Research Proposal for CPSC 490:
Solving Ill-Conditioned Laplacian Systems

Hannah Lawrence
Advisor: Dan Spielman
January 23, 2018

Overview  For this project, I will adapt a recent algorithm for solving ill-conditioned directed Laplacian systems for the case of undirected graphs. This will involve simplifying the theoretical development of the algorithm, as well as actually implementing and testing it in Julia.

1 Background

Systems of equations of the form $Lx = b$, where $L = D - A$ is the Laplacian of a graph, have received significant attention in the research community over the last two decades. Solving such a system has applications to diverse areas and problems, from graph partitioning and random walks to machine learning and medical imaging [1]. Certain applications of Laplacian solvers involve undirected graphs whose edge weights vary over many orders of magnitude, resulting in a high ratio between the Laplacian’s largest and smallest eigenvalues. In such an ill-conditioned system, small perturbations in the input can drastically reduce the accuracy of the numerically computed system; very high precision data types are one way to combat the issue. However, a recent paper by Cohen et al. [2] presents a reduction for approximately solving ill-conditioned, directed Laplacian systems. To do so, it makes $O(\log(n\kappa))$ calls to a solver of well-conditioned Laplacians, where $n$ is the number of nodes in the graph and $\kappa$ is an upper bound on the condition number of $L + L^T$. Thus, the algorithm has only logarithmic dependence on condition number, and should circumvent the challenges of the original ill-conditioned system.

1.1 Existing Algorithm

The iterative algorithm of [2] takes as input an Eulerian Laplacian $L = D - A^T$ and a vector $b \perp 1$, and outputs an approximate solution $x$ to $Lx = b$ that satisfies the error bound

$$||x - L^+b||_{U_L} \leq \frac{1}{2}||L^+b||_{U_L}.$$  

Here, $U_L \triangleq L + L^T$ is a symmetric matrix, and $||x||_{U_L} = \sqrt{x^TU_Lx}$. Intuitively, the algorithm transforms an ill-conditioned Laplacian to one that is at most polynomially ill-conditioned by repeatedly “contracting” edges that are weighted too heavily to lower the highest eigenvalue, and adding a small multiple of a clique to increase the smallest non-zero eigenvalue.

1.2 This Project

With the assistance of Professor Spielman, I will modify the algorithm and its theoretical analysis for the simpler case of undirected weighted graphs, and implement and test it in Julia. Beyond gaining familiarity with the conventions and features of Julia, this will require learning more about the techniques used in analysis of Laplacian solvers, including linear algebra, spectral graph theory, and the contraction and projection operators. While less central to the project, it will likely also be enlightening to read more about numerical analysis (condition numbers, precision), existing
Laplacian solvers, and applications of Laplacian solvers – particularly those in which the systems are usually ill-conditioned. For instance, one application in which an ill-conditioned Laplacian system arose in Professor Spielman’s work is a linear programming problem related to minimum cost flow.

2 Deliverables

2.1 Theory
The algorithm described by Cohen et al. [2] is formulated for the Laplacians of Eulerian directed graphs. While any directed Laplacian will satisfy \( \mathbf{1}^T \mathbf{L} = \mathbf{0} \), Eulerian Laplacians additionally satisfy \( \mathbf{1}^T \mathbf{L} = \mathbf{0} \). However, even these “nicer” directed Laplacians are not as simple to work with as the Laplacians of undirected graphs, which are positive semidefinite. For this reason, the proof of correctness for the algorithm presented in [2] is quite long and involved. While the theoretical results should still hold for undirected graphs, which can be thought of as special cases of directed graphs with symmetric and equally weighted edges, we hope that there may be simplifications possible for undirected Laplacians. For this project, I aim to adapt the algorithm for the simpler case of undirected graphs, potentially improving the error bounds, and write up a simplified proof of correctness.

2.2 Practice
After determining the appropriate adaptation for undirected graphs, I will implement the algorithm in Julia. The code will necessarily include calls to the undirected Laplacian solver in Professor Spielman’s existing Laplacians.jl library. Additional optimizations may be added for speed, and high-precision testing on ill-conditioned Laplacians will likely be necessary to ensure the algorithm works in practice.

3 Timeline

Background Reading  Weeks 1-2
- Study the existing algorithm and accompanying proofs in detail, including results from related papers
- Study the theoretical basis of condition numbers, as well as current bounds on the largest and smallest eigenvalues of the Laplacian

Simplification of Result  Weeks 3-7
- Adapt the algorithm for undirected Laplacians, and follow the structure of [2] to prove the accompanying error bounds
- Find potential improvements for the undirected case

Implementation and Testing  Weeks 8-10
- Implement the algorithm in Julia
- Evaluate its performance on ill-conditioned systems (using high-precision test data), and compare speed and accuracy to existing methods

Theoretical and Practical Improvement  Weeks 11-12
- If time permits, consider modified reduction schemes that may satisfy the points outlined by Cohen et al. [2]
- Make minor alterations to the Julia code based on testing results
References
