An Algorithmist's Toolkit Notes

Instructor: Jonathan Kelner
Notes by: Lichen Zhang
Massachusetts Institute of Technology
18.408 An Algorithmist's Toolkit

Spring 2024

Contents

Overview **3**

1 Spectral Graph Theory **5**
1.1 The Graph Laplacian . . . 5
1.2 Spectra of Common Graphs and Spectral Drawings . . . 13
1.3 Eigenvalue Bounds and the Courant–Fischer Theorem . . . 21
1.4 Graph Cuts and Cheeger's Inequality . . . 25
1.5 Random Walks and Mixing . . . 35
1.6 Approximate Counting . . . 41
1.7 Diameter, Eigenvalues, and Chebyshev Polynomials . . . 52
1.8 Expanders . . . 58
1.9 Spectral Sparsification . . . 62
1.10 Electrical Networks and Effective Resistance . . . 72
1.11 Spanning Trees and the Matrix-Tree Theorem . . . 78

2 Convex Geometry **83**
2.1 Convex Bodies and High-Dimensional Volume . . . 83
2.2 Polarity and Norms . . . 86
2.3 John's Theorem . . . 90
2.4 The Brunn–Minkowski Inequality and Its Applications . . . 97
2.5 Approximating the Volume of a Convex Body . . . 105
2.6 Concentration of Measure . . . 113
2.7 Metric Embeddings and the Johnson–Lindenstrauss Theorem . . . 118

3 Iterative Methods and Optimization **122**
3.1 Iterative Methods for Linear Systems . . . 122
3.2 The Conjugate Gradient Method . . . 129
3.3 Preconditioning . . . 133
3.4 Gradient Descent . . . 135
3.5 Subgradient Descent . . . 139
3.6 Descent in General Norms . . . 142

3.7 Online Linear Optimization and Follow the Regularized Leader 145
3.8 The Accelerated Gradient Method 150

References **157**

Overview

These are notes for MIT 18.408, *An Algorithmist's Toolkit*, taught by Jonathan Kelner in spring 2024. This overview says what the course is for, what it assumes, and which topics it covers.

Goals and prerequisites

Goals. The course has two goals. The first is to learn a collection of powerful, interrelated mathematical techniques for algorithm design. The topics are chosen to be both mathematically interesting and practically useful. There will sometimes be long mathematical interludes, but always with proportional algorithmic payoffs. The second goal is to be able to apply these tools directly to your own research, whether that work is theoretical or applied; open research questions will be pointed out whenever possible.

Prerequisites. This is a graduate course, so a significant level of mathematical maturity is assumed, but there are few formal prerequisites: multivariate calculus, linear algebra, basic algorithms and basic probability. Some algorithms beyond an introductory course would help but are not strictly necessary.

The main topics

The main topics are:

(1) spectral graph theory;

(2) convex geometry;

(3) iterative methods for optimization and linear algebra;

(4) graphs, linear systems, and electrical networks;

(5) graph decomposition, approximation, and embedding;

(6) lattices and basis reduction.

For each topic we list the mathematical theory to be developed, the other fields it connects to, and its applications.

Spectral graph theory.
Theory: graph Laplacians and their eigenvalues; isoperimetric and Cheeger inequalities; recent advances in algorithmic directed spectral graph theory. *Connections:* Markov processes and

random walks; continuous differential and convex geometry; expanders and random graphs. *Applications:* graph cutting, clustering, approximate counting, disjoint path problems, routing, graph drawing, and possibly coding theory and how Google works.

Convex geometry.
Theory: geometric properties of high-dimensional convex bodies; Fritz John's theorem and isotropy; Brunn–Minkowski and isoperimetric inequalities; Dvoretzky's theorem; concentration of measure; connections to probability theory. *Connections:* probability theory and large deviations; spectral graph theory. *Applications:* volume computation; convex programming.

Iterative methods for linear algebra and optimization.
Theory: using geometric information to solve linear systems, eigenvalue problems and convex programs quickly; basic iterative methods; the Lanczos algorithm, conjugate gradients, coordinate descent and the Kaczmarz method, stochastic descent, Nesterov's accelerated gradient method; preconditioning; speeding up iterative methods with spectral graph theory. *Connections:* convex geometry. *Applications:* fast graph algorithms, numerical linear algebra, machine learning, online algorithms, complexity theory.

Graphs, linear systems, and electrical networks.
Theory: linear systems in graph Laplacians; their interpretations in terms of discrete harmonic functions, Green's functions, electrical circuits, and random processes. *Connections:* spectral graph theory; electrical circuits; probability theory. *Applications:* analysis of random processes; random spanning trees.

Graph decomposition, approximation, and embedding.
Theory: techniques for relating the properties of a graph to those of other (ideally smaller or simpler) graphs, or of distributions over smaller or simpler graphs. *Connections:* spectral graph theory; the theory of metric spaces. *Applications:* max flow and min s-t cut; approximation algorithms for NP-hard graph problems; oblivious routing.

Lattices and basis reduction.
Theory: basic properties of lattices; Minkowski's theorem; the LLL algorithm for lattice basis reduction. *Connections:* computational algebra; convex geometry. *Applications:* solving low-dimensional integer programs; solving some NP-hard problems in practice; breaking cryptosystems.

These notes are organized in three chapters, on spectral graph theory (Chapter 1), convex geometry (Chapter 2), and iterative methods and optimization (Chapter 3); electrical networks appear within Chapter 1.

Acknowledgments

These notes are based on Jonathan Kelner's lecture slides from the spring 2022 offering of the course and, for the first five lectures (Sections 1.1 to 1.4 and the start of Section 1.5), on scribe notes from earlier offerings of the course by Tim Abbott, Jesse Geneson, Zeyuan Zhu and David Rolnick; by M. Baym, M. Nachin, T. Tulabandhula and J. Peebles; by Andre Wibisono, Yufei Zhao, Linus Hamilton and S. Silwal; and by A. Schvartzman, Y. Cai, D. Iancu and A. Sidiropoulos. The figures of Section 3.1 and Fig. 3.9 are redrawn after J. R. Shewchuk, "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain" [She94].

Chapter 1

Spectral Graph Theory

The recipe of spectral graph theory is short:

(1) associate a matrix to a graph;

(2) diagonalize the matrix (or do other linear algebra with it);

(3) hope that something useful happens.

The surprise, which this chapter makes precise, is how often step (3) succeeds. We start with the graph Laplacian and the spectra of some common graphs, and develop the main tool for bounding eigenvalues without computing them. We then relate the second eigenvalue to cutting the graph (Cheeger's inequality) and to how fast a random walk on it mixes, and use random walks to count approximately. The rest of the chapter applies these ideas to diameters, expanders, sparsification, electrical networks and spanning trees.

1.1 The Graph Laplacian

This section introduces the matrix we will use most, the *Laplacian* $L_G = D_G - A_G$. We see how it acts as a linear map, write it as a sum of one small matrix per edge, and compute its quadratic form $x^\top L_G x = \sum_{(i,j)\in E} w_{ij}(x_i - x_j)^2$, which is the formula behind almost everything in this chapter. From the quadratic form we derive the basic structural facts: the Laplacian of a disconnected graph splits into blocks, it is positive semidefinite, and the multiplicity of its eigenvalue 0 counts the connected components. Finally we write $L = B^\top B$ for an explicit $m \times n$ matrix B, the incidence matrix, and use this factorization to explain the name: B is a discrete gradient and L is a discrete (negative) second derivative, just like the Laplace operator of calculus.

1.1.1 Reminders about eigenvalues

We assume you have seen eigenvalues before; here are the facts we will use.

Definition 1.1.1. Let M be an $n \times n$ matrix. Suppose that

$$Mx = \lambda x, \qquad x \in \mathbb{R}^n,\ x \neq \mathbf{0}, \qquad \lambda \in \mathbb{R}.$$

We call x an *eigenvector* of M and λ its *eigenvalue*.

Everything good happens for symmetric matrices.

Fact 1.1.2. *Let M be a symmetric $n \times n$ real matrix.*

(a) If v and w are eigenvectors of M with different *eigenvalues, then $v^\top w = 0$.*

(b) If v and w have the same *eigenvalue, then any $q = av + bw$ is an eigenvector too (when it is nonzero). The span of the eigenvectors with a given eigenvalue is called an* eigenspace*.*

(c) M has a full orthonormal basis of eigenvectors $v_1, \ldots, v_n$.

(d) All eigenvalues and eigenvectors are real.

(e) M can be diagonalized*:*

$$M = V \Lambda V^\top = \sum_{i=1}^{n} \lambda_i v_i v_i^\top, \tag{1.1.1}$$

where V is orthogonal ($VV^\top = I$) with columns the eigenvectors $v_1, \ldots, v_n$, and Λ is diagonal with entries the eigenvalues $\lambda_1, \ldots, \lambda_n$.

All bets are off if M is not symmetric: none of these statements need hold. Fortunately, the graph matrices below are symmetric.

1.1.2 Matrices for graphs

We will look at several matrices associated with a graph: the adjacency matrix, the random walk matrix, the Laplacian matrix, and the normalized Laplacian matrix. Different ones are useful at different times. This section introduces the first and third; the random walk matrix appears in Section 1.5.1 and the normalized Laplacian in Section 1.4.8.

Conventions. From now on, unless stated otherwise, $G = (V, E)$ is a graph that is undirected, unweighted, and has no multiple edges or self loops. We write $n = |V|$ for the number of vertices and $m = |E|$ for the number of edges, and we usually take $V = [n]$.

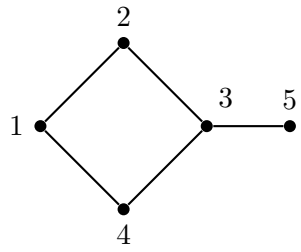
The adjacency matrix

Definition 1.1.3. The *adjacency matrix* $A = A_G$ of G is the $n \times n$ matrix with

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

If G is weighted and the edge (i, j) has weight w_{ij}, then $A_{ij} = w_{ij}$ if $(i, j) \in E$ and $A_{ij} = 0$ otherwise.

Since G is undirected, A is clearly symmetric. Figure 1.1 shows a small example, which we will reuse.



$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Figure 1.1: A graph on five vertices (a 4-cycle $1, 2, 3, 4$ with an extra vertex 5 attached to 3) and its adjacency matrix. Rows and columns are indexed by the vertices in the order $1, \ldots, 5$.

The Laplacian

For our purposes, a slightly different matrix will often be nicer.

Definition 1.1.4. The *Laplacian matrix* $L = L_G$ of G is the $n \times n$ matrix with

$$L_{ij} = \begin{cases} -1 & \text{if } (i,j) \in E, \\ d_i & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases}$$

where d_i is the degree of the ith vertex. If G is weighted,

$$L_{ij} = \begin{cases} -w_{ij} & \text{if } (i,j) \in E, \\ d_i & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases}$$

where now $d_i = \sum_{j:(i,j)\in E} w_{ij}$ is the weighted degree of the ith vertex.

The Laplacian is also clearly symmetric. Equivalently, let $D = D_G$ be the diagonal matrix whose ith diagonal entry is the degree of vertex i. Then

$$L_G = D_G - A_G. \tag{1.1.2}$$

Example 1.1.5. For the graph of Fig. 1.1,

$$D = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \qquad L = D - A = \begin{pmatrix} 2 & -1 & 0 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ -1 & 0 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{pmatrix}.$$

Each row of L sums to zero: the diagonal entry is the number of -1's in its row.

1.1.3 What does the Laplacian do to a vector?

Think of a vector $x \in \mathbb{R}^n$ as a function $x : V \to \mathbb{R}$ that assigns the number $x(i)$ to vertex i. Then the ith entry of $L_G x$ is

$$\begin{aligned}[L_G x]_i &= d_i\, x(i) - \sum_{j:(i,j)\in E} x(j) \\ &= d_i \left(x(i) - \frac{\sum_{j:(i,j)\in E} x(j)}{d_i} \right) \\ &= d_i(x(i) - \text{average of } x \text{ on the neighbors of } i). \end{aligned} \tag{1.1.3}$$

If G is weighted, the same computation holds with the weighted average $\sum_j w_{ij}x(j)/d_i$. So L_G measures, at each vertex, how far the value there is from the average of its neighbors, scaled by the degree.

Example 1.1.6. For the path $1 - 2 - 3$,

$$L_G x = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} x(1) \\ x(2) \\ x(3) \end{pmatrix} = \begin{pmatrix} x(1) - x(2) \\ 2\big(x(2) - \frac{x(1)+x(3)}{2}\big) \\ x(3) - x(2) \end{pmatrix}.$$

Formula (1.1.3) implies that for any graph G,

$$L_G \mathbf{1} = \mathbf{0}, \tag{1.1.4}$$

since a constant function equals the average of its neighbors everywhere. So $x = \mathbf{1}$ is an eigenvector of L_G with eigenvalue 0.

We will see in Corollaries 1.1.12 and 1.1.14 that all the other eigenvalues of L_G are ≥ 0, and > 0 when G is connected. So if we order the eigenvalues as $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, with respective eigenvectors $v_1, \ldots, v_n$, then

$$v_1 = \mathbf{1}, \qquad \lambda_1 = 0.$$

(Here v_1 is not normalized; the unit eigenvector is $\mathbf{1}/\sqrt{n}$.)

1.1.4 The Laplacian as a sum of edge Laplacians

If G and H are two graphs on the same vertex set, let $G \cup H$ be the graph with the edges of both, where the union adds weights (an edge present in both gets the sum of its two weights). Directly from the definition,

$$L_{G\cup H} = L_G + L_H \qquad \text{(additivity)}. \tag{1.1.5}$$

Let L_e be the Laplacian of the graph on n vertices with just the single (unweighted) edge $e = (i, j)$:

$$L_e = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \oplus [\text{a bunch of zeros}],$$

meaning that L_e has the 2×2 block shown in rows and columns i and j, and zeros everywhere else. By additivity,

$$L_G = \sum_{e\in E} w_e L_e. \tag{1.1.6}$$

So we can prove many facts about the Laplacian by proving them for one edge and adding them up. This is the first example of a general technique we will use later: bounding Laplacians by adding up the matrices of substructures.

1.1.5 The Laplacian as a quadratic form

We have seen how the Laplacian works as a linear map. Now let us see how it works when used as a quadratic form, $x \mapsto x^\top L x$. The 2×2 block of an edge Laplacian factors as

$$L_e = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \oplus [\text{a bunch of zeros}] = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 & -1 \end{pmatrix} \oplus [\text{a bunch of zeros}].$$

So if $e = (i, j)$,

$$x^\top L_e x = \begin{pmatrix} x_i & x_j \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} x_i \\ x_j \end{pmatrix} = (x_i - x_j)^2. \tag{1.1.7}$$

Since $L_G = \sum_{e \in E} w_e L_e$, summing (1.1.7) over the edges gives the most important formula of this chapter:

$$x^\top L_G x = x^\top \Big(\sum_{e \in E} w_e L_e \Big) x = \sum_{(i,j) \in E} w_{ij} (x_i - x_j)^2. \tag{1.1.8}$$

This formula has a physical interpretation. Place the vertices of G on the real line, with vertex i at the point x_i, and let the edges be springs. Then $\sum_{(i,j) \in E} (x_i - x_j)^2$ represents the energy stored in the springs by their tension. This also gives an intuitive reason why the spectral drawings of Section 1.2.2 look so tidy: for an eigenvector with a small eigenvalue, such as v_2 or v_3, the value at each vertex tends to be close to the average of its neighbors (by (1.1.3)), and plotting (v_2, v_3) gives a two-dimensional drawing of minimal spring energy, subject to certain constraints. We make the constraints precise with the Courant–Fischer theorem in Corollary 1.3.4: v_2 minimizes the energy among unit vectors orthogonal to $\mathbf{1}$.

1.1.6 The Laplacian of a disjoint union

We now derive the basic structural facts about the Laplacian from the formulas above, starting with graphs that fall into pieces.

Proposition 1.1.7. *If G and H are supported on disjoint sets of vertices, then the Laplacian of their disjoint union $G \sqcup H$ is the direct sum*

$$L_{G \sqcup H} = L_G \oplus L_H = \begin{pmatrix} L_G & 0 \\ 0 & L_H \end{pmatrix},$$

where the vertices of G are listed first.

Proof. An isolated vertex has a zero row and column in the Laplacian. Let G' be G together with the vertices of H added as isolated vertices, and let H' be H together with the vertices of G added as isolated vertices. Then $L_{G'} = \left(\begin{smallmatrix} L_G & 0 \\ 0 & 0 \end{smallmatrix}\right)$ and $L_{H'} = \left(\begin{smallmatrix} 0 & 0 \\ 0 & L_H \end{smallmatrix}\right)$. Since $G \sqcup H = G' \cup H'$, additivity gives $L_{G \sqcup H} = L_{G'} + L_{H'} = L_G \oplus L_H$. □

So the Laplacian of any graph is the direct sum of the Laplacians of its connected components, and its spectrum is the union of theirs.

Proposition 1.1.8. *Suppose G has n vertices and L_G has eigenvectors $v_1, \dots, v_n$ with eigenvalues $\lambda_1, \dots, \lambda_n$, and H has n' vertices and L_H has eigenvectors $v'_1, \dots, v'_{n'}$ with eigenvalues $\lambda'_1, \dots, \lambda'_{n'}$. Then $L_{G \sqcup H}$ has eigenvectors*

$$v_1 \oplus \mathbf{0}, \ \dots, \ v_n \oplus \mathbf{0}, \qquad \mathbf{0} \oplus v'_1, \ \dots, \ \mathbf{0} \oplus v'_{n'}$$

with corresponding eigenvalues $\lambda_1, \dots, \lambda_n, \lambda'_1, \dots, \lambda'_{n'}$. Here $v \oplus \mathbf{0}$ denotes the vector v followed by zeros, $\binom{v}{\mathbf{0}}$.

Proof. By Proposition 1.1.7,

$$L_{G \sqcup H}(v_i \oplus \mathbf{0}) = \begin{pmatrix} L_G & 0 \\ 0 & L_H \end{pmatrix} \begin{pmatrix} v_i \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \lambda_i v_i \\ \mathbf{0} \end{pmatrix} = \lambda_i (v_i \oplus \mathbf{0}),$$

and in the same way $L_{G \sqcup H}(\mathbf{0} \oplus v'_j) = \lambda'_j (\mathbf{0} \oplus v'_j)$. □

1.1.7 Positive semidefiniteness

Definition 1.1.9. A symmetric matrix M is *positive semidefinite* (PSD) if

$$x^\top M x \geq 0 \qquad \text{for all } x \in \mathbb{R}^n.$$

It is *positive definite* (PD) if the inequality is strict for all $x \neq \mathbf{0}$.

Fact 1.1.10. *A symmetric matrix M is PSD if and only if all its eigenvalues are ≥ 0, and PD if and only if all its eigenvalues are > 0.*

Proof. Diagonalize $M = V \Lambda V^\top$ as in Fact 1.1.2. With $y = V^\top x$, which ranges over all of $\mathbb{R}^n$ as x does, we get $x^\top M x = y^\top \Lambda y = \sum_i \lambda_i y_i^2$. This is ≥ 0 for all y if and only if every $\lambda_i \geq 0$, and > 0 for all $y \neq \mathbf{0}$ if and only if every $\lambda_i > 0$. □

Fact 1.1.11. *A symmetric matrix M is PSD if and only if it can be written as $M = X^\top X$. Here X can be $k \times n$ for any k; it need not be square. Such an X always exists for PSD M, by diagonalizing M, but it is not unique.*

Proof. If $M = X^\top X$, then $x^\top M x = (Xx)^\top (Xx) = \|Xx\|^2 \geq 0$. Conversely, if M is PSD, diagonalize $M = V \Lambda V^\top$; by Fact 1.1.10 the diagonal matrix Λ has nonnegative entries, so it has a square root $\Lambda^{1/2}$ (entries $\sqrt{\lambda_i}$), and $M = (\Lambda^{1/2} V^\top)^\top (\Lambda^{1/2} V^\top)$. For non-uniqueness: if U is any orthogonal matrix, then $(UX)^\top (UX) = X^\top X$; also one can append rows of zeros to X. □

By (1.1.8), $x^\top L_G x$ is a sum of nonnegative terms. So:

Corollary 1.1.12. *The Laplacian L_G of any graph (with nonnegative weights) is positive semidefinite. Hence all its eigenvalues are nonnegative: $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.*

Corollary 1.1.19 gives a second proof, by writing L_G explicitly as $X^\top X$ for a natural matrix X.

1.1.8 The dimension of the null space

Claim 1.1.13. *If G is connected, the null space of L_G is one-dimensional and spanned by the vector* $\mathbf{1}$.

Proof. Let $x \in \ker(L)$, so $Lx = \mathbf{0}$. Then

$$x^\top L x = \sum_{(i,j)\in E} (x_i - x_j)^2 = 0.$$

Each term is nonnegative, so $x_i = x_j$ for every $(i,j) \in E$. Since G is connected, any two vertices are joined by a path, along which x is constant; so all the x_i are equal, and x is a multiple of $\mathbf{1}$. □

Corollary 1.1.14. *For connected G, $\lambda_2 > 0$.*

All eigenvalues are ≥ 0 by Corollary 1.1.12, and the eigenvalue 0 has multiplicity one by Claim 1.1.13. So the second smallest eigenvalue is positive.

Corollary 1.1.15. $\dim\ker(L_G)$ *equals the number of connected components of G.*

By Proposition 1.1.7, L_G is the direct sum of the Laplacians of the components, so its null space is the direct sum of theirs. By Claim 1.1.13 each of these is one-dimensional, spanned by the indicator vector of the component.

Remark 1.1.16 (Caveat on conventions). We use the ordinary Laplacian, whereas Fan Chung's book on spectral graph theory [Chu97] uses the normalized Laplacian (Definition 1.4.7). She also numbers the eigenvalues starting from λ_0, while we start from λ_1. This makes some equations look slightly different, but all the ideas are unchanged.

1.1.9 Factoring the Laplacian: the incidence matrix

We know from Fact 1.1.11 that, being positive semidefinite, L can be factored as $L = X^\top X$ using its eigenvectors. But there is a nicer factorization. Let χ_i be the ith unit basis vector, and recall from Section 1.1.5 that the edge Laplacian of $e = (i,j)$ is $L_e = (\chi_i - \chi_j)(\chi_i - \chi_j)^\top$, so that (for an unweighted graph)

$$L = \sum_{e\in E} L_e = \sum_{(i,j)\in E} (\chi_i - \chi_j)(\chi_i - \chi_j)^\top. \tag{1.1.9}$$

A sum of outer products $\sum_e b_e b_e^\top$ is the product $B^\top B$ of the matrix B whose rows are the $b_e^\top$ with its transpose. So we can write $L = B^\top B$, where $B^\top$ has the column $\chi_i - \chi_j$ for each $(i,j) \in E$.

Definition 1.1.17. The (signed edge-vertex) *incidence matrix* $B = B_G$ of G is the $m \times n$ matrix, with rows indexed by edges and columns by vertices, given by

$$B_{e,v} = \begin{cases} 1 & \text{if } e = (v,w) \text{ and } v < w, \\ -1 & \text{if } e = (v,w) \text{ and } v > w, \\ 0 & \text{otherwise.} \end{cases}$$

So the row of edge $e = (u,v)$ with $u < v$ is $b_e^\top$, where $b_e = \chi_u - \chi_v$.

The incidence matrix is also written $\nabla = \nabla_G$, because, as we will see in Section 1.1.10, B acts as a discrete gradient. The choice of which endpoint gets the $+1$ is an arbitrary but fixed orientation of each edge. Any choice gives the same $B^\top B$, since $b_e b_e^\top = (-b_e)(-b_e)^\top$.

Proposition 1.1.18. *For every graph G, $L_G = B_G^\top B_G$.*

Proof. $B^\top B = \sum_{e \in E} b_e b_e^\top$, since the rows of B are the vectors $b_e^\top$. By (1.1.9) this is L_G. □

Corollary 1.1.19. *For every $x \in \mathbb{R}^n$,*

$$x^\top L_G x = \|B_G x\|^2 = \sum_{(i,j) \in E} (x_i - x_j)^2,$$

since the entry of $B_G x$ for the edge (i,j) is $\pm(x_i - x_j)$. This gives another proof that L_G is positive semidefinite.

Example 1.1.20. For the graph of Fig. 1.1, with its edges named as in Fig. 1.2, the matrices are

$$L = \begin{pmatrix} 2 & -1 & 0 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ -1 & 0 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{pmatrix}, \qquad B = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{pmatrix} \end{array},$$

and $B^\top$ is the transpose of B. For instance, edge $b = (1,4)$ has the row $\chi_1^\top - \chi_4^\top$ in B, and the column $\chi_1 - \chi_4$ in $B^\top$. One can check entry by entry that $B^\top B = L$: the diagonal entry $(B^\top B)_{vv}$ counts the edges at v, and an off-diagonal entry $(B^\top B)_{uv}$ is -1 exactly when (u,v) is an edge.

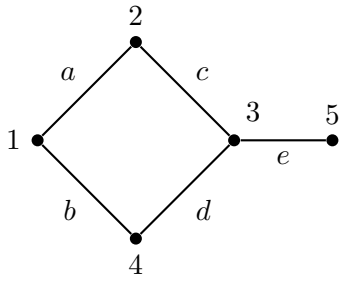


Figure 1.2: The example graph of Fig. 1.1 with named edges: $a = (1,2)$, $b = (1,4)$, $c = (2,3)$, $d = (3,4)$, $e = (3,5)$.

1.1.10 Why is this called the Laplacian?

The continuous Laplacian. In one dimension, the continuous Laplacian is just the second derivative $\frac{d^2}{dx^2}$. In two dimensions it is the operator

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$$

It is the divergence of the gradient, $\Delta f = \nabla \cdot \nabla f$. We now discretize these derivatives and find the graph Laplacian.

Discretizing derivatives: the 1-d case. Suppose $f : \mathbb{R} \to \mathbb{R}$ is a function of one variable, and sample it at the integers, which we think of as the vertices of a (long) path graph P_n (Fig. 1.3). Up to scaling, the first derivative $\frac{df}{dx}$ is approximated by the differences $f(k+1) - f(k)$, one for each edge of the path. These are the entries of $B_{P_n} f$, up to a sign that depends on the orientation of the edges; so B is a discrete gradient. Taking differences once more, the second derivative $\frac{d^2 f}{dx^2}$ at the vertex k is approximated, up to scaling, by

$$\big(f(k+1) - f(k)\big) - \big(f(k) - f(k-1)\big) = f(k+1) + f(k-1) - 2f(k) = -\big[L_{P_n} f\big](k).$$

So the Laplacian of the path is (minus) the discrete second derivative. In the same way, $L = B^\top B$ is (minus) the discrete divergence of the discrete gradient, just like $\Delta = \nabla \cdot \nabla$. The sign is why L is positive semidefinite while $\frac{d^2}{dx^2}$ is a negative operator. For nice f vanishing at infinity, integration by parts gives $\int f f'' \, dx = -\int (f')^2 \, dx \le 0$, the analog of $x^\top(-L)x = -\|Bx\|^2 \le 0$.

