ETH Zurich

Advanced Graph Algorithms and Optimization

Rasmus Kyng & Maximilian Probst Gutenberg

Spring 2023

These notes will be updated throughout the course. They are likely to contain typos, and they may have mistakes or lack clarity in places. Feedback and comments are welcome. Please send to kyng@inf.ethz.ch or, even better, submit a pull request at

`https://github.com/rjkyng/agao23_script`.

We want to thank scribes from the 2020 edition of the course who contributed to these notes: Hongjie Chen, Meher Chaitanya, Timon Knigge, and Tim Taubner – and we're grateful to all the readers who've submitted corrections, including Martin Kucera, Alejandro Cassis, and Luke Volpatti.

A important note: If you're a student browsing these notes to decide whether to take this course, please note that the current notes are incomplete. We will release parts later in the semester. You can take a look at last year's notes for an impression of the what the rest of the course will look like. Find them here:

`https://github.com/rjkyng/agao22_script/raw/main/agao22_script.pdf`

There will, however, be some changes to the content compared to last year.

Contents

1 Course Introduction **6**

1.1 Overview 6

1.2 Electrical Flows and Voltages - a Graph Problem from Middle School? . . . 6

1.3 Convex Optimization 13

1.4 More Graph Optimization Problems 15

I Introduction to Convex Optimization **18**

2 Some Basic Optimization, Convex Geometry, and Linear Algebra **19**

2.1 Overview 19

2.2 Optimization Problems 19

2.3 A Characterization of Convex Functions 21

2.3.1 First-order Taylor Approximation 22

2.3.2 Directional Derivatives 23

2.3.3 Lower Bounding Convex Functions with Affine Functions 23

2.4 Conditions for Optimality 25

3 Convexity, Second Derivatives and Gradient Descent **26**

3.1 A Review of Linear Algebra 26

3.2 Characterizations of Convexity and Optimality via Second Derivatives 28

3.2.1 A Necessary Condition for Local Extrema 29

3.2.2 A sufficient condition for local extrema 30

3.2.3 Characterization of convexity 30
3.3 Gradient Descent - An Approach to Optimization? 32
3.3.1 A Quantitative Bound on Changes in the Gradient 32
3.3.2 Analyzing Gradient Descent 33

II Spectral Graph Theory 36

4 Introduction to Spectral Graph Theory 37

4.1 Recap: Incidence and Adjacency Matrices, the Laplacian Matrix and Electrical Energy 37
4.2 Understanding Eigenvalues of the Laplacian 39
4.2.1 Test Vector Bounds on λ_2 and λ_n 40
4.2.2 Eigenvalue Bounds Beyond Test Vectors 41
4.2.3 The Loewner Order, aka. the Positive Semi-Definite Order 42
4.2.4 Upper Bounding a Laplacian's λ_n Using Degrees 43
4.2.5 The Loewner Order and Laplacians of Graphs. 43
4.2.6 The Path Inequality 44
4.2.7 Lower Bounding λ_2 of a Path Graph 45
4.2.8 Laplacian Eigenvalues of the Complete Binary Tree 46

5 Conductance, Expanders and Cheeger's Inequality 48

5.1 Conductance and Expanders 49
5.2 A Lower Bound for Conductance via Eigenvalues 50
5.3 An Upper Bound for Conductance via Eigenvalues 52
5.4 Conclusion 55

6 Random Walks 56

6.1 A Primer on Random Walks 56
6.2 Convergence Results for Random Walks 57
6.2.1 Making Random Walks Lazy 58

6.2.2 Convergence of Lazy Random Walks 59
6.2.3 The Rate of Convergence 60
6.3 Properties of Random Walks 62
6.3.1 Hitting Times 62
6.3.2 Commute Time 63

Chapter 1

Course Introduction

1.1 Overview

This course will take us quite deep into modern approaches to graph algorithms using convex optimization techniques. By studying convex optimization through the lens of graph algorithms, we'll try to develop an understanding of fundamental phenomena in optimization. Much of our time will be devoted to flow problems on graphs. We will not only be studying these problems for their own sake, but also because they often provide a useful setting for thinking more broadly about optimization.

The course will cover some traditional discrete approaches to various graph problems, especially flow problems, and then contrast these approaches with modern, asymptotically faster methods based on combining convex optimization with spectral and combinatorial graph theory.

1.2 Electrical Flows and Voltages - a Graph Problem from Middle School?

We will dive right into graph problems by considering how electrical current moves through a network of resistors.

First, let us recall some middle school physics. If some of these things don't make sense to you, don't worry, in less than a paragraph from here, we'll be back to safely doing math.

Recall that a typical battery that one buys from Migros has two endpoints, and produces what is called a *voltage difference* between these endpoints.

One end of the battery will have a positive charge (I think that means an excess of positrons[1]), and the other a negative charge. If we connect the two endpoints with a wire, then a current will flow from one end of the battery to the other in an attempt to even out this imbalance of charge.

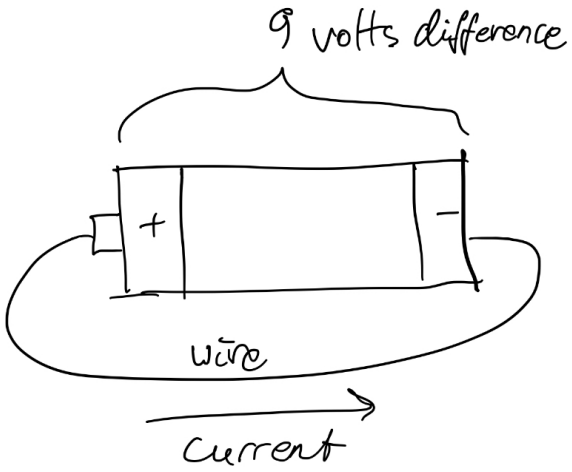

Figure 1.1: A 9 volts battery with a wire attached.

We can also imagine a kind of battery that tries to send a certain amount of current through the wires between its endpoints, e.g. 1 unit of charge per unit of time. This will be a little more convenient to work with, so let us focus on that case.

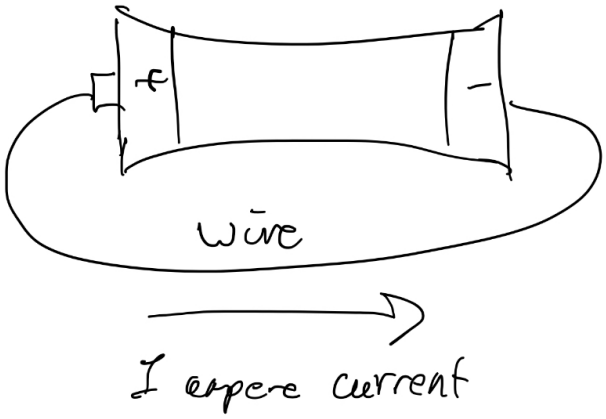

Figure 1.2: A 1 ampere battery with a wire attached.

A *resistor* is a piece of wire that connects two points u and v, and is completely described by a single number r called its *resistance*.

[1] I'm joking, of course! Try Wikipedia if you want to know more. However, you will not need it for this class.

If the voltage difference between the endpoints of the resistor is x, and the resistance is r then this will create a flow of charge per unit of time of $f = x/r$. This is called Ohm's Law.

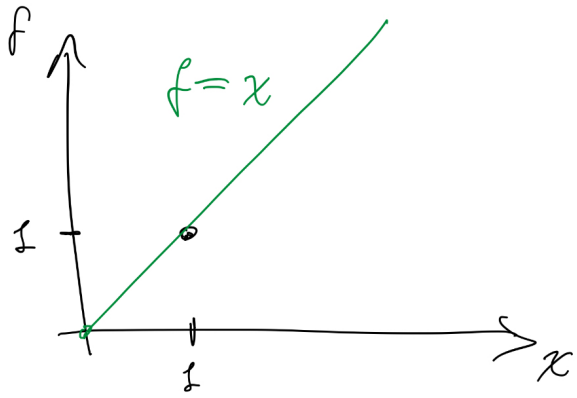

Figure 1.3: Ohm's Law for a resistor with resistance $r = 1$.

Suppose we set up a bunch of wires that route electricity from our current source s to our current sink t in some pattern:

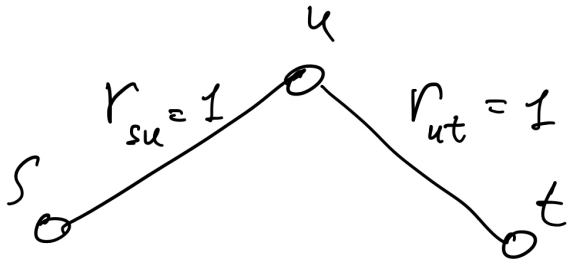

Figure 1.4: A path of two resistors.

We have one unit of charge flowing out of s per unit of time, and one unit coming into t. Because charge is conserved, the current flowing into any other point u must equal the amount flowing out of it. This is called Kirchhoff's Current Law.

To send one unit of current from s to t, we must be sending it first from s to u and then from u to t. So the current on edge (s, u) is 1 and the current on (u, t) is 1. By Ohm's Law, the voltage difference must also be 1 across each of the two wires. Thus, if the voltage is x at s, it must be $x + 1$ at u and $x + 2$ at t. What is x? It turns out it doesn't matter: We only care about the differences. So let us set $x = 0$.

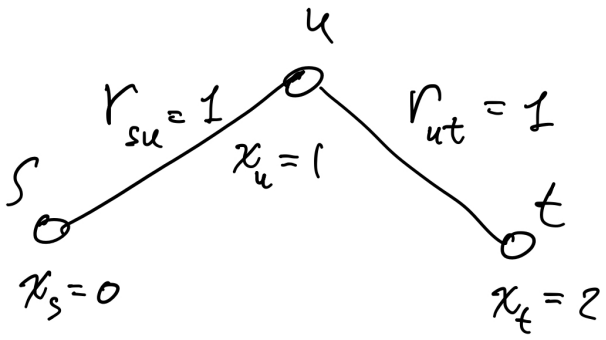

Figure 1.5: A path of two resistors.

Let us try one more example:

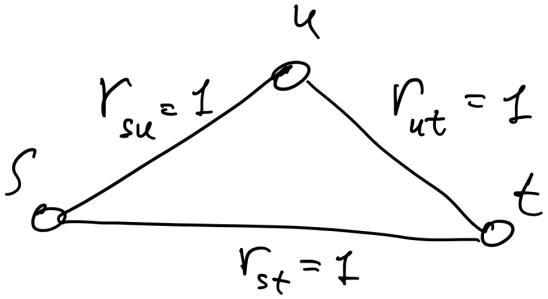

Figure 1.6: A network with three resistors.

How much flow will go directly from s to t and how much via u?

Well, we know what the net current flowing into and out of each vertex must be, and we can use that to set up some equations. Let us say the voltage at s is x_s, at u is x_u and at t is x_t.

- Net current at s: $-1 = (x_s - x_t) + (x_s - x_u)$
- Net current at u: $0 = (x_u - x_s) + (x_u - x_t)$
- Net current at t: $1 = (x_t - x_s) + (x_t - x_u)$

The following is a solution: $x_s = 0$, $x_u = \frac{1}{3}$, $x_t = \frac{2}{3}$. And as before, we can shift all the voltages by some constant x and get another solution $x_s = x + 0$, $x_u = x + \frac{1}{3}$, $x_t = x + \frac{2}{3}$. You might want to convince yourself that these are the only solutions.

Electrical flows in general graphs. Do we know enough to calculate the electrical flow in some other network of resistors? To answer this, let us think about the network as a graph. Consider an undirected graph $G = (V, E)$ with $|V| = n$ vertices and $|E| = m$ edges, and let us assume G is connected. Let's associate a resistance $\boldsymbol{r}(e) > 0$ with every edge $e \in E$.

To keep track of the direction of the flow on each edge, it will be useful to assign an arbitrary direction to every edge. So let's do that, but remember that this is just a bookkeeping tool that helps us track where flow is going.

A *flow* in the graph is a vector $\boldsymbol{f} : \mathbb{R}^E$. The *net flow* of $\boldsymbol{f}$ at a vertex $u \in V$ is defined as $\sum_{v \to u} \boldsymbol{f}(v,u) - \sum_{u \to v} \boldsymbol{f}(u,v)$.

We say a flow routes the demands $\boldsymbol{d} \in \mathbb{R}^V$ if the net flow at every vertex v is $\boldsymbol{d}(v)$.

We can assign a voltage to every vertex $\boldsymbol{x} \in R^V$. Ohm's Law says that the electrical flow induced by these voltages will be $\boldsymbol{f}(u,v) = \frac{1}{\boldsymbol{r}(u,v)}(\boldsymbol{x}(v) - \boldsymbol{x}(u))$.

Say we want to route one unit of current from vertex $s \in V$ to vertex $t \in V$. As before, we can write an equation for every vertex saying that the voltage differences must produce the desired net current:

- Net current at s: $-1 = \sum_{(s,v)} \frac{1}{\boldsymbol{r}(s,v)}(\boldsymbol{x}(v) - \boldsymbol{x}(s))$
- Net current at $u \in V \setminus \{s,t\}$: $0 = \sum_{(u,v)} \frac{1}{\boldsymbol{r}(u,v)}(\boldsymbol{x}(v) - \boldsymbol{x}(u))$
- Net current at t: $1 = \sum_{(t,v)} \frac{1}{\boldsymbol{r}(t,v)}(\boldsymbol{x}(v) - \boldsymbol{x}(t))$

This gives us n constraints, exactly as many as we have voltage variables. However we have to be a little careful when trying to conclude that a solution exists, yielding voltages $\boldsymbol{x}$ that gives induce an electrical flow routing the desired demand.

You will prove in the exercises (Week 1, Exercise 3) that a solution $\boldsymbol{x}$ exists. The proof requires two important observations: Firstly that the graph is connected, and secondly that summed over all vertices, the net demand is zero, i.e. as much flow is coming into the network as is leaving it.

The incidence matrix and the Laplacian matrix. To have a more compact notation for net flow constraints, we also introduce the *edge-vertex incidence matrix* of the graph, $\boldsymbol{B} \in \mathbb{R}^{V \times E}$.

$$\boldsymbol{B}(v,e) = \begin{cases} 1 & \text{if } e = (u,v) \\ -1 & \text{if } e = (v,u) \\ 0 & \text{o.w.} \end{cases}$$

Now we can express the net flow constraint that $\boldsymbol{f}$ routes $\boldsymbol{d}$ by

$$\boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}.$$

This is also called a conservation constraint. In our examples so far, we have $\boldsymbol{d}(s) = -1$, $\boldsymbol{d}(t) = 1$ and $\boldsymbol{d}(u) = 0$ for all $u \in V \setminus \{s,t\}$.

If we let $\boldsymbol{R} = \mathrm{diag}_{e \in E}\, \boldsymbol{r}(e)$ then Ohm's law tells us that $\boldsymbol{f} = \boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{x}$. Putting these observations together, we have $\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{x} = \boldsymbol{d}$. The voltages $\boldsymbol{x}$ that induce $\boldsymbol{f}$ must solve this system of linear equations, and we can use that to compute both $\boldsymbol{x}$ and $\boldsymbol{f}$. It is exactly

the same linear equation as the one we considered earlier. We can show that for a connected graph, a solution $\boldsymbol{x}$ exists if and only if the flow into the graph equals the net flow out, which we can express as $\sum_v \boldsymbol{d}(v) = 0$ or $\mathbf{1}^\top \boldsymbol{d} = 0$. You will show this as part of Exercise 3. This also implies that an electrical flow routing $\boldsymbol{d}$ exists if and only if the net flow into the graph equals the net flow out, which we can express as $\mathbf{1}^\top \boldsymbol{d} = 0$.

The matrix $\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top$ is called the *Laplacian* of the graph and is usually denoted by $\boldsymbol{L}$.

An optimization problem in disguise. So far, we have looked at electrical voltages and flows as arising from a set of linear equations – and it might not be apparent that this has anything to do with optimization. But transporting current through a resistor requires energy, which will be dissipated as heat by the resistor (i.e. it will get hot!). If we send a current of f across a resistor with a potential drop of x, then the amount of energy spent per unit of time by the resistor will be $f \cdot x$. This is called Joule's Law. Applying Ohm's law to a resistor with resistance r, we can also express this energy per unit of time as $f \cdot x = x^2/r = r \cdot f^2$. Since we aren't bothering with units, we will even forget about time, and refer to these quantities as "energy", even though a physicist would call them "power".

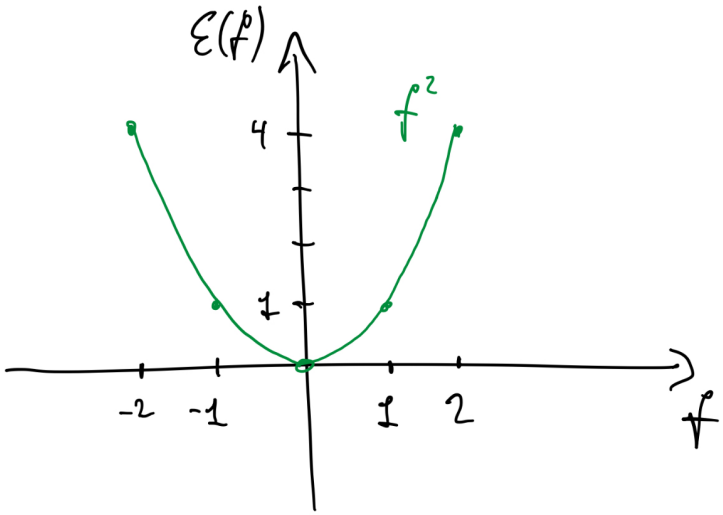

Figure 1.7: Energy as a function of flow in a resistor with resistance $r = 1$.

Now, another interesting question would seem to be: If we want to find a flow routing a certain demand $\boldsymbol{d}$, how should the flow behave in order to minimize the electrical energy spent routing the flow? The electrical energy of a flow vector $\boldsymbol{f}$ is $\mathcal{E}(\boldsymbol{f}) \stackrel{\text{def}}{=} \sum_e \boldsymbol{r}(e)\boldsymbol{f}(e)^2$. We can phrase this as an optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{f} \in \mathbb{R}^E} \mathcal{E}(\boldsymbol{f}) \\ &\text{s.t. } \boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}. \end{aligned}$$

We call this problem *electrical energy-minimizing flow*. As we will prove later, the flow $\boldsymbol{f}^*$ that minimizes the electrical energy among all flows that satisfy $\boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}$ is precisely the electrical flow.

A pair of problems. What about our voltages, can we also get them from some optimization problem? Well, we can work backwards from the fact that our voltages solve the equation $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{d}$. Consider the function $c(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} - \boldsymbol{x}^\top \boldsymbol{d}$. We should ask ourselves some questions about this function $c : \mathbb{R}^V \to \mathbb{R}$. Is it continuous and continuously differentiable? The answer to this is yes, and that is not hard to see. Does the function have a minimum? This is maybe not immediately clear, but the minimum does indeed exist.

When this is minimized, the derivative of $c(\boldsymbol{x})$ with respect to each coordinate of $\boldsymbol{x}$ must be zero. This condition yields exactly the system of linear equations $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{d}$. You will confirm this in Exercise 4 of the first exercise sheet.

Based on our derivative condition for the optimum, we can also express the electrical voltages as the solution to an optimization problem, namely

$$\min_{\boldsymbol{x}\in\mathbb{R}^V} c(\boldsymbol{x})$$

As you are probably aware, having the derivative of each coordinate equal zero is not a sufficient condition for being at the optimum of a function[2].

It is also interesting to know whether *all* solutions to $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{d}$ are in fact minimizers of c. The answer is yes, and we will see some very general tools for proving statements like this in Chapter 2.

Altogether, we can see that routing electrical current through a network of resistors leads to a *pair* of optimization problems, let's call them $\boldsymbol{f}^*$ and $\boldsymbol{x}^*$, and that the solutions to the two problems are related, in our case through the equation $\boldsymbol{f}^* = \boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{x}^*$ (Ohm's Law). But why and how are these two optimization problems related?

Instead of minimizing $c(\boldsymbol{x})$, we can equivalently think about maximizing $-c(\boldsymbol{x})$, which gives the following optimization problem: $\max_{\boldsymbol{x}\in\mathbb{R}^V} -c(\boldsymbol{x})$. In fact, as you will show in the exercises for Week 1, we have $\mathcal{E}(\boldsymbol{f}^*) = -c(\boldsymbol{x}^*)$, so the minimum electrical energy is exactly the maximum value of $-\boldsymbol{c}(\boldsymbol{x})$. More generally for *any* flow that routes $\boldsymbol{d}$ and *any* voltages $\boldsymbol{x}$, we have $\mathcal{E}(\boldsymbol{f}) \geq -c(\boldsymbol{x})$. So, for any $\boldsymbol{x}$, the value of $-c(\boldsymbol{x})$ is a lower bound on the minimum energy $\mathcal{E}(\boldsymbol{f}^*)$.

This turns out to be an instance of a much broader phenomenon, known as Lagrangian duality, which allows us to learn a lot about many optimization problems by studying two related pairs of problems, a minimization problem, and a related maximization problem that gives lower bounds on the optimal value of the minimization problem.

[2]Consider the function in one variable $c(x) = x^3$.

Solving $\boldsymbol{Lx} = \boldsymbol{d}$. Given a graph G with resistances for the edges, and some net flow vector $\boldsymbol{d}$, how quickly can we compute $\boldsymbol{x}$? Broadly speaking, there are two very different families of algorithms we could use to try to solve this problem.

We could solve the linear equation using something like *Gaussian Elimination* to compute an exact solution.

