Advanced Graph Algorithms and Optimization	Spring 2020

Graded Homework 2

Rasmus Kyng — *Graded Homework 2 — Friday, May 8*

- The solution is due on **Sunday, May 24, 2020** by **12:00 noon**. Please email your solution, which should be written in LaTeX, to abdolahad.noori@inf.ethz.ch. We will send out a confirmation that we have received your file. Make sure you receive this confirmation within the day of the due date, otherwise complain timely.
- We would like to stress that the ETH Disciplinary Code applies to this special assignment as it constitutes part of your final grade. The only exception we make to the Code is that we encourage you to verbally discuss the tasks with your colleagues. It is strictly prohibited to share any (hand)written or electronic (partial) solutions with any of your classmates. We are obligated to inform the Rector of any violations of the Code.

Problem 1: Cuts and Quadratic Forms in Graphs (50 points)

Consider a weighted, undirected graph $G = (V, E, \boldsymbol{w})$, with edge-vertex incidence matrix $\boldsymbol{B} \in \mathbb{R}^{V \times E}$ (using arbitrary directions for each edge as usual). The Laplacian of G is $\boldsymbol{L} = \boldsymbol{B}\operatorname{diag}(\boldsymbol{w})\boldsymbol{B}^\top$. We define for $\boldsymbol{x} \in \mathbb{R}^V$,

$$\|\boldsymbol{x}\|_{G,1} = \sum_{(u,v)\in E} \boldsymbol{w}(e)\left|\boldsymbol{x}(v) - \boldsymbol{x}(u)\right|.$$

Part A. (20 points) Consider two weighted, undirected graphs

- $G = (V, E, \boldsymbol{w})$ with edge-vertex incidence matrix $\boldsymbol{B} \in \mathbb{R}^{V \times E}$.
- $\tilde{G} = (V, \tilde{E}, \tilde{\boldsymbol{w}})$ with edge-vertex incidence matrix $\tilde{\boldsymbol{B}} \in \mathbb{R}^{V \times \tilde{E}}$.

Prove that if $\boldsymbol{L}_G \preceq \boldsymbol{L}_{\tilde{G}}$ then for all $\boldsymbol{x} \in \mathbb{R}^V$

$$\|\boldsymbol{x}\|_{G,1} \leq \|\boldsymbol{x}\|_{\tilde{G},1}.$$

Hint: it's easy to see for $\boldsymbol{x} \in \{0,1\}^V$.

Part B. (20 points) In this problem, we are trying to understand how well Gaussian elimination on a Laplacian preserves the minimum cut.

We use the definitions of Star(,) and Clique(,) from Lecture 9.

Using slight abuse of notation, given a vertex $u \in V$, and a vector $\boldsymbol{x} \in \mathbb{R}^V$, we will use

$$\begin{pmatrix} z \\ \boldsymbol{x}(V \setminus \{u\}) \end{pmatrix}$$

to denote the vector $\boldsymbol{y} \in \mathbb{R}^V$ with $\boldsymbol{y}(u) = z$ and $\boldsymbol{y}(v) = \boldsymbol{x}(v)$ for $v \neq u$.

You should identify two universal constants c_1 and c_2 such that for any weighted, undirected graph $G = (V, E, \boldsymbol{w})$, with Laplacian $\boldsymbol{L}$, for all vertices $u \in V$, letting $\boldsymbol{N} = \textsc{Star}(u, L)$ and $\boldsymbol{C} = \textsc{Clique}(u, L)$, for all $\boldsymbol{x} \in \{0,1\}^V$, letting $\boldsymbol{y} = \begin{pmatrix} z \\ \boldsymbol{x}(V \setminus \{u\}) \end{pmatrix}$, we have

$$\left(\min_{z \in \{0,1\}} \boldsymbol{y}^\top \boldsymbol{N} \boldsymbol{y} \right) c_1 \leq \left(\min_{z \in \{0,1\}} \boldsymbol{y}^\top \boldsymbol{C} \boldsymbol{y} \right) \leq c_2 \left(\min_{z \in \{0,1\}} \boldsymbol{y}^\top \boldsymbol{N} \boldsymbol{y} \right).$$

and you should prove that the stated inequalities hold, i.e. don't just state c_1 and c_2 without proof. You should find the tightest possible bounds for c_1 and c_2 (see Part C for further clarification).