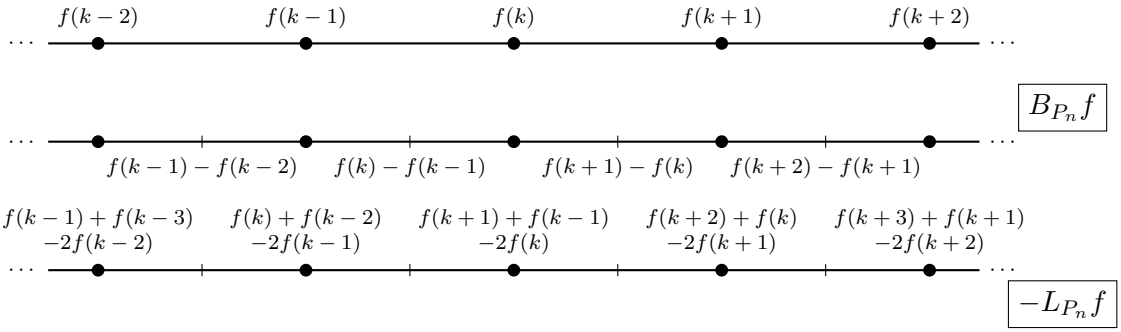


Figure 1.3: Discretizing derivatives in one dimension. Top: samples of f. Middle: first differences, one per edge (up to sign, the entries of $B_{P_n} f$). Bottom: second differences, one per vertex (the entries of $-L_{P_n} f$).

Discretizing derivatives: the 2-d case. Now let $f : \mathbb{R}^2 \to \mathbb{R}$ and sample it on the integer grid, which we think of as a grid graph (Fig. 1.4). Each vertex (j, k) has four neighbors $(j \pm 1, k)$ and $(j, k \pm 1)$. The first differences along the four edges at (j, k) are $f(j,k) - f(j-1,k)$, $f(j+1,k) - f(j,k)$, $f(j,k) - f(j,k-1)$ and $f(j,k+1) - f(j,k)$. Taking differences of differences in each direction and adding, the second differences sum to

$$f(j,k-1) + f(j,k+1) + f(j-1,k) + f(j+1,k) - 4f(j,k) = -\big[L_{\text{grid}} f\big](j,k),$$

the discrete analog of $\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$. This matches the product structure of Theorem 1.2.7. The grid is $P \times P$, and its Laplacian is $L_1 + L_2$, one term for each coordinate direction, just as Δ has one second derivative per coordinate.

1.2 Spectra of Common Graphs and Spectral Drawings

This section computes the full spectrum of several common graphs: the complete graph, the ring, the path, and, through a general theorem on graph products, the grid. These explicit spectra are the examples against which later bounds are tested. We then use the first few eigenvectors as coordinates to draw graphs, and see that they carry a surprising amount of geometric information about the graph.

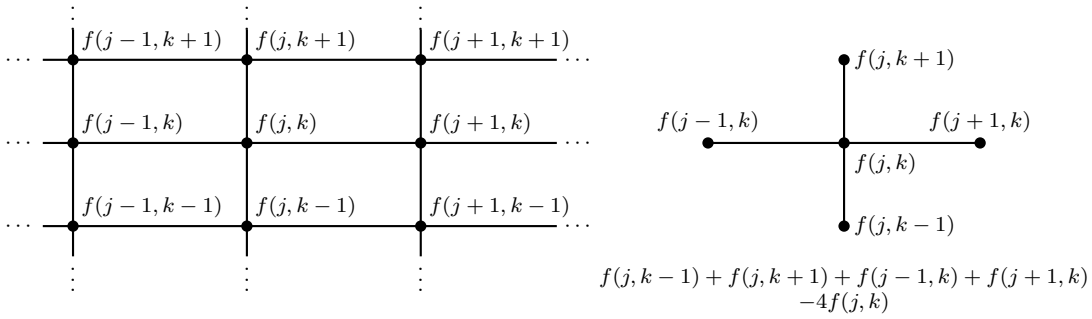


Figure 1.4: Discretizing derivatives in two dimensions. Left: samples of f on the grid. Right: the four neighbors of (j, k), and the sum of second differences at (j, k), which is $-[L_{\text{grid}} f](j, k)$.

1.2.1 Spectra of some common graphs

The complete graph

The complete graph K_n has an edge (i, j) for all $i \neq j$ (Fig. 1.5).

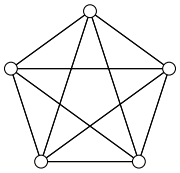


Figure 1.5: The complete graph K_5.

Proposition 1.2.1. *The Laplacian of K_n has $v_1 = \mathbf{1}$, $\lambda_1 = 0$, and the eigenvalue n with multiplicity $n - 1$; the eigenspace of n is $\{x : x^\top \mathbf{1} = 0\}$.*

Proof. K_n is connected, so by Claim 1.1.13 the eigenvalue 0 has multiplicity one. Now take any x with $x^\top \mathbf{1} = \sum_j x_j = 0$. Every vertex has degree $n - 1$ and is adjacent to all the others, so

$$[L_{K_n} x]_i = (n-1)x_i - \sum_{j \neq i} x_j = nx_i - \sum_j x_j = nx_i.$$

So every nonzero vector orthogonal to $\mathbf{1}$ is an eigenvector with eigenvalue n, and these vectors form a space of dimension $n - 1$. □

The ring graph

The ring R_n has the edges $(i, i + 1)$ for $i = 1, \ldots, n - 1$, and $(1, n)$ (Fig. 1.6). It is convenient to read vertex indices modulo n, so that the neighbors of vertex u are $u - 1$ and $u + 1$.

Proposition 1.2.2. *The eigenvectors of L_{R_n} come in pairs: for $0 \leq k \leq n/2$,*

$$x_k(u) = \cos\Big(\frac{2\pi k u}{n}\Big), \qquad y_k(u) = \sin\Big(\frac{2\pi k u}{n}\Big)$$

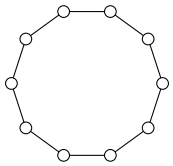


Figure 1.6: The ring graph R_{10}.

are both eigenvectors with eigenvalue

$$2 - 2\cos\Big(\frac{2\pi k}{n}\Big).$$

Note that $x_0 = \mathbf{1}$*, while* $y_0 = \mathbf{0}$ *and, if* n *is even,* $y_{n/2} = \mathbf{0}$*; we ignore these two.*

Proof by picture. Plot R_n with vertex u at the point $(x_k(u), y_k(u))$, as shown for $n = 10$ and $k = 3$ in Fig. 1.7. Every vertex lands on the unit circle, and the two neighbors of each vertex land at angles $\pm 2\pi k/n$ from it. So the picture looks the same from every vertex: going from u to $u + 1$ rotates the whole picture by the angle $2\pi k/n$. In particular, at every vertex the average of the position vectors of the two neighbors is the same multiple of the vertex's own position vector. So it is enough to compute this multiple at a single vertex, say $u = n$. There $x_k(n) = 1$ and $y_k(n) = 0$, while the neighbors $u = n \pm 1$ have $x_k = \cos(2\pi k/n)$ and $y_k = \pm\sin(2\pi k/n)$. Hence

$$\begin{aligned}
[Lx_k](n) &= 2 \cdot 1 - 2\cos(2\pi k/n) = \big(2 - 2\cos(2\pi k/n)\big)\, x_k(n), \\
[Ly_k](n) &= 2 \cdot 0 - \sin(2\pi k/n) + \sin(2\pi k/n) = 0 = \big(2 - 2\cos(2\pi k/n)\big)\, y_k(n),
\end{aligned}$$

and by the symmetry just described the same eigenvalue equations hold at every vertex.

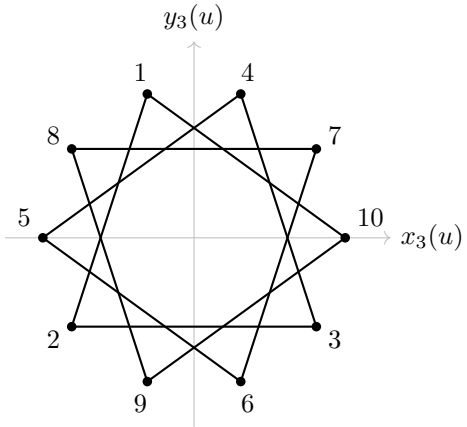


Figure 1.7: The ring R_{10} drawn with coordinates $(x_k(u), y_k(u))$ for $k = 3$: vertex u sits at angle $2\pi \cdot 3u/10$ on the unit circle, and consecutive vertices are 108° apart.

Proof with roots of unity. One can also prove Proposition 1.2.2 easily by writing the eigenvectors in terms of roots of unity, and this shows more: for any $k \in \mathbb{Z}$ and any $c \in \mathbb{R}$,[1] the

[1] The case $c = -k/2$, which need not be an integer, is used for the path in Section 1.2.1.

vector

$$x(u) = \mathrm{Re}\big(e^{2\pi i(ku+c)/n}\big) \tag{1.2.1}$$

is an eigenvector of L_{R_n} with eigenvalue $2 - 2\cos(2\pi k/n)$. Indeed, let $z(u) = e^{2\pi i(ku+c)/n}$. Since k is an integer, $z(u+n) = z(u)$, so z is a well-defined function on the vertices of the ring, and

$$\begin{aligned}[Lz](u) &= 2z(u) - z(u-1) - z(u+1) \\ &= e^{2\pi i(ku+c)/n}\big(2 - e^{-2\pi ik/n} - e^{2\pi ik/n}\big) = \big(2 - 2\cos(2\pi k/n)\big)\, z(u).\end{aligned}$$

Since L is a real matrix, taking real parts of both sides gives $Lx = (2 - 2\cos(2\pi k/n))\, x$. (Taking $c = 0$ gives x_k, and taking imaginary parts instead of real parts gives y_k.)

Remark 1.2.3. The vectors of Proposition 1.2.2 are n nonzero, mutually orthogonal vectors $(x_0, \dots, x_{\lfloor n/2 \rfloor}$ and $y_1, \dots, y_{\lceil n/2 \rceil - 1})$, so they form a complete basis of eigenvectors of L_{R_n}. Most eigenspaces are two-dimensional, so there are many other choices of basis; the vectors (1.2.1) with different c are rotations within an eigenspace.

These eigenvectors look like Fourier coefficients. We explain why at the end of Section 1.2.1.

The path graph

The path P_n has the edges $(i, i+1)$ for $i = 1, \dots, n-1$. The trick for finding its spectrum is to realize P_n as a quotient of the ring R_{2n}, using the projection map $\pi : R_{2n} \to P_n$,

$$\pi(\text{vertex } i) = \begin{cases} \text{vertex } i & \text{if } i \le n, \\ \text{vertex } 2n+1-i & \text{if } i > n. \end{cases}$$

Picture the ring R_{2n} flattened into two rows, vertices $1, \dots, n$ on top and $2n, \dots, n+1$ below them, and fold the bottom row onto the top row (Fig. 1.8).

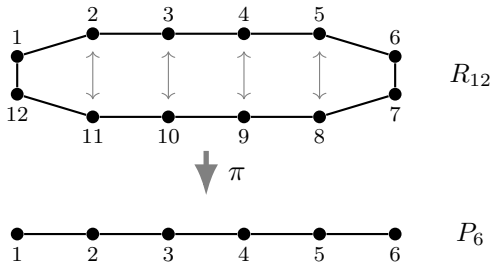


Figure 1.8: The path P_6 as the quotient of the ring R_{12}: the map π identifies vertex i with vertex $13 - i$.

We can use any eigenvector of R_{2n} that respects the quotient. This is an example of a general technique.

Lemma 1.2.4. *Let w be an eigenvector of $L_{R_{2n}}$ with eigenvalue λ that respects the quotient, that is, $w(i) = w(2n+1-i)$ for all i. Let $v \in \mathbb{R}^n$ be its restriction to the vertices $1, \dots, n$. Then v is an eigenvector of L_{P_n} with eigenvalue λ.*

Proof. For an interior vertex $2 \le i \le n-1$, the neighbors of i are $i-1$ and $i+1$ in both graphs, so $[L_{P_n} v](i) = 2w(i) - w(i-1) - w(i+1) = [L_{R_{2n}} w](i) = \lambda v(i)$. At the endpoint $i = 1$, the neighbors in R_{2n} are 2 and $2n$, and $w(2n) = w(1)$, so

$$[L_{R_{2n}} w](1) = 2w(1) - w(2) - w(2n) = w(1) - w(2) = [L_{P_n} v](1).$$

At the endpoint $i = n$, the neighbors in R_{2n} are $n-1$ and $n+1$, and $w(n+1) = w(n)$, so $[L_{R_{2n}} w](n) = w(n) - w(n-1) = [L_{P_n} v](n)$. Hence $L_{P_n} v = \lambda v$ at every vertex. (And $v \neq \mathbf{0}$, since w is determined by v.) □

So we just need eigenvectors v_k of R_{2n} for which $v_k(i) = v_k(2n+1-i)$ for all i. The eigenvectors of Proposition 1.2.2 do not line up with this reflection, but we can rotate them so that they do: take (1.2.1) on R_{2n},

$$\mathrm{Re}\big(e^{2\pi i(ku+c)/2n}\big) \qquad \text{with } c = -\frac{k}{2},$$

which gives

$$v_k(u) = \cos\Big(\frac{\pi k(u-\frac{1}{2})}{n}\Big). \tag{1.2.2}$$

This respects the quotient: $v_k(2n+1-u) = \cos(2\pi k - \pi k(u-\frac{1}{2})/n) = v_k(u)$. So by Lemma 1.2.4, v_k (restricted to $u = 1, \dots, n$) automatically satisfies the eigenvalue equation of P_n, with the same eigenvalue as in R_{2n}, namely $2 - 2\cos(2\pi k/2n) = 2 - 2\cos(\pi k/n)$.

We get n of these (instead of the $2n$ eigenvectors of R_{2n}), one for each $k = 0, \dots, n-1$. Other values of k give nothing new: $k = n$ gives the zero vector, since $\cos(\pi(u-\frac{1}{2})) = 0$, and replacing k by $2n-k$ or by $k+2n$ just changes the sign of v_k. And these n vectors are a complete set of eigenvectors of L_{P_n}. Each is nonzero, since its first entry is $\cos(\pi k/2n) > 0$. Their eigenvalues $2 - 2\cos(\pi k/n)$, $0 \le k < n$, are distinct, since cos is strictly decreasing on $[0, \pi)$. So they are n mutually orthogonal eigenvectors in $\mathbb{R}^n$. To sum up:

Theorem 1.2.5. *The Laplacian of the path P_n has eigenvectors*

$$v_k(u) = \cos\Big(\frac{\pi k(u-\frac{1}{2})}{n}\Big), \qquad u = 1, \dots, n,$$

with eigenvalues

$$2 - 2\cos\Big(\frac{\pi k}{n}\Big) \qquad \textit{(the same as for } R_{2n}\textit{)},$$

for $k = 0, \dots, n-1$.

Here $k = 0$ gives $v_1 = \mathbf{1}$. In particular $\lambda_2(P_n) = 2 - 2\cos(\pi/n)$, which is of order $1/n^2$.

A note on Fourier coefficients. We have seen that rings and paths have eigenvectors that look like Fourier coefficients. The continuous Laplacian of Section 1.1.10 explains this:

$$\frac{d^2}{dx^2}\sin(kx+c) = -k^2\sin(kx+c), \qquad \frac{d^2}{dx^2}\cos(kx+c) = -k^2\cos(kx+c).$$

So $\sin(kx+c)$ and $\cos(kx+c)$ are eigenfunctions of the operator $\frac{d^2}{dx^2}$, the 1-d Laplacian, and the eigenvectors of the ring and the path are their discrete versions. For example, the eigenvector $x_k(u) = \cos(2\pi ku/n)$ of the ring is just $\cos(kt)$ sampled at the points $t = uh$, with spacing $h = 2\pi/n$. Its eigenvalue $2 - 2\cos(kh)$ is close to k^2h^2 for small kh. So up to the factor h^2 and the sign, it is the continuous eigenvalue $-k^2$.

Graph products

The next natural example is the grid graph. Its spectrum will follow from a general theorem about product graphs.

Definition 1.2.6. Let $G = (V, E)$ and $H = (W, F)$. The (Cartesian) product $G \times H$ is the graph with vertex set $V \times W$ and edges

$$\begin{aligned} &((v_1, w), (v_2, w)) && \text{for } (v_1, v_2) \in E \text{ and } w \in W \quad \text{(type 1)}, \\ &((v, w_1), (v, w_2)) && \text{for } (w_1, w_2) \in F \text{ and } v \in V \quad \text{(type 2)}. \end{aligned}$$

So $G \times H$ consists of one copy of G for each vertex of H (the type 1 edges), with the copies joined to each other according to H (the type 2 edges, which join corresponding vertices of two copies). Figure 1.9 shows an example, $K_{10} \times K_{2,8}$, where $K_{2,8}$ has two "apex" vertices each joined to the same eight middle vertices.

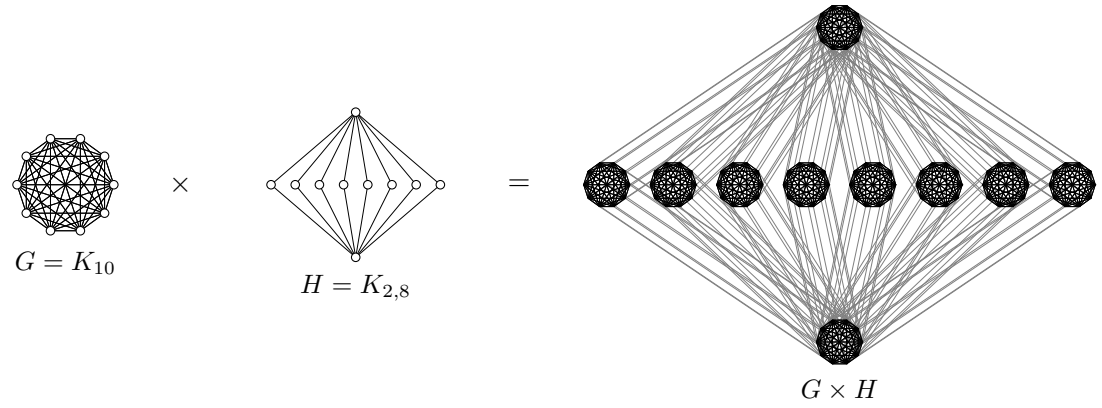


Figure 1.9: Top: the product $K_{10} \times K_{2,8}$ (one copy of K_{10} for each vertex of $K_{2,8}$; gray lines are type 2 edges). Bottom: the nonzero entries of L_1, L_2 and $L_{G\times H} = L_1 + L_2$, with the 100 vertices (v, w) ordered by w (top apex first, bottom apex last) and then by v. Each 10×10 block corresponds to a pair of copies of G: the type 1 edges fill the diagonal blocks, and each type 2 edge pattern is a diagonal line in an off-diagonal block.

By additivity, the Laplacian of the product splits by edge type: $L_{G\times H} = L_1 + L_2$, where L_1 is the Laplacian of the type 1 edges and L_2 that of the type 2 edges. Explicitly,

$$[L_1]_{(v,w),(v',w')} = \begin{cases} [L_G]_{v,v'} & \text{if } w = w', \\ 0 & \text{otherwise,} \end{cases} \qquad [L_2]_{(v,w),(v',w')} = \begin{cases} [L_H]_{w,w'} & \text{if } v = v', \\ 0 & \text{otherwise.} \end{cases} \tag{1.2.3}$$

In words, L_1 has a copy of L_G for each vertex of H, and L_2 has a copy of L_H for each vertex of G. The grid graph is the simplest example (Fig. 1.10): the $n \times m$ grid is

$$G_{n\times m} = P_n \times P_m,$$

and it is the union of m copies of P_n and n copies of P_m.

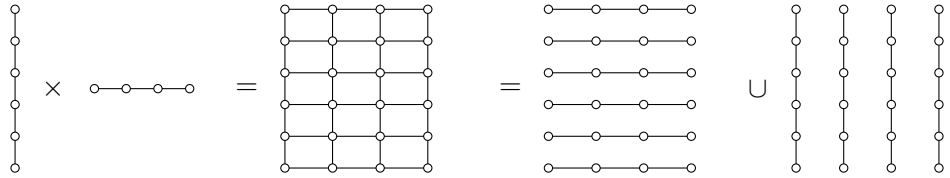


Figure 1.10: The grid graph $P_6 \times P_4$, and its decomposition into the edges of the copies of P_4 (horizontal) and the copies of P_6 (vertical).

Theorem 1.2.7. *Suppose L_G has eigenvalues $\lambda_1, \dots, \lambda_n$ with eigenvectors $x_1, \dots, x_n$, and L_H has eigenvalues $\mu_1, \dots, \mu_k$ with eigenvectors $y_1, \dots, y_k$. (In this subsection μ_j denotes an eigenvalue of L_H.) Then for all $i \in [n]$ and $j \in [k]$, $L_{G\times H}$ has the eigenvector*

$$z_{ij}(v, w) = x_i(v)\, y_j(w)$$

with eigenvalue $\lambda_i + \mu_j$.

Note that the eigenvalues *add*; they do not multiply.

Proof. Using (1.2.3), only the terms with $w' = w$ survive in L_1:

$$\begin{aligned}[L_1 z_{ij}](v, w) &= \sum_{(v',w')} [L_1]_{(v,w),(v',w')}\, z_{ij}(v', w') = \sum_{v'} [L_G]_{v,v'}\, x_i(v') y_j(w) \\ &= y_j(w) \sum_{v'} [L_G]_{v,v'}\, x_i(v') = \lambda_i\, x_i(v) y_j(w).\end{aligned}$$

Similarly, only the terms with $v' = v$ survive in L_2:

$$\begin{aligned}[L_2 z_{ij}](v, w) &= \sum_{(v',w')} [L_2]_{(v,w),(v',w')}\, z_{ij}(v', w') = \sum_{w'} [L_H]_{w,w'}\, x_i(v) y_j(w') \\ &= x_i(v) \sum_{w'} [L_H]_{w,w'}\, y_j(w') = \mu_j\, x_i(v) y_j(w).\end{aligned}$$

Adding,

$$\begin{aligned}[L_{G\times H} z_{ij}](v, w) &= [L_1 z_{ij}](v, w) + [L_2 z_{ij}](v, w) = \lambda_i x_i(v) y_j(w) + \mu_j x_i(v) y_j(w) \\ &= (\lambda_i + \mu_j)\, x_i(v) y_j(w) = (\lambda_i + \mu_j)\, z_{ij}(v, w).\end{aligned}$$

□

Remark 1.2.8. In tensor notation, if the vertices (v, w) are ordered by v and then by w, then $L_{G\times H} = L_G \otimes I_k + I_n \otimes L_H$ and $z_{ij} = x_i \otimes y_j$, and the theorem is the computation $(L_G \otimes I_k + I_n \otimes L_H)(x_i \otimes y_j) = (\lambda_i + \mu_j)(x_i \otimes y_j)$.

The theorem gives the complete spectrum of $G \times H$. If $x_1, \ldots, x_n$ and $y_1, \ldots, y_k$ are orthonormal bases, then $z_{ij}^\top z_{i'j'} = (x_i^\top x_{i'})(y_j^\top y_{j'})$, so the nk vectors z_{ij} are orthonormal and form a basis of $\mathbb{R}^{V \times W}$. Combining this with Theorem 1.2.5, the grid $P_n \times P_m$ has eigenvectors $z_{kl}(a,b) = \cos(\pi k(a - \frac{1}{2})/n) \cos(\pi l(b - \frac{1}{2})/m)$ with eigenvalues

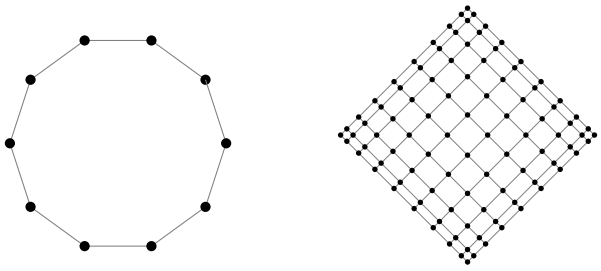
$$\Big(2 - 2\cos\frac{\pi k}{n}\Big) + \Big(2 - 2\cos\frac{\pi l}{m}\Big), \qquad 0 \le k < n,\ 0 \le l < m.$$

For the 10×10 grid of Fig. 1.11, the two smallest nonzero eigenvalues come from $(k,l) = (1,0)$ and $(0,1)$, and they are equal. Their eigenvectors $\cos(\pi(a - \frac{1}{2})/10)$ and $\cos(\pi(b - \frac{1}{2})/10)$, used as coordinates, draw the grid with its lines slightly bunched toward the boundary. This is the picture in Fig. 1.11, turned by 45°: the eigenspace is two-dimensional, and the drawing there uses a basis of it rotated by 45° from this one.

1.2.2 Drawing graphs with eigenvectors

So far we have used eigenvectors to find eigenvalues. Now we look at the eigenvectors themselves. A lot of information is contained in the first few *nontrivial* eigenvectors $v_2, v_3, \ldots$. Each of v_2 and v_3 is a vector, and so gives a map $G \to \mathbb{R}$. Let us use them as coordinates: place vertex u at the point $(v_2(u), v_3(u))$ of the plane, draw the edges as straight segments, and see what happens.

Figure 1.11 shows the result for a ring (cycle) on 10 vertices and for a 10×10 grid. In both cases the drawing is the picture one would draw by hand.



ring, 10 vertices 10×10 grid

Figure 1.11: Vertex u drawn at $(v_2(u), v_3(u))$, for the ring graph on 10 vertices (left) and the 10×10 grid graph (right). These drawings use the exact eigenvectors found in Section 1.2.1. Since $\lambda_2 = \lambda_3$, the eigenspace is two-dimensional, and a different orthonormal basis of it just rotates or reflects the picture.

The same works for graphs that were not drawn on a grid to begin with. Here are two examples of meshes, each first drawn with hand-made coordinates (the coordinates the mesh came with) and then with the coordinates (v_2, v_3); see Figs. 1.12 and 1.13. The spectral drawing produces a recognizable, untangled picture of the same graph, using nothing but its adjacency structure.

Figure 1.14 is the most striking example. We are handed a graph with 9821 vertices whose adjacency matrix has 28262 nonzero entries, with the vertices in a random order. Its sparsity pattern looks like noise, and so does a drawing with random coordinates. Drawn with coordinates (v_2, v_3), the graph turns out to be a planar mesh in which the letters "MIT" stand out, meshed much more finely than the rest.

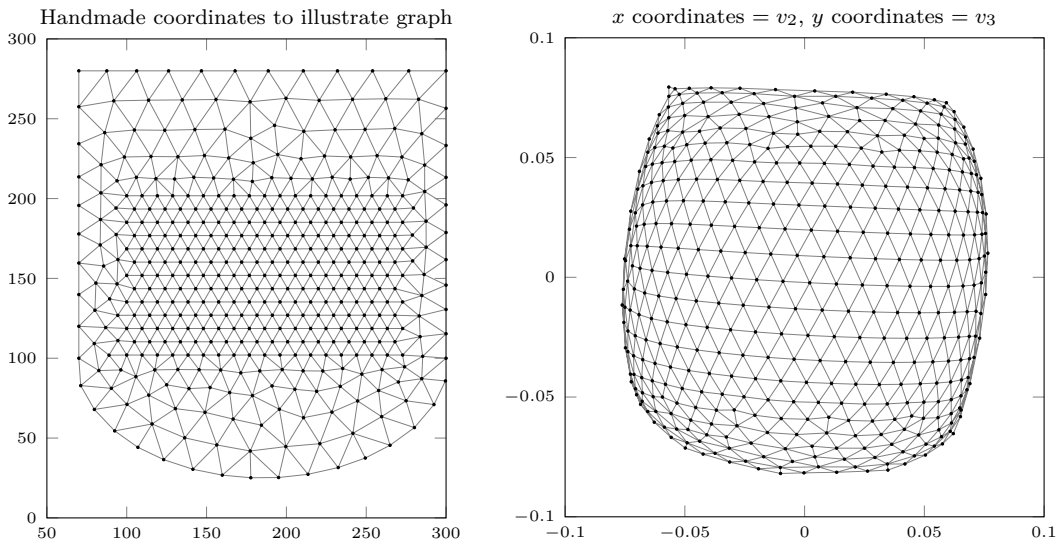


Figure 1.12: A geometric graph (after Dan Spielman), drawn with its own coordinates and with (v_2, v_3).

