U.C. Berkeley — CS270: Algorithms
Professor Vazirani and Professor Rao

Lecture 21
Scribe: Anupam
Last revised

Lecture 21

1 Laplacian systems in nearly linear time

Building upon the ideas introduced in the previous two lectures, we will see an algorithm for solving Laplacian linear systems in time $O(m \log^3 n)$ based on the recent work of Koutis, Miller and Peng [1]. More generally, linear systems in symmetric diagonally dominant matrices reduce to Laplacian systems.

1.1 A recursive solution

The iterative method actually used in the solver is called the recursive preconditioned Chebyshev iteration and it converges in $O(\sqrt{\kappa})$ iterations. Operationally each iteration involves some matrix multiplications so for simplicity we will pretend that the iterative method is gradient descent. Recall the computational steps involved in one iteration of the preconditioned gradient descent method,

$$\begin{aligned} r_i &= B^{-1}b - B^{-1}Ax_i \\ \alpha &= \frac{r_i^T r_i}{r_i^T B^{-1} A r_i} \\ x_{i+1} &= x_i + \alpha r_i \end{aligned} \tag{1}$$

We saw that using low stretch trees as preconditioners yields an algorithm with running time $O(m^{3/2})$. The idea for improving the running time is to add some edges to the low stretch tree to improve the condition number. The preconditioner B is no longer a tree, so there is no linear time algorithm for inverting B, to compute $B^{-1}(Ay)$ we need to solve the linear system $Bz = Ay$. The linear system in B is solved recursively yielding the following recurrence for the total running time,

$$T(m) = \sqrt{\kappa}(T(m') + O(m)) \tag{2}$$

The number of iterations is $\sqrt{\kappa}$, the $T(m')$ is the time required to solve a linear system in B and $O(m)$ is the time required for multiplication by A.

The running time for such recurrences can be analyzed using Master's theorem,

Theorem 1
The solutions to the recurrence $T(n) = aT(\frac{n}{b}) + f(n)$ are given by:
(i) If $f(n) = O(n^{\log_b a - \epsilon})$ then $T(n) = O(n^{\log_b a})$.
(ii) If $f(n) = \Theta(n^{\log_b a} \log^k n)$ then $T(n) = O(n^{\log_b a} \log^{k+1} n)$.

We will show that for all $k \in \mathbb{N}$ it is possible to construct a preconditioner B with $m \log^2 n/k$ edges and $\kappa = \sqrt{k}$ yielding the recurrence,

$$T(m) = \sqrt{k}(T(m \log^2 n/k) + O(m)) \tag{3}$$

Choosing $k = \log^4 n$ we have $a = b = \log^2 n$ and $f(n) = \Theta(m \log^2 n)$ in the recurrence. The running time of the algorithm is $T(m) = O(m \log^3 n)$ by case (ii) of Master's theorem, which is almost linear in m.

1.2 Sparsifiers

The preconditioner B for should have two properties: (i) B should be a sparse graph, the number of edges $m' = m \log^2 n/k$ is small. (ii) The condition number of $B^{-1}A = \sqrt{k}$ is small, this is equivalent to $B \preceq A \preceq \sqrt{k}B$ by the claim in the previous lecture. A sparse graph approximating the spectral properties of G is called a spectral sparsifier. It will be helpful to think about sparsifiers to construct the preconditioner B.

A graph H with is said to be an ϵ-spectral sparsifier for the graph G if for all $x \in \mathbb{R}^n$,

$$(1-\epsilon) \leq \frac{x^T L_H x}{x^T L_G x} \leq (1+\epsilon) \tag{4}$$

The Rayleigh quotient characterization of eigenvalues shows that all the eigenvalues of H are within $(1 \pm \epsilon)$ of the eigenvalues of G. The weaker notion of an ϵ cut sparsifier requires that the weight of every cut $S \subset [n]$ in H is within $(1 \pm \epsilon)$ of the weight of the corresponding cut in G.

$$(1-\epsilon) \leq \frac{w_H(S,\overline{S})}{w_G(S,\overline{S})} \leq (1+\epsilon) \tag{5}$$

If $x \in \{0,1\}^n$ is the characteristic vector of a cut, then $x^T L x = \sum_{(i,j)\in E} w_{ij}(x_i - x_j)^2$ is the weight of the edges across the cut, showing that an ϵ-spectral sparsifier H is also an ϵ-cut sparsifier.

1.2.1 Sparsification:

Consider the following general algorithm for sparsification by sampling:

1. Draw q uniformly random samples from $E(G)$, edge e with weight w_e is sampled with probability p_e and added to the sparsifier with weight w_e/qp_e.