Part C. (10 points) Prove that the constants you found in Part B are tight in the following sense:

- For all $c_1' > c_1$ there exists a graph G s.t. for some $\boldsymbol{x} \in \{0,1\}^V$

$$\left(\min_{z \in \{0,1\}} \boldsymbol{y}^\top \boldsymbol{C} \boldsymbol{y} \right) < \left(\min_{z \in \{0,1\}} \boldsymbol{y}^\top \boldsymbol{N} \boldsymbol{y} \right) c_1'.$$

- For all $c_2' < c_2$ there exists a graph G s.t. for some $\boldsymbol{x}$ $\{0,1\}^V$

$$c_2' \left(\min_{z \in \{0,1\}} \boldsymbol{y}^\top \boldsymbol{N} \boldsymbol{y} \right) < \left(\min_{z \in \{0,1\}} \boldsymbol{y}^\top \boldsymbol{C} \boldsymbol{y} \right).$$

Problem 2: Paper Report (100 points)

You've been assigned a paper to write a report on. In this problem, we ask you to write that report, and give you some pointers about the structure of the report.

You may assume definitions and concepts we have used in the course without restating them, but when you refer to important new concepts from the paper, you should define them.

Your goal should not be to provide a complete restatement of the paper. Instead, imagine you're trying to efficiently communicate key points about the paper to a person who is well-versed in theoretical computer science and optimization, but not familiar with this particular problem and paper.

Part A: Problem statement, contribution, literature review (30 points) Describe the problem that's studied in the paper and the contribution of the paper, i.e. how they advance our understanding of the problem, and what their algorithms/theorems can do. Describe important prior work (based on the background overview from the paper, or your own additional research if necessary). Write 1-2 pages for Part A.

Part B: Technical overview (60 points) Describe the main technical ideas in the paper, including algorithms, theorems, and proofs. Give an outline, which should be top-down and modular:

- Give a high-level, intuitive description of what's going on in the paper.
- Describe the overall structure: Start with main algorithms/proofs, then important subroutines/lemmas that the results rely on.
- If you can identify some key ideas/steps that make the algorithm(s)/proof(s) work out, discuss these.
- If you seems potential mistakes or things that could be made simpler, you can mention these points.

Write 1-4 pages for Part B.

Part C: paper evaluation/"review" (5 points) Think critically about the paper. Is the question interesting? Has the paper made a significant theoretical contribution? Are there indications the authors might be mostly recycling known ideas? Is the contribution likely to be practically relevant[1]? Write up to half a page for Part C.

Part D: open problems (5 points) Are there interesting follow-up questions one could ask about this paper? New areas where these ideas might be useful? Remaining theoretical questions directly related to the problem studied in the paper? Practical uses that could be explored? Write up to half a page for Part D.

[1] That's often hard to know from looking at a theory paper!

Problem 3 (bonus problem!): Laplacians in Disguise (50 points, extra credit!)

This problem is for extra credit. *You can obtain the maximum grade in the course without solving these problems. If you do attempt to solve the problems, points you get will count positively toward your final grade. If you're short on time, make sure you finish Problems 1 and 2 first!*

Consider a symmetric matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$. We say that

- $\boldsymbol{M}$ is *symmetric diagonally dominant* (abbrev. SDD) if for all $i \in [n]$
$$\boldsymbol{M}(i,i) > \sum_{j\neq i} |\boldsymbol{M}(i,j)|$$
- If all off-diagonal entries of $\boldsymbol{M}$ are non-positive (i.e. $\boldsymbol{M}(i,j) \leq 0$ for $i \neq j$), we say that $\boldsymbol{M}$ is a Z-matrix.

In this exercise, you will need a version of the Perron-Frobenius theorem.