Alternatively, we could start with a guess at a solution, e.g. $\boldsymbol{x}_0 = \mathbf{0}$, and then we could try to make a change to $\boldsymbol{x}_0$ to reach a new point $\boldsymbol{x}_1$ with a lower value of $c(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} - \boldsymbol{x}^\top \boldsymbol{d}$, i.e. $c(\boldsymbol{x}_1) < c(\boldsymbol{x}_0)$. If we repeat a process like that for enough steps, say t, hopefully we eventually reach $\boldsymbol{x}_t$ with $c(\boldsymbol{x}_t)$ close to $c(\boldsymbol{x}^*)$, where $\boldsymbol{x}^*$ is a minimizer of $c(\boldsymbol{x})$ and hence $\boldsymbol{L}\boldsymbol{x}^* = \boldsymbol{d}$. Now, we also need to make sure that $c(\boldsymbol{x}_t) \approx c(\boldsymbol{x}^*)$ implies that $\boldsymbol{L}\boldsymbol{x}_t \approx \boldsymbol{d}$ in some useful sense.

One of the most basic algorithms in this framework of "guess and adjust" is called *Gradient Descent*, which we will study in Week 2. The rough idea is the following: if we make a very small step from $\boldsymbol{x}$ to $\boldsymbol{x} + \boldsymbol{\delta}$, then a multivariate Taylor expansion suggests that $c(\boldsymbol{x} + \boldsymbol{\delta}) - c(\boldsymbol{x}) \approx \sum_{v \in V} \boldsymbol{\delta}(v) \frac{\partial c(\boldsymbol{x})}{\partial \boldsymbol{x}(v)}$.

If we are dealing with smooth convex function, this quantity is negative if we let $\boldsymbol{\delta}(v) = -\epsilon \cdot \frac{\partial c(\boldsymbol{x})}{\partial \boldsymbol{x}(v)}$ for some small enough ϵ so the approximation holds well. So we should be able to make progress by taking a small step in this direction. That's Gradient Descent! The name comes from the vector of partial derivatives, which is called the gradient.

As we will see later in this course, understanding electrical problems from an optimization perspective is crucial to develop fast algorithms for computing electrical flows and voltages, but to do very well, we also need to borrow some ideas from Gaussian Elimination.

What running times do different approaches get?

1. Using Gaussian Elimination, we can find $\boldsymbol{x}$ s.t. $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{d}$ in $O(n^3)$ time and with asymptotically faster algorithms based on matrix multiplication, we can bring this down to roughly $O(n^{2.372})$.

2. Meanwhile Gradient Descent will get a running time of $O(n^3 m)$ or so – at least this is a what a simple analysis suggests.

3. However, we can do much better: By combining ideas from both algorithms, and a bit more, we can get $\boldsymbol{x}$ up to very high accuracy in time $O(m \log^c n)$ where c is some small constant.

1.3 Convex Optimization

Recall our plot in Figure 1.7 of the energy required to route a flow f across a resistor with resistance r, which was $\mathcal{E}(f) = r \cdot f^2$. We see that the function has a special structure: the

graph of the function sits below the line joining any two points $(f, \mathcal{E}(f))$ and $(g, \mathcal{E}(g))$. A function $\mathcal{E} : \mathbb{R} \to \mathbb{R}$ that has this property is said to be convex.

Figure 1.8 shows the energy as a function of flow, along with two points $(f, \mathcal{E}(f))$ and $(g, \mathcal{E}(g))$. We see the function sits below the line segment between these points.

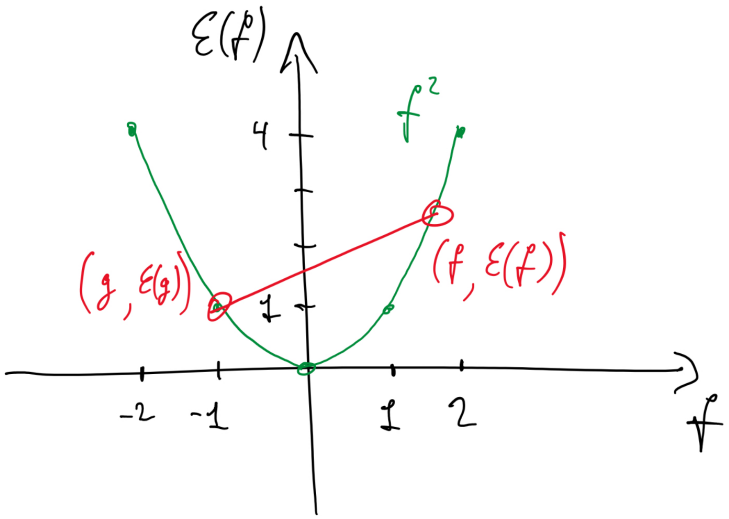

Figure 1.8: Energy as a function of flow in a resistor with resistance $r = 1$. The function is convex.

We can also interpret this condition as saying that for all $\theta \in [0, 1]$

$$\mathcal{E}(\theta f + (1 - \theta)g) \leq \theta \mathcal{E}(f) + (1 - \theta)\mathcal{E}(g).$$

This immediately generalizes to functions $\mathcal{E} : \mathbb{R}^m \to \mathbb{R}$.

A *convex set* is a subset of $S \subseteq \mathbb{R}^m$ s.t. if $\boldsymbol{f}, \boldsymbol{g} \in S$ then for all $\theta \in [0, 1]$ we have $\theta \boldsymbol{f} + (1-\theta)\boldsymbol{g} \in S$.

Figure 1.9 shows some examples of sets that are and aren't convex.

Convex functions and convex sets are central to optimization, because for most problems of minimization of a convex function over a convex set, we can develop fast algorithms [3].

So why convex functions and convex sets? One important reason is that for a convex function defined over a convex feasible set, any local minimum is also a global minimum, and this fact makes searching for an optimal solution computationally easier. In fact, this is closely related to why Gradient Descent works well on many convex functions.

Notice that the set $\{\boldsymbol{f} : \boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}\}$ is convex, i.e. the set of all flows that route a fixed demand

[3] There are some convex optimization problems that are NP-hard. That said, polynomial time algorithms exist for almost any convex problem you can come up with. The most general polynomial time algorithm for convex optimization is probably the Ellipsoid Method.

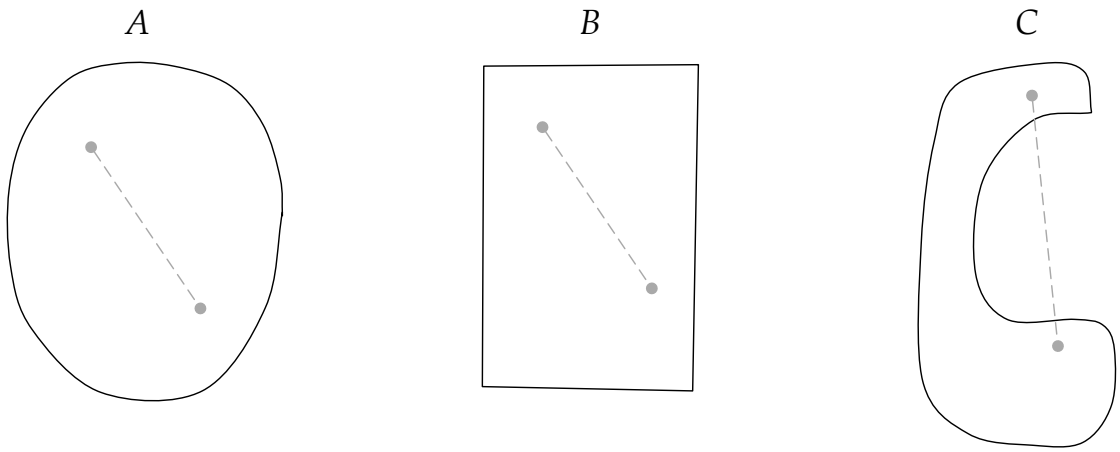

Figure 1.9: A depiction of convex and non-convex sets. The sets A and B are convex since the straight line between any two points inside them is also in the set. The set C is not convex.

$\boldsymbol{d}$ is convex. It is also easy to verify that $\mathcal{E}(\boldsymbol{f}) = \sum_e \boldsymbol{r}(e)\boldsymbol{f}(e)^2$ is a convex function, and hence finding an electrical flow is an instance of convex minmization:

1.4 More Graph Optimization Problems

Maximum flow. Again, let $G = (V, E)$ be an undirected, connected graph with n vertices and m edges. Suppose we want to find a flow $\boldsymbol{f} \in \mathbb{R}^E$ that routes $\boldsymbol{d}$, but instead of trying to minimize electrical energy, we try to pick an $\boldsymbol{f}$ that minimizes the largest amount of flow on any edge, i.e. $\max_e |\boldsymbol{f}_e|$ – which we also denote by $\|\boldsymbol{f}\|_\infty$. We can write this problem as

$$\begin{aligned} &\min_{\boldsymbol{f} \in \mathbb{R}^E} \|\boldsymbol{f}\|_\infty \\ &\text{s.t. } \boldsymbol{B}\boldsymbol{f} = \boldsymbol{d} \end{aligned}$$

This problem is known as the Minimum Congested Flow Problem[4]. It is equivalent to the more famous Maximum Flow Problem.

The behavior of this kind of flow is very different than electrical flow. Consider the question of whether a certain demand can be routed $\|\boldsymbol{f}\|_\infty \leq 1$. Imagine sending goods from a source s to a destination t using a network of train lines that all have the same capacity and asking whether the network is able to route the goods at the rate you want: This boils down to whether routing exists with $\|\boldsymbol{f}\|_\infty \leq 1$, if we set it up right.

We have a very fast, convex optimization-based algorithm for Minimum Congested Flow: In $m\epsilon^{-1}\log^{O(1)} n$ time, we can find a flow $\tilde{\boldsymbol{f}}$ s.t. $\boldsymbol{B}\tilde{\boldsymbol{f}} = \boldsymbol{d}$ and $\left\|\tilde{\boldsymbol{f}}\right\|_\infty \leq (1+\epsilon)\|\boldsymbol{f}^*\|_\infty$, where $\boldsymbol{f}^*$

[4]This version is called undirected, because the graph is undirected, and *uncapacitated* because we are aiming for the same bound on the flow on all edges.

is an optimal solution, i.e. an actual minimum congestion flow routing $\boldsymbol{d}$.

But what if we want ϵ to be very small, e.g. $1/m$? Then this running time isn't so good anymore. But, in this case, we can use other algorithms that find flow $\boldsymbol{f}^*$ *exactly*. Unfortunately, these algorithms take time roughly $m^{4/3+o(1)}$.

Just as the electrical flow problem had a dual voltage problem, so maximum flow has a dual voltage problem, which is know as the s-t minimum cut problem.

Maximum flow, with directions and capacities. We can make the maximum flow problem harder by introducing directed edges: To do so, we allow edges to exist in both directions between vertices, and we require that the flow on a directed edge is always non-negative. So now $G = (V, E)$ is a directed graph. We can also make the problem harder by introducing capacities. We define a capacity vector $\boldsymbol{c} \in \mathbb{R}^E \geq \mathbf{0}$ and try to minimize $\left\| \boldsymbol{C}^{-1}\boldsymbol{f} \right\|_\infty$, where $\boldsymbol{C} = \operatorname{diag}_{e \in E} \boldsymbol{c}(e)$. Then our problem becomes

$$\begin{aligned} \min_{\boldsymbol{f} \in \mathbb{R}^E} & \left\| \boldsymbol{C}^{-1}\boldsymbol{f} \right\|_\infty \\ \text{s.t. } & \boldsymbol{B}\boldsymbol{f} = \boldsymbol{d} \\ & \boldsymbol{f} \geq \mathbf{0}. \end{aligned}$$

For this capacitated, directed maximum flow problem, our best algorithms run in about $O(m\sqrt{n})$ time in sparse graphs and $O(m^{1.483})$ in dense graphs[5], even if we are willing to accept fairly low accuracy solution. If the capacities are allowed to be exponentially large, the best running time we can get is $O(mn)$. For this problem, we do not yet know how to improve over classical combinatorial algorithms using convex optimization.

Multi-commodity flow. We can make the problem even harder still, by simultaneously trying to route two types of flow (imagine pipes with Coke and Pepsi). Our problem now looks like

$$\begin{aligned} \min_{\boldsymbol{f}_1, \boldsymbol{f}_2 \in \mathbb{R}^E} & \left\| \boldsymbol{C}^{-1}(\boldsymbol{f}_1 + \boldsymbol{f}_2) \right\|_\infty \\ \text{s.t. } & \boldsymbol{B}\boldsymbol{f}_1 = \boldsymbol{d}_1 \\ & \boldsymbol{B}\boldsymbol{f}_2 = \boldsymbol{d}_2 \\ & \boldsymbol{f}_1, \boldsymbol{f}_2 \geq \mathbf{0}. \end{aligned}$$

Solving this problem to high accuracy is essentially as hard as solving a general linear program! We should see later in the course how to make this statement precise.

If we in the above problem additionally require that our flows must be integer valued, i.e. $\boldsymbol{f}_1, \boldsymbol{f}_2 \in \mathbb{N}_0$, then the problem becomes NP-complete.

[5]Provided the capacities are integers satisfying a condition like $\boldsymbol{c} \leq n^{100}\mathbf{1}$.

Random walks in a graph. Google famously uses[6] the PageRank problem to help decide how to rank their search results. This problem essentially boils down to computing the *stable distribution* of a random walk on a graph. Suppose $G = (V, E)$ is a directed graph where each outgoing edge (v, u), which we will define as going from u to v, has a transition probability $p_{(v,u)} > 0$ s.t. $\sum_{z \leftarrow u} p_{(z,u)} = 1$. We can take a step of a random walk on the vertex set by starting at some vertex $u_0 = u$, and then randomly pick one of the outgoing edges (v, u) with probability $p_{(v,u)}$ and move to the chosen vertex $u_1 = v$. Repeating this procedure, to take a step from the next vertex u_1, gives us a *random walk* in the graph, a sequence of vertices $u_0, u_1, u_2 \ldots, u_k$.

We let $\boldsymbol{P} \in \mathbb{R}^{V \times V}$ be the matrix of transition probabilities given by

$$\boldsymbol{P}_{vu} = \begin{cases} p_{(v,u)} & \text{for } (u, v) \in E \\ 0 & \text{o.w.} \end{cases}$$

Any probability distribution over the vertices can be specified by a vector $\boldsymbol{p} \in \mathbb{R}^V$ where $\boldsymbol{p} \geq \mathbf{0}$ and $\sum_v \boldsymbol{p}(v) = 1$. We say that probability distribution $\boldsymbol{\pi}$ on the vertices is a *stable distribution* of the random walk if $\boldsymbol{\pi} = \boldsymbol{P}\boldsymbol{\pi}$. A strongly connected graph always has exactly one stable distribution.

How quickly can we compute the stable distribution of a general random walk? Under some mild conditions on the stable distribution[7], we can find a high accuracy approximation of $\boldsymbol{\pi}$ in time $O(m \log^c n)$ for some constant c.

This problem does not easily fit in a framework of convex optimization, but nonetheless, our fastest algorithms for it use ideas from convex optimization.

Topics in this Course

In this course, we will try to address the following questions.

1. What are the fundamental tools of fast convex optimization?
2. What are some problems we can solve quickly on graphs using optimization?
3. What can graphs teach us about convex optimization?
4. What algorithm design techniques are good for getting algorithms that quickly find a crude approximate solution? And what techniques are best when we need to get a highly accurate answer?
5. What is special about flow problems?

[6] At least they did at some point.

[7] Roughly something like $\max_v 1/\boldsymbol{\pi}(v) \leq n^{100}$.

Part I

Introduction to Convex Optimization

Chapter 2

Some Basic Optimization, Convex Geometry, and Linear Algebra

2.1 Overview

In this chapter, we will

1. Start with an overview (i.e. this list).
2. Learn some basic terminology and facts about optimization.
3. Recall our definition of convex functions and see how convex functions can also be understood in terms of a characterization based on first derivatives.
4. See how the first derivatives of a convex function can certify that we are at a global minimum.

2.2 Optimization Problems

Focusing for now on optimization over $\boldsymbol{x} \in \mathbb{R}^n$, we usually write optimization problems as:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \text{ (or } \max) \; & f(\boldsymbol{x}) \\ s.t. \; & g_1(\boldsymbol{x}) \leq b_1 \\ & . \\ & . \\ & . \\ & g_m(\boldsymbol{x}) \leq b_m \end{aligned}$$

where $\{g_i(\boldsymbol{x})\}_{i=1}^m$ encode the constraints. For example, in the following optimization problem from the previous chapter

$$\min_{\boldsymbol{f} \in \mathbb{R}^E} \sum_e \boldsymbol{r}(e)\boldsymbol{f}(e)^2$$
$$\text{s.t. } \boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}$$

we have the constraint $\boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}$. Notice that we can rewrite this constraint as $\boldsymbol{B}\boldsymbol{f} \leq \boldsymbol{d}$ and $-\boldsymbol{B}\boldsymbol{f} \leq -\boldsymbol{d}$ to match the above setting. The set of points which respect the constraints is called the *feasible set*.

Definition 2.2.1. For a given optimization problem the set $\mathcal{F} = \{\boldsymbol{x} \in \mathbb{R}^n \ : \ g_i(\boldsymbol{x}) \leq b_i, \forall i \in [m]\}$ is called the **feasible set**. A point $\boldsymbol{x} \in \mathcal{F}$ is called a **feasible point**, and a point $\boldsymbol{x}' \notin \mathcal{F}$ is called an **infeasible point**.

Ideally, we would like to find optimal solutions for the optimization problems we consider. Let's define what we mean exactly.

Definition 2.2.2. For a *maximization* problem $\boldsymbol{x}^\star$ is called an **optimal solution** if $f(\boldsymbol{x}^\star) \geq f(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{F}$. Similarly, for a *minimization* problem $\boldsymbol{x}^\star$ is an optimal solution if $f(\boldsymbol{x}^\star) \leq f(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{F}$.

What happens if there are *no feasible points*? In this case, an optimal solution cannot exist, and we say the problem is infeasible.

Definition 2.2.3. If $\mathcal{F} = \emptyset$ we say that the optimization problem is **infeasible**. If $\mathcal{F} \neq \emptyset$ we say the optimization problem is **feasible**.

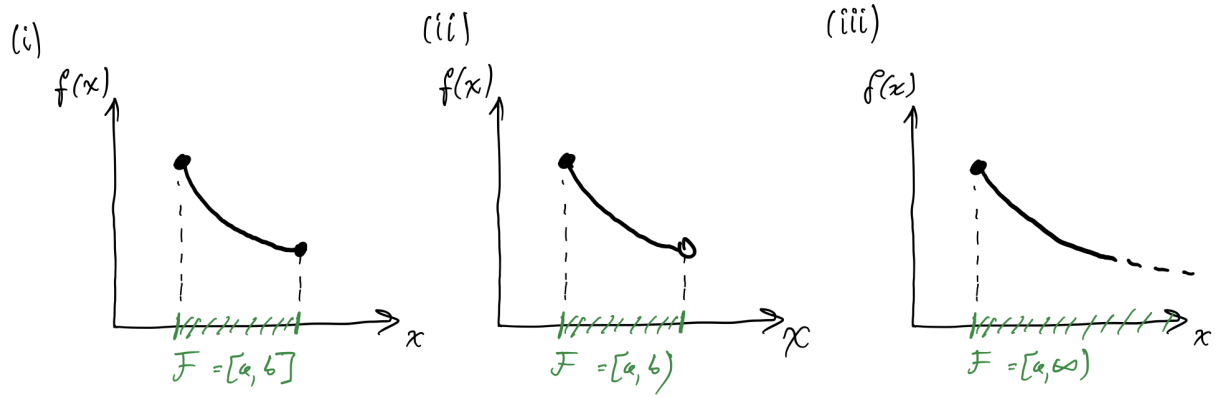

Figure 2.1

Consider three examples depicted in Figure 2.1:

(i) $\mathcal{F} = [a, b]$

(ii) $\mathcal{F} = [a, b)$

(iii) $\mathcal{F} = [a, \infty)$

In the first example, the minimum of the function is attained at b. In the second case the region is open and therefore there is no minimum function value, since for every point we will choose, there will always be another point with a smaller function value. Lastly, in the third example, the region is unbounded and the function decreasing, thus again there will always be another point with a smaller function value.

Sufficient Condition for Optimality. The following theorem, which is a fundamental theorem in real analysis, gives us a sufficient (though not necessary) condition for optimality.

Theorem (Extreme Value Theorem)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function and $\mathcal{F} \subseteq \mathbb{R}^n$ be nonempty, bounded, and closed. Then, the optimization problem $\min f(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{F}$ has an optimal solution.

2.3 A Characterization of Convex Functions

Recall the definitions of convex sets and convex functions that we introduced in Chapter 1:

Definition 2.3.1. A set $S \subseteq \mathbb{R}^n$ is called a **convex set** if any two points in S contain their line, i.e. for any $\boldsymbol{x}, \boldsymbol{y} \in S$ we have that $\theta\boldsymbol{x} + (1-\theta)\boldsymbol{y} \in S$ for any $\theta \in [0, 1]$.

Definition 2.3.2. For a convex set $S \subseteq \mathbb{R}^n$, we say that a function $f : S \to \mathbb{R}$ is **convex on** S if for any two points $\boldsymbol{x}, \boldsymbol{y} \in S$ and any $\theta \in [0, 1]$ we have that:

$$f\left(\theta\boldsymbol{x} + (1-\theta)\boldsymbol{y}\right) \leq \theta f(\boldsymbol{x}) + \Big(1-\theta\Big) f(\boldsymbol{y}).$$

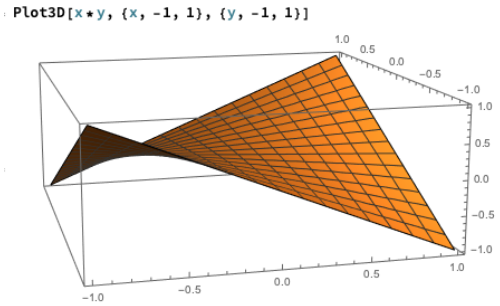

Figure 2.2: This plot shows the function $f(x, y) = xy$. For any fixed y_0, the function $h(x) = f(x, y_0) = xy_0$ is this is linear in x, and so is a convex function in x. But is f convex?

