Coarsening Schemes

July 23, 2020

We are given an undirected, weighted graph $G = (V, E, w)$ where $w : E \to \mathbb{R}$ assigns weights to each edge.

As background, we define $\partial(S)$ as the set of edges with exactly one vertex in S and $d(S)$ as the sum of the weighted degrees of vertices in S for any subset $S \subset V$.

We begin by defining two key graph metrics.[1]

1. **Conductance:** $\phi_G := \min\limits_{S \subset V} \frac{w(\partial(s)))}{\min(d(S), d(V-S))}$
2. **Stretch:** $st_T(e) := w_e \sum\limits_{f \in P} \frac{1}{w_f}$ where P is the set of edges in the path from u to v in the spanning tree T.

Cheeger's inequality $2\phi_G \geq \lambda_2(D^{-1/2} L D^{-1/2}) \geq \frac{\phi_G^2}{2}$ (where λ_2 is the smallest non-zero eigenvalue) suggests seeking to design Laplacian precondioners with high conductance.

We will also see low-stretch subgraphs are good preconditioners.

We now identify 3 possible coarsening schemes:

1. **Ultra-sparsifiers**
2. **Bourgain's embedding into l_1**
3. **Low-stretch subgraphs**

1 Ultra-sparsifiers

Vaidya studied spanning trees as preconditioners and derived an $O(\sqrt{nm})$ upper bound on the number of iterations of PCG. As we know PCG converges in at most n iterations, spanning trees do not lead to provably fast preconditioners.

Instead, Vaidya suggested "starting" with a spanning tree and adding $O(n)$ edges back to it, with the goal to "fix" eigenvalues. We obtain a tree-like graph and may next do Cholesky factorization to eliminate nodes of degree 1 or 2 and recurse to the next coarser level.[1]

2 Bourgain's embedding into l_1

To solve min-cost flow, Sherman first embeds the graph into an l_1 lattice. Next, they satisfy demands on the near the edges and recurse on the middle.[2]

3 Low-stretch subgraphs

Low-strech subgraphs are also known to be good preconditioners. In particular, Sherman uses $j - trees$ (the union of a forest together with another graph) to satisfy max flow demands on the "tree" vertices and recurses on the remaining "interior" graph with more complex structure. Note that low-stretch subgraphs rely heavily on a low-diameter decomposition. [5]

4 References

1. Algorithms, Graph Theory, and Linear Equations in Laplacian Matrices (Spielman) provides an overview of the background, ultra-sparsifiers (Section 7), and low stretch spanning trees (Section 6).

2. Generalized Preconditioning and Undirected Minimum-Cost Flow (Sherman) provides a framework for the composition of solvers (Section 3) and Bourgain's embedding into l_1.

3. Nearly-Linear Work Parallel SDD Solvers, Low-Diameter Decomposition, and Low-Stretch Subgraphs (Spielman)

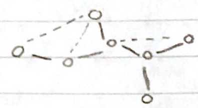
4. From Graphs to Matrices, and Back: New Techniques for Graph Algorithms (Madry)

5. Nearly Maximum Flows in Nearly Linear Time (Sherman)

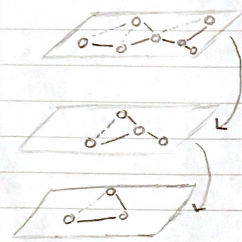
6. Parallel Approximate Undirected Shortest Paths via Low Hop Emulators (Andoni) provides fast parallel algorithms for Bourgain's embedding (Appendix E) and low diameter decomposition (Appendix F).

5 Drawings

ultra-sparsifier : G' with $n + O(n)$ edges is <u>ultra-sparsifier</u> if it is a good approximation of G



(take spanning tree & add edges)
to "fix" eigenvalues

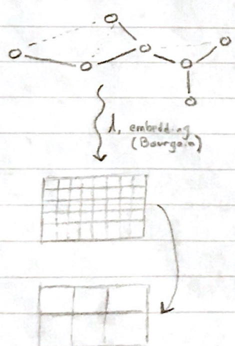


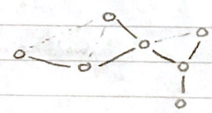
Cholesky factorization
(to eliminate vertices of deg 1 & 2)

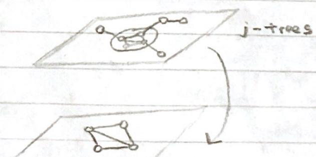
problem: can one design an algorithm for solving linear equations in Laplacian in time $O(m \log \varepsilon^{-1})$.

Sherman :

min-cost flow







$$\tilde{X}_{X \to Y}(A) = \min \{ \|A\| \|G\| : G : Y \to X,\ A \circ G \circ A = A \}.$$

Let F be an $(\alpha, \frac{\beta}{\tilde{\kappa}})$ algorithm then F^* is an $\left(\frac{\alpha}{1-\beta}, \frac{\beta^{\tau}}{\tilde{\kappa}}\right)$ algorithm.
(a final $\circ$ with a $(M, 0)$ solver reduces β-error).

Andoni : where does low diameter decompositions/metric tree embeddings appear?