Finally, let us go back to the ring, now with 20 vertices, and use other pairs of eigenvectors as coordinates (Fig. 1.15). With (v_2, v_3) we again get the ring drawn as a regular polygon. With (v_6, v_7) and with (v_{18}, v_{19}) we get star polygons: consecutive vertices of the ring land farther and farther apart around the circle. Such periodic behavior is explained by the eigenvectors of the ring in Proposition 1.2.2.

1.3 Eigenvalue Bounds and the Courant–Fischer Theorem

For most graphs we cannot compute the spectrum exactly, as we did for the graphs of Section 1.2, so we want bounds. After two simple bounds from the trace, this section develops the main tool for bounding eigenvalues without computing them, the Courant–Fischer theorem: λ_2 is the minimum of a ratio (the Rayleigh quotient) over vectors orthogonal to $\mathbf{1}$, so any single test vector gives an upper bound on λ_2. We then use test vectors to show that $\lambda_{\max}$ is more than the maximum degree, and to bound λ_2 for the path and the complete binary tree.

1.3.1 Bounds from the trace

The simplest bounds come from the trace.

Claim 1.3.1. $\sum_i \lambda_i = \sum_i d_i \le d_{\max} n$, *where* d_i *is the degree of the* i*th vertex and* $d_{\max}$ *the maximum degree.*

Proof. Take the trace of L: it is the sum of the eigenvalues, and also the sum of the diagonal entries, which are the degrees. □

Claim 1.3.2.

$$\lambda_2 \le \frac{\sum_i d_i}{n-1} \le \lambda_n.$$

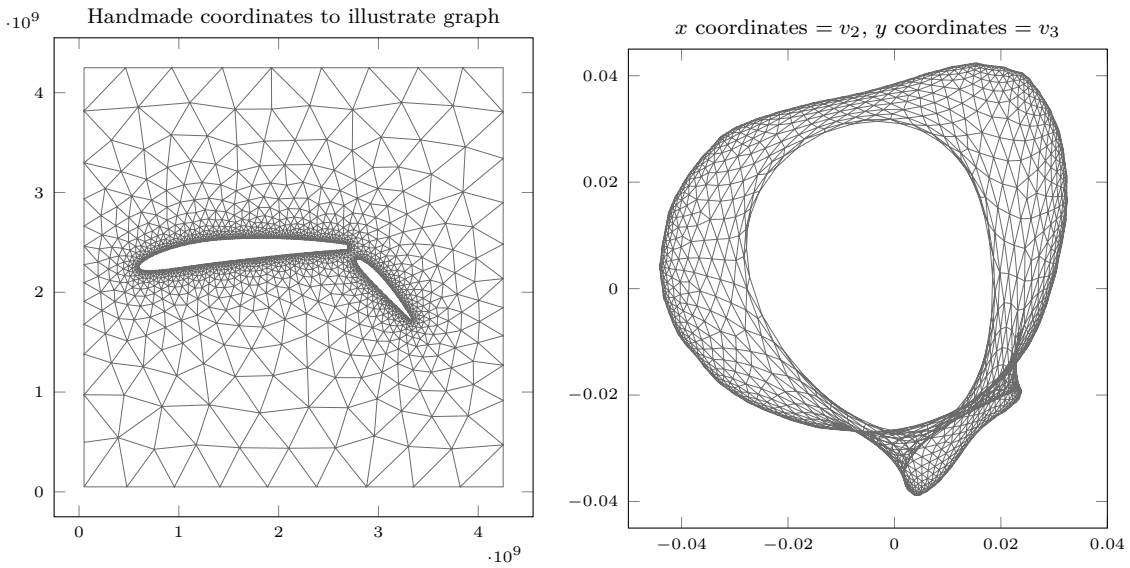


Figure 1.13: The airfoil mesh, drawn with its own coordinates and with (v_2, v_3).

Proof. This follows from Claim 1.3.1 and the fact that $\lambda_1 = 0$: the $n-1$ eigenvalues $\lambda_2 \le \cdots \le \lambda_n$ sum to $\sum_i d_i$, so the smallest of them is at most their average and the largest is at least their average. □

1.3.2 Courant–Fischer and Rayleigh quotients

The main tool for bounding λ_2 and $\lambda_{\max}$ is a variational characterization of eigenvalues. In this subsection A is any symmetric matrix (not an adjacency matrix), with eigenvalues $\lambda_1 \le \cdots \le \lambda_n$ and orthonormal eigenvectors $v_1, \dots, v_n$.

Theorem 1.3.3 (Courant–Fischer formula)**.** *For any $n \times n$ symmetric matrix A,*

$$\lambda_1 = \min_{\|x\|=1} x^\top A x = \min_{x \neq \mathbf{0}} \frac{x^\top A x}{x^\top x}, \qquad \lambda_2 = \min_{\substack{\|x\|=1 \\ x \perp v_1}} x^\top A x = \min_{\substack{x \perp v_1 \\ x \neq \mathbf{0}}} \frac{x^\top A x}{x^\top x},$$

$$\lambda_k = \min_{\substack{\|x\|=1 \\ x \perp \mathrm{span}(v_1,\dots,v_{k-1})}} x^\top A x = \min_{\substack{x \perp \mathrm{span}(v_1,\dots,v_{k-1}) \\ x \neq \mathbf{0}}} \frac{x^\top A x}{x^\top x}, \qquad \lambda_{\max} = \max_{\|x\|=1} x^\top A x = \max_{x \neq \mathbf{0}} \frac{x^\top A x}{x^\top x}.$$

The ratio $\frac{x^\top A x}{x^\top x}$ is called the *Rayleigh quotient* of x. The two forms of each formula agree: scaling x does not change the Rayleigh quotient, so we may take $\|x\| = 1$.

Proof. Write things in the eigenbasis, so that A is diagonal. That is, write $A = V \Lambda V^\top$ and replace x by $V^\top x$; since V is orthogonal, this does not change $\|x\|$, and the eigenvectors become the standard basis vectors. Then

$$x^\top A x = \begin{pmatrix} x_1 & \cdots & x_n \end{pmatrix} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \sum_{i=1}^n \lambda_i x_i^2.$$

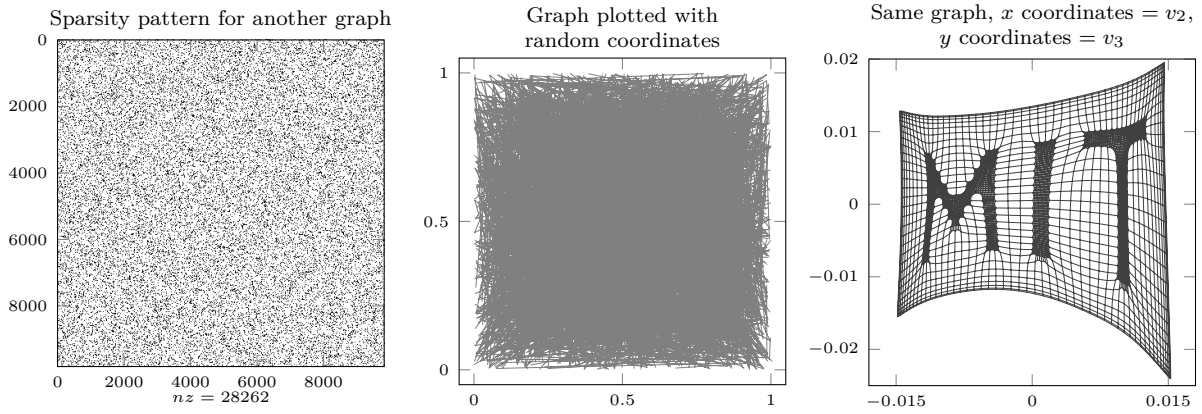


Figure 1.14: The same graph three ways: sparsity pattern, random coordinates, and the spectral coordinates (v_2, v_3).

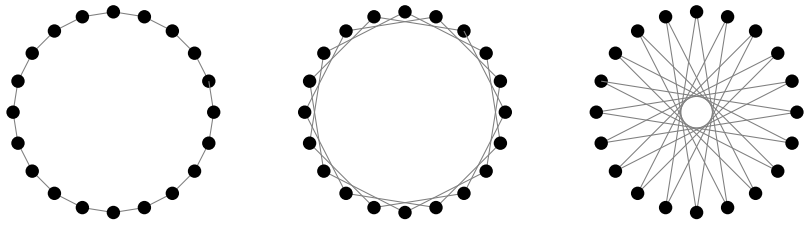


Figure 1.15: The ring graph with 20 vertices, with vertex u drawn at $(v_2(u), v_3(u))$, $(v_6(u), v_7(u))$ and $(v_{18}(u), v_{19}(u))$. These also use the exact eigenvectors of Proposition 1.2.2: the pair (v_{2k}, v_{2k+1}) places vertex u at angle $2\pi ku/20$, here for $k = 1, 3, 9$.

If x is a unit vector orthogonal to the first $k-1$ eigenvectors, then $x_1 = \cdots = x_{k-1} = 0$, and

$$x^\top A x = \sum_{i=1}^{n} \lambda_i x_i^2 = \sum_{i=k}^{n} \lambda_i x_i^2 \geq \lambda_k \sum_{i=k}^{n} x_i^2 = \lambda_k \|x\|^2 = \lambda_k.$$

On the other hand, $x = v_k$ is such a vector and has $v_k^\top A v_k = \lambda_k$, so the minimum is exactly λ_k. The formula for $\lambda_{\max}$ follows by applying the formula for λ_1 to $-A$, whose eigenvalues are the negatives of those of A. □

For the Laplacian, $v_1 = \mathbf{1}$ and $\lambda_1 = 0$, and the quadratic form is $\sum_{(i,j)\in E}(x_i - x_j)^2$. So:

Corollary 1.3.4 (Courant–Fischer for the Laplacian)**.**

$$\lambda_2 = \min_{\substack{x \perp \mathbf{1} \\ x \neq \mathbf{0}}} \frac{x^\top L x}{x^\top x} = \min_{\substack{x \perp \mathbf{1} \\ x \neq \mathbf{0}}} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_{i \in V} x_i^2}, \tag{1.3.1}$$

$$\lambda_{\max} = \max_{x \neq \mathbf{0}} \frac{x^\top L x}{x^\top x} = \max_{x \neq \mathbf{0}} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_{i \in V} x_i^2}. \tag{1.3.2}$$

One can interpret (1.3.1) with springs: put a spring on each edge and minimize the potential energy $\sum_{(i,j)\in E}(x_i - x_j)^2$ of the configuration x, but with slightly weird boundary conditions, namely the normalization $\sum_i x_i^2 = 1$ and $\sum_i x_i = 0$.

Using this. To get an *upper* bound on λ_2, we just need to produce a vector $x \perp \mathbf{1}$ with a small Rayleigh quotient. To get a *lower* bound on $\lambda_{\max}$, we need a vector with a large Rayleigh quotient. No diagonalization is needed. Here are three examples.

Example 1.3.5. Suppose G has a vertex v of degree d. Then $\lambda_{\max} \geq d + 1$.

Proof. Let x be given by

$$x(v) = d, \qquad x(\text{any neighbor of } v) = -1, \qquad x(\text{any other vertex}) = 0.$$

Each of the d edges at v contributes $(d - (-1))^2 = (d+1)^2$ to the numerator of the Rayleigh quotient, and every other edge contributes something nonnegative. The denominator is $d^2 + d \cdot 1$. So by (1.3.2),

$$\lambda_{\max} \geq \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_{i\in V} x_i^2} \geq \frac{d(d+1)^2}{d^2 + d} = d + 1. \qquad \square$$

Example 1.3.6. The path graph P_n has $\lambda_2 \leq O(1/n^2)$. Of course, we already know this from the exact spectrum in Theorem 1.2.5, but this argument is easier and more general. It is a little cleaner to use P_{n+1}, so that there are n edges. Label the vertices $0, 1, \ldots, n$ along the path and let $x_i = 2i - n$, so the values run $-n, -n+2, \ldots, n-2, n$ (Fig. 1.16). Then $\sum_i x_i = 0$, so $x \perp \mathbf{1}$, and each of the n edges contributes $2^2 = 4$ to the numerator:

$$\lambda_2(P_{n+1}) \leq \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_i x_i^2} = \frac{4n}{\sum_{i=0}^{n}(2i-n)^2} = \frac{4n}{\Omega(n^3)} = O\Big(\frac{1}{n^2}\Big).$$

In fact $\sum_{i=0}^{n}(2i-n)^2 = n(n+1)(n+2)/3$, so the bound is $12/((n+1)(n+2))$. The true value, from Theorem 1.2.5, is $2 - 2\cos(\pi/(n+1)) \approx \pi^2/(n+1)^2$.

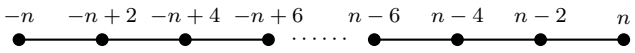


Figure 1.16: The test vector $x_i = 2i - n$ on the path P_{n+1}.

Example 1.3.7. The complete binary tree on $n = 2^h - 1$ nodes has $\lambda_2 \leq \frac{2}{n-1}$. Let x be 0 on the root, -1 on every node of the left subtree of the root, and 1 on every node of the right subtree (Fig. 1.17). The two subtrees have the same number of nodes, so $x \perp \mathbf{1}$. Only the two edges at the root join nodes with different values, and each contributes 1; the denominator counts the $n - 1$ non-root nodes. So

$$\lambda_2 \leq \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_i x_i^2} = \frac{2}{n-1}.$$

This bound is correct up to a constant factor, and we got it with little effort, without diagonalizing a big matrix.

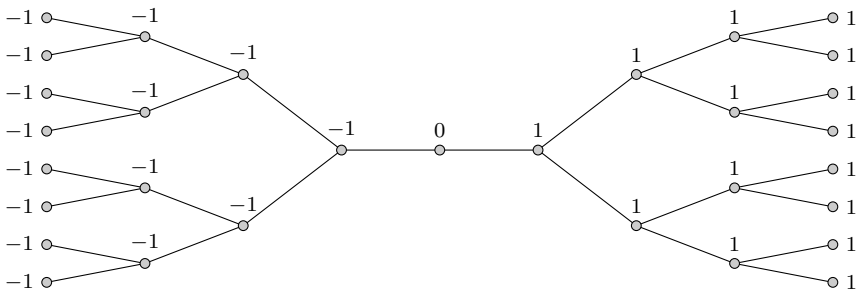


Figure 1.17: The test vector on the complete binary tree with $31 = 2^5 - 1$ nodes: 0 at the root, -1 on the left subtree, 1 on the right subtree.

1.4 Graph Cuts and Cheeger's Inequality

This section connects λ_2 to cutting a graph into two pieces. We first ask what a "good" cut should be, and settle on two measures, the cut ratio and the conductance. Both are NP-hard to minimize, and the main result of the section, Cheeger's inequality, says that λ_2 approximates them. The easy half comes from a general recipe of approximation algorithms: write the problem as an integer program, drop the integrality constraint, and solve the resulting *relaxation*. For the cut ratio the relaxation turns out to be exactly the Courant–Fischer formula for λ_2, which gives $\lambda_2 \leq 2\phi(G)$. The hard half needs a way to turn a vector back into a cut (a *rounding* method), together with a bound on how much rounding costs. The inequality is cleanest for the conductance and the normalized Laplacian, so we introduce that matrix and prove the hard half for it: sort the vertices by their values in a vector, cut at a random threshold, and bound the expected cost. Along the way we watch the construction run on a small example.

1.4.1 What is a good cut?

Two first attempts

We want to cut a graph into two pieces so that both pieces are pretty big, for example for parallel processing, for divide-and-conquer algorithms, or for clustering. What quantity do we want to minimize?

Notation. For a set $S \subseteq V$ of vertices of G: $|S|$ is the number of vertices in S; $\bar{S} = V \setminus S$ is the complement of S, and $|\bar{S}|$ the number of vertices in it; and $e(S) = e(\bar{S})$ is the number of edges between S and $\bar{S}$.

Attempt 1: min-cut. Divide the vertex set V into two pieces S and $\bar{S}$ so as to minimize $e(S)$. This captures the intuition that a good cut should not break too many edges, but by itself it fails: Fig. 1.18 shows a problem. There we would like to cut across the two edges in the middle, between the two dense clusters, but the minimum cut is the single edge on the right, which cuts off one vertex.

Attempt 2: approximate bisection. Cut the vertex set into two sets each of size $n/2$, or maybe each of size at least, say, $n/3$. This takes care of the problem in Fig. 1.18, but it has its

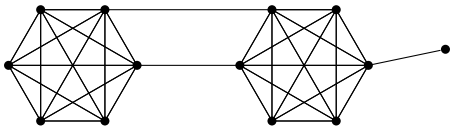


Figure 1.18: The problem with min-cut: two copies of K_6 joined by two edges, plus one extra vertex attached by a single edge. The minimum cut separates the extra vertex.

own problem, shown in Fig. 1.19: there we would like to cut the single edge between the two clusters, but the balance requirement forces us to cut through the dense graph on the left.

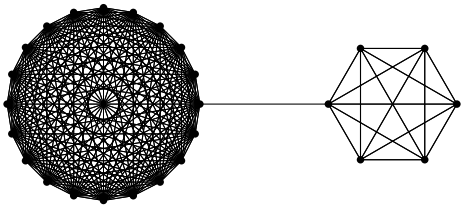


Figure 1.19: The problem with approximate bisection: a complete graph on 20 vertices joined by a single edge to a complete graph on 6 vertices. Any cut with both sides of size at least $n/3$ must cut through the large clique.

The cut ratio and the conductance

The following criteria balance the two attempts: they reward cutting few edges, but penalize cuts with a small side.

Definition 1.4.1. The *cut ratio* of a cut $(S, \bar{S})$ is

$$\phi(S) = \frac{e(S)}{\min(|S|, |\bar{S}|)}.$$

The cut of minimum ratio is the cut that minimizes $\phi(S)$. The *isoperimetric number* of G is the value of this cut:

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

(S must be nonempty and not all of V for $\phi(S)$ to be defined.) For example, in Fig. 1.18 cutting off the extra vertex has ratio $1/1 = 1$, while cutting between the two clusters has ratio $2/6 = 1/3$. In Fig. 1.19 cutting the single edge has ratio $1/6$, while any cut through the large clique cuts many edges.

Alternatively, we could weight the vertices by their degrees. For a set S, let

$$d(S) = \sum_{v \in S} d(v)$$

be the total degree of the vertices in S. (Sometimes $d(S)$, or $d(S)/d(V)$, is written $\mathrm{vol}(S)$.)

Definition 1.4.2. For a subset $S \subseteq V$, the *conductance* of the corresponding cut is

$$\Phi(S) = \frac{e(S)}{\min(d(S), d(\bar{S}))},$$

and the conductance of the graph is $\Phi(G) = \min_{\emptyset \neq S \subsetneq V} \Phi(S)$.

Note that the literature is not very consistent about these names (conductance versus isoperimetric number, and whether the "volume" of a set is its number of vertices or its total degree). It is NP-hard to compute the isoperimetric number $\phi(G)$ or the conductance $\Phi(G)$ exactly. The rest of this section shows that λ_2 approximates them; this is Cheeger's inequality (Theorems 1.4.5 and 1.4.8).

1.4.2 Replacing minima by products

Sometimes the minima in the denominators of $\phi(S)$ and $\Phi(S)$ are replaced by products. Up to normalization, this changes the value by at most a factor of 2.

Lemma 1.4.3. *For every $S \subseteq V$,*

$$\frac{n}{2}\min\bigl(|S|,|\bar{S}|\bigr) \le |S|\cdot|\bar{S}| \le n\,\min\bigl(|S|,|\bar{S}|\bigr), \tag{1.4.1}$$

$$\frac{d(V)}{2}\min\bigl(d(S),d(\bar{S})\bigr) \le d(S)\,d(\bar{S}) \le d(V)\,\min\bigl(d(S),d(\bar{S})\bigr). \tag{1.4.2}$$

Proof. By symmetry we may assume $|S| \le |\bar{S}|$, so that the minimum is $|S|$. Since $|S| + |\bar{S}| = n$, the larger part satisfies $n/2 \le |\bar{S}| \le n$; multiplying by $|S|$ gives (1.4.1). The same argument with $d(S) + d(\bar{S}) = d(V)$ gives (1.4.2). □

Dividing $e(S)$ by the three quantities in (1.4.1), and by those in (1.4.2), shows that

$$\phi(S) \le \frac{n\,e(S)}{|S|\cdot|\bar{S}|} \le 2\phi(S), \qquad \Phi(S) \le \frac{d(V)\,e(S)}{d(S)\,d(\bar{S})} \le 2\Phi(S). \tag{1.4.3}$$

The product has a nice interpretation. Write $[A]$ for the indicator of a statement A (so $[A] = 1$ if A is true and $[A] = 0$ otherwise). Then

$$\begin{aligned} |S|\cdot|\bar{S}| &= \Bigl(\sum_{i\in V}[i\in S]\Bigr)\Bigl(\sum_{j\in V}[j\in\bar{S}]\Bigr) = \sum_{i,j\in V}[i\in S,\ j\in\bar{S}] \\ &= \sum_{i<j}[i,j \text{ on different sides of the cut}] = e_{K_n}(S), \end{aligned} \tag{1.4.4}$$

the number of edges cut by S in the complete graph K_n on the same vertex set.

1.4.3 An interlude on relaxations

The following setup is common in approximation algorithms (Fig. 1.20).

- We want to solve an NP-hard problem of the form: minimize $f(x)$ subject to the constraint $x \in C$. Let p be the minimizer.
- Instead, we minimize $f(x)$ subject to a weaker constraint $x \in C'$, where $C' \supseteq C$. This gives a lower minimum, attained at a point q: $f(q) \le f(p)$.
- We show how to "round" q to a feasible point $q' \in C$.
- To show that this gives an approximation, we prove that $f(q') \le \gamma f(q)$ for some $\gamma \ge 1$ (here γ is an approximation factor; from Section 1.5.4 on, γ denotes the spectral gap).

This implies

$$f(q') \le \gamma f(q) \le \gamma f(p),$$

so the rounded point is a γ-approximation to the true minimum.

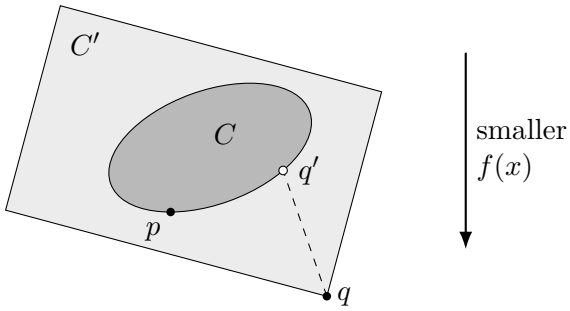


Figure 1.20: Relaxation and rounding. Lower points have smaller values of f. The true minimizer over C is p; the minimizer q over the larger set C' has $f(q) \le f(p)$; rounding moves q to a feasible point $q' \in C$.

1.4.4 An integer program for the cut ratio

Think of a cut as a vector $x \in \{-1, 1\}^n$:

$$x_i = \begin{cases} 1 & \text{if } i \in S, \\ -1 & \text{if } i \in \bar{S}. \end{cases}$$

An edge (i, j) contributes $(x_i - x_j)^2 = 4$ if it is cut and 0 if it is not, so

$$e(S) = \frac{1}{4} \sum_{(i,j)\in E} (x_i - x_j)^2.$$

The same count in the complete graph, together with (1.4.4), gives

$$|S| \cdot |\bar{S}| = e_{K_n}(S) = \frac{1}{4} \sum_{i<j} (x_i - x_j)^2.$$

Therefore

$$\min_{x\in\{-1,1\}^n} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_{i<j}(x_i - x_j)^2} = \min_{S} \frac{e(S)}{|S| \cdot |\bar{S}|}, \tag{1.4.5}$$

where on the left x ranges over the nonconstant vectors, which correspond to the cuts $\emptyset \ne S \subsetneq V$ on the right (a constant x gives $0/0$). By (1.4.3), the value of this integer program lies between $\phi(G)/n$ and $2\phi(G)/n$. So solving it would let us approximate $\phi(G)$ within a factor of 2.

But it is NP-hard to do so. However, if we remove the constraint $x \in \{-1, 1\}^n$, we can solve the program. Dropping the constraint is actually the same as requiring only $x \in [-1, 1]^n$, since we can scale x down without changing the value of the fraction.

1.4.5 Solving the relaxed program

The relaxed program is

$$\min_{\substack{x\in\mathbb{R}^n \\ x\ne c\mathbf{1}}} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_{i<j}(x_i - x_j)^2},$$

where we exclude the constant vectors $x = c\mathbf{1}$ so that the denominator is nonzero. The value of the fraction depends only on the differences $x_i - x_j$, so adding a multiple of $\mathbf{1}$ to x does not

change it. Hence we may suppose without loss of generality that $\sum_i x_i = 0$, that is, $x \perp \mathbf{1}$. For such x,

$$\sum_{i<j}(x_i - x_j)^2 = n\sum_i x_i^2. \tag{1.4.6}$$

This is the Laplacian of the complete graph: the left side is $x^\top L_{K_n} x$, and by Proposition 1.2.1 every vector orthogonal to $\mathbf{1}$ is an eigenvector of L_{K_n} with eigenvalue n, so $x^\top L_{K_n} x = n\, x^\top x$. So, by the Courant–Fischer formula for the Laplacian (Corollary 1.3.4),

$$\min_{\substack{x\in\mathbb{R}^n \\ x\neq c\mathbf{1}}} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_{i<j}(x_i - x_j)^2} = \min_{\substack{x\in\mathbb{R}^n \\ x\perp\mathbf{1},\ x\neq\mathbf{0}}} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{n\sum_i x_i^2} = \frac{\lambda_2}{n}. \tag{1.4.7}$$

1.4.6 The easy half: $\lambda_2 \le 2\phi(G)$

Theorem 1.4.4. *For every graph G, $\lambda_2 \le 2\phi(G)$.*

Proof. We chain together Lemma 1.4.3, the integer program (1.4.5), the relaxation, and (1.4.7):

$$\begin{aligned}\phi(G) = \min_S \frac{e(S)}{\min(|S|,|\bar S|)} &\ge \frac{n}{2}\min_S \frac{e(S)}{|S|\cdot|\bar S|} = \frac{n}{2}\min_{x\in\{-1,1\}^n} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_{i<j}(x_i - x_j)^2} \\ &\ge \frac{n}{2}\min_{\substack{x\in\mathbb{R}^n \\ x\neq c\mathbf{1}}} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_{i<j}(x_i - x_j)^2} = \frac{n}{2}\min_{\substack{x\in\mathbb{R}^n \\ x\perp\mathbf{1}}} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{n\sum_i x_i^2} = \frac{\lambda_2}{2}.\end{aligned}$$

The first inequality is the left half of (1.4.1). The second holds because the relaxed program minimizes over a larger set: every nonconstant $x \in \{-1,1\}^n$ is a nonconstant vector of $\mathbb{R}^n$. □

1.4.7 Cheeger's inequality

The bound $\phi(G) \ge \lambda_2/2$ shows how to construct a relaxation. For the other direction we need two more things:

- a *rounding method*, that is, a way of getting a cut from λ_2 and its eigenvector v_2;
- a bound on how much rounding increases the quantity we are trying to minimize.