We will first give an important characterization of convex function. To do so, we need to characterize multivariate functions via their Taylor expansion.

Notation for this section. In the rest of this section, we frequently consider a multivariate function f whose domain is a set $S \subseteq \mathbb{R}^n$, which we will require to be open. When we additionally require that S is convex, we will specify this. Note that $S = \mathbb{R}^n$ is both open and convex and it suffices to keep this case in mind. Things sometimes get more complicated if S is not open, e.g. when the domain of f has a boundary. We will leave those complications for another time.

2.3.1 First-order Taylor Approximation

Definition 2.3.3. The **gradient** of a function $f : S \to \mathbb{R}$ at point $\boldsymbol{x} \in S$ is denoted $\boldsymbol{\nabla} f(\boldsymbol{x})$ is:

$$\boldsymbol{\nabla} f(\boldsymbol{x}) = \left[\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}(1)}, \ldots, \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}(n)}\right]^\top$$

First-order Taylor expansion. For a function $f : \mathbb{R} \to \mathbb{R}$ of a single variable, differentiable at $x \in \mathbb{R}$

$$f(x + \delta) = f(x) + f'(x)\delta + o(|\delta|)$$

where by definition:

$$\lim_{\delta \to 0} \frac{o(|\delta|)}{|\delta|} = 0.$$

Similarly, a multivariate function $f : S \to \mathbb{R}$ is said to be *(Fréchet) differentiable* at $\boldsymbol{x} \in S$ when there exists $\boldsymbol{\nabla} f(\boldsymbol{x}) \in \mathbb{R}^n$ s.t.

$$\lim_{\boldsymbol{\delta} \to \mathbf{0}} \frac{\left\| f(\boldsymbol{x} + \boldsymbol{\delta}) - f(\boldsymbol{x}) - \boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{\delta} \right\|_2}{\|\boldsymbol{\delta}\|_2} = 0.$$

Note that this is equivalent to saying that $f(\boldsymbol{x} + \boldsymbol{\delta}) = f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{\delta} + o(\|\boldsymbol{\delta}\|_2)$.

We say that f is *continuously differentiable* on a set $S \subseteq \mathbb{R}^n$ if it is differentiable and in addition the gradient is continuous on S. A differentiable convex function whose domain is an open convex set $S \subseteq \mathbb{R}^n$ is always continuously differentiable[1].

Remark. In this course, we will generally err on the side of being informal about functional analysis when we can afford to, and we will not worry too much about the details of different notions of differentiability (e.g. Fréchet and Gateaux differentiability), except when it turns out to be important.

Theorem 2.3.4 (Taylor's Theorem, multivariate first-order remainder form)**.** *If $f : S \to \mathbb{R}$ is continuously differentiable over $[\boldsymbol{x}, \boldsymbol{y}]$, then for some $\boldsymbol{z} \in [\boldsymbol{x}, \boldsymbol{y}]$,*

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top (\boldsymbol{y} - \boldsymbol{x}).$$

[1] See p. 248, Corollary 25.5.1 in *Convex Analysis* by Rockafellar (my version is the Second print, 1972). Rockefellar's corollary concerns finite convex functions, because he otherwise allows convex functions that may take on the values $\pm\infty$.

This theorem is useful for showing that the function f can be approximated by the affine function $\boldsymbol{y} \to f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$ when $\boldsymbol{y}$ is "close to" $\boldsymbol{x}$ in some sense.

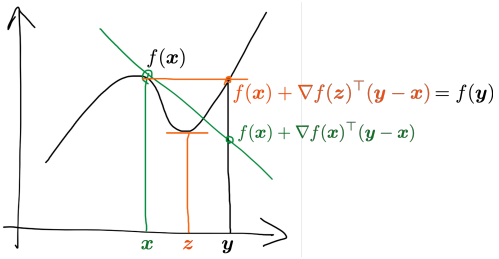

Figure 2.3: The convex function $f(\boldsymbol{y})$ sits above the linear function in $\boldsymbol{y}$ given by $f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$.

2.3.2 Directional Derivatives

Definition 2.3.5. Let $f : S \to \mathbb{R}$ be a function differentiable at $\boldsymbol{x} \in S$ and let us consider $\boldsymbol{d} \in \mathbb{R}^n$. We define the **derivative of f at $\boldsymbol{x}$ in direction $\boldsymbol{d}$** as:

$$Df(\boldsymbol{x})[\boldsymbol{d}] = \lim_{\lambda \to \boldsymbol{0}} \frac{f(\boldsymbol{x} + \lambda \boldsymbol{d}) - f(\boldsymbol{x})}{\lambda}$$

Proposition 2.3.6. $Df(\boldsymbol{x})[\boldsymbol{d}] = \boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{d}$.

Proof. Using the first order expansion of f at $\boldsymbol{x}$:

$$f(\boldsymbol{x} + \lambda \boldsymbol{d}) = f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\lambda \boldsymbol{d}) + o(\|\lambda \boldsymbol{d}\|_2)$$

hence, dividing by λ (and noticing that $\|\lambda \boldsymbol{d}\|_2 = \lambda \|\boldsymbol{d}\|_2$):

$$\frac{f(\boldsymbol{x} + \lambda \boldsymbol{d}) - f(\boldsymbol{x})}{\lambda} = \boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{d} + \frac{o(\lambda \|\boldsymbol{d}\|_2)}{\lambda}$$

letting λ go to 0 concludes the proof. □

2.3.3 Lower Bounding Convex Functions with Affine Functions

In order to prove the characterization of convex functions in the next section we will need the following lemma. This lemma says that any differentiable convex function can be lower bounded by an affine function.

Theorem 2.3.7. *Let S be an open convex subset of $\mathbb{R}^n$, and let $f : S \to \mathbb{R}$ be a differentiable function. Then, f is convex if and only if for any $\boldsymbol{x}, \boldsymbol{y} \in S$ we have that $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$.*

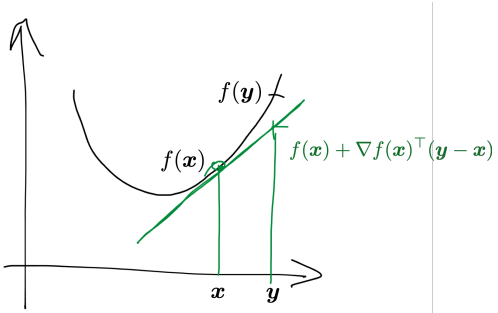

Figure 2.4: The convex function $f(\boldsymbol{y})$ sits above the linear function in $\boldsymbol{y}$ given by $f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top(\boldsymbol{y} - \boldsymbol{x})$.

Proof. [$\implies$] Assume f is convex, then for all $\mathbf{x}, \mathbf{y} \in S$ and $\theta \in [0, 1]$, if we let $\mathbf{z} = \theta\boldsymbol{y} + (1-\theta)\boldsymbol{x}$, we have that

$$f(\mathbf{z}) = f((1-\theta)\boldsymbol{x} + \theta\boldsymbol{y}) \leq (1-\theta)f(\boldsymbol{x}) + \theta f(\boldsymbol{y})$$

and therefore by subtracting $f(\boldsymbol{x})$ from both sides we get:

$$\begin{aligned} f\left(\boldsymbol{x} + \theta(\boldsymbol{y} - \boldsymbol{x})\right) - f(\boldsymbol{x}) &\leq \theta f(\boldsymbol{y}) + (1-\theta)f(\boldsymbol{x}) - f(\boldsymbol{x}) \\ &= \theta f(\boldsymbol{y}) - \theta f(\boldsymbol{x}). \end{aligned}$$

Thus we get that (for $\theta > 0$):

$$\frac{f\left(\boldsymbol{x} + \theta(\boldsymbol{y} - \boldsymbol{x})\right) - f(\boldsymbol{x})}{\theta} \leq f(\boldsymbol{y}) - f(\boldsymbol{x})$$

Applying Proposition 2.3.6 with $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{y}$ we have that:

$$\boldsymbol{\nabla} f(\boldsymbol{x})^\top(\boldsymbol{y} - \boldsymbol{x}) = \lim_{\theta \to 0^+} \frac{f(\boldsymbol{x} + \theta(\boldsymbol{y} - \boldsymbol{x})) - f(\boldsymbol{x})}{\theta} \leq f(\boldsymbol{y}) - f(\boldsymbol{x}).$$

[$\impliedby$] Assume that $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top(\boldsymbol{y} - \boldsymbol{x})$ for all $\mathbf{x}, \mathbf{y} \in S$ and show that f is convex. Let $\mathbf{x}, \mathbf{y} \in S$ and $\mathbf{z} = \theta\boldsymbol{y} + (1-\theta)\boldsymbol{x}$. By our assumption we have that:

$$f(\boldsymbol{y}) \geq f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top(\boldsymbol{y} - \mathbf{z}) \tag{2.1}$$

$$f(\boldsymbol{x}) \geq f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top(\boldsymbol{x} - \mathbf{z}) \tag{2.2}$$

Observe that $\boldsymbol{y} - \boldsymbol{z} = (1-\theta)(\boldsymbol{y} - \boldsymbol{x})$ and $\boldsymbol{x} - \boldsymbol{z} = \theta(\boldsymbol{x} - \boldsymbol{y})$. Thus adding θ times (2.1) to $(1-\theta)$ times (2.2) gives cancellation of the vectors multiplying the gradient, yielding

$$\begin{aligned} \theta f(\boldsymbol{y}) + (1-\theta)f(\boldsymbol{x}) &\geq f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top \mathbf{0} \\ &= f(\theta\boldsymbol{y} + (1-\theta)\boldsymbol{x}) \end{aligned}$$

This is exactly the definition of convexity. □

2.4 Conditions for Optimality

We now want to find necessary and sufficient conditions for local optimality.

Definition 2.4.1. Consider a differentiable function $f : S \to \mathbb{R}$. A point $\boldsymbol{x} \in S$ at which $\boldsymbol{\nabla} f(\boldsymbol{x}) = \mathbf{0}$ is called a **stationary point**.

Proposition 2.4.2. *If $\boldsymbol{x}$ is a local extremum of a differentiable function $f : S \to \mathbb{R}$ then $\boldsymbol{\nabla} f(\boldsymbol{x}) = \mathbf{0}$.*

Proof. Let us assume that $\boldsymbol{x}$ is a local minimum for f. Then for all $\boldsymbol{d} \in \mathbb{R}^n$, $f(\boldsymbol{x}) \leq f(\boldsymbol{x}+\lambda \boldsymbol{d})$ for λ small enough. Hence:

$$0 \leq f(\boldsymbol{x} + \lambda \boldsymbol{d}) - f(\boldsymbol{x}) = \lambda \boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{d} + o(\|\lambda \boldsymbol{d}\|)$$

dividing by $\lambda > 0$ and letting $\lambda \to 0^+$, we obtain $0 \leq \boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{d}$. But, taking $\boldsymbol{d} = -\boldsymbol{\nabla} f(\boldsymbol{x})$, we get $0 \leq -\|\boldsymbol{\nabla} f(\boldsymbol{x})\|_2^2$. This implies that $\boldsymbol{\nabla} f(\boldsymbol{x}) = \mathbf{0}$.

The case where $\boldsymbol{x}$ is a local maximum can be dealt with similarly. □

Remark 2.4.3. For this proposition to hold, it is important that S is open.

For convex functions however it turns out that a stationary point necessarily implies that the function is at its minimum. Together with the proposition above, this says that for a convex function on $\mathbb{R}^n$ a point is optimal if and only if it is stationary.

Proposition 2.4.4. *Let $S \subseteq \mathbb{R}^n$ be an open convex set and let $f : S \to \mathbb{R}$ be a differentiable and convex function. If $\boldsymbol{x}$ is a stationary point then $\boldsymbol{x}$ is a global minimum.*

Proof. From Theorem 3.3.5 we know that for all $\boldsymbol{x}, \boldsymbol{y} \in S$: $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x})$. Since $\boldsymbol{\nabla} f(\boldsymbol{x}) = \mathbf{0}$ this implies that $f(\boldsymbol{y}) \geq f(\boldsymbol{x})$. As this holds for any $\boldsymbol{y} \in S$, $\boldsymbol{x}$ is a global minimum.

□

Chapter 3

Convexity, Second Derivatives and Gradient Descent

Notation for this chapter. In this chapter, we sometimes consider a multivariate function f whose domain is a set $S \subseteq \mathbb{R}^n$, which we will require to be open. When we additionally require that S is convex, we will specify this. Note that $S = \mathbb{R}^n$ is both open and convex and it suffices to keep this case in mind. Things sometimes get more complicated if S is not open, e.g. when the domain of f has a boundary. We will leave those complications for another time.

3.1 A Review of Linear Algebra

Semi-definiteness of a matrix. The following classification of symmetric matrices will be useful.

Definition 3.1.1. Let $\boldsymbol{A}$ by a symmetric matrix in $\mathbb{R}^{n\times n}$. We say that $\boldsymbol{A}$ is:

1. *positive definite* iff $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} > 0$ for all $x \in \mathbb{R}^n \setminus \{0\}$;
2. *positive semidefinite* iff $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} \geq 0$ for all $x \in \mathbb{R}^n$;
3. If neither $\boldsymbol{A}$ nor $-\boldsymbol{A}$ is positive semi-definite, we say that $\boldsymbol{A}$ is *indefinite*.

Example: indefinite matrix. Consider the following matrix $\boldsymbol{A}$:

$$\boldsymbol{A} := \begin{bmatrix} +4 & -1 \\ -1 & -2 \end{bmatrix}$$

For $\boldsymbol{x} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, we have $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} = 4 > 0$. For $\boldsymbol{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ we have $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} = -2 < 0$. $\boldsymbol{A}$ is therefore indefinite.

The following theorem gives a useful characterization of (semi)definite matrices.

Theorem 3.1.2. *Let $\boldsymbol{A}$ be a symmetric matrix in $\mathbb{R}^{n\times n}$.*

1. *$\boldsymbol{A}$ is positive definite iff all its eigenvalues are positive;*
2. *$\boldsymbol{A}$ is positive semidefinite iff all its eigenvalues are non-negative;*

In order to prove this theorem, let us first recall the Spectral Theorem for symmetric matrices.

Theorem 3.1.3 (The Spectral Theorem for Symmetric Matrices)**.** *For all symmetric $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ there exist $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ and a diagonal matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times n}$ s.t.*

1. *$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$.*
2. *$\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}$ (the $n \times n$ identity matrix). I.e. the columns of $\boldsymbol{V}$ form an orthonormal basis. Furthermore, $\boldsymbol{v}_i$ is an eigenvector of $\lambda_i(\boldsymbol{A})$, the ith eigenvalue of $\boldsymbol{A}$.*
3. *$\boldsymbol{\Lambda}_{ii} = \lambda_i(\boldsymbol{A})$.*

Using the Spectral Theorem, we can show the following result:

Theorem 3.1.4 (The Courant-Fischer Theorem)**.** *Let $\boldsymbol{A}$ be a symmetric matrix in $\mathbb{R}^{n\times n}$, with eigenvalues $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$. Then*

1. $$\lambda_i = \min_{\substack{\text{subspace } W\subseteq\mathbb{R}^n \\ \dim(W)=i}} \max_{\boldsymbol{x}\in W,\boldsymbol{x}\neq\boldsymbol{0}} \frac{\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}$$
2. $$\lambda_i = \max_{\substack{\text{subspace } W\subseteq\mathbb{R}^n \\ \dim(W)=n+1-i}} \min_{\boldsymbol{x}\in W,\boldsymbol{x}\neq\boldsymbol{0}} \frac{\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}$$

Theorem 3.1.2 is an immediate corollary of Theorem 4.1.1, since we can see that the minimum value of the quadratic form $\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}$ over $\boldsymbol{x} \in W = \mathbb{R}^n$ is $\lambda_1(\boldsymbol{A})\|\boldsymbol{x}\|_2^2$.

Proof of Theorem 4.1.1. We start by showing Part 1.

Consider letting $W = \operatorname{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_i\}$, and normalize $\boldsymbol{x} \in W$ so that $\|\boldsymbol{x}\|_2 = 1$. Then $\boldsymbol{x} = \sum_{j=1}^{i} \boldsymbol{c}(j)\boldsymbol{v}_j$ for some vector $\boldsymbol{c} \in \mathbb{R}^i$ with $\|\boldsymbol{c}\|_2 = 1$.

Using the decomposition from Theorem 3.1.3 $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$ where $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues of $\boldsymbol{A}$, which we take to be sorted in increasing order. Then $\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}^\top\boldsymbol{V}^\top\boldsymbol{\Lambda}\boldsymbol{V}\boldsymbol{x} = (\boldsymbol{V}\boldsymbol{x})^\top\boldsymbol{\Lambda}(\boldsymbol{V}\boldsymbol{x}) = \sum_{j=1}^{i}\lambda_j\boldsymbol{c}(j)^2 \le \lambda_i\|\boldsymbol{c}\|_2^2 = \lambda_i$. So this choice of W ensures the maximizer cannot achieve a value above λ_i.

But is it possible that the "minimizer" can do better by choosing a different W? Let $T = \text{span}\{\boldsymbol{v}_i, \ldots, \boldsymbol{v}_n\}$. As $\dim(T) = n+1-i$ and $\dim(W) = i$, we must have $\dim(W \cap T) \geq 1$, by a standard property of subspaces. Hence for any W of this dimension,

$$\begin{aligned}\max_{\boldsymbol{x} \in W, \boldsymbol{x} \neq \boldsymbol{0}} \frac{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} &\geq \max_{\boldsymbol{x} \in W \cap T, \boldsymbol{x} \neq \boldsymbol{0}} \frac{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \\ &\geq \min_{\substack{\text{subspace } V \subseteq T \\ \dim(V)=1}} \max_{\boldsymbol{x} \in V, \boldsymbol{x} \neq \boldsymbol{0}} \frac{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \lambda_i,\end{aligned}$$

where the last equality follows from a similar calculation to our first one. Thus, λ_i can always be achieved by the "maximizer" for all W of this dimension.

Part 2 can be dealt with similarly.

□

Example: a positive semidefinite matrix. Consider the following matrix $\boldsymbol{A}$:

$$\boldsymbol{A} := \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

For $\boldsymbol{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, we have $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$, so $\lambda = 0$ is an eigenvalue of $\boldsymbol{A}$. For $\boldsymbol{x} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, we have $\boldsymbol{A}\boldsymbol{x} = \begin{pmatrix} 2 \\ -2 \end{pmatrix} = 2\boldsymbol{x}$, so $\lambda = 2$ is the other eigenvalue of $\boldsymbol{A}$. As both are non-negative, by the theorem above, $\boldsymbol{A}$ is positive semidefinite.

Since we are learning about symmetric matrices, there is one more fact that everyone should know about them. We'll use $\lambda_{\max}(\boldsymbol{A})$ denote maximum eigenvalue of a matrix $\boldsymbol{A}$, and $\lambda_{\min}(\boldsymbol{A})$ the minimum.

Claim 3.1.5. *For a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, $\|\boldsymbol{A}\| = \max(|\lambda_{\max}(\boldsymbol{A})|, |\lambda_{\min}(\boldsymbol{A})|)$.*

3.2 Characterizations of Convexity and Optimality via Second Derivatives

We will now use the second derivatives of a function to obtain characterizations of convexity and optimality. We will begin by introducing the *Hessian*, the matrix of pairwise second derivatives of a function. We will see that it plays a role in approximating a function via a second-order Taylor expansion. We will then use *semi-definiteness* of the Hessian matrix to characterize both conditions of optimality as well as the convexity of a function.

Definition 3.2.1. Given a function $f : S \to \mathbb{R}$ its **Hessian** matrix at point $\boldsymbol{x} \in S$ denoted $\boldsymbol{H}_f(\boldsymbol{x})$ (also sometimes denoted $\boldsymbol{\nabla}^2 f(\boldsymbol{x})$) is:

$$\boldsymbol{H}_f(\boldsymbol{x}) := \begin{bmatrix} \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(1)^2} & \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(1)\partial \boldsymbol{x}(2)} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(1)\partial \boldsymbol{x}(n)} \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(2)\partial \boldsymbol{x}(1)} & \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(2)^2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(2)\partial \boldsymbol{x}(n)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(n)\partial \boldsymbol{x}(1)} & \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(n)\partial \boldsymbol{x}(2)} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(n)^2} \end{bmatrix}$$

Second-order Taylor expansion. When f is twice differentiable it is possible to obtain an approximation of f by quadratic functions. Our definition of $f : S \to \mathbb{R}$ being twice (Fréchet) differentiable at $\boldsymbol{x} \in S$ is that there exists $\boldsymbol{\nabla} f(\boldsymbol{x}) \in \mathbb{R}^n$ and $\boldsymbol{H}_f(\boldsymbol{x}) \in R^{n \times n}$ s.t.

$$\lim_{\boldsymbol{\delta} \to \boldsymbol{0}} \frac{\left\| f(\boldsymbol{x} + \boldsymbol{\delta}) - f(\boldsymbol{x}) - \left(\boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^\top \boldsymbol{H}_f(\boldsymbol{x}) \boldsymbol{\delta} \right) \right\|_2}{\|\boldsymbol{\delta}\|_2^2} = 0.$$

This is equivalent to saying that for all $\boldsymbol{\delta}$

$$f(\boldsymbol{x} + \boldsymbol{\delta}) = f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^\top \boldsymbol{H}_f(\boldsymbol{x}) \boldsymbol{\delta} + o(\|\boldsymbol{\delta}\|_2^2).$$

where by definition:

$$\lim_{\boldsymbol{\delta} \to \boldsymbol{0}} \frac{o(\|\boldsymbol{\delta}\|^2)}{\|\boldsymbol{\delta}\|_2^2} = 0$$

We say that f is *continuously differentiable* on a set $S \subseteq \mathbb{R}^n$ if it is twice differentiable and in addition the gradient and Hessian are continuous on S.