Theorem (The Perron-Frobenius Theorem for Symmetric Matrices)**.** Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be a symmetric matrix with non-negative entries. Furthermore, assume that the non-zeros of $\boldsymbol{A}$ correspond to a connected graph when treating $\boldsymbol{A}$ as the adjacency matrix of an (undirected) graph. Let ρ be the largest eigenvalue of $\boldsymbol{A}$. Then $\rho > 0$, and there exists an eigenvector $\boldsymbol{v} \in \mathbb{R}^n$ of $\boldsymbol{A}$ such that $\boldsymbol{v} > \boldsymbol{0}$, and $\boldsymbol{A}\boldsymbol{v} = \rho\boldsymbol{v}$. Furthermore $\|\boldsymbol{A}\| = \rho$.

Part A (5 points) Let $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ be a symmetric and SDD matrix. Prove that $\boldsymbol{M}$ is positive semi-definite.

Hint: look at how we did this for Laplacians. Alternatively, if you know the Gershgorin Circle Theorem, you may appeal to that.

Part B (20 points) Prove the following theorem.

Theorem. Let $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ be a symmetric matrix. If $\boldsymbol{M}$ is positive definite and a Z-matrix, then there exists a diagonal matrix $\boldsymbol{\mathcal{D}}$ with positive entries on the diagonal, such that $\boldsymbol{\mathcal{D}}\boldsymbol{M}\boldsymbol{\mathcal{D}}$ is SDD.

Hint: Find a way to use the Perron-Frobenius theorem.

Part C. (10 points) Let $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ be a symmetric matrix, which is SDD. Let $\boldsymbol{D}$ be the diagonal matrix given by $\boldsymbol{D}(i,i) = \boldsymbol{M}(i,i)$, and let $\boldsymbol{A}$ be the matrix with zeros on the diagonal and $\boldsymbol{A}(i,j) = \boldsymbol{M}(i,j)$ for $i \neq j$. Then $\boldsymbol{M} = \boldsymbol{D} + \boldsymbol{A}$. Further, let $\boldsymbol{A}_+$ be given by $\boldsymbol{A}_+(i,j) = \max(\boldsymbol{A}(i,j), 0)$ and $\boldsymbol{A}_-(i,j) = \min(\boldsymbol{A}(i,j), 0)$.

Consider the matrix $\boldsymbol{N} \in \mathbb{R}^{2n\times 2n}$ whose blocks are given by
$$\boldsymbol{N} = \begin{pmatrix} \boldsymbol{D} + \boldsymbol{A}_- & -\boldsymbol{A}_+ \\ -\boldsymbol{A}_+ & \boldsymbol{D} + \boldsymbol{A}_- \end{pmatrix}.$$

- Prove that $\boldsymbol{N}$ is SDD and a Z-matrix.
- Consider a linear equation $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{b}$. Construct a linear equation $\boldsymbol{N}\boldsymbol{y} = \boldsymbol{c}$ such that given $\boldsymbol{y}$, we can obtain $\boldsymbol{x}$.

Part D. (15 points) Let $\boldsymbol{N} \in \mathbb{R}^{n \times n}$ be a symmetric Z-matrix which is also SDD.

- Show that for some diagonal matrix $\boldsymbol{T} \in \mathbb{R}^{n \times n}$ with non-negative diagonal entries and some Laplacian $\boldsymbol{L} \in \mathbb{R}^{n \times n}$, we have $\boldsymbol{N} = \boldsymbol{T} + \boldsymbol{L}$.
- Consider the linear equation $\boldsymbol{N}\boldsymbol{y} = \boldsymbol{c}$, and assume a solution $\boldsymbol{y}$ exists. Construct a Laplacian matrix $\tilde{\boldsymbol{L}} \in \mathbb{R}^{(n+1) \times (n+1)}$, such that if we have a solution to the equation

  $$\tilde{\boldsymbol{L}}\boldsymbol{x} = \begin{pmatrix} -\boldsymbol{1}^\top \boldsymbol{c} \\ \boldsymbol{c} \end{pmatrix}$$

  then we can recover in $O(n)$ time a vector $\boldsymbol{y}$ such that $\boldsymbol{N}\boldsymbol{y} = \boldsymbol{c}$. Your algorithm for constructing $\tilde{\boldsymbol{L}}$ must run in linear time in the number of non-zeros of $\boldsymbol{N}$.