Implementation and Testing of Algorithm for Computing Effective Resistances Between Vertices in Graphs

Kwabena Antwi-Boasiako
Advisor: Prof. Daniel Spielman
January 21, 2010

1 Motivation

The usual method of finding distances between vertices in a graph is not informative for certain applications. A different notion of distance is realized by looking at graphs as resistor networks: given a graph, we can treat each edge as a resistor. If the graph is unweighted, we will assume that the resistor has resistance 1. If an edge $e$ has weight $w(e)$, we will give the corresponding resistor resistance $r(e) = 1/w(e)$. The reason is that when the weight of an edge is very small, the edge is barely there, so it should correspond to very high resistance. Having no edge corresponds to having a resistor of infinite resistance [1]. Another characterization is that in a social network with multiple paths connecting vertices, vertices with more paths connecting them can be thought of as having less resistance between them while isolated vertices can be thought of as having a high resistance between them and the rest of the graph.

2 Algorithm

We can compute the effective resistance between two vertices by solving a system of linear equations in the graph Laplacian but this can be time-consuming. The algorithm by Spielman and Srivastava [2] solves $O(\log n)$ linear equations in the graph Laplacian to compute the vectors that we can then use to approximate effective resistances.
3 What I will be doing

I will be implementing the randomized algorithm discovered by Spielman and Srivastava in their paper [2].\(^1\) In so doing, I will be testing how well the algorithm works on graphs generated from real-life data. Some of the real-life graphs will be obtained by crawling data on the Internet. I will also investigate the role constant factors play in the speed of the algorithm; this will be done to determine whether the algorithm is worthwhile in practice. Also, since the code for the algorithm has not been made publicly available yet, this will be a useful set of tools to make publicly available.

4 Deliverables

These are the items I will produce at the end of the semester:

4.1 Code

The code for the algorithm will be written in Python. I will use a Python package for solving systems of linear equations in Laplacian matrices. This will be used to compute the rows of the matrix from which the effective resistances can be queried.

4.2 Written report

I will submit a written report that will summarize my findings on how the algorithm performed on real-life data. This will include a remark on how useful the algorithm is in practice, i.e. whether the constant factor in the proof of the algorithm’s correctness is actually that large.

References


\(^1\)The proof of the algorithm is in Section 4 Lemma 9 of the paper