As for first order expansions, we have a Taylor's Theorem, which we state in the so-called remainder form.

Theorem 3.2.2 (Taylor's Theorem, multivariate second-order remainder form)**.** *If $f : S \to \mathbb{R}$ is twice continuously differentiable over $[\boldsymbol{x}, \boldsymbol{y}]$, then for some $\boldsymbol{z} \in [\boldsymbol{x}, \boldsymbol{y}]$,*

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{1}{2} (\boldsymbol{y} - \boldsymbol{x})^\top \boldsymbol{H}_f(\boldsymbol{z}) (\boldsymbol{y} - \boldsymbol{x})$$

3.2.1 A Necessary Condition for Local Extrema

Recall that in the previous chapter, we show the following proposition.

Proposition 3.2.3. *If $\boldsymbol{x}$ is a local extremum of a differentiable function $f : S \to \mathbb{R}$ then $\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{0}$.*

We can now give the second-order necessary conditions for local extrema via the Hessian.

Theorem 3.2.4. *Let $f : S \to \mathbb{R}$ be a function twice differentiable at $\boldsymbol{x} \in S$. If $\boldsymbol{x}$ is a local minimum, then $\boldsymbol{H}_f(\boldsymbol{x})$ is positive semidefinite.*

Proof. Let us assume that $\boldsymbol{x}$ is a local minimum. We know from Proposition 3.2.3 that $\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{0}$, hence the second-order expansion at $\boldsymbol{x}$ takes the form:

$$f(\boldsymbol{x} + \lambda \boldsymbol{d}) = f(\boldsymbol{x}) + \lambda^2 \frac{1}{2} \boldsymbol{d}^\top \boldsymbol{H}_f(\boldsymbol{x}) \boldsymbol{d} + o(\lambda^2 \left\| \boldsymbol{d} \right\|_2^2)$$

Because $\boldsymbol{x}$ is a local minimum, we can then derive

$$0 \leq \lim_{\lambda \to 0^+} \frac{f(\boldsymbol{x} + \lambda \boldsymbol{d}) - f(\boldsymbol{x})}{\lambda^2} = \frac{1}{2} \boldsymbol{d}^\top \boldsymbol{H}_f(\boldsymbol{x}) \boldsymbol{d}$$

This is true for any $\boldsymbol{d}$, hence $\boldsymbol{H}_f(\boldsymbol{x})$ is positive semidefinite. □

Remark 3.2.5. Again, for this proposition to hold, it is important that S is open.

3.2.2 A sufficient condition for local extrema

A local minimum thus is a stationary point and has a positive semi-definite Hessian. The converse is almost true, but we need to strengthen the Hessian condition slightly.

Theorem 3.2.6. *Let $f : S \to \mathbb{R}$ be a function twice differentiable at a stationary point $\boldsymbol{x} \in S$. If $\boldsymbol{H}_f(\boldsymbol{x})$ is positive definite then $\boldsymbol{x}$ is a local minimum.*

Proof. Let us assume that $\boldsymbol{H}_f(\boldsymbol{x})$ is positive definite. We know that $\boldsymbol{x}$ is a stationary point. We can write the second-order expansion at $\boldsymbol{x}$:

$$f(\boldsymbol{x} + \boldsymbol{\delta}) = f(\boldsymbol{x}) + \frac{1}{2} \boldsymbol{\delta}^\top \boldsymbol{H}_f(\boldsymbol{x}) \boldsymbol{\delta} + o(\left\| \boldsymbol{\delta} \right\|_2^2)$$

Because the Hessian is positive definite, it has a strictly positive minimum eigenvalue $\lambda_{\min}$, we can conclude that $\boldsymbol{\delta}^\top \boldsymbol{H}_f(\boldsymbol{x}) \boldsymbol{\delta} \geq \lambda_{\min} \left\| \boldsymbol{\delta} \right\|_2^2$. From this, we conclude that when $\left\| \boldsymbol{\delta} \right\|_2^2$ is small enough, $f(\boldsymbol{x} + \boldsymbol{\delta}) - f(\boldsymbol{x}) \geq \frac{1}{4} \lambda_{\min} \left\| \boldsymbol{\delta} \right\|_2^2 > 0$. This proves that $\boldsymbol{x}$ is a local minimum. □

Remark 3.2.7. When $\boldsymbol{H}_f(\boldsymbol{x})$ is indefinite at a stationary point $\boldsymbol{x}$, we have what is known as a *saddle point*: $\boldsymbol{x}$ will be a minimum along the eigenvectors of $\boldsymbol{H}_f(\boldsymbol{x})$ for which the eigenvalues are positive and a maximum along the eigenvectors of $\boldsymbol{H}_f(\boldsymbol{x})$ for which the eigenvalues are negative.

3.2.3 Characterization of convexity

Definition 3.2.8. For a convex set $S \subseteq R^n$, we say that a function $f : S \to \mathbb{R}$ is **strictly convex on** S if for any two points $\boldsymbol{x}_1, \boldsymbol{x}_2 \in S$ and any $\theta \in (0, 1)$ we have that:

$$f\left(\theta \boldsymbol{x}_1 + (1 - \theta) \boldsymbol{x}_2\right) < \theta f(\boldsymbol{x}_1) + \Big(1 - \theta\Big) f(\boldsymbol{x}_2).$$

Theorem 3.2.9. *Let $S \subseteq \mathbb{R}^n$ be open and convex, and let $f : S \to \mathbb{R}$ be twice continuously differentiable.*

1. *If $H_f(\boldsymbol{x})$ is positive semi-definite for any $\boldsymbol{x} \in S$ then f is convex on S.*
2. *If $H_f(\boldsymbol{x})$ is positive definite for any $\boldsymbol{x} \in S$ then f is **strictly** convex on S.*
3. *If f is convex, then $H_f(\boldsymbol{x})$ is positive semi-definite $\forall \boldsymbol{x} \in S$.*

Proof.

1. By applying Theorem 3.2.2, we find that for some $\boldsymbol{z} \in [\boldsymbol{x}, \boldsymbol{y}]$:

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{1}{2}\Big((\boldsymbol{y} - \boldsymbol{x})^\top H_f(\boldsymbol{z})(\boldsymbol{y} - \boldsymbol{x})\Big)$$

   If $H_f(\boldsymbol{z})$ is positive semi-definite, this necessarily implies that:

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$$

   and from Theorem 3.3.5 we get that f is convex.

2. if $H_f(\boldsymbol{x})$ is positive definite, we have that:

$$f(\boldsymbol{y}) > f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}).$$

   Applying the same idea as in Theorem 3.3.5 we can show that in this case f is **strictly** convex.

3. Let f be a convex function. For $\boldsymbol{x} \in S$, and some small $\lambda > 0$, for any $\mathbf{d} \in \mathbb{R}^n$ we have that $\boldsymbol{x} + \lambda \mathbf{d} \in S$. From the Taylor expansion of f we get:

$$f(\boldsymbol{x} + \lambda \mathbf{d}) = f(\boldsymbol{x}) + \lambda \boldsymbol{\nabla} f(\boldsymbol{x})^\top \mathbf{d} + \frac{\lambda^2}{2} \mathbf{d}^\top H_f(\boldsymbol{x}) \mathbf{d} + o(\lambda^2 \left\| \boldsymbol{d} \right\|_2^2).$$

   From Lemma 3.3.5 we get that if f is convex then:

$$f(\boldsymbol{x} + \lambda \mathbf{d}) \geq f(\boldsymbol{x}) + \lambda \boldsymbol{\nabla} f(\boldsymbol{x})^\top \mathbf{d}.$$

   Therefore, we have that for any $\mathbf{d} \in \mathbb{R}^n$:

$$\frac{\lambda^2}{2} \mathbf{d}^\top H_f(\boldsymbol{x}) \mathbf{d} + o(||\lambda \mathbf{d}||^2) \geq 0$$

   Dividing by λ^2 and taking $\lambda \to 0^+$ gives us that for any $\mathbf{d} \in \mathbb{R}^n$: $\mathbf{d}^\top H_f(\boldsymbol{x}) \mathbf{d} \geq 0$. □

Remark 3.2.10. It is important to note that if S is open and f is strictly convex, then $\boldsymbol{H}_f(\boldsymbol{x})$ may still (only) be positive semi-definite $\forall \boldsymbol{x} \in S$. Consider $f(x) = x^4$ which is strictly convex, then the Hessian is $\boldsymbol{H}_f(x) = 12x^2$ which equals 0 at $x = 0$.

3.3 Gradient Descent - An Approach to Optimization?

We have begun to develop an understanding of convex functions, and what we have learned already suggests a way for us to try to find an approximate minimizer of a given convex function.

Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable, and we want to solve

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x})$$

We would like to find $\boldsymbol{x}^*$, a global minimizer of f. Suppose we start with some initial guess $\boldsymbol{x}_0$, and we want to update it to $\boldsymbol{x}_1$ with $f(\boldsymbol{x}_1) < f(\boldsymbol{x}_0)$. If we can repeatedly make updates like this, maybe we eventually find a point with nearly minimum function value, i.e. some $\tilde{\boldsymbol{x}}$ with $f(\tilde{\boldsymbol{x}}) \approx f(\boldsymbol{x}^*)$?

Recall that $f(\boldsymbol{x}_0 + \boldsymbol{\delta}) = f(\boldsymbol{x}_0) + \boldsymbol{\nabla} f(\boldsymbol{x}_0)^\top \boldsymbol{\delta} + o(\|\boldsymbol{\delta}\|_2)$. This means that if we choose $\boldsymbol{x}_1 = \boldsymbol{x}_0 - \lambda \boldsymbol{\nabla} f(\boldsymbol{x}_0)$, we get

$$f(\boldsymbol{x}_0 - \lambda \boldsymbol{\nabla} f(\boldsymbol{x}_0)) = f(\boldsymbol{x}_0) - \lambda \|\boldsymbol{\nabla} f(\boldsymbol{x}_0)\|_2^2 + o(\lambda \|\boldsymbol{\nabla} f(\boldsymbol{x}_0)\|_2)$$

And because f is convex, we know that $\boldsymbol{\nabla} f(\boldsymbol{x}_0) \neq \mathbf{0}$ unless we are already at a global minimum. So, for some small enough $\lambda > 0$, we should get $f(\boldsymbol{x}_1) < f(\boldsymbol{x}_0)$ unless we're already at a global minimizer. This idea of taking a step in the direction of $-\boldsymbol{\nabla} f(\boldsymbol{x}_0)$ is what is called *Gradient Descent*. But how do we choose λ each time? And does this lead to an algorithm that quickly reaches a point with close to minimal function value? To get good answers to these questions, we need to assume more about the function f that we are trying to minimize.

In the following subsection, we will see some conditions that suffice. But there are also many other settings where one can show that some form of gradient descent converges.

3.3.1 A Quantitative Bound on Changes in the Gradient

Definition 3.3.1. Let $f : S \to \mathbb{R}$ be a differentiable function, where $S \subseteq \mathbb{R}^n$ is convex and open. We say that f is *β-gradient Lipschitz* iff for all $\boldsymbol{x}, \boldsymbol{y} \in S$

$$\|\boldsymbol{\nabla} f(\boldsymbol{x}) - \boldsymbol{\nabla} f(\boldsymbol{y})\|_2 \leq \beta \|\boldsymbol{x} - \boldsymbol{y}\|_2 .$$

We also refer to this as f being *β-smooth*.

Proposition 3.3.2. *Consider a twice continuously differentiable $f : S \to \mathbb{R}$. Then f is β-gradient Lipschitz if and only if for all $\boldsymbol{x} \in S$, $\|\boldsymbol{H}_f(\boldsymbol{x})\| \leq \beta$.*

You will prove this in Exercise 13 (Week 2) of the first exercise set.

Proposition 3.3.3. *Let $f : S \to \mathbb{R}$ be a β-gradient Lipschitz function. Then for all $\boldsymbol{x}, \boldsymbol{y} \in S$,*

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{\beta}{2} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

To prove this proposition, we need the following result from multi-variate calculus. This is a restricted form of the fundamental theorem of calculus for line integrals.

Proposition 3.3.4. *Let $f : S \to \mathbb{R}$ be a differentiable function, and consider $\boldsymbol{x}, \boldsymbol{y}$ such that $[\boldsymbol{x}, \boldsymbol{y}] \in S$. Let $\boldsymbol{x}_\theta = \boldsymbol{x} + \theta(\boldsymbol{y} - \boldsymbol{x})$. Then*

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \int_{\theta=0}^{1} \boldsymbol{\nabla} f(\boldsymbol{x}_\theta)^\top (\boldsymbol{y} - \boldsymbol{x}) d\theta$$

Now, we're in a position to show Proposition 3.3.3

Proof of Proposition 3.3.3. Let $f : S \to \mathbb{R}$ be a β-gradient Lipschitz function. Consider arbitrary $\boldsymbol{x}, \boldsymbol{y} \in S$ such that $[\boldsymbol{x}, \boldsymbol{y}] \in S$

$$\begin{aligned}
f(\boldsymbol{y}) &= f(\boldsymbol{x}) + \int_{\theta=0}^{1} \boldsymbol{\nabla} f(\boldsymbol{x}_\theta)^\top (\boldsymbol{y} - \boldsymbol{x}) d\theta \\
&= f(\boldsymbol{x}) + \int_{\theta=0}^{1} \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) d\theta + \int_{\theta=0}^{1} (\boldsymbol{\nabla} f(\boldsymbol{x}_\theta) - \boldsymbol{\nabla} f(\boldsymbol{x}))^\top (\boldsymbol{y} - \boldsymbol{x}) d\theta \\
&\qquad \text{Next we use Cauchy-Schwarz pointwise.} \\
&\qquad \text{We also evaluate the first integral.} \\
&\leq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \int_{\theta=0}^{1} \|\boldsymbol{\nabla} f(\boldsymbol{x}_\theta) - \boldsymbol{\nabla} f(\boldsymbol{x})\| \, \|\boldsymbol{y} - \boldsymbol{x}\| \, d\theta \\
&\qquad \text{Then we apply } \beta\text{-gradient Lipschitz and note } \boldsymbol{x}_\theta - \boldsymbol{x} = \theta(\boldsymbol{y} - \boldsymbol{x}). \\
&\leq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \int_{\theta=0}^{1} \beta\theta \, \|\boldsymbol{y} - \boldsymbol{x}\|^2 \, d\theta. \\
&= f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{\beta}{2} \|\boldsymbol{y} - \boldsymbol{x}\|^2 .
\end{aligned}$$

□

3.3.2 Analyzing Gradient Descent

It turns out that just knowing a function $f : \mathbb{R}^n \to \mathbb{R}$ is convex and β-gradient Lipschitz is enough to let us figure out a reasonable step size for Gradient Descent and let us analyze its convergence.

We start at a point $\boldsymbol{x}_0 \in \mathbb{R}^n$, and we try to find a point $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{\delta}$ with lower function value. We will let our upper bound from Proposition 3.3.3 guide us, in fact, we could ask, what is the *best* update for minimizing this upper bound, i.e. a $\boldsymbol{\delta}$ solving

$$\min_{\boldsymbol{\delta} \in \mathbb{R}^n} f(\boldsymbol{x}_0) + \boldsymbol{\nabla} f(\boldsymbol{x}_0)^\top \boldsymbol{\delta} + \frac{\beta}{2} \|\boldsymbol{\delta}\|^2$$

We can compute the best according to this upper bound by noting first that function is convex and continuously differentiable, and hence will be minimized at any point where the gradient is zero. Thus we want $\mathbf{0} = \boldsymbol{\nabla}_{\boldsymbol{\delta}} \left(f(\boldsymbol{x}_0) + \boldsymbol{\nabla} f(\boldsymbol{x}_0)^\top \boldsymbol{\delta} + \frac{\beta}{2} \|\boldsymbol{\delta}\|^2 \right) = \boldsymbol{\nabla} f(\boldsymbol{x}_0) + \beta \boldsymbol{\delta}$, which occurs at $\boldsymbol{\delta} = -\frac{1}{\beta} \boldsymbol{\nabla} f(\boldsymbol{x}_0)$.

Plugging in this value into the upper bound, we get that $f(\boldsymbol{x}_1) \leq f(\boldsymbol{x}_0) - \|\boldsymbol{\nabla} f(\boldsymbol{x}_0)\|_2^2 / 2\beta$.

Now, as our algorithm, we will start with some guess $\boldsymbol{x}_0$, and then at every step we will update our guess using the best step based on our Proposition 3.3.3 upper bound on f at $\boldsymbol{x}_i$, and so we get

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \frac{1}{\beta} \boldsymbol{\nabla} f(\boldsymbol{x}_i) \text{ and } f(\boldsymbol{x}_{i+1}) \leq f(\boldsymbol{x}_i) - \frac{\|\boldsymbol{\nabla} f(\boldsymbol{x}_i)\|_2^2}{2\beta}. \tag{3.1}$$

Let us try to prove that our algorithm converges toward an $\boldsymbol{x}$ with low function value.

We will measure this by looking at

$$\text{gap}_i = f(\boldsymbol{x}_i) - f(\boldsymbol{x}^*)$$

where $\boldsymbol{x}^*$ is a global minimizer of f (note that there may not be a unique minimizer of f). We will try to show that this function value gap grows small. Using $f(\boldsymbol{x}_{i+1}) - f(\boldsymbol{x}_i) = \text{gap}_{i+1} - \text{gap}_i$, we get

$$\text{gap}_{i+1} - \text{gap}_i \leq -\frac{\|\boldsymbol{\nabla} f(\boldsymbol{x}_i)\|_2^2}{2\beta} \tag{3.2}$$

If the gap_i value is never too much bigger than $\frac{\|\boldsymbol{\nabla} f(\boldsymbol{x}_i)\|_2^2}{2\beta}$, then this should help us show we are making progress. But how much can they differ? We will now try to show a limit on this.

Recall that in the previous chapter we showed the following theorem.

Theorem 3.3.5. *Let S be an open convex subset of $\mathbb{R}^n$, and let $f : S \to \mathbb{R}$ be a differentiable function. Then, f is convex if and only if for any $\boldsymbol{x}, \boldsymbol{y} \in S$ we have that $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$.*

Using the convexity of f and the lower bound on convex functions given by Theorem 3.3.5, we have that

$$f(\boldsymbol{x}^*) \geq f(\boldsymbol{x}_i) + \boldsymbol{\nabla} f(\boldsymbol{x}_i)^\top (\boldsymbol{x}^* - \boldsymbol{x}_i) \tag{3.3}$$

Rearranging gets us

$$\begin{aligned} \text{gap}_i &\leq \nabla f(\boldsymbol{x}_i)^\top (\boldsymbol{x}_i - \boldsymbol{x}^*) && (3.4) \\ &\leq \|\nabla f(\boldsymbol{x}_i)\|_2 \|\boldsymbol{x}_i - \boldsymbol{x}^*\|_2 && \text{by Cauchy-Schwarz.} \end{aligned}$$

At this point, we are essentially ready to connect Equation (3.2) with Equation (3.4) and analyze the convergence rate of our algorithm.

However, at the moment, we see that the change $\text{gap}_{i+1} - \text{gap}_i$ in how close we are to the optimum function value is governed by the norm of the gradient $\|\nabla f(\boldsymbol{x}_i)\|_2$, while the size of the gap is related to *both* this quantity and the distance $\|\boldsymbol{x}_i - \boldsymbol{x}^*\|_2$ between the current solution $\boldsymbol{x}_i$ and an optimum $\boldsymbol{x}^*$. Do we need both or can we get rid of, say, the distance? Unfortunately, with this algorithm and for this class of functions, a dependence on the distance is necessary. However, we can simplify things considerably using the following observation, which you will prove in the exercises (Exercise 2):

Claim 3.3.6. *When running Gradient Descent as given by the step in Equation* (3.1)*, for all* i $\|\boldsymbol{x}_i - \boldsymbol{x}^*\|_2 \leq \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2$.

Combining this Claim with Equation (3.2) and Equation (3.4),

$$\text{gap}_{i+1} - \text{gap}_i \leq -\frac{1}{2\beta} \cdot \left(\frac{\text{gap}_i}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2} \right)^2 \tag{3.5}$$

At this point, a simple induction will complete the proof of following result.

Theorem 3.3.7. *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be a* β*-gradient Lipschitz, convex function. Let* $\boldsymbol{x}_0$ *be a given starting point, and let* $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x})$ *be a minimizer of* f*. The Gradient Descent algorithm given by*

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \frac{1}{\beta} \nabla f(\boldsymbol{x}_i)$$

ensures that the k*th iterate satisfies*

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) \leq \frac{2\beta \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2^2}{k+1}.$$

Carrying out this induction is one of the exercises in problem set 2.

Part II

Spectral Graph Theory

Chapter 4

Introduction to Spectral Graph Theory

In this chapter, we will study graphs through linear algebra. This approach is known as Spectral Graph Theory and turns out to be surprisingly powerful. An in-depth treatment of many topics in this area can be found in [**?**].

4.1 Recap: Incidence and Adjacency Matrices, the Laplacian Matrix and Electrical Energy

In Chapter 1, we looked at undirected graphs and we introduce the incidence matrix and the Laplacian of the graph. Let us recall these.

We consider an undirected weighted graph $G = (V, E, \boldsymbol{w})$, with $n = |V|$ vertices and $m = |E|$ edges, where $\boldsymbol{w} \in \mathbb{R}_+^E$ assigns positive weight for every edge. Let's assume G is connected.