The bound on the performance of the rounding will come from showing that it gives a cut whose ratio ϕ satisfies $\phi^2/(2d_{\max}) \le \lambda_2$. Since $\phi(G)$ is at most the ratio of any cut, this gives the left half of the following theorem.

Theorem 1.4.5 (Cheeger's inequality)**.** *For every graph G with maximum degree $d_{\max}$,*

$$\frac{\phi(G)^2}{2d_{\max}} \le \lambda_2 \le 2\phi(G). \tag{1.4.8}$$

The right half is Theorem 1.4.4. For the left half we switch to the normalized Laplacian, where the inequality is cleaner; see Section 1.4.8 and Remark 1.4.9.

Remark 1.4.6 (Both halves are tight). Neither half of (1.4.8) can be improved by more than a constant factor. The path P_n and the complete binary tree on n nodes both have isoperimetric number $\Theta(1/n)$: in each, cutting a single edge in the middle splits the graph into two pieces of

nearly equal size. Conversely, every cut of a connected graph cuts at least one edge and has a side with at most $n/2$ vertices, so $\phi(G) \geq 2/n$. By Theorem 1.2.5 and Example 1.3.6, the path has $\lambda_2 = \Theta(1/n^2)$, so for the path the left half is tight up to a constant ($d_{\max} = 2$). For the binary tree, $\lambda_2 \leq 2/(n-1)$ (Example 1.3.7), and this bound is correct up to a constant factor, so for the tree the right half is tight up to a constant.

1.4.8 The normalized Laplacian

Cheeger's inequality is actually a little cleaner for the conductance,

$$\Phi(S) = \frac{e(S)}{\min\bigl(d(S), d(\bar{S})\bigr)},$$

which weights each vertex by its degree. The analogous matrix is the normalized Laplacian. From now on in this section, G has no isolated vertices, so that D is invertible.

Definition 1.4.7. The *normalized Laplacian* of G is

$$N = D^{-1/2} L D^{-1/2}.$$

Think of the change of variables $x = D^{1/2} y$. Then

$$\frac{x^\top N x}{x^\top x} = \frac{x^\top D^{-1/2} L D^{-1/2} x}{(D^{1/2}y)^\top (D^{1/2}y)} = \frac{y^\top L y}{y^\top D y}. \tag{1.4.9}$$

In the denominator, the entries of y are weighted by the degrees of the vertices, which is what the conductance needs.

1.4.9 Cheeger's inequality for the normalized Laplacian

Let $0 = \nu_1 \leq \nu_2 \leq \cdots \leq \nu_n$ be the eigenvalues of N. (We save μ_i for the eigenvalues of the walk matrix in Section 1.5.3.) Write $\mathbf{d} = D\mathbf{1}$ for the vector of degrees, and $\mathbf{d}^{1/2} = D^{1/2}\mathbf{1}$ for the vector of their square roots. The vector corresponding to $\nu_1 = 0$ is $\mathbf{d}^{1/2}$, since $N\mathbf{d}^{1/2} = D^{-1/2}L\mathbf{1} = \mathbf{0}$. Under the change of variables $x = D^{1/2}y$, the condition $x \perp \mathbf{d}^{1/2}$ becomes $y^\top D^{1/2} D^{1/2}\mathbf{1} = y^\top \mathbf{d} = 0$. So by Courant–Fischer (Theorem 1.3.3) and (1.4.9),

$$\nu_2 = \min_{x \perp \mathbf{d}^{1/2}} \frac{x^\top N x}{x^\top x} = \min_{y \perp \mathbf{d}} \frac{y^\top L y}{y^\top D y}, \tag{1.4.10}$$

both minima over nonzero vectors.

Theorem 1.4.8 (Cheeger's inequality for N)**.** *For every graph G without isolated vertices,*

$$\frac{\Phi(G)^2}{2} \leq \nu_2 \leq 2\Phi(G).$$

The proof that $\nu_2 \leq 2\Phi(G)$ is analogous to the proof of Theorem 1.4.4: use the test vector $y = \chi_S - \alpha\mathbf{1}$ with

$$\alpha = \frac{\mathbf{d}^\top \chi_S}{\mathbf{d}^\top \mathbf{1}} = \frac{d(S)}{d(V)},$$

chosen so that $y \perp \mathbf{d}$.

Proof of the upper bound. Fix a cut S and let $y = \chi_S - \alpha\mathbf{1}$ as above; $y \neq \mathbf{0}$ since S is a nonempty proper subset. Since $L\mathbf{1} = \mathbf{0}$, the numerator of (1.4.10) is $y^\top L y = \chi_S^\top L \chi_S = e(S)$. Using $1 - \alpha = d(\bar{S})/d(V)$, the denominator is

$$y^\top D y = \sum_{v \in V} d(v)(\chi_S(v) - \alpha)^2 = d(S)(1-\alpha)^2 + d(\bar{S})\,\alpha^2 = \frac{d(S)\,d(\bar{S})}{d(V)}.$$

So by (1.4.10) and (1.4.3),

$$\nu_2 \leq \frac{d(V)\,e(S)}{d(S)\,d(\bar{S})} \leq 2\Phi(S),$$

and minimizing over S gives $\nu_2 \leq 2\Phi(G)$. □

The lower bound $\Phi(G)^2/2 \leq \nu_2$ is the hard direction. Its proof occupies Section 1.4.10.

Remark 1.4.9 (Normalized versus unnormalized). Theorem 1.4.8 implies the unnormalized version, Theorem 1.4.5. For d-regular graphs this is immediate. Then $D = dI$, so $N = L/d$, $\nu_2 = \lambda_2/d$ and $\Phi(S) = \phi(S)/d$, and the two theorems say the same thing. For a general graph, comparing the Rayleigh quotients in (1.4.10) and in the Courant–Fischer formula for λ_2 gives $d_{\min}\nu_2 \leq \lambda_2 \leq d_{\max}\nu_2$. To see this, shift the optimal vector of one problem by a multiple of $\mathbf{1}$ to make it feasible for the other, as in Claim 1.4.11. Comparing denominators gives $\phi(S)/d_{\max} \leq \Phi(S) \leq \phi(S)/d_{\min}$. With Theorem 1.4.8 these give (1.4.8) only up to a factor $d_{\max}/d_{\min}$; for instance $\lambda_2 \geq d_{\min}\nu_2 \geq d_{\min}\Phi(G)^2/2 \geq (d_{\min}/d_{\max}) \cdot \phi(G)^2/(2d_{\max})$. The left half of (1.4.8) exactly as stated comes from the same proof as the normalized one, with every vertex weight $d(v)$ replaced by 1. That is, replace D by I, so that $y \perp \mathbf{1}$, and the two claims below read $\mathbb{E}[\min(|S_i|, |\bar{S}_i|)] = z^\top z$ and $\mathbb{E}[e(S_i)] \leq \sqrt{2d_{\max}\rho}\, z^\top z$. The factor $d_{\max}$ comes in through $\sum_u d(u)z(u)^2 \leq d_{\max} z^\top z$ in the proof of the second claim.

1.4.10 Proof of the hard direction

How to get a cut from v_2 and ν_2

In this subsection v_2 denotes an eigenvector of N (not of L) for the eigenvalue ν_2. Let $y \in \mathbb{R}^n$ be any vector with $y \perp \mathbf{d}$. We use y to get a map $V \to \mathbb{R}$, sending vertex u to $y(u)$, and cutting $\mathbb{R}$ at a point gives a partition of V. Formally:

- order the vertices so that $y_1 \leq y_2 \leq \cdots \leq y_n$;
- let $S = \{1, \ldots, k\}$ for some k;
- use this S for the cut.

Theorem 1.4.10. *Let $y \perp \mathbf{d}$, $y \neq \mathbf{0}$, and number the vertices so that $y_1 \leq y_2 \leq \cdots \leq y_n$. Then there is some i with $1 \leq i \leq n-1$ for which*

$$\Phi^2(\{1, \ldots, i\}) \leq \frac{2\,y^\top L y}{y^\top D y}, \qquad \textit{equivalently} \qquad \Phi(\{1, \ldots, i\}) \leq \sqrt{\frac{2\,y^\top L y}{y^\top D y}}.$$

This implies the lower bound of Cheeger's inequality for N: let $y = D^{-1/2}v_2$. Then $y \perp \mathbf{d}$ (because $v_2 \perp \mathbf{d}^{1/2}$), and by (1.4.9) the ratio $y^\top L y / y^\top D y$ equals $v_2^\top N v_2 / v_2^\top v_2 = \nu_2$. So $\Phi(G) \leq \Phi(\{1, \ldots, i\}) \leq \sqrt{2\nu_2}$, that is, $\Phi(G)^2/2 \leq \nu_2$.

But the theorem gives even more:

- it actually gives us the cut, not just a bound on $\Phi(G)$;
- it works even if we do not exactly have v_2 and ν_2: any $y \perp \mathbf{d}$ with a small ratio $y^\top Ly / y^\top Dy$ yields a cut of small conductance.

Preprocessing

Suppose $y^\top \mathbf{d} = 0$, $y \neq \mathbf{0}$, and let

$$\rho = \frac{y^\top Ly}{y^\top Dy}.$$

We will find a cut with $\Phi \leq \sqrt{2\rho}$. Reorder the coordinates so that $y_1 \leq \cdots \leq y_n$, and let j be the smallest value such that

$$\sum_{i=1}^{j} d_i \geq \frac{d(V)}{2}.$$

Define a new vector z by

$$z = y - y_j \mathbf{1}.$$

Now $z_j = 0$, (approximately) half of the total degree is to the left of 0, and (approximately) half is to the right.

Claim 1.4.11. $\dfrac{z^\top Lz}{z^\top Dz} \leq \rho.$

Proof. Since $L\mathbf{1} = \mathbf{0}$, the numerators agree: $z^\top Lz = y^\top Ly$. For the denominators, $y^\top D\mathbf{1} = y^\top \mathbf{d} = 0$, so for every $\alpha \in \mathbb{R}$

$$(y + \alpha\mathbf{1})^\top D(y + \alpha\mathbf{1}) = y^\top Dy + 2\alpha\, y^\top \mathbf{d} + \alpha^2\, \mathbf{1}^\top D\mathbf{1} = y^\top Dy + \alpha^2 d(V),$$

which is minimized when $\alpha = 0$. Taking $\alpha = -y_j$ gives $z^\top Dz \geq y^\top Dy$, and hence $z^\top Lz / z^\top Dz \leq y^\top Ly / y^\top Dy = \rho$. □

How should we choose our cut?

We will choose a set $S_i = \{1, \ldots, i\}$, but for what i? The idea is *randomized rounding*:

- choose i randomly from some distribution;
- show that

$$\mathbb{E}[e(S_i)] \leq \sqrt{2\rho}\, \mathbb{E}[\min(d(S_i), d(\bar{S}_i))]. \tag{1.4.11}$$

Why is this enough? (1.4.11) says that $\mathbb{E}[e(S_i) - \sqrt{2\rho}\, \min(d(S_i), d(\bar{S}_i))] \leq 0$, so for at least one value of i that occurs with positive probability, $e(S_i) - \sqrt{2\rho}\, \min(d(S_i), d(\bar{S}_i)) \leq 0$, that is, $\Phi(S_i) \leq \sqrt{2\rho}$. For the distribution below, $S_i = V$, where Φ is undefined, has probability zero, and S_i always contains vertex 1. So this S_i is a cut, with $1 \leq i \leq n-1$.

Here is the distribution. Let

$$w(u) = \operatorname{sgn}(z(u)) \cdot z(u)^2,$$

and then scale z by a positive constant so that $w(n) - w(1) = 1$. (Since $z(1) \le z_j = 0 \le z(n)$, this means $z(1)^2 + z(n)^2 = 1$. Scaling z changes neither the order of the vertices nor the ratio in Claim 1.4.11.) Choose t uniformly from $[w(1), w(n)]$, and choose i so that

$$S_i = \{u \mid w(u) \le t\}.$$

Since w is nondecreasing along our numbering of the vertices, this set is a prefix $\{1, \dots, i\}$: we place the vertices on the real line at the points $w(u)$ and cut the line at the random point t. Then (1.4.11) follows from the next two claims.

Claim 1.4.12. $\mathbb{E}\big[\min(d(S_i), d(\bar{S}_i))\big] = z^\top D z$.

Claim 1.4.13. $\mathbb{E}\big[e(S_i)\big] \le \sqrt{2\rho}\, z^\top D z$.

We prove both claims in Section 1.4.10, after an example; this completes the proof of Theorem 1.4.10 and hence of Cheeger's inequality for N.

Remark 1.4.14. The uniform choice of t is only needed to make the analysis work. In practice one simply sweeps over all $n-1$ prefix cuts $\{1, \dots, i\}$ and returns the best one, which is at least as good as the random one.

An example

Before the proofs, here is the construction on the twelve-vertex graph of Fig. 1.21. Each vertex is labeled with its value of $y = D^{-1/2}x$, where x is a unit eigenvector of N for ν_2. The vertices are numbered in increasing order of y, as in Table 1.1.

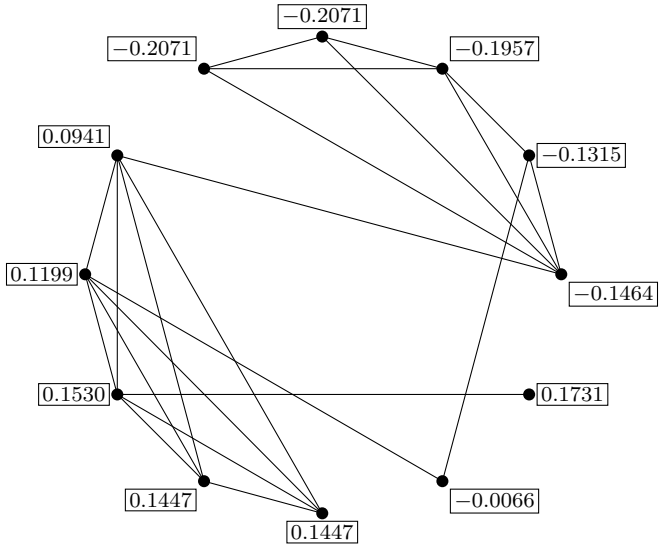


Figure 1.21: The example graph, with each vertex labeled by its value of $y = D^{-1/2}x$, where x is the eigenvector of N for ν_2. It has 22 edges and $d(V) = 44$.

Here $j = 7$, the vertex with $y_7 = 0.0941$. The running sums of the degrees are 3, 6, 10, 15, 18, 20, 25, ..., and the seventh is the first to reach $d(V)/2 = 22$. Then $z = y - 0.0941 \cdot \mathbf{1}$, and after squaring with signs and rescaling we get the vector w in the last column of Table 1.1; the distance from the leftmost to the rightmost point is $0.0644 - (-0.9356) = 1$. Figure 1.22

vertex i	y_i	d_i	$w(i)$
1	-0.2071	3	-0.9356
2	-0.2071	3	-0.9356
3	-0.1957	4	-0.8663
4	-0.1464	5	-0.5967
5	-0.1315	3	-0.5248
6	-0.0066	2	-0.1045
7	0.0941	5	0
8	0.1199	5	0.0069
9	0.1447	4	0.0264
10	0.1447	4	0.0264
11	0.1530	5	0.0358
12	0.1731	1	0.0644

Table 1.1: The vector y of Fig. 1.21 in sorted order, the degrees, and the rescaled vector w (with $j = 7$).

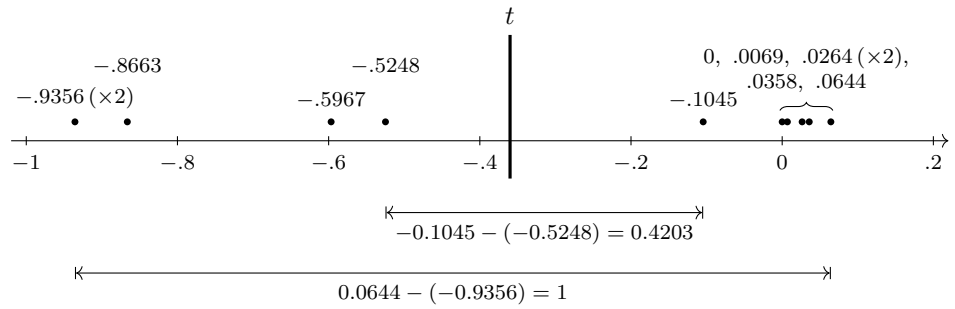


Figure 1.22: The vertices of Fig. 1.21 placed at their values of w, and one threshold t. The cut S_i consists of the vertices left of t. The edge between the vertices at -0.5248 and -0.1045 is cut exactly when t lands between them, which happens with probability 0.4203.

places the vertices on the line at their values of w. A threshold t is drawn uniformly from $[w(1), w(n)] = [-0.9356, 0.0644]$, and S_i is everything to the left of it.

For the threshold drawn in Fig. 1.22, S_i is the first five vertices: $e(S_i) = 2$ and $d(S_i) = 18 < d(\bar{S}_i) = 26$, so $\Phi(S_i) = 1/9$. The best of the eleven prefix cuts is $\{1, \dots, 6\}$, with $e = 2$ and $d = 20$, so $\Phi = 0.1$. This happens to be $\Phi(G)$. For this graph $\nu_2 \approx 0.116$, so Theorem 1.4.8 reads $0.005 \le 0.116 \le 0.2$ here.

Proof of the two claims

Proof of Claim 1.4.12. First, if $t < 0$ then $d(S_i)$ is the smaller of $d(S_i)$ and $d(\bar{S}_i)$, and if $t \ge 0$ then $d(\bar{S}_i)$ is the smaller one. If $t < 0$, every $u \in S_i$ has $w(u) \le t < 0$, so $z(u) < 0$, that is, $y_u < y_j$, and so $u \le j - 1$. By the choice of j, this gives $d(S_i) \le \sum_{\ell=1}^{j-1} d_\ell < d(V)/2$. If $t \ge 0$, every $u \in \bar{S}_i$ has $z(u) > 0$, so $u \ge j + 1$, and $d(\bar{S}_i) \le d(V) - \sum_{\ell=1}^{j} d_\ell \le d(V)/2$.

So vertex u lies on the smaller side exactly when t lies between $w(u)$ and 0: for $t < 0$ this means $w(u) \le t < 0$, and for $t \ge 0$ it means $0 \le t < w(u)$. This interval lies inside $[w(1), w(n)]$,

since $w(1) \le 0 \le w(n)$. Since t is uniform on an interval of length 1, the probability of this event is $|w(u)| = z(u)^2$. Therefore

$$\mathbb{E}\Big[\min\big(d(S_i), d(\bar{S}_i)\big)\Big] = \sum_u \Pr[u \text{ is on the smaller side}] \cdot d(u) = \sum_u z(u)^2 d(u) = z^\top D z. \qquad \square$$

Proof of Claim 1.4.13. An edge (u, v) is cut exactly when t lies between $w(u)$ and $w(v)$, so the probability that it is cut equals its length

$$|w(u) - w(v)| = \begin{cases} |z(u)^2 - z(v)^2| & \text{if } \operatorname{sgn} z(u) = \operatorname{sgn} z(v), \\ z(u)^2 + z(v)^2 & \text{if } \operatorname{sgn} z(u) \neq \operatorname{sgn} z(v), \end{cases}$$

and in both cases

$$|w(u) - w(v)| \le |z(u) - z(v)| \cdot \big(|z(u)| + |z(v)|\big). \tag{1.4.12}$$

If the signs agree, $|z(u)^2 - z(v)^2| = |z(u) - z(v)|\,|z(u) + z(v)|$ and $|z(u) + z(v)| = |z(u)| + |z(v)|$. If they differ, $|z(u) - z(v)| = |z(u)| + |z(v)|$ and $z(u)^2 + z(v)^2 \le \big(|z(u)| + |z(v)|\big)^2$. Summing over the edges and applying the Cauchy–Schwarz inequality,

$$\begin{aligned} \mathbb{E}\big[e(S_i)\big] &= \sum_{(u,v)\in E} |w(u) - w(v)| \le \sum_{(u,v)\in E} |z(u) - z(v)| \cdot \big(|z(u)| + |z(v)|\big) \\ &\le \sqrt{\sum_{(u,v)\in E} \big(z(u) - z(v)\big)^2} \sqrt{\sum_{(u,v)\in E} \big(|z(u)| + |z(v)|\big)^2} \\ &\le \sqrt{\rho\, (z^\top D z)} \sqrt{\sum_{u\in V} 2 z(u)^2 d(u)} = \sqrt{2\rho}\; z^\top D z. \end{aligned}$$

In the last line, the first factor uses $\sum_{(u,v)\in E} (z(u) - z(v))^2 = z^\top L z \le \rho\, z^\top D z$, which is Claim 1.4.11. The second uses $(|a| + |b|)^2 \le 2a^2 + 2b^2$, and the fact that each vertex u lies on $d(u)$ edges, so summing $2z(u)^2 + 2z(v)^2$ over the edges gives $\sum_u 2 z(u)^2 d(u)$. $\square$

Why the claims are enough. By the two claims,

$$\mathbb{E}\Big[e(S_i) - \sqrt{2\rho}\, \min\big(d(S_i), d(\bar{S}_i)\big)\Big] \le \sqrt{2\rho}\, z^\top D z - \sqrt{2\rho}\, z^\top D z = 0,$$

so some value of t makes the quantity inside the expectation at most 0. We may take $t < w(n)$, since $t = w(n)$ has probability zero. Then S_i contains vertex 1 but not vertex n, so it is a nonempty proper subset. Since G has no isolated vertices, $\min(d(S_i), d(\bar{S}_i)) > 0$, and dividing gives $\Phi(S_i) \le \sqrt{2\rho}$. This proves Theorem 1.4.10, and with it the lower bound of Theorem 1.4.8.

Remark 1.4.15. As noted in Section 1.4.10, the proof does more than bound ν_2: it is an algorithm. Sort the vertices by y and try the $n - 1$ prefix cuts $\{1, \dots, i\}$; one of them has conductance at most $\sqrt{2\rho}$. And it works for any $y \perp \mathbf{d}$, so an approximate eigenvector with a small Rayleigh quotient is just as good.

1.5 Random Walks and Mixing

This section turns to random walks on graphs. We set up the walk matrix and the stationary distribution, see that a plain random walk need not converge, and fix this with the lazy walk. We

then translate the facts we know about Laplacian eigenvalues into facts about the eigenvalues of the walk, and prove that the lazy walk approaches its stationary distribution at a rate governed by the spectral gap. Through Cheeger's inequality that rate is controlled by the conductance of the graph. These bounds are what make sampling by random walks work in Section 1.6.

1.5.1 The walk matrix and the stationary distribution

Let $G = (V, E)$ be an undirected graph. A *random walk* on G starts at some vertex v and repeats the following step: move to a uniformly random neighbor of the current vertex (Fig. 1.23, left).

Probability vectors. Let $p_t(v)$ denote the probability that the walk is at vertex v at time t. We think of p_t as a vector in $\mathbb{R}^n$, with $\sum_{v \in V} p_t(v) = 1$. If the walk is at vertex u at time t, then at time $t+1$ it is at each neighbor v of u with probability $1/d(u)$. So

$$p_{t+1}(v) = \sum_u \Pr[\text{at } u \text{ at time } t] \cdot \Pr[\text{go from } u \text{ to } v \text{ at time } t+1] = \sum_{(u,v) \in E} p_t(u) \cdot \frac{1}{d(u)}.$$

We can write this with a matrix $W = W_G$:

$$[W_G]_{i,j} = \begin{cases} \dfrac{1}{d(j)} & (i,j) \in E, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{that is,} \qquad W_G = AD^{-1},$$

and then $p_{t+1} = W p_t$. Note the unfortunate ordering: $[W_G]_{i,j}$ is the probability of going from j to i.

The stationary distribution. Let

$$\pi(u) = \frac{d(u)}{\sum_{v \in V} d(v)}.$$

Claim 1.5.1. *π is a probability vector.*

Proof. Its coordinates are positive, and they sum to $\sum_u d(u) / \sum_v d(v) = 1$. □

Claim 1.5.2. *$W\pi = \pi$.*

Proof. For a vertex k, the entry W_{ki} is $1/d(i)$ when $(i,k) \in E$ and 0 otherwise, so

$$[W\pi](k) = \sum_i W_{ki}\, \pi(i) = \frac{1}{\sum_{v \in V} d(v)} \sum_{(i,k) \in E} \frac{1}{d(i)}\, d(i) = \frac{1}{\sum_{v \in V} d(v)}\, d(k) = \pi(k).$$ □

So if the walk has distribution π at step t, it has the same distribution at step $t+1$. This means that π is an eigenvector of W with eigenvalue 1. Since π is a multiple of the degree vector $\mathbf{d} = D\mathbf{1}$, we could also take $\mathbf{d}$ as the eigenvector.

1.5.2 Periodicity and the lazy walk

A natural guess is that if we walk around long enough, we eventually end up arbitrarily close to the stationary distribution, whatever the starting vertex. This is not true. Start the walk on the 6-cycle at a single vertex (Fig. 1.23, right). The graph is bipartite, and if the walk is on the left side at time 0, it is on the left side at every even time and on the right side at every odd time. So the distribution can never converge to the stationary distribution.

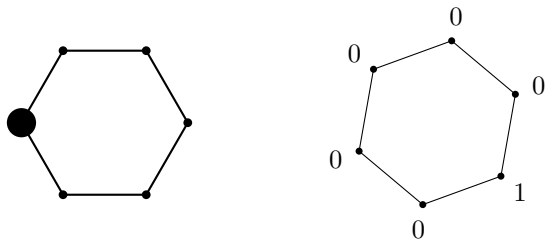


Figure 1.23: Left: a random walk on the 6-cycle; the large dot is the current vertex. Right: the starting distribution p_0, with all of its mass on one vertex. Because the 6-cycle is bipartite, p_t alternates between the two sides and does not converge.

The easy fix: a lazy random walk. At each time t:

- with probability $1/2$, take a step of the random walk;
- with probability $1/2$, stay where you are.

One can show that this breaks the periodicity. The lazy walk has the transition matrix

$$W' = \frac{W + I}{2} = \frac{I + AD^{-1}}{2}.$$

1.5.3 The normalized walk matrices

The matrices $W = AD^{-1}$ and $W' = (I + AD^{-1})/2$ are not symmetric, which could make things complicated. So we define symmetric versions. The *normalized walk matrix* is

$$W_N = D^{-1/2} W D^{1/2} = D^{-1/2} A D^{-1/2},$$

and the *normalized lazy walk matrix* is

$$W'_N = D^{-1/2} W' D^{1/2} = \frac{I + D^{-1/2} A D^{-1/2}}{2}.$$

Claim 1.5.3. *W_N and W have the same eigenvalues, and related eigenvectors: v is an eigenvector of W_N with eigenvalue θ if and only if $D^{1/2}v$ is an eigenvector of W with eigenvalue θ.*

Proof. Suppose $W_N v = \theta v$ and let $q = D^{1/2}v$. Then

$$\theta v = W_N v = D^{-1/2} W D^{1/2} v = D^{-1/2} W q.$$

Multiplying on the left by $D^{1/2}$ gives $Wq = \theta D^{1/2} v = \theta q$, so q is an eigenvector of W with eigenvalue θ. The same computation run backwards, starting from $Wq = \theta q$ and $v = D^{-1/2}q$, gives the converse. □

Since W has the eigenvector $\mathbf{d} = D\mathbf{1}$ with eigenvalue 1 (Claim 1.5.2), the normalized walk matrix has the eigenvector $D^{-1/2}\mathbf{d} = D^{1/2}\mathbf{1}$ with eigenvalue 1. Likewise for W'_N and W'.

