$ allowing decrements as well as increments.

The constructive Lovasz Local lemma

- Suppose we have a series of random events E_1, …,E_m with Prob{E_i] <= 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 at most (1-p)^m. We, of course, often use this to amplify probabilities.
- Suppose now that these events are not independent but rather just limited dependence. Namely suppose that each E_i is dependent on at most d other events. Then the Lovasz local lemma (LLL) states that if ep(d+1) is at most 1, then there is a non zero probability that none of the events E_i occurred.

An application of the LLL

- Let F = C_1 ^ C_2 …^C_m be a an exact k CNF formula. From our previous discussions we know that if m < 2^k, F must be satisfiable.
- Suppose instead that we have an arbitrary number of clauses but now for each clause C, at most d 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 d+1 <=2^k/e, then F is satisfiable.

A constructive proof

- Here we will follow a somewhat weaker version (for d<=2^k/8) proven by Moser (2010) and then improved by Moser and G. Tardos to give the LLL bound. This is a constructive proof in that there is a randomized algorithm (which can be de-randomized) that with high prob (given the limited dependence) will terminate and produce a satisfying assignment in O(m log m) evaluations of the formula.
- Both the algorithm and the analysis are very elegant. The algorithm is in essence a local search search algorithm and it seems that this kind of analysis (an information theoretic argument using Kolmogorov complexity) should be more widely applicable.

The algorithm

SOLVE

    Let tau be a random assignment.
    While there is a clause C not satisfied by tau
        Call FIX(C)
    End While

FIX(C)

    Randomly set the variable in C
    While there is a neighbouring unsatisfied clause D
        Call FIX(D)
    End While

The analysis

- Suppose that the algorithm uses at least s calls and then note that up to the s th call that the algorithm is using (and is described) by n + sk random bits.
- Kolmogorov complexity tells us that most random strings (say of length t) are incompressible and basically require t bits to be described.
- The algorithm gives us a compressed way to describe these bits (if s is large enough a d small enough).
- The key idea is that we need to denote with a constant number c of bits that a call has terminated and for any call FIX(D), if we know (log r bits) which neighbour C is of D, then we know (since C was unsatsified) how its bits were set upon entering C.
- Hence we must have $n + m \log m + s (\log r + c) >= n + sk$

Begin spectral methods

- This like other topics in the course 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) recent lecture notes are by Lap Chi Lau and that is what I am relying on. These notes follow those of Dan Spielman who has been central to the recent activity in this area.
- I will introduce terminology as needed and will basically skip proofs in favour of giving some sense of the importance of the spectral graph methods.

Introducing spectral graph theory

- Since we will be focusing on 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 = λx. (The spectrum of A or a graph G refers to the set of eigenvalues of A, 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 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-λ 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 λ= 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 $\lambda_1 \leq \lambda_2 \ldots \leq \lambda_n$ we can think of the two smallest eigenvalues being close iff the graph is "close" to being disconnected iff there is a "sparse cut".

Sparsest cut

- A cut in a graph G = (V,E) is a partition of V
- Into S and (V-S) or equivalently the set cut(S) of edges e = (u,v) with u in S, v in V-S.
- What is a sparse cut or the sparsest cut? We want in some sense to view the size of a cut relative to the sizes of S and V-S. This is what one needs for decomposing a graph.
- The conductance phi(S) of a set S is defined as |cut(S)|/min{vol(S), vol(V-S)} where the volume vol(S) = sum_{u in S} deg(S)
- The conductance phi(G) = min_{S:|S| <= n/2} of phi(S) is one well studied and NP-hard concept of sparsest cut and the best known approximation is O(sqrt{n}).

Cheeger's inequality

- To nicely state this Cheeger's inequality (which Lau states as perhaps the most important result in spectral graph theory), it useful to consider the normalized adjacency and Laplacians; namely A' = D^{-1/2} A D^{-1/2} and L' = D^{-1/2} L D^{-1/2} where D^{-1/2} is the diagonal matrix with entries (d_i)^{-1/2} .
- Letting $\{\alpha_i\}$ (resp. λ'_i) denote the eigenvalues of A' and L', it follows that $1 \geq \alpha_1 \geq \alpha_n \geq -1$ and $0 = \lambda'_1 \leq \lambda'_2 \ldots \leq \lambda'_n \leq 2$

Cheeger's inequality

- The spectral gap is the difference between α_1 and α_2 (or between λ_1 and λ_2)
- The spectral gap is closely related to conductance as well as expansion properties and random walk properties.
- Theorem (Cheeger): $\lambda_2'/2 \leq \phi(G) \leq \sqrt{2\lambda_2'}$
- The lower bound on the conductance comes from a linear programming relaxation of the integer program formulation of the problem.

Other implications of the spectral gap

- Expander graphs have many applications (e.g. to coding theory, random walks, correctness amplification and de-randomization) and there are various combinatorial parameterized definitions.
- But intuitively, expander graphs G = (V,E) satisfy the property that for all (not too large) subsets S of V, the neighbourhood of S is suitably larger than S.
- This is a property (whp) of random graphs and in a sense expander graphs often act as surrogates for random graphs and there has been much research on the explicit construction of small degree random graphs.
- Algebraically, expander graphs can also be characterized as graphs with suitable spectral gap and equivalently as graphs having rapid O(log n) mixing time (to equilibrium in a random walk).

Application of the smallest eigenvalue

- Consider the normalized adjacency matrix A'(G) with eigenvalues $\{\alpha_i\}$.
- It can be shown that G is biparitite iff $\alpha_1 = -\alpha_n$
- Considering I + A' and recalling that the eigenvalues of A' are in [1,-1], I+A has eigenvalues in [0,2].
- A graph is "close to biparitite" if the smallest eigenvalue of I + A' is close to 0. But another way to think about being close to bipartite is to have a large maximum cut(S) relative to the |E|.
- The best approximation for this NP hard problem is the same .878 by the same kind of SDP analysis as we saw for Max-2-Sat.
- The obvious greedy algorithm (or a random S) gives an approximation of ½ and it remained an open problem to find a combinatorial algorithm with ratio better than ½. Trevisan uses a spectral based algorithm that achieves ratio .531 which was then improved by Sato to .614

A few other spectral applications

- The SDP approximation for max cut is the best known and assuming the UGC is the best possible. Steurer recently has given evidence both for and against the UGC. The evidence against is an improved algorithm exploiting the entire spectrum.
- More classical results go back to Hoffman who relates the independence number $\alpha(G)$ and $\chi(G)$ chromatic number of a graph to the spectrum.
- Namely, for $\{\lambda_i\}$ again being the eigenvalues of the adjacency matrix, $\alpha(G) \leq \frac{-\lambda_n}{d_{max} - \lambda_n}$
- $\chi(G) \geq 1 - \frac{\lambda_1}{\lambda_n}$