To introduce the *edge-vertex incidence matrix* of the graph, we first have to associate an arbitrary direction to every edge. We then let $\boldsymbol{B} \in \mathbb{R}^{V \times E}$.

$$\boldsymbol{B}(v, e) = \begin{cases} 1 & \text{if } e = (u, v) \\ -1 & \text{if } e = (v, u) \\ 0 & \text{o.w.} \end{cases}$$

The edge directions are only there to help us track the meaning of signs of quantities defined on edges: The math we do should not depend on the choice of sign.

Let $\boldsymbol{W} \in \mathbb{R}^{E \times E}$ be the diagonal matrix given by $\boldsymbol{W} = \text{diag}(\boldsymbol{w})$, i.e $\boldsymbol{W}(e, e) = \boldsymbol{w}(e)$. We define the Laplacian of the graph as $\boldsymbol{L} = \boldsymbol{B}\,\boldsymbol{W}\,\boldsymbol{B}^\top$. Note that in the first chapter, we defined the Laplacian as $\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top$, where $\boldsymbol{R}$ is the diagonal matrix with edge resistances on the diagonal. We want to think of high *weight* on an edge as expressing that two vertices are

highly connected, whereas we think of high resistance on an edge as expressing that the two vertices are poorly connected, so we let $\boldsymbol{w}(e) = 1/\boldsymbol{R}(e,e)$.

The weighted adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{V \times V}$ of a graph is given by

$$\boldsymbol{A}(u,v) = \begin{cases} \boldsymbol{w}(u,v) & \text{if } \{u,v\} \in E \\ 0 & \text{otherwise.} \end{cases}$$

Note that we treat the edges as undirected here, so $\boldsymbol{A}^\top = \boldsymbol{A}$. The weighted degree of a vertex is defined as $\boldsymbol{d}(v) = \sum_{\{u,v\} \in E} w(u,v)$. Again we treat the edges as undirected. Let $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{d})$ be the diagonal matrix in $\mathbb{R}^{V \times V}$ with weighted degrees on the diagonal.

In Problem Set 1, you showed that $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, and that for $\boldsymbol{x} \in \mathbb{R}^V$,

$$\boldsymbol{x}^\top \boldsymbol{L} \boldsymbol{x} = \sum_{\{a,b\} \in E} \boldsymbol{w}(a,b)(\boldsymbol{x}(a) - \boldsymbol{x}(b))^2.$$

Now we can express the net flow constraint that $\boldsymbol{f}$ routes $\boldsymbol{d}$ by

$$\boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}.$$

This is also called a conservation constraint. In our examples so far, we have $\boldsymbol{d}(s) = -1$, $\boldsymbol{d}(t) = 1$ and $\boldsymbol{d}(u) = 0$ for all $u \in V \setminus \{s,t\}$.

If we let $\boldsymbol{R} = \mathrm{diag}_{e \in E}\, \boldsymbol{r}(e)$ then Ohm's law tells us that electrical voltages $\boldsymbol{x}$ will induce an electrical flow $\boldsymbol{f} = \boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{x}$. We defined the electrical energy of a flow $\boldsymbol{f} \in \mathbb{R}^E$ to be

$$\mathcal{E}(\boldsymbol{f}) = \sum_e \boldsymbol{r}(e)\boldsymbol{f}(e)^2 = \boldsymbol{f}^\top \boldsymbol{R} \boldsymbol{f}.$$

And, from Ohm's Law, we can then see that

$$\mathcal{E}(\boldsymbol{f}) = \boldsymbol{f}^\top \boldsymbol{R}\boldsymbol{f} = \boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x}.$$

Hence, define the electrical energy associated with a set of voltages to be

$$\mathcal{E}(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x}.$$

The Courant-Fisher Theorem. Let us also recall the Courant-Fischer theorem, which we proved in Chapter 3 (Theorem 4.1.1).

Theorem 4.1.1 (The Courant-Fischer Theorem)**.** *Let $\boldsymbol{A}$ be a symmetric matrix in $\mathbb{R}^{n \times n}$, with eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. Then*

1.

$$\lambda_i = \min_{\substack{\text{subspace } W \subseteq \mathbb{R}^n \\ \dim(W) = i}} \max_{\boldsymbol{x} \in W, \boldsymbol{x} \neq \boldsymbol{0}} \frac{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$$

2.

$$\lambda_i = \max_{\substack{\text{subspace } W \subseteq \mathbb{R}^n \\ \dim(W) = n+1-i}} \min_{\boldsymbol{x} \in W, \boldsymbol{x} \neq \boldsymbol{0}} \frac{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$$

In fact, from our proof of the Courant-Fischer theorem in Chapter 3, we can also extract a slightly different statement:

Theorem 4.1.2 (The Courant-Fischer Theorem, eigenbasis version)**.** *Let $\boldsymbol{A}$ be a symmetric matrix in $\mathbb{R}^{n \times n}$, with eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$, and corresponding eigenvectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ which form an othernormal basis. Then*

1.

$$\lambda_i = \min_{\substack{\boldsymbol{x} \perp \boldsymbol{x}_1, \ldots \boldsymbol{x}_{i-1} \\ \boldsymbol{x} \neq \boldsymbol{0}}} \frac{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$$

2.

$$\lambda_i = \max_{\substack{\boldsymbol{x} \perp \boldsymbol{x}_{i+1}, \ldots \boldsymbol{x}_n \\ \boldsymbol{x} \neq \boldsymbol{0}}} \frac{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$$

Of course, we also have $\lambda_i(\boldsymbol{A}) = \frac{\boldsymbol{x}_i^\top \boldsymbol{A} \boldsymbol{x}_i}{\boldsymbol{x}_i^\top \boldsymbol{x}_i}$.

4.2 Understanding Eigenvalues of the Laplacian

We would like to understand the eigenvalues of the Laplacian matrix of a graph.

But first, why should we care? It turns out that Laplacian eigenvalues can help us understand many properties of a graph. But we are going to start off with simple motivating observation: Electrical voltages $\boldsymbol{x} \in \mathbb{R}^V$ consume electrical energy $\mathcal{E}(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{L} \boldsymbol{x}$. This means that by the Courant-Fischer Theorem

$$\mathcal{E}(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{L} \boldsymbol{x} \leq \lambda_n(L) \boldsymbol{x}^\top \boldsymbol{x}$$

And, for any voltages $\boldsymbol{x} \perp \boldsymbol{1}$,

$$\mathcal{E}(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{L} \boldsymbol{x} \geq \lambda_2(L) \boldsymbol{x}^\top \boldsymbol{x}.$$

Thus, we can use the eigenvalues to give upper and lower bounds on how much electrical energy will be consumed by the flow induced by $\boldsymbol{x}$, in terms compared to $\boldsymbol{x}^\top \boldsymbol{x} = \|\boldsymbol{x}\|_2^2$.

In a couple of chapters, we will also prove the following claim, which shows that the Laplacian eigenvalues can directly tell us about the electrical energy that is required to route a given demand.

Claim 4.2.1. *Given a demand vector $\boldsymbol{d} \in \mathbb{R}^V$ such that $\boldsymbol{d} \perp \boldsymbol{1}$, the electrical voltages $\boldsymbol{x}$ that route $\boldsymbol{d}$ satisfy $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{d}$ and the electrical energy of these voltages satifies*

$$\frac{\|\boldsymbol{d}\|_2^2}{\lambda_n} \leq \mathcal{E}(\boldsymbol{x}) \leq \frac{\|\boldsymbol{d}\|_2^2}{\lambda_2}$$

Eigenvalues of the Laplacian of a Complete Graph. To get a sense of how Laplacian eigenvalues behave, let us start by considering the n vertex complete graph with unit weights, which we denote by K_n. The adjacency matrix of K_n is $\boldsymbol{A} = \mathbf{1}\mathbf{1}^\top - \boldsymbol{I}$, since it has ones everywhere, except for the diagonal, where entries are zero. The degree matrix $\boldsymbol{D} = (n-1)\boldsymbol{I}$. Thus the Laplacian is $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} = n\boldsymbol{I} - \mathbf{1}\mathbf{1}^\top$.

Thus for any $\boldsymbol{y} \perp \mathbf{1}$, we have $\boldsymbol{y}^\top \boldsymbol{L}\boldsymbol{y} = n\boldsymbol{y}^\top\boldsymbol{y} - (\mathbf{1}^\top\boldsymbol{y})^2 = n\boldsymbol{y}^\top\boldsymbol{y}$.

From this, we can conclude that any $\boldsymbol{y} \perp \mathbf{1}$ is an eigenvector of eigenvalue n, and that all $\lambda_2 = \lambda_3 = \ldots = \lambda_n = n$.

Next, let us try to understand λ_2 and λ_n for P_n, the n vertex path graph with unit weight edges. I.e. the graph has edges $E = \{\{i, i+1\} \text{ for } i = 1 \text{ to } (n-1)\}$.

This is in a sense the least well-connected unit weight graph on n vertices, whereas K_n is the most well-connected.

4.2.1 Test Vector Bounds on λ_2 and λ_n

We can use the eigenbasis version of the Courant-Fisher theorem to observe that the second-smallest eigenvalue of the Laplacian is given by

$$\lambda_2(\boldsymbol{L}) = \min_{\substack{\boldsymbol{x}\neq\mathbf{0} \\ \boldsymbol{x}^\top\mathbf{1}=0}} \frac{\boldsymbol{x}^\top\boldsymbol{L}\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}. \tag{4.1}$$

We can get a better understanding of this particular case through a couple of simple observations. Suppose $\boldsymbol{x} = \boldsymbol{y} + \alpha\mathbf{1}$, where $\boldsymbol{y} \perp \mathbf{1}$. Then $\boldsymbol{x}^\top\boldsymbol{L}\boldsymbol{x} = \boldsymbol{y}^\top\boldsymbol{L}\boldsymbol{y}$, and $\|\boldsymbol{x}\|_2^2 = \|\boldsymbol{y}\|^2 + \alpha^2\|\mathbf{1}\|^2$. So for any given vector, you can increase the value of $\frac{\boldsymbol{x}^\top\boldsymbol{L}\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}$, by instead replacing $\boldsymbol{x}$ with the component orthogonal to $\mathbf{1}$, which we denoted by $\boldsymbol{y}$.

We can conclude from Equation (4.1) that for *any* vector $\boldsymbol{y} \perp \mathbf{1}$,

$$\lambda_2 \leq \frac{\boldsymbol{y}^\top\boldsymbol{L}\boldsymbol{y}}{\boldsymbol{y}^\top\boldsymbol{y}}$$

When we use a vector $\boldsymbol{y}$ in this way to prove a bound on an eigenvalue, we call it a *test vector*.

Now, we'll use a test vector to give an upper bound on $\lambda_2(\boldsymbol{L}_{P_n})$. Let $\boldsymbol{x} \in \mathbb{R}^V$ be given by $\boldsymbol{x}(i) = (n+1) - 2i$, for $i \in [n]$. This vector satisfies $\boldsymbol{x} \perp \mathbf{1}$. We picked this because we wanted a sequence of values growing linearly along the path, while also making sure that the vector

is orthogonal to $\mathbf{1}$. Now

$$\begin{aligned}\lambda_2(\boldsymbol{L}_{P_n}) &\leq \frac{\sum_{i\in[n-1]}(\boldsymbol{x}(i)-\boldsymbol{x}(i+1))^2}{\sum_{i=1}^n \boldsymbol{x}(i)^2} \\ &= \frac{\sum_{i=1}^{n-1} 2^2}{\sum_{i=1}^n (n+1-2i)^2} \\ &= \frac{4(n-1)}{(n+1)n(n-1)/3} \\ &= \frac{12}{n(n+1)} \leq \frac{12}{n^2}.\end{aligned}$$

Later, we will prove a lower bound that shows this value is right up to a constant factor. But the test vector approach based on the Courant-Fischer theorem doesn't immediately work when we want to prove lower bounds on $\lambda_2(\boldsymbol{L})$.

We can see from either version of the Courant-Fischer theorem that

$$\lambda_n(\boldsymbol{L}) = \max_{\boldsymbol{v}\neq\mathbf{0}} \frac{\boldsymbol{v}^\top \boldsymbol{L}\boldsymbol{v}}{\boldsymbol{v}^\top \boldsymbol{v}}. \tag{4.2}$$

Thus for *any* vector $\boldsymbol{y} \neq 0$,

$$\lambda_n \geq \frac{\boldsymbol{y}^\top \boldsymbol{L}\boldsymbol{y}}{\boldsymbol{y}^\top \boldsymbol{y}}.$$

This means we can get a test vector-based lower bound on λ_n. Let us apply this to the Laplacian of P_n. We'll try the vector $\boldsymbol{x} \in \mathbb{R}^V$ given by $\boldsymbol{x}(1) = -1$, and $\boldsymbol{x}(n) = 1$ and $\boldsymbol{x}(i) = 0$ for $i \neq 1, n$.

Here we get

$$\lambda_n(\boldsymbol{L}_{P_n}) \geq \frac{\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \frac{2}{2} = 1.$$

Again, it's not clear how to use the Courant-Fischer theorem to prove an upper bound on $\lambda_n(\boldsymbol{L})$. But, later we'll see how to prove an upper that shows that for P_n, the lower bound we obtained is right up to constant factors.

4.2.2 Eigenvalue Bounds Beyond Test Vectors

In the previous sections, we first saw a complete characterization of the eigenvalues and eigenvectors of the unit weight complete graph on n vertices, K_n. Namely, $\boldsymbol{L}_{K_n} = n\boldsymbol{I} - \mathbf{1}\mathbf{1}^\top$, and this means that *every* vector $\boldsymbol{y} \perp \mathbf{1}$ is an eigenvector of eigenvalue n.

We then looked at eigenvalues of P_n, the unit weight path on n vertices, and we showed using *test vector* bounds that

$$\lambda_2(\boldsymbol{L}_{P_n}) \leq \frac{12}{n^2} \text{ and } 1 \leq \lambda_n(\boldsymbol{L}_{P_n}). \tag{4.3}$$

Ideally we would like to prove an almost matching lower bound on λ_2 and an almost matching upper bound on λ_n, but it is not clear how to get that from the Courant-Fischer theorem.

To get there, we start we need to introduce some more tools.

4.2.3 The Loewner Order, aka. the Positive Semi-Definite Order

We'll now introduce an ordering on symmetric matrices called the *Loewner order*, which I also like to just call the positive semi-definite order. As we will see in a moment, it is a partial order on symmetric matrices, we denote it by "$\preceq$". For conveniece, we allow ourselves to both write $\boldsymbol{A} \preceq \boldsymbol{B}$ and equivalently $\boldsymbol{B} \succeq \boldsymbol{A}$.

For a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ we define that

$$\boldsymbol{A} \succeq \boldsymbol{0}$$

if and only if $\boldsymbol{A}$ is positive semi-definite.

More generally, when we have two symmetric matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n \times n}$, we will write

$$\boldsymbol{A} \preceq \boldsymbol{B} \text{ if and only if for all } \boldsymbol{x} \in \mathbb{R}^n \text{ we have } \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{x} \tag{4.4}$$

This is a partial order, because it satisfies the three requirements of

1. Reflexivity: $\boldsymbol{A} \preceq \boldsymbol{A}$.
2. Anti-symmetry: $\boldsymbol{A} \preceq \boldsymbol{B}$ and $\boldsymbol{B} \preceq \boldsymbol{A}$ implies $\boldsymbol{A} = \boldsymbol{B}$
3. Transitivity: $\boldsymbol{A} \preceq \boldsymbol{B}$ and $\boldsymbol{B} \preceq \boldsymbol{C}$ implies $\boldsymbol{A} \preceq \boldsymbol{C}$

Check for yourself that these properties hold!

The PSD order has other very useful properties: $\boldsymbol{A} \preceq \boldsymbol{B}$ implies $\boldsymbol{A} + \boldsymbol{C} \preceq \boldsymbol{B} + \boldsymbol{C}$ for any symmetric matrix $\boldsymbol{C}$. Convince yourself of this too!

And, combining this observation with transitivity, we can see that $\boldsymbol{A} \preceq \boldsymbol{B}$ and $\boldsymbol{C} \preceq \boldsymbol{D}$ implies $\boldsymbol{A} + \boldsymbol{C} \preceq \boldsymbol{B} + \boldsymbol{D}$.

Here is another useful property: If $\boldsymbol{0} \preceq \boldsymbol{A}$ then for all $\alpha \geq 1$

$$\frac{1}{\alpha}\boldsymbol{A} \preceq \boldsymbol{A} \preceq \alpha \boldsymbol{A}.$$

Here is another one:

Claim 4.2.2. *If $\boldsymbol{A} \preceq \boldsymbol{B}$, then for all i*

$$\lambda_i(\boldsymbol{A}) \leq \lambda_i(\boldsymbol{B}).$$

Proof. We can prove this Claim by applying the subspace version of the Courant-Fischer theorem.

$$\lambda_i(\boldsymbol{A}) = \min_{\substack{\text{subspace } W \subseteq \mathbb{R}^n \\ \dim(W)=i}} \max_{\boldsymbol{x} \in W, \boldsymbol{x} \neq \boldsymbol{0}} \frac{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \leq \min_{\substack{\text{subspace } W \subseteq \mathbb{R}^n \\ \dim(W)=i}} \max_{\boldsymbol{x} \in W, \boldsymbol{x} \neq \boldsymbol{0}} \frac{\boldsymbol{x}^\top \boldsymbol{B} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \lambda_i(\boldsymbol{B}).$$

□

Note that the converse of Clam 4.2.2 is very much false, for example the matrices $\boldsymbol{A} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$ and $\boldsymbol{B} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ have equal eigenvalues, but both $\boldsymbol{A} \not\preceq \boldsymbol{B}$ and $\boldsymbol{B} \not\preceq \boldsymbol{A}$.

Remark 4.2.3. It's useful to get used to and remember some of the properties of the Loewner order, but all the things we have established so far are almost immediate from the basic characterization in Equation (4.4). So, ideally, don't memorize all these facts, instead, try to see that they are simple consequences of the definition.

4.2.4 Upper Bounding a Laplacian's λ_n Using Degrees

In an earlier chapter, we observed that for any graph $G = (V, E, \boldsymbol{w})$, $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} \succeq \boldsymbol{0}$. We can see this from $\boldsymbol{x}^\top(\boldsymbol{D} - \boldsymbol{A})\boldsymbol{x} = \sum_{(u,v)\in E} \boldsymbol{w}(u,v)(\boldsymbol{x}(u) - \boldsymbol{x}(v))^2 \geq 0$. Similarly $\boldsymbol{D} + \boldsymbol{A} \succeq \boldsymbol{0}$. because $\boldsymbol{x}^\top(\boldsymbol{D} + \boldsymbol{A})\boldsymbol{x} = \sum_{(u,v)\in E} \boldsymbol{w}(u,v)(\boldsymbol{x}(u) + \boldsymbol{x}(v))^2 \geq 0$. But this means that $-\boldsymbol{A} \preceq \boldsymbol{D}$ and hence $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} \preceq 2\boldsymbol{D}$.

So, for the path graph P_n, we have $\boldsymbol{L}_{P_n} \preceq \boldsymbol{D} - \boldsymbol{A} \preceq 2\boldsymbol{D} \preceq 4\boldsymbol{I}$. So by Claim 4.2.2

$$\lambda_n(\boldsymbol{L}_{P_n}) \leq 4. \tag{4.5}$$

We can see that our test vector-based lower bound on $\lambda_n(\boldsymbol{L}_{P_n})$ from Equation (4.3) is tight up to a factor 4.

Since this type of argument works for any unit weight graph, it proves the following claim.

Claim 4.2.4. *For any unit weight graph G, $\lambda_n(\boldsymbol{L}_G) \leq 2 \max_{v \in V} \mathit{degree}(v)$.*

This is tight on a graph consisting of a single edge.

4.2.5 The Loewner Order and Laplacians of Graphs.

It's sometimes convenient to overload the for the PSD order to also apply to graphs. We will write

$$G \preceq H$$

if $\boldsymbol{L}_G \preceq \boldsymbol{L}_H$.

For example, given two unit weight graphs $G = (V, E)$ and $H = (V, F)$, if $H = (V, F)$ is a subgraph of G, then

$$\boldsymbol{L}_H \preceq \boldsymbol{L}_G.$$

We can see this from the Laplacian quadratic form:

$$\boldsymbol{x}^\top \boldsymbol{L}_G \boldsymbol{x} = \sum_{(u,v)\in E} \boldsymbol{w}(u, v)(\boldsymbol{x}(u) - \boldsymbol{x}(v))^2.$$

Dropping edges will only decrease the value of the quadratic form. The same is for decreasing the weights of edges. The graph order notation is especially useful when we allow for scaling a graph by a constant, say $c > 0$,

$$c \cdot H \preceq G$$

What is $c \cdot H$? It is the same graph as H, but the weight of every edge is multiplied by c. Now we can make statements like $\frac{1}{2}H \preceq G \preceq 2H$, which turn out to be useful notion of the two graphs approximating each other.

4.2.6 The Path Inequality

Now, we'll see a general tool for comparing two graphs G and H to prove inequalities like $cH \preceq G$ for some constant c. Our tools won't necessarily work well for all cases, but we'll see some examples where they do.

In the rest of the chapter, we will often need to compare two graphs defined on the same vertex set $V = \{1, \ldots, n\} = [n]$.

We use $G_{i,j}$ to denote the unit weight graph on vertex set $[n]$ consisting of a single edge between vertices i and j.