Claim 1.5.4. $W_N = I - N$.

Proof. $W_N = D^{-1/2} A D^{-1/2} = D^{-1/2}(D - L)D^{-1/2} = I - D^{-1/2} L D^{-1/2} = I - N$. □

So the eigenvalues of W_N are $1 -$ (eigenvalues of N), and it makes sense to order them the opposite way:

$$1 = \mu_1 \geq \mu_2 \geq \cdots \geq \mu_n, \qquad \mu_i = 1 - \nu_i.$$

We can now translate theorems about Laplacians into theorems about these μ_i.

Proposition 1.5.5. *Let μ_i be the eigenvalues of W_N and μ'_i the ith eigenvalue of W'_N, both in decreasing order.*

(a) All μ_i lie in $[-1, 1]$.

(b) If G is connected then $\mu_2 < 1$. This implies that the stationary distribution is unique.

(c) The eigenvalue -1 occurs only for bipartite graphs (exercise).

(d) All μ'_i lie in $[0, 1]$, and if G is connected then $\mu'_2 < 1$.

Proof. (a) Since $\mu_i = 1 - \nu_i$, this says $0 \leq \nu_i \leq 2$. With $x = D^{1/2}y$, the Rayleigh quotient of N is $\sum_{(u,v)\in E}(y(u) - y(v))^2 / \sum_u d(u)y(u)^2$. This is at least 0, and it is at most 2 since $(a-b)^2 \leq 2a^2 + 2b^2$. (b) If G is connected, the kernel of L is spanned by $\mathbf{1}$ (Claim 1.1.13), so the kernel of $N = D^{-1/2}LD^{-1/2}$ is spanned by $D^{1/2}\mathbf{1}$. So $\nu_2 > 0$ and $\mu_2 < 1$. Then the eigenvalue 1 of W_N has a one-dimensional eigenspace, and by Claim 1.5.3 so does the eigenvalue 1 of W. This eigenspace contains only one probability vector. (d) $W'_N = (I + W_N)/2$, so $\mu'_i = (1 + \mu_i)/2$, and (a) and (b) give the claims. Part (c) is left as an exercise. □

Exercise 1.5.6. Show that if -1 is an eigenvalue of W_N, then G is bipartite (if G is not connected: some connected component of G is bipartite).

1.5.4 ℓ_2 convergence

Let the *spectral gap* be

$$\gamma = 1 - \mu'_2$$

(we write γ, since λ is reserved for Laplacian eigenvalues). Define the ℓ_2 distance between probability distributions p and q by

$$\|p - q\|_2 = \sqrt{\sum_i (p_i - q_i)^2}.$$

Theorem 1.5.7. *Let p_0 be an initial probability distribution, and let $p_t = (W')^t p_0$ be the distribution after t steps of the lazy random walk. Then*

$$\|p_t - \pi\|_2 \leq (1 - \gamma)^t \sqrt{\frac{\max_x d(x)}{\min_y d(y)}}.$$

We prove Theorem 1.5.7 when G is regular; the irregular case is similar and is left as an exercise (Exercise 1.5.9).

Proof for regular G. If G is d-regular then $\pi = \frac{1}{n}\mathbf{1}$, and the degree term $\sqrt{\max_x d(x)/\min_y d(y)}$ equals 1, so we must show $\|p_t - \pi\|_2 \leq (1 - \gamma)^t$. Since $D = dI$, the matrix $W' = (I + A/d)/2 =$

W'_N is symmetric, so it has an orthonormal basis of eigenvectors $v_1, \ldots, v_n$ with eigenvalues $1 = \mu'_1 \geq \mu'_2 \geq \cdots \geq \mu'_n \geq 0$, and $v_1 = \frac{1}{\sqrt{n}}\mathbf{1}$.

Write p_0 in the eigenbasis of W':

$$p_0 = \sum_{i=1}^{n} c_i v_i = c_1 v_1 + \sum_{i=2}^{n} c_i v_i, \qquad \text{where } c_i = p_0^\top v_i.$$

Then

$$(W')^t p_0 = c_1 v_1 + \sum_{i=2}^{n} c_i \, \mu_i'^t \, v_i.$$

Here $c_1 v_1 = (\frac{1}{\sqrt{n}}\mathbf{1}^\top p_0)\frac{1}{\sqrt{n}}\mathbf{1} = \frac{1}{n}\mathbf{1} = \pi$, because the entries of p_0 sum to 1. So, using orthonormality of the v_i,

$$\|p_t - \pi\|_2 = \Big\|\sum_{i=2}^{n} c_i \, \mu_i'^t \, v_i\Big\|_2 = \sqrt{\sum_{i=2}^{n} c_i^2 \, \mu_i'^{2t}}.$$

For $i \geq 2$ we have $0 \leq \mu'_i \leq \mu'_2$, so

$$\|p_t - \pi\|_2 \leq \mu_2'^t \sqrt{\sum_{i=2}^{n} c_i^2} \leq \mu_2'^t \sqrt{\sum_{i=1}^{n} (v_i^\top p_0)^2} \leq \mu_2'^t = (1-\gamma)^t.$$

The last inequality uses $\sum_{i=1}^{n} (v_i^\top p_0)^2 = \|p_0\|_2^2 \leq \|p_0\|_1^2 = 1$. □

1.5.5 ℓ_∞ convergence

Theorem 1.5.8. *For any vertex v,*

$$|p_t(v) - \pi(v)| \leq (1-\gamma)^t \sqrt{\frac{d(v)}{\min_y d(y)}}.$$

The proof is pretty much identical to that of Theorem 1.5.7; we leave it, and the irregular case of Theorem 1.5.7, as an exercise.

Exercise 1.5.9. Prove Theorem 1.5.8, and prove Theorem 1.5.7 for graphs that are not regular.

1.5.6 Conductance controls mixing

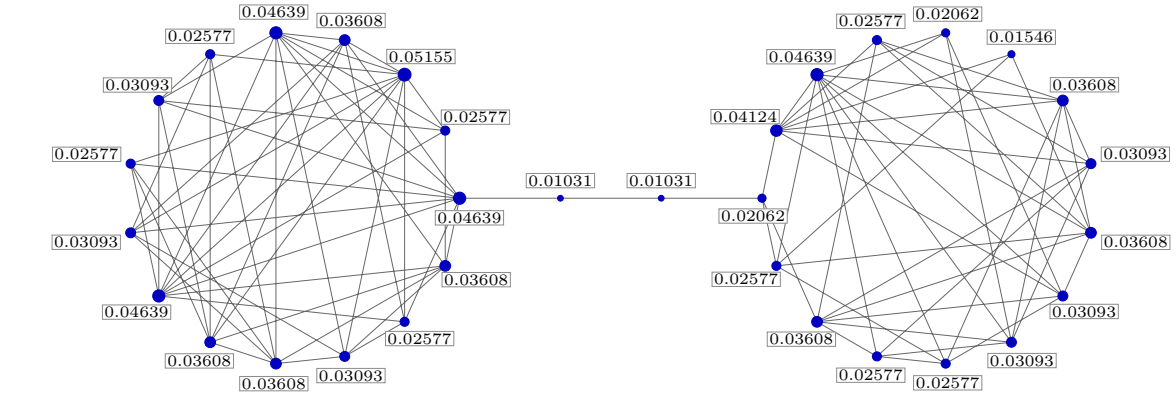
Cheeger's inequality for the normalized Laplacian (Theorem 1.4.8) says $\Phi(G)^2/2 \leq \nu_2 \leq 2\Phi(G)$, where ν_2 is the second smallest eigenvalue of N. Since $\mu'_2 = (1+\mu_2)/2$ and $\mu_2 = 1 - \nu_2$, we have $\nu_2 = 2(1-\mu'_2) = 2\gamma$. So in terms of the walk, Cheeger's inequality becomes

$$\frac{\Phi(G)^2}{2} \leq 2(1-\mu'_2) \leq 2\Phi(G), \qquad \text{that is,} \qquad \frac{\Phi(G)^2}{4} \leq \gamma \leq \Phi(G). \tag{1.5.1}$$

So $\Phi(G)$ is related to the rate of convergence: by Theorem 1.5.7 and Theorem 1.5.8, the distance to π shrinks by a factor $1 - \gamma \leq 1 - \Phi(G)^2/4$ per step. Proving lower bounds on Φ lets us prove that a walk mixes quickly, and this is often *much* easier than computing eigenvalues. For example, one can show this way that random walks on unweighted undirected graphs mix in polynomial time. If G is connected with m edges, every nonempty proper S has $e(S) \geq 1$ and $\min(d(S), d(\bar{S})) \leq d(V)/2 = m$, so $\Phi(G) \geq 1/m$ and $\gamma \geq 1/(4m^2)$. By Theorem 1.5.8, $|p_t(v) - \pi(v)| \leq e^{-\gamma t}\sqrt{n}$, so $O(m^2 \log(n/\epsilon))$ steps bring every entry within ϵ of π.

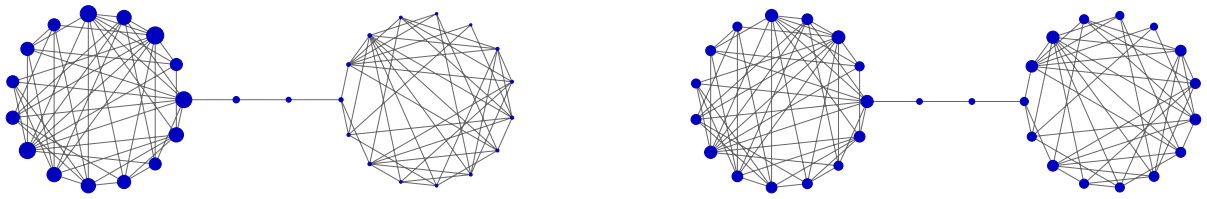
Why should cuts be related to mixing? Pictures help (Fig. 1.24).

(a) Stationary distribution $\pi(v) = d(v)/2m$



(b) The walk at $t = 20$

Stationary distribution, drawn the same way



(c) The walk at $t = 90$, labeled with $p_t(v)/\pi(v)$

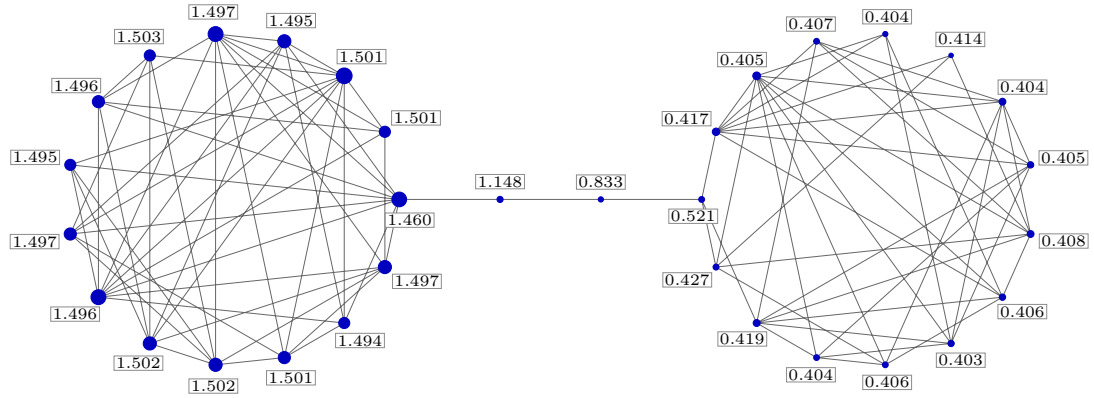


Figure 1.24: Three snapshots of a random walk on a graph with one very sparse cut. The graph consists of two dense, randomly wired clusters of fifteen vertices each, drawn as circles, joined by a path of three edges through two middle vertices of degree 2. (a) Each vertex is labeled with its stationary probability, which is proportional to its degree (values from about 0.015 to 0.052; the two path vertices have 0.01031). (b) The walk started inside the left cluster, at $t = 20$, drawn with dot areas proportional to $p_t(v)$, next to the stationary distribution drawn the same way: almost all of the mass (about 94%) is still in the left cluster. (c) The same walk at $t = 90$, with each vertex labeled by $p_t(v)/\pi(v)$: about 1.49–1.50 throughout the left cluster (1.46 at the vertex attached to the path), 1.15 and 0.83 on the two path vertices, and 0.40–0.43 throughout the right cluster (0.52 at the vertex attached to the path). Within each cluster the walk has already equilibrated, but mass leaks across the sparse cut slowly. The walk is the simple walk $p_t = W^t p_0$ with $W = AD^{-1}$, started at one vertex; the lazy walk gives the same pictures at times 40 and 180.

1.6 Approximate Counting

Many counting problems are hard to solve exactly but can be solved approximately by sampling: draw random elements of a set that contains the objects we want to count, and use the fraction of samples that are such objects. This section develops the idea. The Chernoff bound says how many samples we need, and it also shows when the naive method breaks down: when the objects are an exponentially small fraction of what we can sample from, or when we cannot sample uniformly at all. For the solutions of a DNF formula, a better choice of the set to sample from fixes the first problem. For the permanent of a dense 0-1 matrix, that is, the number of perfect matchings of a dense bipartite graph, we need more: we count in stages, and we sample with a random walk whose mixing we control through its conductance, as in Section 1.5.

1.6.1 Monte Carlo methods

Everyone's favorite example. Compute the number $\pi \approx 3.14$ by throwing darts at a dartboard (Fig. 1.25, left). If you pick a point uniformly at random in the square $[-1,1]^2$, the probability that it lands in the unit circle is

$$\frac{\text{area(circle)}}{\text{area(square)}} = \frac{\pi}{4}.$$

Suppose you pick N points in $[-1,1]^2$. (In this section N is a number of samples, not the normalized Laplacian.) Then $\mathbb{E}[\text{number in unit circle}] = N\pi/4$, so we return

$$\pi \approx \frac{4 \cdot (\#\text{points in circle})}{N}.$$

Question: how close will this be to the right answer?

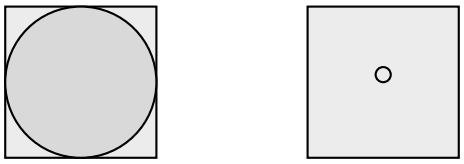


Figure 1.25: Left: the dartboard for estimating π. Right: a big dartboard with a small circle, where almost every dart misses.

Chernoff bounds

Suppose we have a random variable $r \in \{0,1\}$ with $\Pr(r=0) = 1-p$ and $\Pr(r=1) = p$. Draw N independent samples $r_1, \ldots, r_N$ and let $R = \sum_i r_i$. Then

$$\mathbb{E}[R] = \mathbb{E}\Big[\sum_i r_i\Big] = \sum_i \mathbb{E}[r_i] = Np.$$

We say that R *ϵ-approximates* $\mathbb{E}[R]$ if

$$(1-\epsilon)\mathbb{E}[R] \le R \le (1+\epsilon)\mathbb{E}[R].$$

So this is a multiplicative error measure.

Theorem 1.6.1 (one version of the Chernoff bound)**.** *The probability that R fails to ϵ-approximate $\mathbb{E}[R]$ obeys the bound*

$$\Pr(|R - \mathbb{E}[R]| \geq \epsilon \mathbb{E}[R]) \leq 2e^{-\Theta(1)\, Np\epsilon^2} = 2e^{-\Theta(1)\, \mathbb{E}[R]\, \epsilon^2}.$$

(Proof omitted.) Here $\Theta(1)$ stands for an absolute constant. For $0 < \epsilon \leq 1$ one can take it to be $1/3$.

Some notes about this bound:

- It is pretty near tight.
- It needs independent trials; without independence it is no longer true.
- The error is multiplicative, not additive.
- For fixed ϵ, the failure probability falls off exponentially in the number of trials. So if some number N of trials gives failure probability 0.5, then Nk trials give failure probability $\frac{1}{2^k}$.
- The bound $\Pr(|R - \mathbb{E}[R]| \geq \epsilon \mathbb{E}[R]) \leq 2e^{-\Theta(1)\mathbb{E}[R]\epsilon^2}$ remains true if we only require the variables to lie in $[0, 1]$ (or in any fixed interval $[a, b]$).

Smaller ϵ requires more trials. To get an ϵ-approximation with probability $1 - \delta$, we need

$$N \geq \Theta\Big(\frac{\log(1/\delta)}{p\, \epsilon^2}\Big). \tag{1.6.1}$$

In words: we need enough trials to get $\Theta(\log(1/\delta)/\epsilon^2)$ successes in expectation. It is easy to make δ smaller and harder to make ϵ smaller.

When do we run into trouble?

- *If we are bad at darts:* a big dartboard with a small circle (Fig. 1.25, right). If $p =$ 1/exponential, we need exponentially many trials to expect a constant number of successes.
- *If we have trouble throwing darts at all:* that is, if it is hard to draw samples uniformly from the ambient space.

We will run into both problems, and develop some techniques for (sometimes) fixing them.

A little terminology.

Definition 1.6.2. An (ϵ, δ) *approximation scheme* for some quantity is an algorithm that computes an ϵ-approximation of it with probability at least $1 - \delta$. A *fully polynomial randomized approximation scheme (FPRAS)* is an (ϵ, δ) approximation scheme that runs in time $\text{poly}(n, \frac{1}{\epsilon}, \log \frac{1}{\delta})$, where n is the size of the input.

1.6.2 Counting DNF solutions

We have Boolean variables $x_1, \ldots, x_n$. A *literal* is an x_i or its negation $\bar{x}_i$. Recall that a *DNF formula* F is a disjunction (OR) of conjunctive (AND) clauses,

$$F = C_1 \vee \cdots \vee C_m,$$

where each C_i is an AND of literals; for example,

$$F = (x_1 \wedge \bar{x}_3) \vee (x_2) \vee (x_2 \wedge \bar{x}_1) \vee (x_3 \wedge x_4).$$

It is easy to check whether a satisfying assignment exists. A clause can be satisfied unless it contains both a variable and its negation, and F is satisfiable if and only if one of its clauses is. It is really hard to count them exactly: the problem is #P-hard, so at least as hard as NP-complete problems. We will see how to get, in polynomial time, an ϵ-approximation with probability greater than $1 - \frac{1}{2^n}$, that is, an FPRAS.

How not to do it. The naive Monte Carlo algorithm repeats the following trial N times: assign 0 or 1 to each x_i uniformly at random, and check whether this is a satisfying assignment. If it gets k successes, it estimates

$$\#\text{solutions} \approx \frac{k}{N}\, 2^n.$$

Why not? Because $\#\text{solutions}/2^n$ *can be exponentially small* (for example, $F = x_1 \wedge x_2 \wedge \cdots \wedge x_n$ has one solution), and then by (1.6.1) we need exponentially many trials.

How to fix the problem. Make a table whose rows are the 2^n assignments and whose columns are the m clauses, with a 1 in row a, column i when assignment a satisfies clause C_i (Fig. 1.26, left). The number of solutions is the number of rows that contain a 1. In each such row, color the first 1 yellow, and color every other 1 blue (in Fig. 1.26 the blue entries are rewritten as 0, to show that they are not counted). We want to count the yellow boxes, since there is exactly one per satisfying assignment.

We will:

- sample uniformly from $X = \{\text{blue}\} \cup \{\text{yellow}\}$, the set of all (assignment, clause) pairs where the assignment satisfies the clause;
- count the fraction of samples that are yellow;
- multiply by $|X|$.

Now the success probability is at least $\frac{1}{m}$, since each row has at most m boxes in X and exactly one of them is yellow (if any are in X).

To sample, go by column instead of by row. If clause i has k_i variables in it, its column has 2^{n-k_i} satisfying assignments, and we know what they are: the k_i variables are forced, and the others are free. A clause containing both a variable and its negation has no satisfying assignments, and we can delete it first. Hence

$$|X| = \sum_i 2^{n-k_i}.$$

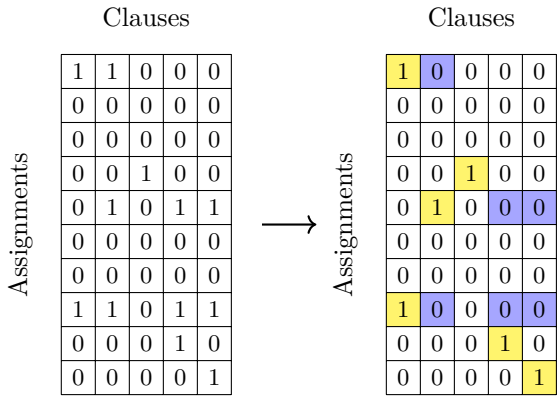


Figure 1.26: The assignment-by-clause table (a few of the 2^n rows). Left: a 1 marks an assignment that satisfies a clause. Right: in each row the first 1 is yellow; the other 1s are blue and rewritten as 0. The yellow boxes are in one-to-one correspondence with the satisfying assignments.

So sample each column i with probability proportional to 2^{n-k_i}, then pick a uniformly random satisfying assignment of this clause, and check whether clause i is the first clause this assignment satisfies (that is, whether the box is yellow).

The resulting pair is uniform on X. So the expected fraction of yellow samples is the number of solutions divided by $|X|$, which is at least $1/m$. Multiplying the observed fraction by $|X|$ gives an unbiased estimate of the number of solutions. By (1.6.1) with $p \geq 1/m$, it is enough to take $N = \Theta(m \log(1/\delta)/\epsilon^2)$ samples. With $\delta = 2^{-n}$ this is $\Theta(mn/\epsilon^2)$ samples, each taking polynomial time. So this is an FPRAS.

1.6.3 The permanent

The permanent. The determinant of an $n \times n$ matrix $M = (m_{ij})$ can easily be computed in polynomial time:

$$\det(M) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} m_{i,\sigma(i)}.$$

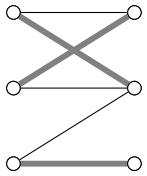
We get the *permanent* by dropping the signs:

$$\operatorname{per}(M) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} m_{i,\sigma(i)}.$$

The problem becomes much harder: computing the permanent is #P-complete, so there is no polynomial-time algorithm unless P = NP.

0-1 permanents and matchings. We are going to look at matrices M all of whose entries are 0 or 1; this is still #P-hard. Such a permanent has an interpretation as the number of perfect matchings in a bipartite graph: put the rows on one side and the columns on the other, with an edge from row i to column j when $m_{ij} = 1$ (Fig. 1.27). A permutation σ contributes 1 to $\operatorname{per}(M)$ exactly when all the edges $(i, \sigma(i))$ are present, that is, when it describes a perfect matching, and 0 otherwise. This is the quantity we will try to approximate.

Surprising result: although it is #P-hard to compute the permanent of a 0-1 matrix exactly, it has an FPRAS.



$$\begin{pmatrix} 1 & \boxed{1} & 0 \\ \boxed{1} & 1 & 0 \\ 0 & 1 & \boxed{1} \end{pmatrix}$$

Figure 1.27: A 0-1 matrix and its bipartite graph (rows on the left, columns on the right). The thick edges form a perfect matching, which corresponds to the boxed entries; this matrix has permanent 2.

A (very) little history.

- 1989: Jerrum and Sinclair [JS89] gave an FPRAS for the permanent of *dense* graphs, those in which every vertex has degree at least $n/2$. This case is still #P-complete.
- 2001: Jerrum, Sinclair and Vigoda [JSV01] gave one for arbitrary graphs, and thus for any matrix with nonnegative entries.

We will just do the dense case.

What won't work. No naive Monte Carlo algorithm will do: for example, guess a random permutation and see whether it gives a matching. Anything like this will require estimating exponentially small quantities, which by Section 1.6.1 needs exponentially many trials. Instead we count in stages, writing the answer as a product of ratios that are each only polynomially large, so that each ratio can be estimated by sampling. To sample, we run a random walk on the set of matchings and show that it mixes in polynomially many steps by bounding its conductance (Section 1.5.6). The conductance bound uses a general technique, *canonical paths*.

1.6.4 Counting matchings in stages

Setting. From here on $G = (V, E)$ is a bipartite graph with n vertices on each side, so that a perfect matching has n edges, and $|E|$ is its number of edges. "Dense" means that every vertex of G has degree at least $n/2$. A matching is denoted by m (it is a set of edges of G, not a number of edges), and u, v, w are vertices of G.

We will look at all (possibly partial) matchings, not just perfect ones.

Definition 1.6.3. For $0 \le k \le n$, let M_k be the set of matchings of G of size k (that is, with exactly k edges), and for $1 \le k \le n$ let

$$r_k = \frac{|M_k|}{|M_{k-1}|}.$$

So $|M_n| = \mathrm{per}(M)$ is the number we want, and $|M_1| = |E|$ is easy to compute. The plan is as follows.

- Suppose we had a black box for sampling from $M_k \cup M_{k-1}$ uniformly at random, for any k.
- Suppose also that $1/\alpha \le r_k \le \alpha$ for some α of polynomial size.
- Then we can estimate each r_k within relative error $1 + 1/n^2$ with very high probability.

- Estimate the answer using the telescoping product

$$|M_n| = |M_1| \cdot \prod_{i=2}^{n} r_i. \tag{1.6.2}$$

Why does the third step work? A uniform sample from $M_k \cup M_{k-1}$ lies in M_k with probability

$$q_k = \frac{|M_k|}{|M_k| + |M_{k-1}|} = \frac{r_k}{1 + r_k},$$

and both q_k and $1 - q_k = 1/(1 + r_k)$ are at least $1/(1 + \alpha)$. So by the Chernoff bound (Theorem 1.6.1), polynomially many samples estimate q_k and $1 - q_k$ to within any inverse-polynomial relative error, with exponentially small failure probability. Then $r_k = q_k/(1 - q_k)$. If each of the $n - 1$ ratios in (1.6.2) is within a factor $1 + 1/n^2$ of the truth, the product is within a factor $(1 + 1/n^2)^n \le e^{1/n} = 1 + O(1/n)$. Asking for relative error $1 + \epsilon/(2n)$ per ratio instead gives a factor at most $e^{\epsilon/2} \le 1 + \epsilon$ for $\epsilon \le 1$, which is what an FPRAS needs.

Technical issues

The main question is how to sample from $M_k \cup M_{k-1}$. There are also some other issues.

- We will actually only sample *approximately* uniformly. This is not a big deal, as long as we are close enough.
- All of our approximations will have some probability of failure. We just make all of these numbers really small. There are only polynomially many of them, so by the union bound the probability that any one fails stays small.
- We do not know that there is a polynomial bound on the r_k. This will be guaranteed by our density assumption (Section 1.6.5).

1.6.5 Bounding the ratios

Theorem 1.6.4. *Let G be a bipartite graph with minimum degree at least $n/2$. Then*

$$\frac{1}{n^2} \le r_k \le n^2 \qquad \text{for all } k.$$

Proof of the upper bound. For every matching in M_k, pick (arbitrarily) a submatching in M_{k-1} as its representative, say by deleting one of its edges. No matching $m' \in M_{k-1}$ can be picked more than $(n - k + 1)^2 \le n^2$ times. This is because a matching that has m' as its representative consists of m' plus one new edge, and that edge has to join two vertices left unmatched by m', one on each side; m' leaves $n - k + 1$ vertices unmatched on each side. Hence $|M_k| \le n^2 |M_{k-1}|$. □

The lower bound follows from the next lemma. Recall that an *augmenting path* for a matching m' is a path whose two end vertices are unmatched by m' and whose edges alternate between edges not in m' and edges in m'; exchanging the two kinds of edges along it gives a matching with one more edge.