Why is this a reasonable algorithm? One way to see this is that the expected weight of edges across cut $(S,\overline{S})$ in H is equal to $w_G(S,\overline{S})$,

$$E[w_H(S,\overline{S})] = \sum_{e \in E_G(S,\overline{S})} qp_e \frac{w_e}{qp_e} = w_G(S,\overline{S}) \tag{6}$$

The samples are independent so for large enough q, we expect the weight of cuts in H to be concentrated around the expectation.

With what probabilities should the edges be sampled? For a dumbbell shaped graph with exactly one edge (u, v) between two connected components, the edge should be chosen with probability 1 in the sparsifier. The example suggests that the probability of sampling an edge should be proportional to the effective resistance of the edge.

The effective resistance $R(u, v)$ is the potential difference across u, v when 1 unit current is inserted at u and withdrawn at v. The effective resistance between (u, v) is computed by solving a Laplacian linear system,

$$R(u, v) = (\chi_u - \chi_v)^T L^+ (\chi_u - \chi_v) \tag{7}$$

Here χ_u, χ_v are the characteristic vectors of vertices u, v and the equation follows from the action of the Laplacian in on potential and current vectors from lecture 19. Computing the effective resistances is as hard as solving linear systems, explicitly computing the effective resistances is avoided using the oversampling theorem,

Theorem 2
If edges of G are sampled with probabilities proportional to $\widetilde{p}_e \geq w_e R_e$ and $t = \sum_e \widetilde{p}_e$ then the graph H obtained from the sparsification algorithm with $q = \widetilde{O}(t \log t)$ satisfies $G/2 \preceq H \preceq 3G/2$ with high probability.

The oversampling theorem shows that instead of computing the effective resistances explicitly it suffices to find $\widetilde{p}_e$ such that $\widetilde{p}_e \geq w_e R_e$ provided $\sum \widetilde{p}_e$ can be bounded. Low stretch spanning trees can be used to compute suitable $\widetilde{p}_e$.

1.3 Oversampling low stretch trees

The stretch of an edge $e = (u, v)$ with respect to a spanning tree T is defined as $st(e) = w_e \sum_{i \in [k]} \frac{1}{w_i}$, where w_i is the weight of the i-th edge in T on the unique path between u and v. For example, the stretch of a tree edge is 1, if the tree edges have resistances $1/w_i$ the stretch is the effective resistance between u and v in the tree. The total stretch is the sum of the stretches of all the edges in G.

Given a weighted graph graph G a spanning tree T with total stretch $O(m \log n)$ can be constructed in nearly linear time. The stretch of edge e satisfies the conditions of the oversampling theorem as $st(e) = w_e R_T(e)$ where $R_T(e)$ is the effective resistance in the tree. The effective resistance can only increase if we remove edges from a graph, so $R_T(e) > R_e$. (See next lecture for proof).

Add a copy of the low stretch spanning tree T scaled by a factor of k to G to obtain $G' = G + (k-1)T$. As $G \subset G' \subset kG$ we have $G \preceq G' \preceq kG$. The edges of G' are sampled with probabilities proportional to $st(e)/k$, the condition of the oversampling theorem is satisfied as kT is a spanning tree for G'.

As we multiplied the weights of the tree edges by k the total stretch of the non tree edges decreases to $O(m \log n/k)$. The sum of the sampling probabilities $t = O(m \log n/k + (n-1))$, the oversampling theorem tells us that if we sample $O(t \log t) = O(m \log^2 n/k)$ edges then with high probability we have graph H such that,

$$G \preceq G' \preceq 2H \preceq 3G' \preceq 3kG \tag{8}$$

The oversampling procedure produces a preconditioner with $O(m \log^2 n/k)$ edges and condition number $O(k)$ as required in the analysis of the recurrence in section 1.

1.4 Proof of the oversampling theorem

The oversampling theorem was not proved in lecture, the theorem was first proved in [2], we insert the proof for completeness.

1.4.1 Factorization of the Laplacian

The Laplacian is positive semidefinite and we know that all positive semidefinite matrices can be written as BB^t. A natural factorization for the Laplacian is obtained by fixing an arbitrary orientation for the edges of G and condsidering the $n \times m$ vertex edge incidence matrix B:

$$B_{ve} = \begin{cases} 1 & \text{if } e = (v,w) \\ -1 & \text{if } e = (w,v) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

It can be easily verified that BB^t is the Laplacian of the unweighted graph G, irrespective of the chosen orientation. The Laplacian of a weighted graph is given by BWB^t where W is an $m \times m$ diagonal matrix having edge weights w_e on the diagonal.