Lemma 4.2.5 (The Path Inequality)**.**

$$(n - 1) \cdot P_n \succeq G_{1,n},$$

Proof. We want to show that for every $\boldsymbol{x} \in \mathbb{R}^n$,

$$(n - 1) \cdot \sum_{i=1}^{n-1} (\boldsymbol{x}(i + 1) - \boldsymbol{x}(i))^2 \geq (\boldsymbol{x}(n) - \boldsymbol{x}(1))^2.$$

For $i \in [n - 1]$, set

$$\boldsymbol{\Delta}(i) = \boldsymbol{x}(i + 1) - \boldsymbol{x}(i).$$

The inequality we want to prove then becomes

$$(n-1)\sum_{i=1}^{n-1}(\boldsymbol{\Delta}(i))^2 \geq \left(\sum_{i=1}^{n-1}\boldsymbol{\Delta}(i)\right)^2.$$

But, this is immediate from the Cauchy-Schwarz inequality $\boldsymbol{a}^\top \boldsymbol{b} \leq \|\boldsymbol{a}\|_2 \|\boldsymbol{b}\|_2$:

$$\begin{aligned}(n-1)\sum_{i=1}^{n-1}(\boldsymbol{\Delta}(i))^2 &= \|\mathbf{1}_{n-1}\|^2 \cdot \|\boldsymbol{\Delta}\|^2 \\ &= (\|\mathbf{1}_{n-1}\| \cdot \|\boldsymbol{\Delta}\|)^2 \\ &\geq (\mathbf{1}_{n-1}^\top \boldsymbol{\Delta})^2 \\ &= (\sum_{i=1}^{n-1}\boldsymbol{\Delta}(i))^2\end{aligned}$$

□

4.2.7 Lower Bounding λ_2 of a Path Graph

We will now use Lemma 4.2.5 to prove a lower bound on $\lambda_2(\boldsymbol{L}_{P_n})$. Our strategy will be to prove that the path P_n is at least some multiple of the complete graph K_n, measured by the Loewner order, i.e. $K_n \preceq f(n)P_n$ for some function $f : \mathbb{N} \to \mathbb{R}$. We can combine this with our observation earlier that $\lambda_2(\boldsymbol{L}_{K_n}) = n$ to show that

$$f(n)\lambda_2(\boldsymbol{L}_{P_n}) \geq \lambda_2(\boldsymbol{L}_{K_n}) = n, \tag{4.6}$$

and this will give our lower bound on $\lambda_2(\boldsymbol{L}_{P_n})$. When establishing the inequality between P_n and K_n, we can treat each edge of the complete graph separately, by first noting that

$$\boldsymbol{L}_{K_n} = \sum_{i<j}\boldsymbol{L}_{G_{i,j}}$$

For every edge (i, j) in the complete graph, we apply the Path Inequality, Lemma 4.2.5:

$$\begin{aligned}G_{i,j} &\preceq (j-i)\sum_{k=i}^{j-1}G_{k,k+1} \\ &\preceq (j-i)P_n\end{aligned}$$

This inequality says that $G_{i,j}$ is at most $(j-i)$ times the part of the path connecting i to j, and that this part of the path is less than the whole.

Summing inequality (4.3) over all edges $(i, j) \in K_n$ gives

$$K_n = \sum_{i<j}G_{i,j} \preceq \sum_{i<j}(j-i)P_n.$$

To finish the proof, we compute

$$\sum_{i<j}(j-i) \leq \sum_{i<j} n \leq n^3$$

So

$$L_{K_n} \preceq n^3 \cdot L_{P_n}.$$

Plugging this into Equation (4.6) we obtain

$$\frac{1}{n^2} \leq \lambda_2(P_n).$$

This only differs from our test vector-based upper bound in Equation (4.3) by a factor 12.

We could make this considerably tighter by being more careful about the sums.

4.2.8 Laplacian Eigenvalues of the Complete Binary Tree

Let's do the same analysis with the complete binary tree with unit weight edges on $n = 2^{d+1} - 1$ vertices, which we denote by T_d.

T_d is the balanced binary tree on this many vertices, i.e. it consists of a root node, which has two children, each of those children have two children and so on until we reach a depth of d from the root, at which point the child vertices have no more children. A simple induction shows that indeed $n = 2^{d+1} - 1$.

We can also describe the edge set by saying that each node i has edges to its children $2i$ and $2i + 1$ whenever the node labels do not exceed n. We emphasize that we still think of the graph as undirected.

The largest eigenvalue. We'll start by above bounding $\lambda_n(\boldsymbol{L}_{T_d})$ using a test vector.

We let $\boldsymbol{x}(i) = 0$ for all nodes that have a child node, and $\boldsymbol{x}(i) = -1$ for even-numbered leaf nodes and $\boldsymbol{x}(i) = +1$ for odd-numbered leaf nodes. Note that there are $(n+1)/2$ leaf nodes, and every leaf node has a single edge, connecting it to a parent with value 0. Thus

$$\lambda_n(\boldsymbol{L}) = \max_{\boldsymbol{v}\neq\boldsymbol{0}} \frac{\boldsymbol{v}^\top \boldsymbol{L}\boldsymbol{v}}{\boldsymbol{v}^\top \boldsymbol{v}} \geq \frac{\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \frac{(n+1)/2}{(n+1)/2} = 1. \tag{4.7}$$

Meanwhile, every vertex has degree at most 3, so by Claim 4.2.4, $\lambda_n(\boldsymbol{L}) \leq 6$. So we can bound the largest eigenvalue above and below by a constant.

λ_2 and diameter in any graph. The following lemma gives a simple lower bound on λ_2 for any graph.

Lemma 4.2.6. *For any unit weight graph G with diameter D,*

$$\lambda_2(\boldsymbol{L}_G) \geq \frac{1}{nD}.$$

Proof. We will again prove a lower bound comparing G to the complete graph. For each edge $(i,j) \in K_n$, let $G^{i,j}$ denote a shortest path in G from i to j. This path will have length at most D. So, we have

$$\begin{aligned} K_n &= \sum_{i<j} G_{i,j} \\ &\preceq \sum_{i<j} DG^{i,j} \\ &\preceq \sum_{i<j} DG \\ &\preceq n^2 DG. \end{aligned}$$

So, we obtain the bound

$$n^2 D\lambda_2(G) \geq n,$$

which implies our desired statement. □

λ_2 in a tree. Since a complete binary tree T_d has diameter $2d \leq 2\log_2(n)$, by Lemma 4.2.6, $\lambda_2(\boldsymbol{L}_{T_d}) \geq \frac{1}{2n\log_2(n)}$.

Let us give an upper bound on λ_2 of the tree using a test vector. Let $\boldsymbol{x} \in \mathbb{R}^v$ have $\boldsymbol{x}(1) = 0$ and $\boldsymbol{x}(i) = -1$ for i in the left subtree and $\boldsymbol{x}(i) = +1$ in the right subtree. Then

$$\lambda_2(\boldsymbol{L}_{T_d}) = \min_{\substack{\boldsymbol{v}\neq\boldsymbol{0} \\ \boldsymbol{v}^\top\boldsymbol{1}=0}} \frac{\boldsymbol{v}^\top\boldsymbol{L}\boldsymbol{v}}{\boldsymbol{v}^\top\boldsymbol{v}} \leq \frac{\boldsymbol{x}^\top\boldsymbol{L}\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}} = \frac{2}{n-1}.$$

So, we have shown $\frac{1}{2n\log_2(n)} \leq \lambda_2(\boldsymbol{L}_{T_d}) \leq \frac{2}{n-1}$, and unlike the previous examples, the gap is more than a constant.

In the exercises for Week 3, I will ask you to improve the lower bound to $1/(cn)$ for some constant c.

Chapter 5

Conductance, Expanders and Cheeger's Inequality

A common algorithmic problem that arises is the problem of partitioning the vertex set V of a graph G into clusters $X_1, X_2, \ldots, X_k$ such that

- for each i, the *induced* graph $G[X_i] = (X_i, E \cap (X_i \times X_i))$ is "well-connected", and
- only an ϵ-fraction of edges e are not contained in any induced graph $G[X_i]$ (where ϵ is a very small constant).

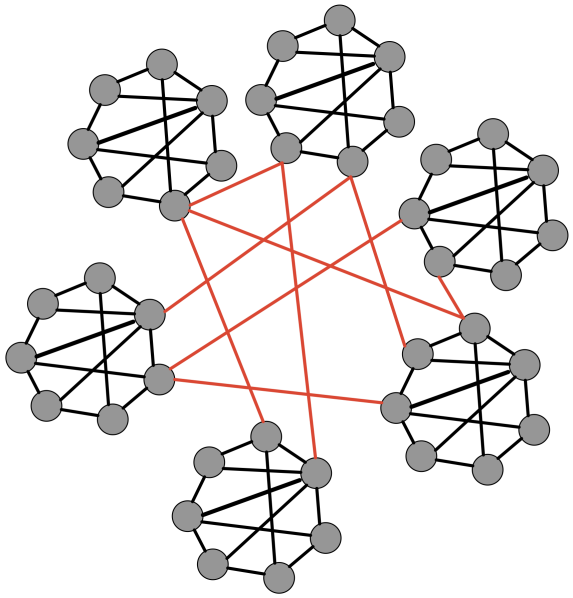

Figure 5.1: After removing the red edges (of which there are few in relation to the total number of edges), each connected component in G is "well-connected".

In this lecture, we make precise what "well-connected" means by introducing the notions of *conductance* and *expanders*.

Building on the chapter, we show that the second eigenvalue of the Laplacian L associated with graph G can be used to certify that a graph is "well-connected" (more precisely the second eigenvalue of a normalized version of the Laplacian). This result, called Cheeger's inequality, is one of the key tools in Spectral Graph Theory. Moreover, it can be turned into an algorithm that computes the partition efficiently!

5.1 Conductance and Expanders

Graph Definitions. In this lecture, we let $G = (V, E)$ be unweighted[1] and always be *connected*, and let $\boldsymbol{d}(v)$ be the degree of a vertex v in G. We define the *volume* $\mathrm{vol}(S)$ for any vertex subset $S \subseteq V$, to be the sum of degrees, i.e. $\mathrm{vol}(S) = \sum_{v \in S} \boldsymbol{d}(v)$.

For any $A, B \subseteq V$, we define $E(A, B)$ to be the set of edges in $E \cap (A \times B)$, i.e. with one endpoint in A and one endpoint in B. We let $G[A]$ be the *induced* graph G by $A \subseteq V$, which is the graph G restricted to the vertices A, i.e. an edge e in G is in $G[A]$ iff both endpoints are in A.

Conductance. Given set $\emptyset \subset S \subset V$, then we define the conductance $\phi(S)$ of S by

$$\phi(S) = \frac{|E(S, V \setminus S)|}{\min\{\mathrm{vol}(S), \mathrm{vol}(V \setminus S)\}}.$$

It can be seen that $\phi(\cdot)$ is symmetric in the sense that $\phi(S) = \phi(V \setminus S)$. We define the conductance of the graph G denoted $\phi(G)$ by

$$\phi(G) = \min_{\emptyset \subset S \subset V} \phi(S).$$

We note that finding the conductance of a graph G is NP-hard. However, good approximations can be found as we will see today (and in a later lecture).

Expander and Expander Decomposition. For any $\phi \in (0, 1]$, we say that a graph G is a *ϕ-expander* if $\phi(G) \geq \phi$. We say that the partition $X_1, X_2, \ldots, X_k$ of the vertex set V is a *ϕ-expander decomposition* of quality q if

- each induced graph $G[X_i]$ is a ϕ-expander, and
- the number of edges not contained in any $G[X_i]$ is at most $q \cdot \phi \cdot m$.

[1] Everything we present here also works for weighted graphs, however, we focus on unweighted graphs for simplicity.

Today, we obtain a ϕ-expander decomposition of quality $q = O(\phi^{-1/2} \cdot \log n)$. In a few lectures, we revisit the problem and obtain quality $q = O(\log^c n)$ for some small constant c. In practice, we mostly care about values $\phi \approx 1$.

An Algorithm to Compute Conductance and Expander Decomposition. In this lecture, the main focus is *not* to obtain an algorithm to compute conductance but rather only to show that the conductance of a graph can be approximated using the eigenvalues of the "normalized" Laplacian.

However, this proof gives then rise to an algorithm CertifyOrCut(G, ϕ) that given a graph G and a parameter ϕ either:

- *Certifies* that G is a ϕ-expander, or
- Presents a *cut* S such that $\phi(S) \leq \sqrt{2\phi}$.

Careful but rather straight-forward recursive application of this procedure then yields an algorithm to compute a ϕ-expander decomposition.

5.2 A Lower Bound for Conductance via Eigenvalues

An Alternative Characterization of Conductance. Let us now take a closer look at the definition of conductance and observe that if a set S has $\text{vol}(S) \leq \text{vol}(V)/2$ then

$$\phi(S) = \frac{|E(S, V \setminus S)|}{\min\{\text{vol}(S), \text{vol}(V \setminus S)\}} = \frac{|E(S, V \setminus S)|}{\text{vol}(S)} = \frac{\mathbf{1}_S^\top \boldsymbol{L} \mathbf{1}_S}{\mathbf{1}_S^\top \boldsymbol{D} \mathbf{1}_S}.$$

To see this, observe that we can rewrite the numerator above using the Laplacian of G as

$$|E(S, V \setminus S)| = \sum_{(u,v) \in E} (\mathbf{1}_S(u) - \mathbf{1}_S(v))^2 = \mathbf{1}_S^\top \boldsymbol{L} \mathbf{1}_S$$

where $\mathbf{1}_S$ is the characteristic vector of S. Further, we can rewrite the denominator as

$$\text{vol}(S) = \mathbf{1}_S^\top \boldsymbol{d} = \mathbf{1}_S^\top \boldsymbol{D} \mathbf{1}_S$$

where $\boldsymbol{D} = \text{diag}(\boldsymbol{d})$ is the degree-matrix. We can now alternatively define the graph conductance of G by

$$\phi(G) = \min_{\substack{\emptyset \subset S \subset V, \\ \text{vol}(S) \leq \text{vol}(V)/2}} \frac{\mathbf{1}_S^\top \boldsymbol{L} \mathbf{1}_S}{\mathbf{1}_S^\top \boldsymbol{D} \mathbf{1}_S} \tag{5.1}$$

where we use that $\phi(S) = \phi(V \setminus S)$ such that the objective value is unchanged as long as for each set $\emptyset \subset S \subset V$ either S or $V \setminus S$ is in the set that we minimize over.

The Normalized Laplacian. Let us next define the *normalized* Laplacian

$$\boldsymbol{N} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}.$$

To learn a bit about this new matrix, let us first look at the first eigenvalue where we use the test vector $y = \boldsymbol{D}^{1/2}\mathbf{1}$, to get by Courant-Fischer (see Theorem 4.1.2) that

$$\lambda_1(\boldsymbol{N}) = \min_{\boldsymbol{x}\neq\mathbf{0}} \frac{\boldsymbol{x}^\top \boldsymbol{N}\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}} \leq \frac{\boldsymbol{y}^\top \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}\boldsymbol{y}}{\boldsymbol{y}^\top\boldsymbol{y}} = \frac{\mathbf{1}^\top\boldsymbol{L}\mathbf{1}}{\boldsymbol{y}^\top\boldsymbol{y}} = 0 \tag{5.2}$$

because $\boldsymbol{D}^{-1/2}\boldsymbol{D}^{1/2} = I$ and $\boldsymbol{L}\mathbf{1} = 0$ (for the former we use the assumption that G is connected). Since $\boldsymbol{N}$ is PSD (as you will show in the exercises), we also know $\lambda_1(\boldsymbol{N}) \geq 0$, so $\lambda_1(\boldsymbol{N}) = 0$.

Let us use Courant-Fischer again to reason a bit about the second eigenvalue of $\boldsymbol{N}$:

$$\lambda_2(\boldsymbol{N}) = \min_{\substack{\boldsymbol{x}\perp\boldsymbol{D}^{1/2}\mathbf{1}\\ \boldsymbol{x}\neq\mathbf{0}}} \frac{\boldsymbol{x}^\top\boldsymbol{N}\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}} = \min_{\substack{\boldsymbol{z}\perp\boldsymbol{d}\\ \boldsymbol{z}\neq\mathbf{0}}} \frac{\boldsymbol{z}^\top\boldsymbol{D}^{1/2}\boldsymbol{N}\boldsymbol{D}^{1/2}\boldsymbol{z}}{\boldsymbol{z}^\top\boldsymbol{D}^{1/2}\boldsymbol{D}^{1/2}\boldsymbol{z}} = \min_{\substack{\boldsymbol{z}\perp\boldsymbol{d}\\ \boldsymbol{z}\neq\mathbf{0}}} \frac{\boldsymbol{z}^\top\boldsymbol{L}\boldsymbol{z}}{\boldsymbol{z}^\top\boldsymbol{D}\boldsymbol{z}}. \tag{5.3}$$

Relating Conductance to the Normalized Laplacian. At this point, it might become clearer why $\boldsymbol{N}$ is a natural matrix to consider when arguing about conductance: if we could argue that for every $\emptyset \subset S \subset V, \mathrm{vol}(S) \leq \mathrm{vol}(V)/2$, we have $\mathbf{1}_S \perp \boldsymbol{d}$, then it would be easy to see that taking the second eigenvalue of $\boldsymbol{N}$ in equation 5.3 is a relaxation of the minimization problem 5.1 defining $\phi(G)$.

While this is clearly not true, we can still argue along these lines.

Theorem 5.2.1 (Cheeger's Inequality, Lower Bound)**.** *We have* $\frac{\lambda_2(\boldsymbol{N})}{2} \leq \phi(G)$.

Proof. Instead of using $\mathbf{1}_S$ directly, we shift $\mathbf{1}_S$ by $\mathbf{1}$ such that it is orthogonal to $\boldsymbol{d}$: we define $\boldsymbol{z}_S = \mathbf{1}_S - \alpha\mathbf{1}$ where α is the scalar that solves

$$\begin{aligned}
0 &= \boldsymbol{d}^\top\boldsymbol{z}_S \\
\iff 0 &= \boldsymbol{d}^\top(\mathbf{1}_S - \alpha\mathbf{1}) \\
\iff 0 &= \boldsymbol{d}^\top\mathbf{1}_S - \alpha\boldsymbol{d}^\top\mathbf{1} \\
\iff \alpha &= \frac{\boldsymbol{d}^\top\mathbf{1}_S}{\boldsymbol{d}^\top\mathbf{1}} = \frac{\mathrm{vol}(S)}{\mathrm{vol}(V)}.
\end{aligned}$$

To conclude the proof, it remains to argue that $\frac{\mathbf{1}_S^\top\boldsymbol{L}\mathbf{1}_S}{\mathbf{1}_S^\top\boldsymbol{D}\mathbf{1}_S} \geq \frac{1}{2}\cdot\frac{\boldsymbol{z}_S^\top\boldsymbol{L}\boldsymbol{z}_S}{\boldsymbol{z}_S^\top\boldsymbol{D}\boldsymbol{z}_S}$:

- Numerator: since $\mathbf{1}^\top\boldsymbol{L}\mathbf{1} = 0$, we have that $\mathbf{1}_S^\top\boldsymbol{L}\mathbf{1}_S = \boldsymbol{z}_S^\top\boldsymbol{L}\boldsymbol{z}_S$.

- Denominator: observe by straight-forward calculations that

$$\begin{aligned}\boldsymbol{z}_S^\top \boldsymbol{D}\boldsymbol{z}_S &= \mathrm{vol}(S)\cdot(1-\alpha)^2 + \mathrm{vol}(V\setminus S)\cdot(-\alpha)^2 \\ &= \mathrm{vol}(S) - 2\,\mathrm{vol}(S)\cdot\alpha + \mathrm{vol}(V)\cdot\alpha^2 \\ &= \mathrm{vol}(S) - \frac{\mathrm{vol}(S)^2}{\mathrm{vol}(V)} \\ &= \mathrm{vol}(S) - \mathrm{vol}(S)\cdot\frac{\mathrm{vol}(S)}{\mathrm{vol}(V)} \\ &\geq \frac{1}{2}\,\mathrm{vol}(S) = \frac{1}{2}\mathbf{1}_S^\top \boldsymbol{D}\mathbf{1}_S\end{aligned}$$

  where we use the assumption that $\mathrm{vol}(S) \leq \mathrm{vol}(V)/2$.

□

5.3 An Upper Bound for Conductance via Eigenvalues

Slightly more surprisingly, we can also show that the second eigenvalue $\lambda_2(\boldsymbol{N})$ can be used to upper bound the conductance.

Theorem 5.3.1 (Cheeger's Inequality, Upper Bound)**.** *We have* $\phi(G) \leq \sqrt{2\cdot\lambda_2(\boldsymbol{N})}$.

Proof. To prove the theorem, we want to show that for *any* $\boldsymbol{z} \perp \boldsymbol{d}$, we can find a set $\emptyset \subset S \subset V$ with $\mathrm{vol}(S) \leq \mathrm{vol}(V)/2$, such that

$$\frac{\mathbf{1}_S^\top \boldsymbol{L}\mathbf{1}_S}{\mathbf{1}_S^\top \boldsymbol{D}\mathbf{1}_S} \leq \sqrt{2\cdot\frac{\boldsymbol{z}^\top\boldsymbol{L}\boldsymbol{z}}{\boldsymbol{z}^\top\boldsymbol{D}\boldsymbol{z}}}. \tag{5.4}$$

As a first step, we would like to change $\boldsymbol{z}$ slightly to make it more convenient to work with:

- we *renumber* the vertices in V such that we have

$$\boldsymbol{z}(1) \leq \boldsymbol{z}(2) \leq \cdots \leq \boldsymbol{z}(n).$$

- we *center* $\boldsymbol{z}$, that is we let $\boldsymbol{z}_c = \boldsymbol{z} - \alpha\mathbf{1}$ where α is chosen such that

$$\sum_{\boldsymbol{z}_c(i)<0} \boldsymbol{d}(i) < \mathrm{vol}(V)/2 \text{ and } \sum_{\boldsymbol{z}_c(i)\leq 0} \boldsymbol{d}(i) \geq \mathrm{vol}(V)/2$$

  i.e. $\sum_{\boldsymbol{z}_c(i)>0} \boldsymbol{d}(i) \leq \mathrm{vol}(V)/2$.