Lemma 1.6.5. *In a bipartite graph of minimum degree at least $n/2$, every partial matching in M_{k-1} ($k \le n$) has an augmenting path of length at most 3. That is, it can be augmented by either*

- *adding an edge to the existing matching, or*
- *removing one edge and then adding two more, one hitting each endpoint of the removed edge.*

We leave the proof as an exercise.

Exercise 1.6.6. Prove Lemma 1.6.5.

Proof of the lower bound of Theorem 1.6.4. Map every $m' \in M_{k-1}$ to the matching in M_k obtained by augmenting m' along one (fixed) shortest augmenting path, which by Lemma 1.6.5 has length 1 or 3. Fix some $m \in M_k$; we count the matchings mapped to it.

- At most k matchings in M_{k-1} can be augmented by a path of length 1 to equal m. (Such an m' is m minus one of its k edges.)
- At most $k(k-1)$ can be augmented by a path of length 3. If m' has the augmenting path a, b, c, d, with $(b, c) \in m'$, then $m = m' \setminus \{(b, c)\} \cup \{(a, b), (c, d)\}$. So we recover m' from the ordered pair of edges $((a, b), (c, d))$ of m by deleting both and adding (b, c), and there are at most $k(k-1)$ such pairs.

So

$$|M_{k-1}| \leq (k + k(k-1))|M_k| \leq n^2|M_k|,$$

that is, $r_k \geq 1/n^2$. □

1.6.6 Sampling by a random walk

All that is left is to show how to sample from

$$C_k = M_k \cup M_{k-1}.$$

We will do this by running a random walk on C_k whose stationary distribution is uniform, and showing that it mixes quickly. By the convergence theorems (Theorems 1.5.7 and 1.5.8) and Cheeger's inequality in the form (1.5.1), it is enough to bound the conductance $\Phi(C_k)$ from below. We will do so with a general technique known as *canonical paths*.

Canonical paths

Let $H = (U, F)$ be any graph for which we want to bound $\Phi(H)$. (We will apply this to the exponentially large graph C_k.) For every two vertices $v, w \in U$ we specify a *canonical path* in H from v to w.

Claim 1.6.7. *Suppose no edge of H occurs on more than $b|U|$ canonical paths. Then*

$$\Phi(H) \geq \frac{1}{2b\, d_{\max}},$$

where $d_{\max}$ is the maximum degree of H.

Proof. Recall that

$$\Phi(H) = \min_{S \subseteq U} \frac{e(S)}{\min\{\sum_{v \in S} d(v),\ \sum_{v \in \bar{S}} d(v)\}}.$$

Since S and $\bar{S}$ give the same ratio, we may assume $|S| \le |U|/2$. For any such $S \subseteq U$, the number of canonical paths crossing the cut is

$$|S|(|U| - |S|) \ge |S| \cdot \frac{|U|}{2}.$$

These are the paths from a vertex of S to a vertex of $\bar{S}$, and each of them uses at least one edge leaving S. At most $b|U|$ paths go through each edge, so the number of edges crossing the cut is

$$e(S) \ge \frac{|S|\,|U|/2}{b|U|} = \frac{|S|}{2b}.$$

We bound the denominator using the maximum degree: it is at most $\sum_{v \in S} d(v) \le |S|\, d_{\max}$. Hence every S has $\Phi(S) \ge 1/(2b\, d_{\max})$. □

The graph C_k

We will just do C_n for now; Section 1.6.7 returns to general k. The vertices of C_n are the matchings in $M_n \cup M_{n-1}$, and its (directed) edges are of four kinds.

- **Reduce:** if $m \in M_n$ and $e \in m$, go to $m' = m \setminus e$.
- **Augment:** if $m \in M_{n-1}$ and u and v are unmatched (and $(u, v) \in E$), move to $m' = m \cup (u, v)$.
- **Rotate:** if $m \in M_{n-1}$, u is matched to w, and v is unmatched (and $(u, v) \in E$), move to $m' = m \cup (u, v) \setminus (u, w)$.
- **Self loops:** stay where you are. Add enough of them that every vertex has degree $2|E|$, so that the stationary distribution is uniform.

We want an undirected graph, and indeed a Reduce one way is an Augment the other way. The reverse of a Rotate is again a Rotate: from $m \cup (u, v) \setminus (u, w)$, the vertex u is matched to v and w is unmatched, and rotating adds (u, w) back and removes (u, v). Every move that is not a self loop is determined by the edge of G that it removes (Reduce) or adds (Augment, Rotate). So each matching has at most $|E|$ such moves. This makes the degree $2|E|$ possible, and it means that at least half of the edges at every vertex are self loops, so the walk is automatically lazy. Figure 1.28 shows C_2 for $G = K_{2,2}$.

The canonical paths

To every node $s \in M_n \cup M_{n-1}$ we associate a *partner* $s' \in M_n$:

- if $s \in M_n$, then $s' = s$;
- if $s \in M_{n-1}$ and s has an augmenting path of length 1, add the corresponding edge to get s';

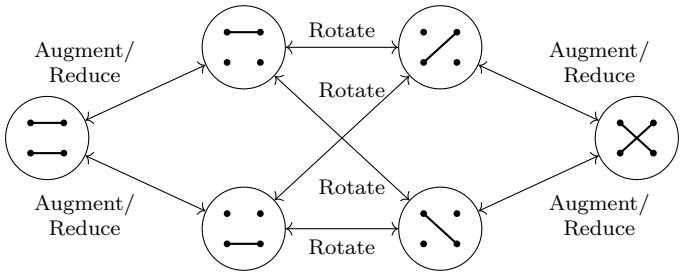


Figure 1.28: The graph C_2 for $G = K_{2,2}$, with self loops omitted. In each state the two dots on the left are one side of G and the two on the right are the other; the drawn edges form the matching. The two perfect matchings are at the far left and far right.

- if $s \in M_{n-1}$ and its shortest augmenting path has length 3, augment along it to get s'.

When there are several such augmenting paths, we fix one choice for each s once and for all. By Lemma 1.6.5, one of the last two cases always applies when $s \in M_{n-1}$. The canonical path from s to t will have three segments:

1. from s to s' (type A);
2. from s' to t' (type B);
3. from t' to t (type A, traversed backwards).

Type A paths connect a vertex s to its partner s'.

- If $s \in M_n$, then $s = s'$ and the path is empty.
- If $s \in M_{n-1}$ with an augmenting path of length one, the path has length 1: it is the edge of the corresponding Augment.
- If $s \in M_{n-1}$ with an augmenting path of length 3, the path has length 2: first Rotate, then Augment (Fig. 1.29).

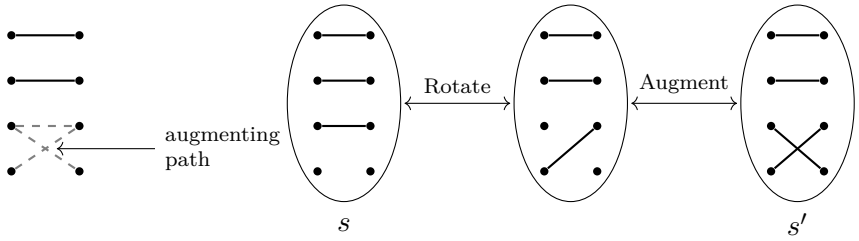


Figure 1.29: A type A path of length 2. Left: a matching $s \in M_{n-1}$ (the two solid edges together with the dashed horizontal edge) and its augmenting path of length 3 (dashed). Right: the type A path from s to its partner s', one Rotate followed by one Augment.

Type B paths. Here s and t are each perfect matchings. Let $s \triangle t$ be their symmetric difference. It consists of a collection of disjoint, even-length, alternating cycles of length at least 4: an edge of s, then an edge of t, then of s, then of t, and so on (Fig. 1.30). Every vertex lies on exactly one edge of s and one edge of t. If these are different edges, the vertex has degree 2 in $s \triangle t$, and otherwise degree 0. The path from s to t will "unwind" each cycle.

- We order the cycles somehow, so that there is a canonical order in which to do this.
- We also need to specify canonically a "start" vertex in each cycle.

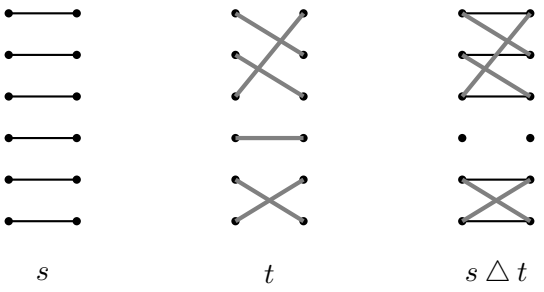


Figure 1.30: Two perfect matchings s (thin) and t (thick gray) on $6 + 6$ vertices, and their symmetric difference. The fourth edge is common to s and t, so it is not in $s \triangle t$; the rest forms one alternating cycle of length 6 (first three rows) and one of length 4 (last two rows).

Unwinding a single cycle. Figure 1.31 shows the unwinding of a cycle of length 8. In general, take a cycle of $s \triangle t$ with 2ℓ edges and its start vertex. First Reduce: remove the edge of s at the start vertex. Then Rotate $\ell - 1$ times, going around the cycle: each Rotate adds the edge of t at the vertex that has just become unmatched and removes the edge of s at the other end of the new edge. Finally Augment: add the edge of t at the start vertex. Afterwards the matching agrees with t on the cycle and is unchanged elsewhere.

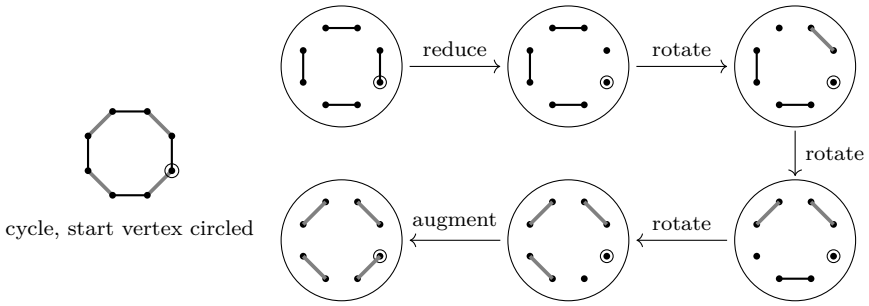


Figure 1.31: Unwinding a cycle of length 8 of $s \triangle t$. Left: the cycle, with the edges of s thin and the edges of t thick gray, and its start vertex circled. Right: the six matchings along the path, from all of s (top left) to all of t (bottom left), restricted to the cycle.

Unwinding a collection of cycles. If $s \triangle t$ consists of the cycles $\Gamma_0, \Gamma_1, \Gamma_2, \ldots$ in their canonical order, the type B path unwinds Γ_0 as above, then Γ_1, then Γ_2, and so on, each from its start vertex. Along the way every matching is in M_n or M_{n-1}, so this is a path in C_n.

Counting canonical paths through an edge

Lemma 1.6.8. *Let $s' \in M_n$. At most $O(n^2)$ nodes $s \in M_n \cup M_{n-1}$ have s' as their partner.*

Proof. There is only one such $s \in M_n$, namely s' itself. Otherwise $s \in M_{n-1}$, and s' is reached from s by either an Augment edge or a (Rotate, Augment) pair of edges, so s is reached from s' by going back along one edge (a Reduce) or two edges. There are at most $O(n)$ ways to Reduce, and $O(n^2)$ ways to undo a (Rotate, Augment) pair. To undo the Augment, remove one of the n edges of s'. Then undo the Rotate. It is a Rotate out of a matching with two unmatched vertices, and it is determined by one of them and an edge of G at it, which is matched at its other end. So there are at most $2n$ choices. □

That lets us understand type A paths. Now let us look at type B paths. Fix one transition T, that is, an edge of C_n. We count how many pairs $(s,t) \in M_n \times M_n$ contain T on their type B canonical path.

Claim 1.6.9. *The number of such pairs is at most $|C_n|$.*

Before discussing the proof, here is what the claim gives. It implies that T is on at most $O(n^2 \cdot |C_n| \cdot n^2)$ canonical paths. The canonical path from u to w uses T in its type B segment only if (u', w') is one of the at most $|C_n|$ pairs of Claim 1.6.9. By Lemma 1.6.8, each of u' and w' is the partner of only $O(n^2)$ nodes. A path can also use T in one of its two type A segments. But a type A segment has length at most 2, so T lies on the type A segment of only $O(n)$ nodes, and these contribute only $O(n|C_n|)$ further paths. So Claim 1.6.7 applies with $b = O(n^4)$, and since every vertex of C_n has degree $d_{\max} = 2|E| \le 2n^2$,

$$\Phi(C_n) \ge \frac{1}{2n^4 d_{\max}} = \Theta\Big(\frac{1}{n^6}\Big) \tag{1.6.3}$$

(writing n^4 for b and suppressing the constant in $b = O(n^4)$). This means that the lazy random walk mixes in a polynomial number of steps, which is what we wanted. Quantitatively, Cheeger's inequality in the form (1.5.1) gives a spectral gap $\gamma \ge \Phi(C_n)^2/4 = \Omega(n^{-12})$. Since C_n is regular, the ℓ_∞ bound of Theorem 1.5.8 gives $|p_t(v) - \pi(v)| \le (1-\gamma)^t$. And $\log|C_n| = O(n \log n)$, so $O(n^{12}(n \log n + \log(1/\epsilon)))$ steps bring every probability within a factor $1 \pm \epsilon$ of the uniform value $1/|C_n|$.

Proof idea for Claim 1.6.9. We bound the number of such pairs by mapping them injectively into C_n. So we need a map $\sigma_T(s,t)$ that takes a pair (starting matching, ending matching) to an element of $M_n \cup M_{n-1}$, such that, given T and the result, we can figure out which s and t it came from. Loosely:

- σ_T agrees with s on the cycles we have already unwound, and on the part of the current cycle that we have already unwound;
- σ_T agrees with t elsewhere.

So $\sigma_T(s,t)$ is "complementary" to the matching at which the path from s to t takes the step T: that matching agrees with t on the part already unwound and with s on the rest. There are three cases:

- T is the Reduce at the beginning of an unwinding;

- T is the Augment at the end of an unwinding;
- T is a Rotate in the middle.

Figure 1.32 shows the simplest case: at the very first step of the path, before anything is unwound, the current matching is s on every cycle, and $\sigma_T(s,t)$ is t on every cycle. *(Proof omitted; we give only this sketch, not the precise definition of σ_T in the three cases or the check that it is injective.)*

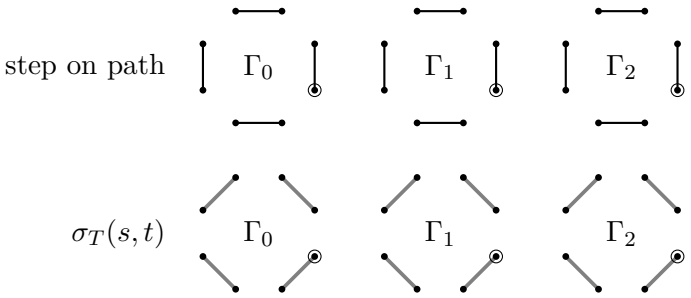


Figure 1.32: The map σ_T when T is the first step of the path from s to t, and $s \triangle t$ consists of three cycles of length 8 with marked start vertices. Top: the matching at which the path takes the step (all of s). Bottom: $\sigma_T(s,t)$ (all of t).

1.6.7 Dealing with smaller matchings

This was for perfect matchings, that is, for the ratio r_n, but we need r_k for all k. We could do a similar argument and show mixing on every C_k. But there is a better trick: reduce one problem to the other. Create a (sufficiently dense) graph G_k whose perfect and near-perfect matchings let us count the k- and $(k-1)$-matchings of G. *(Proof omitted; we do not give the construction of G_k.)*

1.6.8 Summary of the argument

- General principle: if we can sample, we can (often) approximately count.
- There are lots of exponential gaps, so we need to go in stages.
- Estimate the ratios $|M_k|/|M_{k-1}|$ by sampling from $M_k \cup M_{k-1}$.
- Sampling directly is hard, so instead take a random walk.
- Show that the random walk mixes (that is, gets close to the uniform distribution) in a polynomial number of steps.

1.7 Diameter, Eigenvalues, and Chebyshev Polynomials

Eigenvalues are related to cuts (Section 1.4) and to the mixing of random walks (Section 1.5), and they give a lot more structural information about a graph. This section shows that they bound the diameter. A first bound comes for free from the mixing bound of Theorem 1.5.8. Looking at what that proof really used, namely that a polynomial in the walk matrix has no zero

entries, leads to a much better bound, obtained by choosing a better polynomial: a shifted and scaled Chebyshev polynomial. These polynomials come from approximation theory, and they return in the iterative methods for linear systems of Chapter 3.

1.7.1 Diameters and eigenvalues

The intuition is simple:

- well-connected graphs have big λ_2;
- well-connected graphs have small diameter;
- so graphs with big λ_2 should have small diameter.

Setting. In this section δ denotes the diameter of G, and we assume that G is d-regular; this simplifies the arguments but does not change the results. Let

$$M = \frac{A}{2d} + \frac{I}{2}$$

be the lazy random walk matrix. For a d-regular graph, $M = W' = W'_N$ in the notation of Section 1.5.3, so M is symmetric and its eigenvalues are $1 = \mu'_1 \geq \mu'_2 \geq \cdots \geq \mu'_n \geq 0$. Let

$$\gamma = 1 - \mu'_2$$

be the spectral gap.

1.7.2 A first bound

Claim 1.7.1. *For any connected d-regular G,*

$$\delta \leq \frac{\ln n}{\gamma}.$$

Proof. We have already done it! Let u and v be the vertices that are farthest from each other, so that $\operatorname{dist}(u, v) = \delta$. Start a lazy random walk at u, and let $p_t(v)$ be the probability that the walk is at v at time t. By the ℓ_∞ convergence theorem (Theorem 1.5.8),

$$|p_t(v) - \pi(v)| \leq (1-\gamma)^t \sqrt{\frac{d(v)}{\min_y d(y)}} = (1-\gamma)^t,$$

since all degrees are equal. If $p_t(v) > 0$, then $\delta \leq t$. Each step moves the walk along at most one edge, so it cannot reach v in fewer than δ steps. Since $\pi(v) = 1/n$, if $|p_t(v) - \pi(v)| < \frac{1}{n}$ then $p_t(v) > 0$, and so $\delta \leq t$. If $t = \frac{\ln n}{\gamma}$, then

$$(1-\gamma)^t = (1-\gamma)^{\frac{\ln n}{\gamma}} \leq \left(\frac{1}{e}\right)^{\ln n} = \frac{1}{n}.$$

This inequality is in fact strict, since $1 - \gamma < e^{-\gamma}$ for $\gamma > 0$, and the strict version is what we need above. □

We have ignored rounding here. The walk only has integer times, so the proof applies to $t = \lceil \ln n/\gamma \rceil$ and shows $\delta \leq \lceil \ln n/\gamma \rceil$. The same rounding applies to Theorem 1.7.4 below. The rounding can matter: for $G = K_2$ we have $n = 2$, $M = \frac{1}{2}J$, $\mu'_2 = 0$ and $\gamma = 1$, so $\ln n/\gamma = \ln 2 < 1 = \delta$, and Claim 1.7.1 fails without the ceiling.

1.7.3 Can we do better?

What just happened? If the (u, v)th entry of A^k is nonzero, there is a path of length at most k from u to v. Replacing A with M does not change this. Since $M = (A + dI)/(2d)$, the matrix M^k is a combination of $I, A, \ldots, A^k$ with positive coefficients. So its (u, v) entry is positive exactly when $\operatorname{dist}(u, v) \le k$. Let χ_u and χ_v be the corresponding basis vectors. What the proof showed is that, for $k = \ln n/\gamma$,

$$\left|\chi_v^\top M^k \chi_u - \frac{1}{n}\right| < \frac{1}{n},$$

and in particular $\chi_v^\top M^k \chi_u \neq 0$.

Using other polynomials. Let $p(x) = \sum_{j=0}^k c_j x^j$ be a polynomial of degree at most k. We can apply it to the matrix M:

$$p(M) = \sum_{j=0}^k c_j M^j.$$

If $p(M)$ has no zero entries, then $\delta \le k$. Why? The (v, u) entry of $p(M)$ is $\sum_j c_j (M^j)_{vu}$. If $\operatorname{dist}(u, v) > k$, every $(M^j)_{vu}$ with $j \le k$ is zero, and then so is the (v, u) entry of $p(M)$.

Claim 1.7.2. *Let p be a polynomial of degree at most k. Suppose $p(1) = 1$ and $|p(\mu_i')| < \frac{1}{n}$ for all $i \ge 2$. Then $\delta \le k$.*

Proof. Let $v_1, \ldots, v_n$ be an orthonormal basis of eigenvectors of M, with $Mv_i = \mu_i' v_i$ and $v_1 = \frac{1}{\sqrt{n}}\mathbf{1}$. Then $p(M) = \sum_i p(\mu_i') v_i v_i^\top$, so for any two vertices a and b,

$$\begin{aligned}
\chi_a^\top p(M) \chi_b &= \chi_a^\top \Big(\sum_i p(\mu_i') v_i v_i^\top\Big) \chi_b = \sum_i p(\mu_i')\, v_i(a)\, v_i(b) \\
&= \frac{1}{n} + \sum_{i=2}^n p(\mu_i')\, v_i(a)\, v_i(b) \ge \frac{1}{n} - \sum_{i=2}^n |p(\mu_i')|\, |v_i(a)|\, |v_i(b)| \\
&\ge \frac{1}{n} - \max_{i \ge 2} |p(\mu_i')| \Big(\sum_{i=2}^n |v_i(a)|^2\Big)^{1/2} \Big(\sum_{i=2}^n |v_i(b)|^2\Big)^{1/2} \\
&\ge \frac{1}{n} - \max_{i \ge 2} |p(\mu_i')| > 0.
\end{aligned}$$

In the second line, the $i = 1$ term is $p(1)\, v_1(a)\, v_1(b) = 1 \cdot \frac{1}{\sqrt{n}} \cdot \frac{1}{\sqrt{n}}$. The third line is the Cauchy–Schwarz inequality. The last line uses $\sum_{i=1}^n v_i(a)^2 = 1$: the matrix with columns $v_1, \ldots, v_n$ is orthogonal, so its rows are unit vectors too. So $p(M)$ has no zero entries, and $\delta \le k$. □

With $p(x) = x^k$ this is the first bound again: $|(\mu_i')^k| \le (1 - \gamma)^k$ for $i \ge 2$. The question is whether some other polynomial with $p(1) = 1$ is small on all the other eigenvalues at a lower degree.

1.7.4 Magic polynomials

We assert the existence of magic polynomials with the following properties. They come back in the analysis of iterative methods for linear systems (Sections 3.1.6 and 3.2.4).

Claim 1.7.3. *For any $t \in (0,1)$ and any $k \geq 0$ there exists a polynomial $p_k^{(t)}$ such that*

- *$p_k^{(t)}$ is of degree k;*
- *$p_k^{(t)}(1) = 1$;*
- *$|p_k^{(t)}(x)| \leq 2(1+2\sqrt{t})^{-k}$ for any $x \in [0, 1-t]$.*

The proof is the construction in Section 1.7.6. Compare this with x^k, which also has degree k and value 1 at $x = 1$. On $[0, 1-t]$ it gets as large as $(1-t)^k \approx e^{-tk}$, so it is small only once k is of order $1/t$. But $2(1+2\sqrt{t})^{-k}$ is already small once k is of order $1/\sqrt{t}$. Figure 1.33 shows some examples.

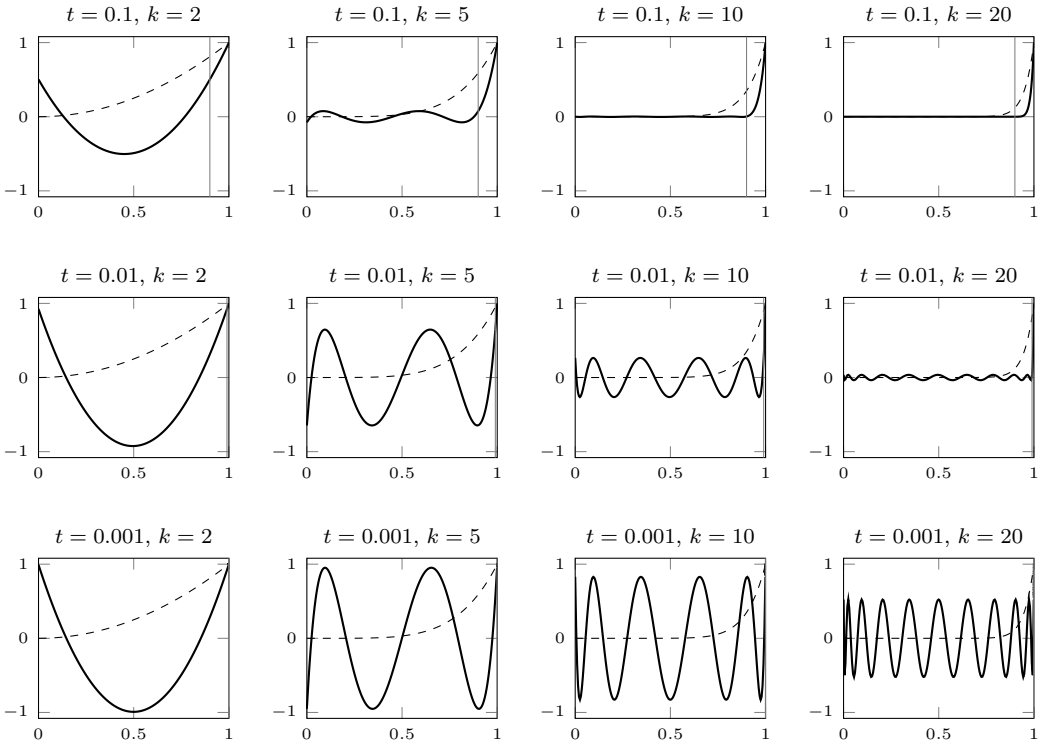


Figure 1.33: The magic polynomial $p_k^{(t)}$ (solid) and x^k (dashed) on $[0,1]$, for $t = 0.1$, 0.01, 0.001 (rows) and $k = 2, 5, 10, 20$ (columns); the vertical line is at $x = 1-t$. Both polynomials equal 1 at $x = 1$, but on $[0, 1-t]$ the magic polynomial is much smaller once k is large compared with $1/\sqrt{t}$.

1.7.5 The better bound

If we set $t = \gamma$ in Claim 1.7.3, we get a polynomial p of degree k such that

- $p(1) = 1$;
- $|p(x)| \leq 2(1+2\sqrt{\gamma})^{-k}$ for any $x \in [0, \mu_2']$.