1.4.2 Energy expressed in two ways

The total energy dissipated in the electrical network can be expressed both in terms of potentials and currents. The expression on terms of potentials is equal to the value of the Laplacian quadratic form,

$$E(\phi) = \sum_{i \sim j} \frac{(\phi_i - \phi_j)^2}{R_{ij}} = \sum_{i \sim j} w_{ij}(\phi_i - \phi_j)^2 = \phi^T L \phi \tag{10}$$

Using the factorization $L = BWB^t$ the energy of the quadratic form can be re-written as the norm of an m dimensional vector $y \in Im(W^{1/2}B^t)$.

$$E(\phi) = \phi^t B W^{1/2} W^{1/2} B^t \phi = y^t y \tag{11}$$

The Laplacian for a candidate sparsifier is of the form $L_H = BW^{1/2}SW^{1/2}B^t$ as a sparsifier is obtained by re-weighting the edges. The diagonal entries of S are $\frac{n_e}{qp_e}$ where p_e is the probability of sampling e and n_e is the number of times e gets sampled.

The energy of the electrical flow in H corresponding to the potential vector ϕ is y^tSy where $y = W^{1/2}B^t\phi$. The spectral sparsification condition (4) states that the y^ty and y^tSy are approximately equal for all $y \in Im(W^{1/2}B^t)$, the condition is removed by considering the projector Π onto $Im(W^{1/2}B^t)$,

$$\forall y \in \mathbb{R}^m, \quad 1 - \epsilon \leq \frac{y^t \Pi S \Pi y}{y^t \Pi \Pi y} \leq 1 + \epsilon \tag{12}$$

It suffices to approximate find an approximation $|\Pi S \Pi - \Pi \Pi|_2 \leq \epsilon$ to produce a spectral sparsifier. Denoting $W^{1/2}B^t$ by $L^{1/2}$ as it is a square root of L, it is easy to see that the projector $\Pi = L^{1/2}L^+(L^{1/2})^t$.

1.4.3 Properties of the projection Π

Here are some observations about Π: (i) Π is isospectral to LL^+ by the cyclic property, the rank of Π is $n-1$ as LL^+ is identity on the $n-1$ dimensional subspace orthogonal to $\vec{1}$. (ii) The diagonal entry $\Pi_{ee} = w_e R_e$ where $R_e = (\chi_u - \chi_v)^T L^+ (\chi_u - \chi_v)$ is the effective resistance across the end points of e, this follows by by explicitly computing the diagonal entry $e_i^t \Pi e_i$ for $\Pi = W^{1/2} B^t L^+ B W^{1/2}$.

Let us consider some general properties of projection matrices. A projector P with column vectors v_i can be expressed as follows:

$$P = \sum_{i \in [n]} v_i e_i^t \tag{13}$$

Squaring and using orthogonality of the e_i we obtain a decomposition of P in terms of the column vectors,

$$P = P^2 = \sum_{i \in [n]} v_i v_i^t \tag{14}$$

The expression allows us to compute decompositions for $\Pi\Pi$ and $\Pi S \Pi$ in terms of $y_e = v_e / \sqrt{p_e}$, the normalized column vectors of Π,

$$\Pi\Pi \;=\; \sum_e p_e y_e y_e^t = E_p[yy^t] \tag{15}$$

To evaluate $\Pi S \Pi$ we expand each term in the product using the expansion of Π from equation (13) and expand S recalling that the diagonal entries of S are $\frac{n_j}{qp_j}$,

$$\begin{aligned} \Pi S \Pi &= \sum_{i \in [m]} v_i e_i^t \sum_{j \in [m]} \frac{n_j}{qp_j} e_j e_j^t \sum_{k \in [m]} e_k v_k^t \\ &= \frac{1}{q} \sum_{e \in [q]} y_e y_e^t \end{aligned} \tag{16}$$

The closeness of $\Pi\Pi$ and $\Pi S \Pi$ follows from the following concentration result for matrices,

Theorem 3
If $\sup |y_i| \leq M$ and $y_1, y_2, \cdots, y_q$ are independent samples drawn from the distribution p then,

$$E \left| E_p[yy^t] - \frac{1}{q} \sum_{e \in [q]} y_e y_e^t \right|_2 \leq CM \sqrt{\frac{\log q}{q}}$$

The upper bound M can be obtained as follows: $|y_i| = \frac{|v_i|}{\sqrt{p_e}} \leq \sqrt{t} \sqrt{\frac{w_e R_e}{\widetilde{p}_e}} \leq \sqrt{t}$ by the assumption $\widetilde{p}_e \geq w_e R_e$, using the fact that $|v_i| = w_e R_e$. Choosing $q = O(t \log t)$ samples ensures that the left hand side of the concentration bound is a constant.

The number of edges in the spectral sparsifier is $\widetilde{O}(m \log^2 n)$, one log factor comes from the stretch of the spanning tree $t = \widetilde{O}(m \log n)$ while the second one arises due to the concentration bound.

References

[1] I. Koutis, G.L. Miller, and R. Peng. Approaching optimality for solving SDD linear systems. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 235–244. IEEE, 2010.

[2] D.A. Spielman and N. Srivastava. Graph sparsification by effective resistances. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 563–568. ACM, 2008.