- we *scale*, let $\boldsymbol{z}_{sc} = \beta\boldsymbol{z}_c$ for some scalar β such that $\boldsymbol{z}_{sc}(1)^2 + \boldsymbol{z}_{sc}(n)^2 = 1$.

In the exercises, you will show that changing $\boldsymbol{z}$ to $\boldsymbol{z}_{sc}$ can only make the ratio we are interested in smaller, i.e. that $\frac{\boldsymbol{z}^\top \boldsymbol{L}\boldsymbol{z}}{\boldsymbol{z}^\top \boldsymbol{D}\boldsymbol{z}} \geq \frac{\boldsymbol{z}_{sc}^\top \boldsymbol{L}\boldsymbol{z}_{sc}}{\boldsymbol{z}_{sc}^\top \boldsymbol{D}\boldsymbol{z}_{sc}}$. Thus, if we can show that equation 5.4 holds for $\boldsymbol{z}_{sc}$ in place of $\boldsymbol{z}$, then it also follows for $\boldsymbol{z}$ itself.

We now arrive at the main idea of the proof: we define the set $S_\tau = \{i \in V \mid \boldsymbol{z}_{sc}(i) < \tau\}$ for some random variable τ with probability density function

$$p(t) = \begin{cases} 2 \cdot |t| & t \in [\boldsymbol{z}_{sc}(1), \boldsymbol{z}_{sc}(n)], \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

So, we have probability $\mathbb{P}[a < \tau < b] = \int_{t=a}^{b} p(t)\, dt$.

Since the volume incident to S_τ might be quite large, let us define S for convenience by

$$S = \begin{cases} S_\tau & \mathrm{vol}(S_\tau) < \mathrm{vol}(V)/2, \\ V \setminus S_\tau & \text{otherwise.} \end{cases}$$

Claim 5.3.2. *We have* $\frac{\mathbb{E}_\tau\left[\mathbf{1}_S^\top \boldsymbol{L}\mathbf{1}_S\right]}{\mathbb{E}_\tau\left[\mathbf{1}_S^\top \boldsymbol{D}\mathbf{1}_S\right]} \leq \sqrt{2 \cdot \frac{\boldsymbol{z}_{sc}^\top \boldsymbol{L}\boldsymbol{z}_{sc}}{\boldsymbol{z}_{sc}^\top \boldsymbol{D}\boldsymbol{z}_{sc}}}$.

Proof. Recall $\mathbf{1}_S^\top \boldsymbol{L}\mathbf{1}_S = E(S_\tau, V \setminus S_\tau)$, and by choice of τ, we have for any edge $e = \{i, j\} \in E$ where $\boldsymbol{z}_{sc}(i) \leq \boldsymbol{z}_{sc}(j)$,

$$\begin{aligned} \mathbb{P}_\tau[e \in E(S_\tau, V \setminus S_\tau)] &= \mathbb{P}_\tau[\boldsymbol{z}_{sc}(i) < \tau \leq \boldsymbol{z}_{sc}(j)] \\ &= \int_{t=i}^{j} 2|t|\, dt = \mathrm{sgn}(j) \cdot \boldsymbol{z}_{sc}(j)^2 - \mathrm{sgn}(i) \cdot \boldsymbol{z}_{sc}(i)^2. \end{aligned}$$

Distinguishing by cases, we get

$$\mathrm{sgn}(j) \cdot \boldsymbol{z}_{sc}(j)^2 - \mathrm{sgn}(i) \cdot \boldsymbol{z}_{sc}(i)^2 = \begin{cases} |\boldsymbol{z}_{sc}(i)^2 - \boldsymbol{z}_{sc}(j)^2| & \mathrm{sgn}(i) = \mathrm{sgn}(j), \\ \boldsymbol{z}_{sc}(i)^2 + \boldsymbol{z}_{sc}(j)^2 & \text{otherwise.} \end{cases}$$

We can further upper bound either case by $|\boldsymbol{z}_{sc}(i) - \boldsymbol{z}_{sc}(j)| \cdot (|\boldsymbol{z}_{sc}(i)| + |\boldsymbol{z}_{sc}(j)|)$ (we leave this as an exercise).

Using our new upper bound, we can sum over all edges $e \in E$ to conclude that

$$\begin{aligned} \mathbb{E}_\tau[|E(S_\tau, V \setminus S_\tau)|] &\leq \sum_{i \sim j} |\boldsymbol{z}_{sc}(i) - \boldsymbol{z}_{sc}(j)| \cdot (|\boldsymbol{z}_{sc}(i)| + |\boldsymbol{z}_{sc}(j)|) \\ &\leq \sqrt{\sum_{i \sim j} (\boldsymbol{z}_{sc}(i) - \boldsymbol{z}_{sc}(j))^2 \cdot \sum_{i \sim j} (|\boldsymbol{z}_{sc}(i)| + |\boldsymbol{z}_{sc}(j)|)^2} \end{aligned}$$

where the last line follows from $\langle \boldsymbol{x}, \boldsymbol{y} \rangle^2 \leq \langle \boldsymbol{x}, \boldsymbol{x} \rangle \cdot \langle \boldsymbol{y}, \boldsymbol{y} \rangle$ (i.e. Cauchy-Schwarz).

The first sum should look familiar by now: it is simply the Quadratic Laplacian Form $\sum_{i\sim j}(\boldsymbol{z}_{sc}(i)-\boldsymbol{z}_{sc}(j))^2 = \boldsymbol{z}_{sc}^\top \boldsymbol{L}\boldsymbol{z}_{sc}$.

It is not hard to reason about the second term either

$$\sum_{i\sim j}(|\boldsymbol{z}_{sc}(i)|+|\boldsymbol{z}_{sc}(j)|)^2 \leq 2\sum_{i\sim j}\boldsymbol{z}_{sc}(i)^2+\boldsymbol{z}_{sc}(j)^2 = 2\sum_{i\in V}\boldsymbol{d}(i)\boldsymbol{z}_{sc}(i)^2 = 2\boldsymbol{z}_{sc}^\top \boldsymbol{D}\boldsymbol{z}_{sc}.$$

Putting everything together, we obtain

$$\mathbb{E}_\tau[|E(S_\tau, V\setminus S_\tau)|] \leq \sqrt{\boldsymbol{z}_{sc}^\top \boldsymbol{L}\boldsymbol{z}_{sc}\cdot 2\boldsymbol{z}_{sc}^\top \boldsymbol{D}\boldsymbol{z}_{sc}} = \sqrt{2\cdot\frac{\boldsymbol{z}_{sc}^\top \boldsymbol{L}\boldsymbol{z}_{sc}}{\boldsymbol{z}_{sc}^\top \boldsymbol{D}\boldsymbol{z}_{sc}}}\cdot \boldsymbol{z}_{sc}^\top \boldsymbol{D}\boldsymbol{z}_{sc} \tag{5.6}$$

While this almost looks like what we want, we still have to argue that $\boldsymbol{z}_{sc}^\top \boldsymbol{D}\boldsymbol{z}_{sc} = \mathbb{E}_\tau[\mathbf{1}_S^\top \boldsymbol{D}\mathbf{1}_S]$ to finish the proof.

To this end, when unrolling the expectation, we use a simple trick that splits by cases:

$$\begin{aligned}
\mathbb{E}_\tau[\mathbf{1}_S^\top \boldsymbol{D}\mathbf{1}_S] &= \sum_{i\in V}\boldsymbol{d}(i)\cdot\mathbb{P}[i\in S] \\
&= \sum_{i\in V,\boldsymbol{z}_{sc}(i)<0}\boldsymbol{d}(i)\cdot\mathbb{P}[i\in S\wedge S=S_\tau] + \sum_{i\in V,\boldsymbol{z}_{sc}(i)\geq 0}\boldsymbol{d}(i)\cdot\mathbb{P}[i\in S\wedge S\neq S_\tau] \\
&= \sum_{i\in V,\boldsymbol{z}_{sc}(i)<0}\boldsymbol{d}(i)\cdot\mathbb{P}[\boldsymbol{z}_{sc}(i)<\tau\wedge\tau<0] + \sum_{i\in V,\boldsymbol{z}_{sc}(i)\geq 0}\boldsymbol{d}(i)\cdot\mathbb{P}[\boldsymbol{z}_{sc}(i)\geq\tau\wedge\tau\geq 0]
\end{aligned}$$

where we use the centering of $\boldsymbol{z}_{sc}$ the definition of S and that the event $\{i\in S\wedge S=S_\tau\}$ can be rewritten as the event $\{i<\tau\wedge\tau<0\}$ (the other case is analogous).

Let i be a vertex with $\boldsymbol{z}_{sc}(i)<0$, then the probability $\mathbb{P}[i\in S\wedge S=S_\tau]$ is exactly $\boldsymbol{z}_{sc}(i)^2$ by choice of the density function of τ (again the case for i with $\boldsymbol{z}_{sc}(i)$ non-negative is analgous). Thus, summing over all vertices, we obtain

$$\begin{aligned}
\mathbb{E}_\tau[\mathbf{1}_S^\top \boldsymbol{D}\mathbf{1}_S] &= \sum_{i\in V,\boldsymbol{z}_{sc}(i)<0}\boldsymbol{d}(i)\cdot\mathbb{P}[\boldsymbol{z}_{sc}(i)<\tau\wedge\tau<0] + \sum_{i\in V,\boldsymbol{z}_{sc}(i)\geq 0}\mathbb{P}[\boldsymbol{z}_{sc}(i)\geq\tau\wedge\tau\geq 0] \\
&= \sum_{i\in V}\boldsymbol{d}(i)\cdot\boldsymbol{z}_{sc}(i)^2 = \boldsymbol{z}_{sc}^\top \boldsymbol{D}\boldsymbol{z}_{sc}.
\end{aligned}$$

Therefore, we can plug in our result directly into Equation 5.6 and the proof is completed by dividing both sides by $\mathbb{E}_\tau[\mathbf{1}_S^\top \boldsymbol{D}\mathbf{1}_S]$. □

While Theorem 5.3.2 only ensures our claim in expectation, this is already sufficient to conclude that there exists some set S that satisfies the same guarantees deterministically, as you will prove in Problem Set 4. This is often called the *probabilistic method of expectation* and can be seen from the definition of expectation. We have thus proven the upper bound of Cheeger's inequality. □

5.4 Conclusion

Today, we have introduced the concepts of conductance and formalized expanders and expander decompositions. These are crucial concepts that you will encounter often in literature and also again in this course. They are a key tool in many recent breakthroughs in Theoretical Computer Science.

In the second part of the lecture (the main part), we discussed Cheeger's inequality which allows to relate the second eigenvalue of the normalized Laplacian to a graphs conductance. We summarize the full statement here.

Theorem 5.4.1 (Cheeger's Inequality)**.** *We have* $\frac{\lambda_2(\boldsymbol{N})}{2} \leq \phi(G) \leq \sqrt{2 \cdot \lambda_2(\boldsymbol{N})}$.

We point out that this Theorem is tight as you will show in the exercises. The proof for Cheeger's inequality is probably the most advanced proof, we have seen so far in the course. The many tricks that make the proof work might sometimes seem a bit magical but it is important to remember that they are a result of many people polishing this proof over and over. The proof techniques used are extremely useful and can be re-used in various contexts. We therefore strongly encourage you to really understand the proof yourself!

Chapter 6

Random Walks

Today, we talk about random walks on graphs and how the spectrum of the Laplacian guides convergence of random walks. We start by giving the definition of a random walk on a weighted graph $G = (V, E, w)$.

6.1 A Primer on Random Walks

Random Walk Basics. We call a random sequence of vertices $v_0, v_1, \ldots$ a *random walk* on G, if v_0 is a vertex in G chosen according to some probability distribution $\boldsymbol{p}_0 \in \mathbb{R}^V$; and for any $t \geq 0$, we have

$$\mathbb{P}[v_{t+1} = v \mid v_t = u] = \begin{cases} \boldsymbol{w}(u,v)/\boldsymbol{d}(u) & \text{if } \{u,v\} \in E, \\ 0 & \text{otherwise.} \end{cases}$$

To gain some intuition for the definition, assume first that the graph G is undirected. Consider a particle that is placed at a random vertex v_0 initially. Then at each step the particle is moved to a neighbor of the current vertex it is resting at, where the neighbor is chosen uniformly at random.

If the graph is weighted, then instead of choosing a neighbor v_{t+1} of a vertex v_t at each step uniformly at random, one chooses a neighbor v of v_t with probability $\boldsymbol{w}(v, v_t)$ divided by the degree $\boldsymbol{d}(v_t)$.

The Random Walk Matrix. We now define the random walk matrix $\boldsymbol{W}$ by

$$\boldsymbol{W} = \boldsymbol{A}\boldsymbol{D}^{-1}$$

and observe that for all vertices $u, v \in V$ (and any t), we have that

$$\boldsymbol{W}_{vu} = \begin{cases} \boldsymbol{w}(u,v)/\boldsymbol{d}(u) & \text{if } \{u,v\} \in E, \\ 0 & \text{otherwise.} \end{cases}$$

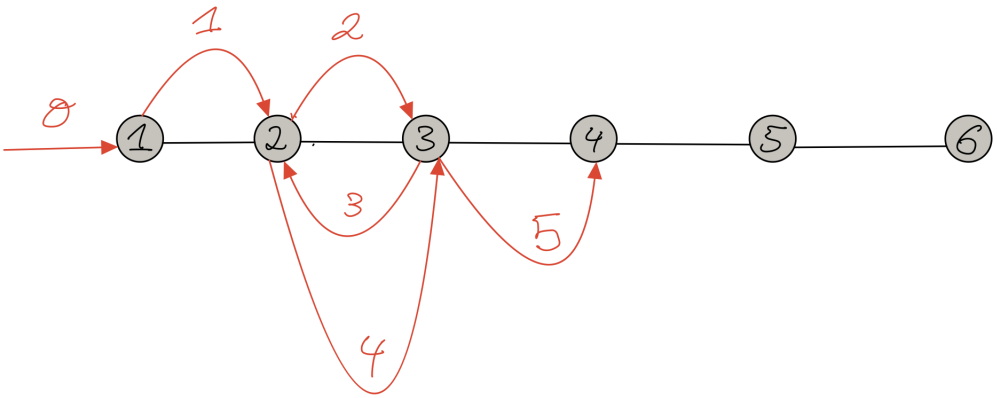

Figure 6.1: A (possibly random) walk where the red edges indicate the edges that the particle moves along. Here the walk visits the vertices $v_0 = 1, v_1 = 2, v_2 = 3, v_3 = 2, v_4 = 3, v_5 = 4$.

Thus, $\boldsymbol{W}_{vu} = \mathbb{P}[v_{t+1} = v \mid v_t = u]$ (for any t).

Therefore, $W\mathbf{1}_u$ is the distribution over the vertices that the random walk visits them at the next time step, given that it currently is at u. More generally, we can now compute the distribution $\boldsymbol{p}_1$ over the vertices that they are visited at time 1 by $\boldsymbol{W}\boldsymbol{p}_0$, the distribution $\boldsymbol{p}_2$ by $\boldsymbol{W}\boldsymbol{p}_1 = \boldsymbol{W}(\boldsymbol{W}\boldsymbol{p}_0)$ and so on. Another way of writing this is $\boldsymbol{p}_t = \boldsymbol{W}^t\boldsymbol{p}_0$.

6.2 Convergence Results for Random Walks

In this first part of the chapter, we are interested mostly in convergence of random walks. We want to answer the following two questions:

- How does a random walk behave after a large number of steps are taken?
- How many steps does it take asymptotically until the random walk behaves as if an infinite number of steps were taken?

To start shedding some light on these questions, we introduce stationary distributions.

Stationary Distribution. We call a distribution $\boldsymbol{\pi} \in \mathbb{R}^V$, a *stationary distribution* if $\boldsymbol{W}\boldsymbol{\pi} = \boldsymbol{\pi}$. That is $\boldsymbol{\pi}$ is an eigenvector of $\boldsymbol{W}$ associated with eigenvalue 1. It turns out such a stationary distribution always exists.

Lemma 6.2.1. *Every graph G has a stationary distribution.*

Proof. Let $\boldsymbol{\pi} = \frac{\boldsymbol{d}}{\mathbf{1}^\top \boldsymbol{d}}$. Clearly, we have that $\|\boldsymbol{\pi}\|_1 = \sum_{v \in V} \boldsymbol{d}(v)/\mathbf{1}^\top \boldsymbol{d} = \frac{1}{\mathbf{1}^\top \boldsymbol{d}} \sum_{v \in V} \boldsymbol{d}(v) = 1$,

so $\boldsymbol{\pi}$ is indeed a distribution. Further note that

$$\boldsymbol{W}\boldsymbol{\pi} = \boldsymbol{A}\boldsymbol{D}^{-1} \cdot \frac{\boldsymbol{d}}{\mathbf{1}^\top \boldsymbol{d}} = \frac{\boldsymbol{A}\mathbf{1}}{\mathbf{1}^\top \boldsymbol{d}} = \frac{\boldsymbol{d}}{\mathbf{1}^\top \boldsymbol{d}} = \boldsymbol{\pi}.$$

□

For many graphs one can show that for $t \to \infty$, we have that $\boldsymbol{p}_t \to \boldsymbol{\pi}$, i.e. that independent of the starting distribution $\boldsymbol{p}_0$, the random walk always converges to distribution $\boldsymbol{\pi}$.

Unfortunately, this is not true for all graphs: take the graph of two vertices connected by a single edge with $\boldsymbol{p}_0$ being 1 at one vertex and 0 at the other.

6.2.1 Making Random Walks Lazy

Lazy Random Walks. Luckily, we can overcome this issue by using a *lazy random walk*. A lazy random walk behaves just like a random walk, however, at each time step, with probability $\frac{1}{2}$ instead of transitioning to a neighbor, it simply stays put. We give the lazy random walk matrix by

$$\tilde{\mathbf{W}} = \frac{1}{2}\boldsymbol{I} + \frac{1}{2}\boldsymbol{W} = \frac{1}{2}\left(\boldsymbol{I} + \boldsymbol{A}\boldsymbol{D}^{-1}\right).$$

It is not hard to see that the stationary distribution $\boldsymbol{\pi}$ for $\boldsymbol{W}$, is also a stationary distribution for $\tilde{\mathbf{W}}$.

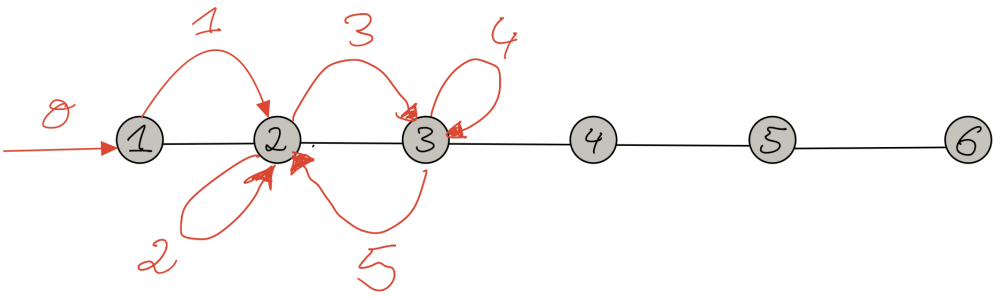

Figure 6.2: A lazy random walk where the red edges indicate the edges that the particle moves along. Here the lazy walk visits the vertices $v_0 = 1, v_1 = 2, v_2 = 2, v_3 = 3, v_4 = 3, v_5 = 2$.

Lazy Random Walks and the Normalized Laplacian. Recall that we defined $\boldsymbol{N} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2} \iff \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \boldsymbol{N}$. We can therefore derive

$$\begin{aligned}
\tilde{\mathbf{W}} &= \frac{1}{2}\boldsymbol{I} + \frac{1}{2}\boldsymbol{A}\boldsymbol{D}^{-1} \\
&= \frac{1}{2}\boldsymbol{I} + \frac{1}{2}\boldsymbol{D}^{1/2}\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}\boldsymbol{D}^{-1/2} \\
&= \frac{1}{2}\boldsymbol{I} + \frac{1}{2}\boldsymbol{D}^{1/2}(\boldsymbol{I} - \boldsymbol{N})\boldsymbol{D}^{-1/2} \\
&= \frac{1}{2}\boldsymbol{I} + \frac{1}{2}\boldsymbol{D}^{1/2}\boldsymbol{I}\boldsymbol{D}^{-1/2} - \frac{1}{2}\boldsymbol{D}^{1/2}\boldsymbol{N}\boldsymbol{D}^{-1/2} \\
&= \boldsymbol{I} - \frac{1}{2}\boldsymbol{D}^{1/2}\boldsymbol{N}\boldsymbol{D}^{-1/2}
\end{aligned}$$

We will now start to reason about the eigenvalues and eigenvectors of $\tilde{\mathbf{W}}$ in terms of the normalized laplacian $\boldsymbol{N}$ that we are already familiar with.

For the rest of the lecture, we let $\nu_1 \leq \nu_2 \leq \cdots \leq \nu_n$ be the eigenvalues of $\boldsymbol{N}$ associated with the orthogonal eigenvectors $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_n$ where we know that such eigenvectors exist by the Spectral Theorem. We note in particular that from the last lecture, we have that $\boldsymbol{\psi}_1 = \frac{\boldsymbol{d}^{1/2}}{(\mathbf{1}^\top \boldsymbol{d})^{1/2}}$ (see Equation 5.2 where we added a normalization such that $\boldsymbol{\psi}_1^\top \boldsymbol{\psi}_1 = 1$).