Here $[0, 1-\gamma] = [0, \mu_2']$ contains all the eigenvalues $\mu_2', \dots, \mu_n'$, which are nonnegative since the walk is lazy. If $k \geq \left(1 + \frac{1}{2\sqrt{\gamma}}\right) \ln(2n)$, we can show that $|p(x)| < \frac{1}{n}$ for all $x \in [0, \mu_2']$. Write $a = 2\sqrt{\gamma}$. Since $\ln(1+a) > \frac{a}{1+a}$ for $a > 0$,

$$2(1+a)^{-k} = 2e^{-k\ln(1+a)} < 2e^{-ka/(1+a)} \leq 2e^{-\ln(2n)} = \frac{1}{n}.$$

Claim 1.7.2 then gives:

Theorem 1.7.4. *For any connected d-regular graph G,*

$$\delta \leq \left(1 + \frac{1}{2\sqrt{\gamma}}\right) \ln(2n).$$

For small γ this is about $\ln(2n)/(2\sqrt{\gamma})$, compared with $\ln n/\gamma$ in Claim 1.7.1: the dependence on the spectral gap improves from $1/\gamma$ to $1/\sqrt{\gamma}$.

1.7.6 Chebyshev polynomials

There are a couple of equivalent definitions; we start with the following. In this subsection d denotes the degree of a polynomial, not the degree of a vertex.

Definition 1.7.5. The dth *Chebyshev polynomial* T_d is the polynomial such that

$$\cos(d\theta) = T_d(\cos\theta) \qquad \text{for all } \theta.$$

It is not obvious that this is a polynomial (yet). Let $z = e^{i\theta} = x + iy$. Then

$$x = \cos\theta = \tfrac{1}{2}(z + \bar{z}) = \tfrac{1}{2}\Big(z + \frac{1}{z}\Big), \qquad T_d(x) = \cos(d\theta) = \mathrm{Re}(e^{id\theta}) = \tfrac{1}{2}\Big(z^d + \frac{1}{z^d}\Big).$$

Claim 1.7.6. *$T_d(x) = 2x\,T_{d-1}(x) - T_{d-2}(x)$, or equivalently $T_d(x) + T_{d-2}(x) = 2x\,T_{d-1}(x)$.*

Proof. We have

$$z^d + z^{d-2} = z^{d-1}\Big(z + \frac{1}{z}\Big) \qquad \text{and} \qquad \frac{1}{z^d} + \frac{1}{z^{d-2}} = \frac{1}{z^{d-1}}\Big(\frac{1}{z} + z\Big).$$

Adding the two and dividing by 2,

$$\tfrac{1}{2}\Big(z^d + \frac{1}{z^d}\Big) + \tfrac{1}{2}\Big(z^{d-2} + \frac{1}{z^{d-2}}\Big) = \tfrac{1}{2}\Big(z^{d-1} + \frac{1}{z^{d-1}}\Big)\Big(z + \frac{1}{z}\Big),$$

which is $T_d(x) + T_{d-2}(x) = T_{d-1}(x) \cdot 2x$. □

Since $T_0(x) = 1$ and $T_1(x) = x$, the recurrence shows that these are polynomials, T_d of degree d:

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_2(x) = 2x^2 - 1, \quad T_3(x) = 4x^3 - 3x, \quad T_4(x) = 8x^4 - 8x^2 + 1.$$

Figure 1.34 shows a few of them. On $[-1, 1]$ they oscillate between -1 and 1; outside $[-1, 1]$ they grow very fast.

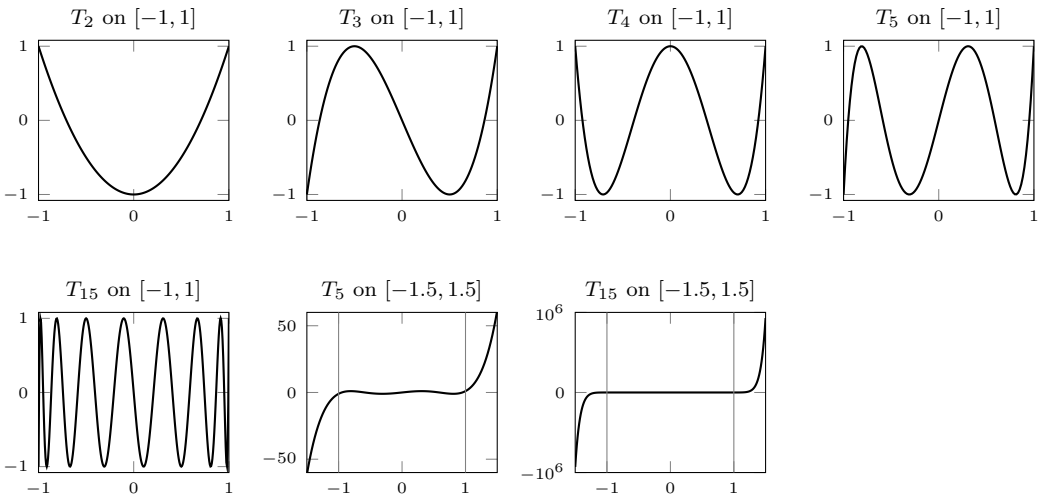


Figure 1.34: Chebyshev polynomials. Top: $T_2, \dots, T_5$ on $[-1,1]$. Bottom: T_{15} on $[-1,1]$, and T_5 and T_{15} on $[-1.5,1.5]$, with vertical lines at ± 1; note the vertical scales ($T_{15}(1.5) \approx 9 \cdot 10^5$).

A closed form. Since $x^2 + y^2 = 1$, we have $y = \sqrt{1-x^2}$ and $iy = \sqrt{x^2-1}$, so

$$T_d(x) = \tfrac{1}{2}\big(x + \sqrt{x^2-1}\big)^d + \tfrac{1}{2}\big(x - \sqrt{x^2-1}\big)^d. \tag{1.7.1}$$

Both sides are polynomials in x, since the odd powers of $\sqrt{x^2-1}$ cancel. They agree on $[-1,1]$, so they agree everywhere. Some immediate properties:

- If $x \in [-1,1]$, then $T_d(x) \in [-1,1]$. (Write $x = \cos\theta$.)
- If $x \ge 1 + \eta$ for some $\eta \ge 0$, then

$$T_d(x) \ge \tfrac{1}{2}\Big(1 + \eta + \sqrt{2\eta + \eta^2}\Big)^d \ge \tfrac{1}{2}\big(1 + \sqrt{2\eta}\big)^d. \tag{1.7.2}$$

  For $x \ge 1$ both terms in (1.7.1) are nonnegative, so $T_d(x) \ge \frac{1}{2}(x + \sqrt{x^2-1})^d$, and $x + \sqrt{x^2-1}$ is increasing in x.

Shifting and scaling these gives our magic polynomials. Let ℓ be the linear function that maps $[0, 1-t]$ onto $[-1,1]$:

$$\ell(x) = \frac{2x}{1-t} - 1, \qquad \ell(0) = -1, \quad \ell(1-t) = 1, \quad \ell(1) = 1 + \frac{2t}{1-t},$$

and define

$$p_k^{(t)}(x) = \frac{T_k(\ell(x))}{T_k(\ell(1))}. \tag{1.7.3}$$

Proof of Claim 1.7.3. The polynomial (1.7.3) has degree k since ℓ is linear, and $p_k^{(t)}(1) = 1$. For $x \in [0, 1-t]$ we have $\ell(x) \in [-1,1]$, so $|T_k(\ell(x))| \le 1$. And $\ell(1) = 1 + \eta$ with $\eta = \frac{2t}{1-t} \ge 2t$, so by (1.7.2)

$$T_k(\ell(1)) \ge \tfrac{1}{2}\big(1 + \sqrt{2\eta}\big)^k \ge \tfrac{1}{2}\big(1 + 2\sqrt{t}\big)^k.$$

Dividing, $|p_k^{(t)}(x)| \le 2(1 + 2\sqrt{t})^{-k}$ on $[0, 1-t]$. □

1.7.7 Polynomials and the power method

Why can't this speed up the mixing of every random walk? We leave this as a question.

The argument also foreshadows the iterative methods of Chapter 3. Suppose you have a symmetric matrix M and want its biggest eigenvector (say normalized so that the eigenvalue is 1), and let γ be the gap between 1 and the next eigenvalue. An easy way to get an approximate answer is to compute $M^k x$ for a pretty big k and a random x. Why does this work?

- The big eigenvector stays, and the smaller ones drop out.
- The error term falls off like $(1-\gamma)^k$, so it takes about $1/\gamma$ steps to get close.
- We need to be a little careful to keep things from getting too big.

We just found a much faster algorithm:

- assuming we know some good bound on γ,
- compute $p_k^{(\gamma)}(M)x$ instead of $M^k x$;
- it converges way faster (about $1/\sqrt{\gamma}$ steps instead of $1/\gamma$, by Claim 1.7.3);
- it obeys a nice three-term recurrence, which depends on the last two values, not just the last one. By Claim 1.7.6, $T_k(\ell(M))x = 2\ell(M)\,T_{k-1}(\ell(M))x - T_{k-2}(\ell(M))x$, so each new vector costs one multiplication by M.

Sections 3.1 and 3.2 develop this idea for solving linear systems.

1.8 Expanders

An *expander* is a sparse graph that nevertheless has no small cuts. Expanders turn up *everywhere*: in routing, lower bounds, derandomization, counterexamples to many conjectures, and PCPs, and more recently in algorithms for sparsest cut, graph connectivity, and more. This section defines expander families, through λ_2 and through cuts, and asks how large λ_2 can be. It then proves the expander mixing lemma, which says that in an expander the number of edges between any two sets of vertices is close to what one would expect in a random graph of the same density, and uses it to show that small sets of vertices have many neighbors. We assume throughout that all graphs are d-regular.

1.8.1 Definitions

Being an expander is an asymptotic property, so we really need to study a family of d-regular graphs $(G_n)_n$ as n goes to infinity, and you should think of d as a constant.

Definition 1.8.1. A family $(G_n)_n$ of d-regular graphs is an *expander family* if $\lambda_2(G_n) \geq c$ for some constant $c > 0$ and for all n.

Expanders and cuts. We know from Cheeger's inequality (Theorem 1.4.5) that $\frac{\lambda_2}{2} \le \phi(G)$. So for an expander family $\phi(G_n) \ge \frac{c}{2}$. This means that any set S of at most $\frac{n}{2}$ vertices has at least $(\frac{c}{2})|S|$ edges leaving it, so there are no small cuts. We also know that $\frac{\Theta(1)\,\phi^2}{d} \le \lambda_2$ (Cheeger's inequality gives the constant $\frac{1}{2}$), so $\phi(G) \le \Theta(1)\sqrt{d\lambda_2}$. In particular a lower bound $\phi(G_n) \ge c'$ gives $\lambda_2(G_n) \ge \Theta(1)\,c'^2/d$, which is again a constant, since d is. So we could alternatively define:

Definition 1.8.2. A family $(G_n)_n$ of d-regular graphs is an *expander family* if $\phi(G_n) \ge c'$ for some constant $c' > 0$ and all n.

1.8.2 Do expanders exist?

A natural question is for what parameters expanders exist, if any.

Claim 1.8.3. *For any d-regular G,*

$$\lambda_2 \le d - 2\sqrt{d-1} + o_n(1).$$

(Proof omitted; we will instead show the bound without the 2, in Claim 1.8.4 below.) Note that $\lambda_2 \le d$ is easy. Actually $\lambda_2 \le \frac{nd}{n-1} = d + \frac{d}{n-1}$, since λ_2 is at most the average of $\lambda_2, \dots, \lambda_n$, and these sum to $\operatorname{tr} L = nd$. The extra $\frac{d}{n-1}$ is needed: the complete graph K_n has $d = n-1$ and $\lambda_2 = n = d+1$. There actually exist families of graphs, called *Ramanujan graphs*, meeting this bound, and a random regular graph is an expander.

Claim 1.8.4. *For any d-regular G with diameter at least 4,*

$$\lambda_2 \le d - \sqrt{d}.$$

Proof. Let a, b be vertices with $\operatorname{dist}(a,b) \ge 4$. Create a test vector x with

- $x(a) = 1$, and $x(b) = -1$;
- $x(u) = 1/\sqrt{d}$ for every neighbor u of a, and $x(u) = -1/\sqrt{d}$ for every neighbor u of b;
- $x(u) = 0$ for every other vertex u.

(See Fig. 1.35.) Then $x \perp \mathbf{1}$ (the entries sum to $1 + d\cdot\frac{1}{\sqrt{d}} - 1 - d\cdot\frac{1}{\sqrt{d}} = 0$), so by Courant–Fischer

$$\lambda_2 \le \frac{x^\top L_G x}{x^\top x} = \sum_{(i,j)\in E} \frac{(x_i - x_j)^2}{x^\top x} \le \frac{2\Big(d(1-\frac{1}{\sqrt{d}})^2 + d(d-1)(\frac{1}{\sqrt{d}})^2\Big)}{2\Big(1 + d\cdot(\frac{1}{\sqrt{d}})^2\Big)}$$
$$= \frac{2\big((d+1-2\sqrt{d}) + (d-1)\big)}{4} = \frac{4(d-\sqrt{d})}{4} = d - \sqrt{d}.$$

For the numerator: the d edges at a each contribute $(1-1/\sqrt{d})^2$. Each neighbor of a has $d-1$ further edges, going to another neighbor of a (contributing 0) or to a vertex at distance 2 from a. Since $\operatorname{dist}(a,b) \ge 4$, such a vertex is neither b nor a neighbor of b, so it has value 0 and the edge contributes $(1/\sqrt{d})^2$. So at most $d(d-1)$ edges contribute $1/d$ each. The same holds around b. No edge joins the two neighborhoods, since that would make $\operatorname{dist}(a,b) \le 3$, and all other edges contribute 0. The denominator is $x^\top x = 2(1 + d\cdot\frac{1}{d}) = 4$. □

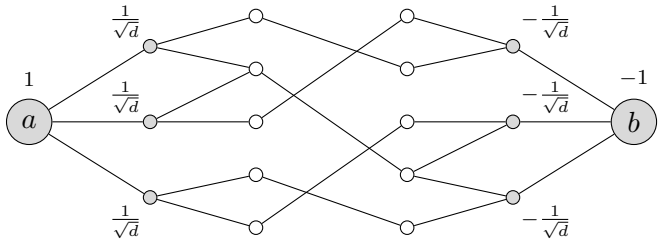


Figure 1.35: The test vector in the proof of Claim 1.8.4, drawn schematically for $d = 3$: value 1 at a, $1/\sqrt{d}$ on its neighbors, -1 at b, $-1/\sqrt{d}$ on its neighbors, and 0 elsewhere (white).

1.8.3 Expanders and randomness: the expander mixing lemma

Expanders are all over the study of randomness; here we study one interesting property.

Adjacency eigenvalues. For a d-regular graph G with adjacency matrix A, write the eigenvalues of A as $d = \alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n$, so that $\alpha_i = d - \lambda_i$, and let

$$\bar{\alpha} := \max(|\alpha_2|, |\alpha_n|)$$

be the second-largest absolute value of an eigenvalue of the adjacency matrix.[2] Since A is symmetric and $A\mathbf{1} = d\mathbf{1}$, the matrix A maps the subspace $\mathbf{1}^\perp$ to itself, and its eigenvalues there are $\alpha_2, \ldots, \alpha_n$. So $\|Aw\| \leq \bar{\alpha}\|w\|$ for every $w \perp \mathbf{1}$.

Comparison with a random graph. Suppose we make a graph by including each possible edge independently with probability d/n, so that degrees are about d. Then the expected number of edges between any two sets S and T is about $d|S||T|/n$. The expander mixing lemma says that in a graph with small $\bar{\alpha}$ the actual number is close to this, for *every* pair of sets. Here $e(S,T) = \chi_S^\top A \chi_T$ counts the ordered pairs $(u,v) \in S \times T$ that are edges, so an edge with both ends in $S \cap T$ is counted twice.

Theorem 1.8.5 (Expander mixing lemma)**.** *Let G be d-regular. For all $S, T \subseteq V$,*

$$\left| e(S,T) - \frac{d|S||T|}{n} \right| \leq \frac{\bar{\alpha}}{n}\sqrt{|S||\bar{S}||T||\bar{T}|} \leq \bar{\alpha}\sqrt{\min(|S|,|\bar{S}|)\cdot\min(|T|,|\bar{T}|)}. \tag{1.8.1}$$

Proof. Let

$$v = \chi_S - \frac{|S|}{n}\mathbf{1}, \qquad w = \chi_T - \frac{|T|}{n}\mathbf{1},$$

so that $v, w \perp \mathbf{1}$. Their squared lengths are

$$\|v\|^2 = |S| - \Big(\frac{|S|}{n}\Big)^2 \mathbf{1}^\top\mathbf{1} = \frac{1}{n}|S|(n-|S|) = \frac{1}{n}|S||\bar{S}|, \qquad \|w\|^2 = \frac{1}{n}|T||\bar{T}|.$$

[2] Since $\alpha_2 = d - \lambda_2$, we have $\bar{\alpha} = d - \lambda_2$ when $|\alpha_n| \leq \alpha_2$; in general $\bar{\alpha} \geq d - \lambda_2$.

Because G is d-regular, $A\mathbf{1} = d\mathbf{1}$ and $\mathbf{1}^\top A = d\mathbf{1}^\top$, so $v^\top A\mathbf{1} = d\, v^\top \mathbf{1} = 0$ and $\mathbf{1}^\top Aw = d\,\mathbf{1}^\top w = 0$. Expanding,

$$e(S,T) = \chi_S^\top A \chi_T = \Big(v + \frac{|S|}{n}\mathbf{1}\Big)^\top A\Big(w + \frac{|T|}{n}\mathbf{1}\Big)$$
$$= v^\top Aw + \frac{|S|}{n}\mathbf{1}^\top Aw + \frac{|T|}{n} v^\top A\mathbf{1} + \frac{|S|}{n}\frac{|T|}{n}\, d\,\mathbf{1}^\top\mathbf{1} = v^\top Aw + \frac{d|S||T|}{n},$$

since the two middle terms vanish and $\mathbf{1}^\top\mathbf{1} = n$. Therefore

$$e(S,T) - \frac{d|S||T|}{n} = v^\top Aw,$$

and by the Cauchy–Schwarz inequality and $\|Aw\| \le \bar{\alpha}\|w\|$ (as $w \perp \mathbf{1}$),

$$|v^\top Aw| \le \|v\|\,\|Aw\| \le \bar{\alpha}\,\|v\|\,\|w\| = \frac{\bar{\alpha}}{n}\sqrt{|S||\bar{S}||T||\bar{T}|}.$$

This is the first inequality in (1.8.1). For the second, $|S||\bar{S}| \le n\min(|S|,|\bar{S}|)$ since $\max(|S|,|\bar{S}|) \le n$, and similarly for T. □

One can read Theorem 1.8.5 in two ways: as saying that expanders are like random graphs, or as saying that expanders are like rescalings of the complete graph. Example 1.9.12 makes the second view precise.

Some properties we already know. Random walks on expanders mix in a logarithmic number of steps (Theorem 1.5.8); expanders have logarithmic diameter (Claim 1.7.1); and expanders have constant isoperimetric number (Section 1.8.1).

1.8.4 Vertex expansion

For a set of vertices X, let $N(X)$ be its neighbor set, the set of vertices adjacent to at least one vertex of X.

Claim 1.8.6. *Let G be d-regular. For every nonempty $X \subseteq V$,*

$$|N(X)| \ge \frac{d^2|X|}{\bar{\alpha}^2 + (d^2 - \bar{\alpha}^2)|X|/n}. \tag{1.8.2}$$

Proof. The proof plugs two sets into the expander mixing lemma and does the algebra. The first set is X. The second set is $Y = V \setminus (N(X) \cup X)$, so that $e(X,Y) = 0$. With this Y, the computation below gives (1.8.2) with $|N(X) \cup X|$ on the left. To get $|N(X)|$ itself, take instead $Y = V \setminus N(X)$. This set also has $e(X,Y) = 0$, since every neighbor of a vertex of X lies in $N(X)$.

Now (1.8.1) with $e(X,Y) = 0$ gives

$$\frac{d|X||Y|}{n} \le \frac{\bar{\alpha}}{n}\sqrt{|X|(n-|X|)|Y|(n-|Y|)}, \quad \text{so} \quad d^2|X||Y| \le \bar{\alpha}^2(n-|X|)(n-|Y|).$$

Write $s = n - |Y| = |N(X)|$. Then $d^2|X|(n-s) \le \bar{\alpha}^2(n-|X|)s$, that is,

$$d^2|X|\,n \le s(\bar{\alpha}^2(n-|X|) + d^2|X|) = s\,n(\bar{\alpha}^2 + (d^2-\bar{\alpha}^2)|X|/n),$$

which is (1.8.2). □

If $|X|/n$ is small and $\bar{\alpha} = 2\sqrt{d-1}$ (a Ramanujan graph), the right side of (1.8.2) is about $\frac{d^2}{4(d-1)}|X|$, which exceeds $\frac{d}{4}|X|$, so $|N(X)| \geq \frac{d}{4}|X|$. Since $|N(X)| \leq d|X|$ for every X, this is quite strong.

Setting X to be a single vertex gives another way to see that the second-largest eigenvalue cannot be much smaller than $\sqrt{d}$. Then $|N(X)| = d$, and (1.8.2) becomes $\bar{\alpha}^2 + (d^2 - \bar{\alpha}^2)/n \geq d$, that is, $\bar{\alpha}^2 \geq \frac{d(n-d)}{n-1}$, so $\bar{\alpha} \geq \sqrt{d} - o_n(1)$.[3]

1.8.5 Other notions of expansion

Several related definitions come up in the literature.

- If $|N(S)| \geq \beta|S|$ whenever $|S| \leq n/2$, one says that the *vertex expansion* of G is at least β.
- Sometimes one only cares about the expansion of smaller sets (for example, sets of size at most $0.01n$).

Definition 1.8.7 ((α, β)-expander)**.** G is an (α, β)-*expander* if all sets S of size at most αn have $|N(S)| \geq \beta|S|$.

In this subsection α and β are the two parameters of this definition, not adjacency eigenvalues. The definition only makes sense if $\alpha\beta < 1$ (roughly because $|N(S)| \leq n$).

- We showed above that Ramanujan graphs are $(\alpha, \frac{d}{4})$-expanders for some constant α (for instance, $\alpha \leq 4/(d-2)^2$ suffices in (1.8.2) with $\bar{\alpha}^2 = 4(d-1)$).
- Some applications need expansion greater than $d/2$, but with a smaller (constant) α. Such graphs exist, but one cannot prove expansion better than $d/2$ with spectral techniques. Random graphs work, and in 2002 Capalbo, Reingold, Vadhan, and Wigderson [CRVW02] gave an explicit construction with expansion $d - o(1)$.
- Many of the applications of expanders use *bipartite expanders*, which are easier to show exist (this was a homework problem).

Definition 1.8.8 (Bipartite expander)**.** A d-regular bipartite graph is an (α, β)-*expander* if every set S on the left side with $|S| \leq \alpha n$ has $|N(S)| \geq \beta|S|$.

Fact 1.8.9. *Whenever $\alpha\beta < 1$, there exists some d such that almost all d-regular graphs on n nodes (n sufficiently large) are (α, β)-expanders.*

(Proof omitted.)

1.9 Spectral Sparsification

A *sparsifier* of a graph G is a sparse weighted graph on the same vertices that approximates G, so that algorithms can run on it instead of on G. This section first sketches the random-sampling construction of Benczúr and Karger [BK96], which approximately preserves every cut, and shows by an example that preserving every cut does not preserve the quadratic form of the Laplacian. It then sets up the Loewner order on symmetric matrices and uses it to define and

[3] This bounds $\bar{\alpha} = \max(|\alpha_2|, |\alpha_n|)$. It gives $\lambda_2 = d - \alpha_2 \leq d - \sqrt{d} + o_n(1)$ when $\alpha_2 = \bar{\alpha}$, but not in general. Compare the bound $\lambda_2 \leq d - \sqrt{d}$ for graphs of diameter at least 4 in Claim 1.8.4.

build *spectral* sparsifiers, whose Laplacians approximate that of G. The algorithm has the same structure as Benczúr–Karger (sample edges independently, reweight by the inverse probability), but its probabilities are defined through linear algebra rather than through cuts. The tools are pseudoinverses and a matrix tail bound of Rudelson and Vershynin [RV07], and the analysis goes through a question of independent interest: how many samples does one need to estimate a covariance matrix? The sampling probabilities turn out to be effective resistances (Section 1.10).

1.9.1 Cut sparsifiers by random sampling

Motivation

Suppose we have a graph G with $m = \Theta(n^2)$ edges and want to approximately solve a cut problem on it, such as sparsest cut, minimum cut, or minimum s-t cut. The running times of most algorithms grow with m, so they are much slower for dense graphs than for sparse ones. It would be very nice if we could somehow throw out most of the edges and still get a similar answer: the running time would then be like that for a sparse graph, at the price, of course, of getting only an approximate answer. The idea is random sampling. The construction below is not spectral, but Section 1.9.4 uses spectral techniques to improve it.

A first cut: uniform sampling

Create a new graph G' by keeping every edge of G independently with probability p. The expected number of edges of G' is mp. Consider a cut $V = S \cup \bar{S}$ with $e_G(S) = c$ crossing edges. Each edge of the cut is kept with probability p, so $\mathbb{E}[e_{G'}(S)] = pc$, and by a Chernoff bound[4]

$$\Pr\left[|e_{G'}(S) - pc| \geq \epsilon pc\right] \leq 2e^{-\epsilon^2 pc/3}. \tag{1.9.1}$$

So we are very likely to get approximately the right answer (relative to the expectation pc) *for sufficiently large cuts*. To handle all cuts at once we need to know how many cuts there are of each size.

Theorem 1.9.1 (Karger)**.** *If G has minimum cut c, then the number of cuts of size less than αc is less than $n^{2\alpha}$.*

(Proof omitted.)

In the next paragraph d denotes a constant of our choosing in an exponent, not a degree, and α is as in Theorem 1.9.1. If $p = \Omega(\frac{d \log n}{\epsilon^2 c})$, then by (1.9.1) a cut of size αc is within a $(1+\epsilon)$ factor of its expectation with probability $1 - n^{-d}$, for whatever constant d in the exponent we want. The failure probability for a cut of size αc is at most $2n^{-\Omega(d\alpha)}$. This decays with α fast enough to beat the fewer than $n^{2\alpha}$ such cuts in a union bound. So we can choose the constants so that every cut of size αc is right, and so we can show that every cut is within a $(1+\epsilon)$ factor of its correct value with probability $1 - n^{-d}$. If c is small, however, all bets are off.

Fixing the problem: importance sampling

The following fix is due to Benczúr and Karger [BK96]. Suppose the graph has a small cut, of size c, but a particular edge e is only involved in cuts of size at least k. (We will actually

[4] This is the two-sided bound for a sum of independent indicator variables with mean pc, for $0 < \epsilon < 1$. The lower tail alone is at most $e^{-\epsilon^2 pc/2}$.

need a slightly stronger assumption.) Then we should only need to sample e as if the graph had minimum cut k: every edge in a cut of size k would then be sampled at least enough to give a small probability of failure. So maybe we should sample nonuniformly. For example, think of two dense clusters joined by a single edge. That edge has to be kept with probability close to 1, or the sample is disconnected. But the edges inside the clusters lie only in large cuts, so they can be sampled sparsely.

The problem is that with nonuniform probabilities the expected size of a cut is no longer the right quantity. The solution: when we do sample an edge e, give it weight $w_e = 1/p_e$. Then each edge contributes $p_e \cdot (1/p_e) = 1$ to a cut in expectation, and every cut has the correct expectation. This is *importance sampling*.

So perhaps we should take w_e equal to the size of the minimum cut containing e. There are several problems with this.