Lemma 6.2.2. *For the i^{th} eigenvalue ν_i of $\boldsymbol{N}$ associated with eigenvector $\boldsymbol{\psi}_i$, we have that $\tilde{\mathbf{W}}$ has an eigenvalue of $(1 - \frac{1}{2}\nu_i)$ associated with eigenvector $\boldsymbol{D}^{1/2}\boldsymbol{\psi}_i$.*

Proof. The proof is by straight-forward calculations

$$\begin{aligned}
\tilde{\mathbf{W}}\boldsymbol{D}^{1/2}\boldsymbol{\psi}_i &= (\boldsymbol{I} - \frac{1}{2}\boldsymbol{D}^{1/2}\boldsymbol{N}\boldsymbol{D}^{-1/2})\boldsymbol{D}^{1/2}\boldsymbol{\psi}_i \\
&= \boldsymbol{D}^{1/2}\boldsymbol{\psi}_i - \frac{1}{2}\boldsymbol{D}^{1/2}\boldsymbol{N}\boldsymbol{\psi}_i \\
&= \boldsymbol{D}^{1/2}\boldsymbol{\psi}_i - \frac{1}{2}\boldsymbol{D}^{1/2}\boldsymbol{\psi}_i\nu_i = \boldsymbol{D}^{1/2}\boldsymbol{\psi}_i(1 - \frac{1}{2}\nu_i).
\end{aligned}$$

□

Corollary 6.2.3. *Every eigenvalue of $\tilde{\mathbf{W}}$ is in $[0, 1]$.*

Proof. Recall that $\boldsymbol{L} \preccurlyeq 2\boldsymbol{D}$ which implies that $\boldsymbol{N} \preccurlyeq 2\boldsymbol{I}$. But this implies that every eigenvalue of $\boldsymbol{N}$ is in $[0, 2]$. Thus, using Lemma 6.2.2, the corollary follows. □

6.2.2 Convergence of Lazy Random Walks

We have now done enough work to obtain an interesting result. We can derive an alternative characterization of $\boldsymbol{p}_t$ by expanding $\boldsymbol{p}_0$ along an orthogonal eigenvectors basis and then we can repeatedly apply $\tilde{\mathbf{W}}$ by taking powers of the eigenvalues.

Unfortunately, $\tilde{\mathbf{W}}$ is not symmetric so its eigenvectors are not necessarily orthogonal. Instead, we use a simple trick that allows to expand along the eigenvectors of $\boldsymbol{N}$

$$\forall i, \boldsymbol{\psi}_i^\top \boldsymbol{D}^{-1/2}\boldsymbol{p}_0 = \alpha_i \iff \boldsymbol{D}^{-1/2}\boldsymbol{p}_0 = \sum_{i=1}^n \alpha_i \boldsymbol{\psi}_i \iff \boldsymbol{p}_0 = \sum_{i=1}^n \alpha_i \boldsymbol{D}^{1/2}\boldsymbol{\psi}_i. \tag{6.1}$$

The above equivalences are best understood if you start from the middle. To get to the left side, you need to observe that multiplying both sides by $\boldsymbol{\psi}_i^\top$ cancels all terms $\boldsymbol{\psi}_j$ with $j \neq i$ in the sum by orthogonality. To get the right hand side expression, one can simply left-multiply by $\boldsymbol{D}^{1/2}$. Technically, we have to show that $\boldsymbol{D}^{-1/2}\boldsymbol{p}_0$ lives in the eigenspace of $\boldsymbol{N}$ but we leave this as an exercise.

This allows us to express a right multiplication by $\tilde{\mathbf{W}}$ as

$$\boldsymbol{p}_1 = \tilde{\mathbf{W}}\boldsymbol{p}_0 = \sum_{i=1}^n \alpha_i \tilde{\mathbf{W}}\boldsymbol{D}^{1/2}\boldsymbol{\psi}_i = \sum_{i=1}^n \alpha_i \left(1 - \frac{\nu_i}{2}\right) \boldsymbol{D}^{1/2}\boldsymbol{\psi}_i.$$

And as promised, if we apply $\tilde{\mathbf{W}}$, the lazy random walk operator, t times, we now obtain

$$\boldsymbol{p}_t = \sum_{i=1}^n \alpha_i \left(1 - \frac{\nu_i}{2}\right)^t \boldsymbol{D}^{1/2}\boldsymbol{\psi}_i = \alpha_1 \boldsymbol{D}^{1/2}\boldsymbol{\psi}_1 + \sum_{i=2}^n \alpha_i \left(1 - \frac{\nu_i}{2}\right)^t \boldsymbol{D}^{1/2}\boldsymbol{\psi}_i. \tag{6.2}$$

where we use in the last equality that $\nu_1 = 0$. Using this simple characterization, we immediately get that $\boldsymbol{p}_t \to \boldsymbol{\pi}$ if $\nu_i > 0$ for all $i \geq 2$ (which is exactly when the graph is connected as you will prove in an exercise). To see this, observe that as t grows sum vanishes. We have that

$$\lim_{t\to\infty} \boldsymbol{p}_t = \alpha_1 \boldsymbol{D}^{1/2}\boldsymbol{\psi}_1 = \boldsymbol{\pi}.$$

where we use that $\alpha_1 = \boldsymbol{\psi}_1^\top D^{-1/2}\boldsymbol{p}_0 = \frac{\mathbf{1}^\top \boldsymbol{p}_0}{(\boldsymbol{d}^\top \mathbf{1})^{1/2}} = \frac{1}{(\boldsymbol{d}^\top \mathbf{1})^{1/2}}$ by the definition of α_1 (from 6.1), by plugging in the value of $\boldsymbol{\psi}_1$ and by recalling that $\boldsymbol{p}_0$ is a distribution and therefore $\mathbf{1}^\top \boldsymbol{p}_0 = 1$; and using that $\boldsymbol{D}^{1/2}\boldsymbol{\psi}_1 = \frac{\boldsymbol{d}}{(\mathbf{1}^\top \boldsymbol{d})^{1/2}}$.

Theorem 6.2.4. *For any connected graph G, we have that the lazy random walk converges to the stationary distribution of G.*

6.2.3 The Rate of Convergence

Let us now come to the main result that we want to prove in this lecture.

Theorem 6.2.5. *In any unweighted (a.k.a. unit weight) connected graph G, for any $\boldsymbol{p}_0$, at any time step t, we have for $\boldsymbol{p}_t = \tilde{\mathbf{W}}^t \boldsymbol{p}_0$ that*

$$\|\boldsymbol{p}_t - \boldsymbol{\pi}\|_\infty \leq e^{-\nu_2 \cdot t/2}\sqrt{n}$$

Instead of proving the theorem above, we prove the lemma below which gives point-wise convergence. This makes it more convenient to derive a proof and it is not hard to deduce the theorem above as a corollary.

Lemma 6.2.6. *In any weighted connected graph G, for all $a, b \in V$, and any time step t, we have for $\boldsymbol{p}_0 = \mathbf{1}_a$ and $\boldsymbol{p}_t = \tilde{\mathbf{W}}^t \boldsymbol{p}_0$ that*

$$|\boldsymbol{p}_t(b) - \boldsymbol{\pi}(b)| \leq e^{-\nu_2 \cdot t/2} \sqrt{\boldsymbol{d}_b / \boldsymbol{d}_a}$$

From Equation 6.2, we obtain that

$$\boldsymbol{p}_t(b) - \boldsymbol{\pi}(b) = \mathbf{1}_b^\top (\boldsymbol{p}_t - \boldsymbol{\pi}) = \mathbf{1}_b^\top \left(\sum_{i=2}^{n} \alpha_i \left(1 - \frac{\nu_i}{2}\right)^t \boldsymbol{D}^{1/2} \boldsymbol{\psi}_i \right) \tag{6.3}$$

$$= \sum_{i=2}^{n} \alpha_i \left(1 - \frac{\nu_i}{2}\right)^t \mathbf{1}_b^\top \boldsymbol{D}^{1/2} \boldsymbol{\psi}_i \leq \left(1 - \frac{\nu_2}{2}\right)^t \cdot \sum_{i=2}^{n} \alpha_i \mathbf{1}_b^\top \boldsymbol{D}^{1/2} \boldsymbol{\psi}_i \tag{6.4}$$

Taking the absolute value on both sides, we obtain that

$$|\boldsymbol{p}_t(b) - \boldsymbol{\pi}(b)| \leq \left(1 - \frac{\nu_2}{2}\right)^t \sum_{i=2}^{n} \left| \alpha_i \mathbf{1}_b^\top \boldsymbol{D}^{1/2} \boldsymbol{\psi}_i \right| \leq \left(1 - \frac{\nu_2}{2}\right)^t \sqrt{\left(\sum_{i=2}^{n} \alpha_i^2 \right) \left(\sum_{i=2}^{n} \beta_i^2 \right)}$$

where we use Cauchy-Schwarz in the last inequality, i.e. $|\langle \boldsymbol{u}, \boldsymbol{v} \rangle|^2 \leq \langle \boldsymbol{u}, \boldsymbol{u} \rangle \cdot \langle \boldsymbol{v}, \boldsymbol{v} \rangle$ and define $\beta_i = \mathbf{1}_b^\top \boldsymbol{D}^{1/2} \boldsymbol{\psi}_i$ for each i. Let us finally bound the two sums:

- We show that $\sum_{i=2}^{n} \alpha_i^2 \leq \|\boldsymbol{D}^{-1/2} \mathbf{1}_a\|_2^2 = 1/\boldsymbol{d}_a$. The last equality is trivial, therefore we only prove the inequality. To see this, observe that by 6.1, we have

  $$\|\boldsymbol{D}^{-1/2} \mathbf{1}_a\|_2^2 = (\boldsymbol{D}^{-1/2} \mathbf{1}_a)^\top (\boldsymbol{D}^{-1/2} \mathbf{1}_a) = \left(\sum_{i=1}^{n} \alpha_i \boldsymbol{\psi}_i^\top \right) \cdot \left(\sum_{i=1}^{n} \alpha_i \boldsymbol{\psi}_i \right).$$

  But since the vectors $\boldsymbol{\psi}_i$ form an orthonormal basis, we must further have

  $$\left(\sum_{i=1}^{n} \alpha_i \boldsymbol{\psi}_i^\top \right) \cdot \left(\sum_{i=1}^{n} \alpha_i \boldsymbol{\psi}_i \right) = \sum_{i=1}^{n} \alpha_i^2 \leq \sum_{i=2}^{n} \alpha_i^2.$$

- Finally, we need to show that $\sum_{i=2}^{n} \beta_i^2 \leq \|\boldsymbol{D}^{1/2} \mathbf{1}_b\|_2^2 = \boldsymbol{d}_b$. Here, we first observe that

  $$\boldsymbol{D}^{1/2} \mathbf{1}_b = \sum_{i=1}^{n} \beta_i \boldsymbol{\psi}_i \iff \boldsymbol{\psi}_i^\top \boldsymbol{D}^{1/2} \mathbf{1}_b = \beta_i \iff \mathbf{1}_b^\top \boldsymbol{D}^{1/2} \boldsymbol{\psi}_i = \beta_i$$

  and then, we can again exploit that the vectors $\boldsymbol{\psi}_i$ form an orthonormal basis and therefore

  $$\|\boldsymbol{D}^{1/2} \mathbf{1}_b\|_2^2 = (\boldsymbol{D}^{1/2} \mathbf{1}_b)^\top (\boldsymbol{D}^{1/2} \mathbf{1}_b) = \left(\sum_{i=1}^{n} \beta_i \boldsymbol{\psi}_i^\top \right) \left(\sum_{i=1}^{n} \beta_i \boldsymbol{\psi}_i \right) \leq \sum_{i=2}^{n} \beta_i^2.$$

Putting everything together (and using $1 + x \leq e^x, \forall x \in \mathbb{R}$), we obtain

$$|\boldsymbol{p}_t(b) - \boldsymbol{\pi}(b)| \leq \left(1 - \frac{\nu_2}{2}\right)^t \sqrt{\boldsymbol{d}_b / \boldsymbol{d}_a} \leq e^{-\nu_2 \cdot t/2} \sqrt{\boldsymbol{d}_b / \boldsymbol{d}_a}$$

6.3 Properties of Random Walks

We now shift our focus away from convergence of random walks and consider some interesting properties of random walks[1]. Here, we are no longer interested in lazy random walks, although all proofs can be straight-forwardly adapted. While in the previous section, we relied on computing the second eigenvalue of the Normalized Laplacian efficiently, here, we will discover that solving Laplacian systems, that is finding an $\boldsymbol{x}$ such that $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{b}$ can solve a host of problems in random walks.

6.3.1 Hitting Times

One of the most natural questions one can ask about a random walk starting in a vertex a (i.e. $\boldsymbol{p}_0 = \mathbf{1}_a$) is how many steps it takes to get to a special vertex s. This quantity is called the *hitting time* from a to s and we denote it by $H_{a,s} = \inf\{t \mid \boldsymbol{v}_t = s\}$. For the rest of this section, we are concerned with computing the expected hitting time, i.e. $\mathbb{E}[H_{a,s}]$.

It turns out, that it is more convenient to compute *all* expected hitting times $H_{a,s}$ for vertices $a \in V$ to a fixed s. We denote by $\boldsymbol{h} \in \mathbf{R}^V$, the vector with $\boldsymbol{h}(a) = \mathbb{E}[H_{a,s}]$. We now show that we can compute $\boldsymbol{h}$ by solving a Laplacian system $\boldsymbol{L}\boldsymbol{h} = \boldsymbol{b}$. We will see later in the course that such systems (spoiler alert!) can be solved in time $\tilde{O}(m)$, so this will imply a near-linear time algorithm to compute the hitting times.

Hitting Time and the Random Walk Matrix. Let us first observe that if $s = a$, then the answer becomes trivially 0, i.e. $\boldsymbol{h}(s) = 0$.

We compute the rest of the vector by writing down a system of equations that recursively characterizes $\boldsymbol{h}$. Observe therefore first that for any $a \neq s$, we have that the random walks starting at a will next visit a neighbor b of a. If the selected neighbor $b = s$, the random walks stops; otherwise, the random walks needs in expectation $\mathbb{E}[H_{b,s}]$ time to move to s.

We can express this algebraically by

$$\boldsymbol{h}_a = 1 + \sum_{a \sim b} \mathbb{P}[v_{t+1} = b \mid v_t = a] \cdot \boldsymbol{h}(b) = 1 + \sum_{a \sim b} \frac{\boldsymbol{w}(a,b)}{\boldsymbol{d}(a)} \cdot \boldsymbol{h}(b) = 1 + (\boldsymbol{W}\mathbf{1}_a)^\top \boldsymbol{h} = 1 + \mathbf{1}_a^\top \boldsymbol{W}^\top \boldsymbol{h}.$$

Using that $\boldsymbol{h}(a) = \mathbf{1}_a^\top \boldsymbol{h} = \mathbf{1}_a^\top \boldsymbol{I}\boldsymbol{h}$, we can rewrite this as

$$1 = \mathbf{1}_a^\top (\boldsymbol{I} - \boldsymbol{W}^\top)\boldsymbol{h}.$$

This gives a system of (linear) equations, that can be neatly summarized by

$$\mathbf{1} - \alpha \cdot \mathbf{1}_s = (\boldsymbol{I} - \boldsymbol{W}^\top)\boldsymbol{h}$$

[1]Note that this part of the script is in large part inspired by Aaron Sidford's lecture notes on the same subject, in his really interesting course Discrete Mathematics and Algorithms.

where we have an extra degree of freedom in choosing α in formulating a constraint $1-\alpha = \mathbf{1}_s^\top(\boldsymbol{I}-\boldsymbol{W}^\top)\boldsymbol{h}$. This extra degree of freedom stems from the fact that $n-1$ equations suffice for us to enforce that the returned vector $\boldsymbol{x}$ from the system is indeed the hitting times (possibly shifted by the value assigned to coordinate t).

Finding Hitting Times via Laplacian System Solve. Since we assume G connected, we have that multiplying with $\boldsymbol{D} = \boldsymbol{D}^\top$ preserves equality. Further since $\boldsymbol{W} = \boldsymbol{A}\boldsymbol{D}^{-1}$, we obtain

$$\boldsymbol{d} - \alpha \cdot \boldsymbol{d}(s) \cdot \mathbf{1}_s = (\boldsymbol{D}-\boldsymbol{A})\boldsymbol{h}.$$

Defining $\boldsymbol{b} = \boldsymbol{d} - \alpha \cdot \boldsymbol{d}(s) \cdot \mathbf{1}_s$, and observing $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, we have $\boldsymbol{L}\boldsymbol{h} = \boldsymbol{b}$.

Finally, we observe that we only have a solution to the above system if and only if $\boldsymbol{b} \in \ker(\boldsymbol{L})^\perp = \operatorname{span}(\mathbf{1})^\perp$. We thus have to set α such that

$$\mathbf{1}^\top(\boldsymbol{d} - \alpha \cdot \boldsymbol{d}(s) \cdot \mathbf{1}_s) = \|\boldsymbol{d}\|_1 - \alpha \cdot \boldsymbol{d}(s) \iff \alpha = \|\boldsymbol{d}\|_1 / \boldsymbol{d}(s).$$

We have now formalized $\boldsymbol{L}$ and $\boldsymbol{b}$ completely. A last detail that we should not forget about is that any solution $\boldsymbol{x}$ to such system $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{b}$ is not necessarily equal $\boldsymbol{h}$ but has the property that it is shifted from $\boldsymbol{h}$ by the all-ones vector. Since we require $\boldsymbol{h}(s) = 0$, we can reconstruct $\boldsymbol{h}$ from $\boldsymbol{x}$ straight-forwardly by subtracting $\boldsymbol{x}(s)\mathbf{1}$.

Theorem 6.3.1. *Given a connected graph G, a special vertex $s \in V$. Then, we can formalize a Laplacian system $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{b}$ (where $\boldsymbol{L}$ is the Laplacian of G) such that the expected hitting times to s are given by $\boldsymbol{h} = \boldsymbol{x} - \boldsymbol{x}(s)\mathbf{1}$. We can reconstruct $\boldsymbol{h}$ from $\boldsymbol{x}$ in time $O(n)$.*

Hitting Times and Electrical Networks. Seeing that hitting times can be computed by formulating a Laplacian system $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{b}$. You might remember that in the first lecture, we argued that a system $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{b}$ also solves the problem of routing a demand $\boldsymbol{b}$ via an electrical flow with voltages $\boldsymbol{x}$.

Indeed, we can interpret computing expected hitting times $\boldsymbol{h}$ to a special vertex s as the problem of computing the electrical voltages $\boldsymbol{x}$ where we insert (or more technically correct apply) $\boldsymbol{d}(a)$ units of current at every vertex $a \neq s$ and where we remove $\mathbf{1}^\top\boldsymbol{d} - \boldsymbol{d}(s)$ units of current at the vertex s. Then, we can express expected hitting time to some vertex a as the voltage difference to s: $\mathbb{E}[H_{a,s}] = \boldsymbol{h}(a) = \boldsymbol{x}(a) - \boldsymbol{x}(s)$.

6.3.2 Commute Time

A topic very related to hitting times are *commute times*. That is for two vertices a, b, the commute time is the time in a random walk starting in a to visit b and then to return to a again. Thus, it can be defined $C_{a,b} = H_{a,b} + H_{b,a}$.

Commute Times via Electric Flows. Recall that expected hitting times have an electric flow interpretation.

Now, let us denote by $\boldsymbol{x}$ a solution to the Laplacian system $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{b}_b$ where the demand is $\boldsymbol{b}_b = \boldsymbol{d} - \boldsymbol{d}^\top \mathbf{1} \cdot \mathbf{1}_b \in \ker(\mathbf{1})^\perp$. Recall that we have $\mathbb{E}[H_{z,b}] = \boldsymbol{x}(z) - \boldsymbol{x}(b)$ for all z. Similarly, we can compute voltages $\boldsymbol{y}$ to the Laplacian system $\boldsymbol{L}\boldsymbol{y} = \boldsymbol{b}_a$ where $\boldsymbol{b}_a = \boldsymbol{d} - \boldsymbol{d}^\top \mathbf{1} \cdot \mathbf{1}_a \in \ker(\mathbf{1})^\perp$. Again, $\mathbb{E}[H_{z,a}] = \boldsymbol{y}(z) - \boldsymbol{y}(a)$ for all z.

But observe that this allows us to argue by linearity that $\mathbb{E}[C_{a,b}] = \mathbb{E}[H_{a,b} + H_{b,a}] = \mathbb{E}[H_{a,b}] + \mathbb{E}[H_{b,a}] = \boldsymbol{x}(a) - \boldsymbol{x}(b) + \boldsymbol{y}(b) - \boldsymbol{y}(a) = (\mathbf{1}_a - \mathbf{1}_b)^\top (\boldsymbol{x} - \boldsymbol{y})$. But using linearity again, we can also argue at the same time that we obtain the vector $\boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{y})$ as a solution to the problem $\boldsymbol{L}\boldsymbol{z} = \boldsymbol{b}_b - \boldsymbol{b}_a = \boldsymbol{d}^\top \mathbf{1}(\mathbf{1}_a - \mathbf{1}_b)$. That is the flow that routes $\|\boldsymbol{d}\|_1$ units of flow from b to a.

Theorem 6.3.2. *Given a graph $G = (V, E)$, for any two fixed vertices $a, b \in V$, the expected commute time $C_{a,b}$ is given by the voltage difference between a and b for any solution $\boldsymbol{z}$ to the Laplacian system $\boldsymbol{L}\boldsymbol{z} = \|\boldsymbol{d}\|_1 \cdot (\mathbf{1}_b - \mathbf{1}_a)$.*

We note that the voltage difference between a and b in an electrical flow routing demand $\mathbf{1}_b - \mathbf{1}_a$ is also called the *effective resistance* $R_{\text{eff}}(a, b)$. This quantity will play a crucial role in the next roles. In the next lecture, we introduce $R_{\text{eff}}(a, b)$ slightly differently as the energy required by the electrical flow that routes $\mathbf{1}_b - \mathbf{1}_a$, however, it is not hard to show that these two definitions are equivalent.

Our theorem can now be restated as saying that the expected commute time $\mathbb{E}[C_{a,b}] = \|\boldsymbol{d}\|_1 \cdot R_{\text{eff}}(a, b)$. This is a classic result.