- **Problem:** we want to take a union bound over all cuts to show that every cut is right. This does not quite work, since there are a lot of cuts of large size.
- **Also:** we do not know w_e; and we need to prove that we do not take too many edges.

The rough ideas of the fix are that one can use approximate values of w_e instead of exact ones, and compute these quickly, and that one can use a slightly more conservative weighting scheme. We will not cover the details of how to fix these problems (see the original paper [BK96]); instead, Section 1.9.8 gives the details of a stronger scheme.

The result

When an edge appears in G', we count it with weight w_e. The outcome of the construction is the following.

Definition 1.9.2 (Combinatorial sparsifier)**.** A weighted graph G' on the vertex set of G is a *combinatorial sparsifier* of G (with parameter ϵ) if the size of every cut in G' is an ϵ-approximation of its size in G: $(1-\epsilon)e_{G'}(S) \le e_G(S) \le (1+\epsilon)e_{G'}(S)$ for every $S \subseteq V$, where $e_{G'}(S)$ is the total weight of the edges of G' leaving S.

Theorem 1.9.3 (Benczúr–Karger)**.** *The importance-sampling scheme above, suitably refined, produces with high probability a weighted graph G' with $O(n \log n/\epsilon^2)$ edges in which every cut is approximately the same as in G, that is, a combinatorial sparsifier of G.*

(Proof omitted.)

1.9.2 From cuts to quadratic forms

Let G be our original graph and G' our (weighted) combinatorial sparsifier. For $x \in \{-1,1\}^n$, let $S = \{i : x_i = 1\}$; then $x^\top L_G x = \sum_{(i,j)\in E} w_{ij}(x_i - x_j)^2 = 4e_G(S)$, and similarly for G'. So the condition that all cuts are ϵ-approximated is just

$$(1-\epsilon)\, x^\top L_{G'} x \le x^\top L_G x \le (1+\epsilon)\, x^\top L_{G'} x \qquad \text{for all } x \in \{-1,1\}^n. \tag{1.9.2}$$

It would be even better if this worked for all $x \in \mathbb{R}^n$. By scaling, that is the same as asking for it for all $x \in [-1,1]^n$.

Definition 1.9.4 (Spectral sparsifier)**.** A weighted graph G' is a *spectral sparsifier* of G (with parameter ϵ) if (1.9.2) holds for all $x \in \mathbb{R}^n$.

A reasonable conjecture is that combinatorial sparsifiers are spectral sparsifiers. This is reasonable, but false!

Example 1.9.5 (A combinatorial sparsifier that is not spectral)**.** Let n be even and let k be small compared with n. Put the vertices $0, 1, \ldots, n-1$ around a circle, join i and j whenever $|i-j| \le k \pmod n$, and add one more edge, from 0 to $n/2$. Call this graph G (Fig. 1.36). Its minimum cut is $2k$. So removing the one edge $(0, n/2)$ gives a combinatorial sparsifier G' of G with $\epsilon = \frac{1}{2k}$. Removing one edge changes each cut by at most 1, and every cut of G and of G' has at least $2k$ edges.

Now let x record the distance from vertex 0 around the circle: $x = (0, 1, \ldots, \frac{n}{2}, \frac{n}{2}-1, \ldots, 1)$, that is, $x_i = \min(i, n-i)$. Then

$$x^\top L_{G'} x = \sum_{(i,j)\in E(G')} (x_i - x_j)^2 = \Theta(nk^3),$$

while

$$x^\top L_G x = \sum_{(i,j)\in E(G')} (x_i - x_j)^2 + \left(\frac{n}{2}\right)^2 = \Theta(nk^3) + \frac{n^2}{4}.$$

For the first estimate, note that each vertex has neighbors at circular distance $1, \ldots, k$ on each side. And $|x_i - x_j|$ is just the circular distance, except on the $O(k^2)$ edges that straddle vertex 0 or vertex $n/2$. So the sum is $\Theta(n\sum_{j=1}^{k} j^2) = \Theta(nk^3)$. So for G' to satisfy (1.9.2) for this x we need $\epsilon = \Omega(\frac{n}{k^3})$, which can be very big (for example, for constant k).

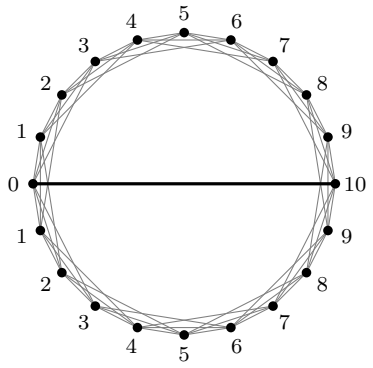


Figure 1.36: The graph of Example 1.9.5 for $n = 20$, $k = 3$: each vertex is joined to the vertices at circular distance at most k, plus the chord from vertex 0 to vertex $n/2$ (thick). Vertices are labeled by their entry of the test vector x, the distance from vertex 0 around the circle.

1.9.3 Matrix and graph inequalities

To compare graphs spectrally we will compare their Laplacians, so we need an answer to the following question: how do you define the notion that one symmetric matrix is "bigger" than another? Everything is symmetric unless otherwise noted. In this subsection M and N denote arbitrary symmetric $n \times n$ matrices (N is not the normalized Laplacian here).

The most obvious answer is entrywise: $M \succeq N$ if $m_{ij} \geq n_{ij}$ for all i, j. This is not great for our purposes: it is not basis independent, and it says nothing about eigenvalues.

Another try: $M \succeq N$ if the ith eigenvalue of M is at least the ith eigenvalue of N, for all i. This is better, but it is *too* basis independent: it ignores the eigenvectors altogether. Do we really want to say both

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \geq \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad \text{and} \qquad \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \geq \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}?$$

Both matrices have eigenvalues 1 and -1, so this definition makes each of them at least as big as the other, although they are different.

Definition 1.9.6 (Loewner order)**.** We say $M \succeq N$ if $x^\top M x \geq x^\top N x$ for all $x \in \mathbb{R}^n$.

Some properties (several are on the problem set):

- If $M \succeq N$ and $N \succeq M$, then $M = N$.
- $M \succeq 0$ if and only if M is positive semidefinite.
- $M \succeq N$ if and only if $M - N$ is positive semidefinite.
- If $M_1 \succeq N_1$ and $M_2 \succeq N_2$, then $M_1 + M_2 \succeq N_1 + N_2$.
- Other nice properties, introduced as needed.

Claim 1.9.7. *Let M have eigenvalues $\lambda_1(M) \leq \cdots \leq \lambda_n(M)$ and N have eigenvalues $\lambda_1(N) \leq \cdots \leq \lambda_n(N)$. If $M \succeq N$, then $\lambda_i(M) \geq \lambda_i(N)$ for all i.*

(Proof omitted; the proof is on the problem set.)

We add one more property, which we use repeatedly.

Proposition 1.9.8. *If $M \succeq N$, then $X^\top M X \succeq X^\top N X$ for any matrix X (of compatible dimensions).*

Proof. For every vector y, $y^\top X^\top M X y = (Xy)^\top M (Xy) \geq (Xy)^\top N (Xy) = y^\top X^\top N X y$. □

We now know how to define inequalities between graphs G and H on the same vertex set.

Definition 1.9.9 (Graph inequalities)**.** We say $G \succeq H$ if $L_G \succeq L_H$.

Claim 1.9.10. *Let $G = (V, E_G, w_G)$ and $H = (V, E_H, w_H)$ be weighted graphs on the same vertex set such that $w_G(i,j) \geq w_H(i,j)$ for all pairs (i,j) (with $w(i,j) = 0$ when (i,j) is not an edge). Then $G \succeq H$.*

Exercise 1.9.11. Prove Claim 1.9.10.

Example 1.9.12 (Expanders and the complete graph)**.** Let $H = \frac{d}{n} K_n$, the complete graph with every edge weighted d/n. Let G be a d-regular expander, with adjacency eigenvalues $d = \alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n$ as in Section 1.8.3, and suppose $\alpha_2 = d - \lambda_2 \leq c = \epsilon d$. Let us require the "two-sided" expander property[5]

$$\bar{\alpha} = \max(|\alpha_2|, |\alpha_n|) \leq \epsilon d.$$

[5] A two-sided condition has to bound the most negative eigenvalue α_n too (a bipartite graph has $\alpha_n = -d$). The next sentence needs $|\alpha_i| \leq \epsilon d$ for all $i \geq 2$.

This means that G is not close to being disconnected or bipartite. It implies that all nonzero eigenvalues of L_G lie in $[(1-\epsilon)d, (1+\epsilon)d]$, and therefore

$$(1-\epsilon)H \preceq G \preceq (1+\epsilon)H. \tag{1.9.3}$$

This makes precise the view (Section 1.8.3) that expanders behave like rescaled complete graphs.

Exercise 1.9.13. Prove (1.9.3), and use it to prove the expander mixing lemma (Theorem 1.8.5).

1.9.4 Spectral sparsification

With the Loewner order we can state the goal in matrix form. We have a dense graph G and want to produce a sparse graph H such that

$$(1-\epsilon)L_H \preceq L_G \preceq (1+\epsilon)L_H. \tag{1.9.4}$$

Here "sparse" means $n \operatorname{poly}(\log n, 1/\epsilon)$ edges. Below we prove the version with the roles of G and H exchanged,

$$(1-\epsilon)L_G \preceq L_H \preceq (1+\epsilon)L_G. \tag{1.9.5}$$

The two are equivalent up to a constant factor in ϵ. For $\epsilon \le 1/2$, (1.9.5) gives $L_G \preceq \frac{1}{1-\epsilon}L_H \preceq (1+2\epsilon)L_H$ and $L_G \succeq \frac{1}{1+\epsilon}L_H \succeq (1-\epsilon)L_H$. The converse is the same computation.

Recall that the Benczúr–Karger sparsifier did not quite get this: it gave $(1-\epsilon)x^\top L_H x \le x^\top L_G x \le (1+\epsilon)x^\top L_H x$ for $\{\pm 1\}$-vectors, and we need it for all vectors.

In the Benczúr–Karger analysis the easy case was "well-connected" graphs. The same is true for us, but with a slightly different notion of connectedness. Instead of graphs with no small cuts, we look at graphs with no sparse cuts, or, equivalently-ish, at graphs with big λ_2. It turns out that uniform random sampling works for graphs with good expansion, but problems analogous to those of Benczúr–Karger appear when there are sparse cuts. (Well, almost: one needs to sample with probability inversely proportional to the degree. This will follow from the proof of the general scheme.)

What we will show, and what is true.

- We will show how to construct spectral sparsifiers with $O(n \log n)$ edges (for constant ϵ) in polynomial time. The analysis is cleaner and more geometric than the Benczúr–Karger analysis, and it is a good example of how generalizing can sometimes make things easier, not harder.

- One can actually construct them in nearly linear time (Section 1.9.9). The main tool is an algorithm for quickly solving linear systems involving graph Laplacians; the construction also uses a theorem on concentration of measure, the Johnson–Lindenstrauss lemma (Section 2.7).

- With more work (and a little more running time), one can construct spectral sparsifiers with $O(n)$ edges [BSS12], a result of Batson, Spielman, and Srivastava. Note that such a sparsifier of a complete graph is an expander. Well, almost: expanders have constant maximum degree, whereas this only guarantees constant average degree.

1.9.5 The algorithm

The algorithm is very simple, and it has the same structure as Benczúr–Karger, just with different probabilities.

Algorithm 1.9.14 (Sparsification by sampling)**.**

1. Compute a probability p_e for each edge e.
2. Sample each edge with probability p_e, independently of the others.
3. If an edge is selected, include it in H with weight $1/p_e$.

The probabilities are based on a linear-algebraic notion of importance, instead of a graph-theoretic one; they have a very nice interpretation in terms of effective resistances of circuits (Section 1.10.3). The main tool in the analysis is a matrix analogue of the Chernoff–Hoeffding bound, and the technical tool is the pseudoinverse. (In the analysis of Section 1.9.8, a fixed number N of edges is drawn with replacement, rather than each edge being kept independently.)

1.9.6 Pseudoinverses

How do you invert a matrix that is not invertible? Let M be an $n \times n$ symmetric matrix and diagonalize it:

$$M = \sum_{i=1}^{n} \lambda_i v_i v_i^\top,$$

with $v_1, \dots, v_n$ orthonormal eigenvectors. If all eigenvalues are nonzero,

$$M^{-1} = \sum_{i=1}^{n} \frac{1}{\lambda_i} v_i v_i^\top.$$

When M is degenerate, we get the pseudoinverse by throwing away the zero eigenvalues.

Definition 1.9.15 (Moore–Penrose pseudoinverse)**.** For symmetric M as above,

$$M^\dagger = \sum_{i:\lambda_i \neq 0} \frac{1}{\lambda_i} v_i v_i^\top.$$

One can define the pseudoinverse for nonsquare and nonsymmetric matrices too, but we will not need it. It has lots of nice properties; we will use these:

- $\ker(M) = \ker(M^\dagger)$;
- $MM^\dagger = \sum_{i:\lambda_i \neq 0} v_i v_i^\top$, the projection onto the span of the eigenvectors with nonzero eigenvalues, which we denote $I_{\mathrm{im}(M)}$;
- so $MM^\dagger$ equals the identity when restricted to $\mathrm{im}(M)$.

For positive semidefinite M we also use the square root $(M^\dagger)^{1/2} = \sum_{i:\lambda_i \neq 0} \lambda_i^{-1/2} v_i v_i^\top$, which satisfies $(M^\dagger)^{1/2} M (M^\dagger)^{1/2} = I_{\mathrm{im}(M)}$, and similarly $M^{1/2} = \sum_i \lambda_i^{1/2} v_i v_i^\top$.

1.9.7 Estimating a covariance matrix

Suppose we are given a distribution on random vectors $y \in \mathbb{R}^n$. How many independent samples do we need to estimate the covariance matrix $M = \mathbb{E}[yy^\top]$? That is, if we get samples $y_1, \dots, y_q$, how big do we need to make q so that $\frac{1}{q}\sum_i y_i y_i^\top$ is close to M?

What is the right notion of relative error? Since M is positive semidefinite, we can define it as the ϵ such that

$$(1-\epsilon)M \preceq \frac{1}{q}\sum_i y_i y_i^\top \preceq (1+\epsilon)M. \tag{1.9.6}$$

Rescaling to isotropic position. If M is full rank, (1.9.6) is equivalent to

$$(1-\epsilon)M^{-1/2}MM^{-1/2} \preceq \frac{1}{q}\sum_i M^{-1/2}y_i y_i^\top M^{-1/2} \preceq (1+\epsilon)M^{-1/2}MM^{-1/2}$$

(apply Proposition 1.9.8 with $X = M^{-1/2}$, and with $X = M^{1/2}$ for the converse), that is,

$$(1-\epsilon)I \preceq \frac{1}{q}\sum_i z_i z_i^\top \preceq (1+\epsilon)I, \qquad \text{where } z = M^{-1/2}y,\ \mathbb{E}[zz^\top] = I.$$

So the question becomes the following.

> **Question.** If $\mathbb{E}[zz^\top] = I$, how many independent samples $z_1, \dots, z_q$ are needed to get $(1-\epsilon)I \preceq \frac{1}{q}\sum_i z_i z_i^\top \preceq (1+\epsilon)I$?

Intuition: the diagonal case. Suppose $y = r_i\chi_i$ with probability $p_i = 1/r_i^2$ (so $\sum_i 1/r_i^2 = 1$). Then

$$\mathbb{E}[yy^\top] = \operatorname{diag}(p_1 r_1^2, \dots, p_n r_n^2) = I.$$

Every sample is a multiple of a standard basis vector, so the empirical matrix $\frac{1}{q}\sum_j y_j y_j^\top$ is diagonal, and spectral approximation is equivalent to approximating each diagonal entry.

- To ϵ-approximate the ith diagonal entry with probability at least $1-\delta$, take $q = \Theta(\frac{1}{p_i}\frac{\log(1/\delta)}{\epsilon^2})$. The ith entry is r_i^2/q times the number of samples equal to $r_i\chi_i$, which is binomial with mean qp_i. By the Chernoff bound, it is within a factor $1 \pm \epsilon$ of its mean with probability $1-\delta$ once $qp_i\epsilon^2 \gtrsim \log(1/\delta)$.
- To get all entries, make $\delta \ll 1/n$ for every i and take a union bound:

$$q = \Theta\Big(\frac{1}{\min_i p_i}\frac{\log n}{\epsilon^2}\Big).$$

- **Best case:** $p_i = 1/n$ for all i (equivalently, $\|y\|^2 = n$ always). Then $q = \Theta(\frac{n\log n}{\epsilon^2})$. The factor $n\log n$ comes from coupon collection (every direction must be sampled), and the factor $1/\epsilon^2$ is what is needed to get the right variance.
- **Bad case:** some p_i is small, that is, the contribution in some direction comes from rare, large vectors.

A matrix tail bound. For general distributions we use the following theorem.

Theorem 1.9.16 (Rudelson–Vershynin [RV07])**.** *There is an absolute constant k with the following property. Let y be a random vector with $\|y\| \le t$ and $\|\mathbb{E}[yy^\top]\| \le 1$, and let $y_1, \dots, y_q$ be independent samples of y. Then*

$$\mathbb{E}\left[\left\|\mathbb{E}[yy^\top] - \frac{1}{q}\sum_{i=1}^{q} y_i y_i^\top\right\|\right] \le kt\sqrt{\frac{\log q}{q}}. \tag{1.9.7}$$

(Proof omitted; it is a concentration-of-measure theorem; results of this kind are the subject of Section 2.6.)

The bound (1.9.7) is less than ϵ for $q \ge \Omega(\frac{t^2}{\epsilon^2}\log\frac{t^2}{\epsilon^2})$ (and actually for slightly smaller q). In the diagonal example, $\|y\|^2 = r_i^2 = 1/p_i$, so $t^2 = 1/\min_i p_i$. In the best case $t^2 = n$, this essentially matches the coupon-collection count $\Theta(n \log n/\epsilon^2)$.

Exercise 1.9.17. Some questions about Theorem 1.9.16:

1. Why do we need the assumption $\|y\| \le t$?
2. Why is the theorem similar to the Chernoff bound?
3. What is the right equivalent of the importance sampling of Benczúr–Karger? (Answered in Section 1.9.8.)

1.9.8 Sparsification by sampling

Throughout this subsection G is a connected unweighted graph, so $\operatorname{rank} L_G = n-1$ and $\operatorname{im}(L_G) = \mathbf{1}^\perp$. We want a weighted graph H with

$$\begin{aligned} & (1-\epsilon)x^\top L_G x \le x^\top L_H x \le (1+\epsilon)x^\top L_G x \quad \forall x \\ \iff & (1-\epsilon)L_G \preceq L_H \preceq (1+\epsilon)L_G \\ \iff & \|(L_G^\dagger)^{1/2} L_H (L_G^\dagger)^{1/2} - I_{\operatorname{im}(L_G)}\| \le \epsilon. \end{aligned} \tag{1.9.8}$$

For the last equivalence, write $\Pi = (L_G^\dagger)^{1/2} L_H (L_G^\dagger)^{1/2}$. Conjugating the middle condition by $(L_G^\dagger)^{1/2}$ (Proposition 1.9.8) gives $(1-\epsilon)I_{\operatorname{im}(L_G)} \preceq \Pi \preceq (1+\epsilon)I_{\operatorname{im}(L_G)}$. Both Π and $I_{\operatorname{im}(L_G)}$ vanish on $\mathbf{1}$ and map $\mathbf{1}^\perp$ to itself, so this just says $\|\Pi - I_{\operatorname{im}(L_G)}\| \le \epsilon$. Conjugating back by $L_G^{1/2}$ gives the converse, since $L_G^{1/2}\Pi L_G^{1/2} = I_{\operatorname{im}(L_G)} L_H I_{\operatorname{im}(L_G)} = L_H$ (as $L_H\mathbf{1} = 0$) and $L_G^{1/2} I_{\operatorname{im}(L_G)} L_G^{1/2} = L_G$.

Putting the edges in isotropic position. Write

$$L_G = \sum_{e\in E} L_e = \sum_{e\in E} b_e b_e^\top,$$

where $b_e = \chi_u - \chi_v$ for $e = (u, v)$. Then

$$I_{\operatorname{im}(L_G)} = (L_G^\dagger)^{1/2} L_G (L_G^\dagger)^{1/2} = \sum_{e\in E} q_e q_e^\top, \qquad \text{where } q_e = (L_G^\dagger)^{1/2} b_e,$$

and

$$\|q_e\|^2 = b_e^\top (L_G^\dagger)^{1/2} (L_G^\dagger)^{1/2} b_e = b_e^\top L_G^\dagger b_e. \tag{1.9.9}$$

We want all the vectors to have the same length, so let

$$\tau_e = \sqrt{\frac{n-1}{\|q_e\|^2}}\, q_e, \qquad \|\tau_e\| = \sqrt{n-1},$$

and make a distribution that picks τ_e with probability

$$p_e = \frac{\|q_e\|^2}{n-1}. \tag{1.9.10}$$

These probabilities add up to 1 because

$$\sum_e \|q_e\|^2 = \sum_{e\in E} q_e^\top q_e = \operatorname{tr}\Big(\sum_{e\in E} q_e q_e^\top\Big) = \operatorname{tr}\big(I_{\operatorname{im}(L_G)}\big) = n-1.$$

If τ denotes a vector drawn from this distribution, then

$$\mathbb{E}[\tau\tau^\top] = \sum_e p_e \tau_e \tau_e^\top = \sum_e q_e q_e^\top = I_{\operatorname{im}(L_G)}.$$

Conclusion of the proof. Pick N vectors τ_e independently, with replacement, from this distribution. (In this subsection N is the number of samples, not the normalized Laplacian.) Give each edge e the weight

$$\frac{n-1}{N\|q_e\|^2} \times (\text{number of times } e \text{ is chosen}) = \frac{\text{number of times } e \text{ is chosen}}{Np_e}, \tag{1.9.11}$$

and let H be the resulting weighted graph.[6] Note that edges with bigger vectors q_e get picked with higher probability, but are scaled down more, just as in Benczúr–Karger. With these weights, since $\tau_e\tau_e^\top = \frac{1}{p_e} q_e q_e^\top = \frac{1}{p_e}(L_G^\dagger)^{1/2} L_e (L_G^\dagger)^{1/2}$, the average of the sampled matrices is

$$\frac{1}{N}\sum_{j=1}^{N} \tau_{e_j}\tau_{e_j}^\top = (L_G^\dagger)^{1/2} L_H (L_G^\dagger)^{1/2},$$

where $e_1, \dots, e_N$ are the sampled edges. By the theorem of Rudelson–Vershynin (Theorem 1.9.16, with $t = \sqrt{n-1}$ and $\|\mathbb{E}[\tau\tau^\top]\| = \|I_{\operatorname{im}(L_G)}\| = 1$),

$$\mathbb{E}\Big[\big\|(L_G^\dagger)^{1/2} L_H (L_G^\dagger)^{1/2} - I_{\operatorname{im}(L_G)}\big\|\Big] \le k\sqrt{n-1}\sqrt{\frac{\log N}{N}} < \epsilon \qquad \text{if } N = \Theta\Big(\frac{n\log n}{\epsilon^2}\Big). \tag{1.9.12}$$

By Markov's inequality, the norm is at most 2ϵ with probability at least $1/2$. Then H is a spectral sparsifier with parameter 2ϵ by (1.9.8), and it has at most $N = O(n\log n/\epsilon^2)$ edges.

This still leaves some questions:

- How should we interpret the p_e? (Corollary 1.10.3: up to the factor $n-1$, they are effective resistances.)
- Can we compute the p_e quickly enough for this to be useful?

[6] With the factor $n-1$ in the weight, $\mathbb{E}[L_H] = \sum_e \frac{Np_e}{Np_e} L_e = L_G$.

1.9.9 Algorithmics of the construction

It is easy to see that the construction runs in polynomial time: the whole procedure is constructive and uses only standard linear algebra operations. The bottleneck is computing the p_e, which involves a matrix (pseudo)inversion and a few multiplications. One can actually do it in nearly linear time. This requires two main components:

- a nearly linear time algorithm for solving linear systems of the form $Lx = b$ when L is a Laplacian, which lets one compute a single effective resistance very quickly;
- a way to compute all the p_e by solving logarithmically many linear systems, which uses the Johnson–Lindenstrauss lemma.

1.9.10 The general setup: leverage scores

A quick comment: the argument did not really use much about graphs. In this subsection $v_1, \dots, v_m$ are arbitrary vectors in $\mathbb{R}^n$ (not vertices or eigenvectors). The general setup is a matrix

$$M = \sum_{t=1}^{m} v_t v_t^\top, \qquad v_t \in \mathbb{R}^n, \quad m \gg n,$$

which we want to approximate using fewer of the vectors v_t. One can get a spectral ϵ-approximation of M from $O(n \log n/\epsilon^2)$ samples by sampling with probabilities proportional to the *leverage scores*

$$v_t^\top M^{-1} v_t.$$

The proof is just the one in Section 1.9.8, with L_G replaced by M and b_e by v_t. If M is invertible, the vectors $q_t = M^{-1/2} v_t$ satisfy $\sum_t q_t q_t^\top = I$ and $\|q_t\|^2 = v_t^\top M^{-1} v_t$. So the leverage scores add up to $\operatorname{tr}(I) = n$. We sample $\tau_t = \sqrt{n/\|q_t\|^2}\, q_t$ with probability $\|q_t\|^2/n$ and apply the Rudelson–Vershynin bound with $t = \sqrt{n}$. If M is singular, $M^\dagger$ replaces M^{-1} and $\operatorname{rank} M$ replaces n. The graph case is $M = L_G$ and $v_e = b_e$, with $\operatorname{rank} L_G = n - 1$.

1.10 Electrical Networks and Effective Resistance

Spectral sparsification samples each edge e with probability proportional to $b_e^\top L_G^\dagger b_e$ (Section 1.9.8). This section explains what these numbers mean: if we view the graph as an electrical network with a unit resistor on every edge, $b_e^\top L_G^\dagger b_e$ is the *effective resistance* of e. We review electrical networks, write Ohm's law and conservation of current with the incidence matrix and the Laplacian, and describe effective resistance both as a potential difference and as an energy. We then connect electrical potentials to random walks: the probability that a random walk reaches one vertex before another is given by electrical potentials. Section 1.11 gives one more application of the electrical view.

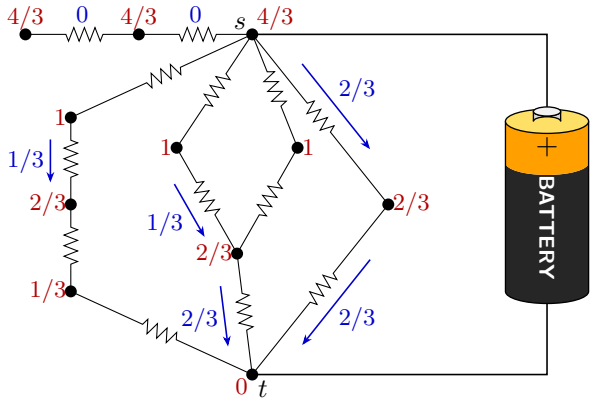
1.10.1 Electrical networks

Graphs will become circuits: an edge e with weight w_e becomes a resistor with resistance $r_e = 1/w_e$. (We also write $c_e = w_e$, the *conductance* of the resistor.) Each vertex a has a

potential, or voltage, $\psi(a)$, and each edge e carries a *current* $i(e)$. For an edge $e = (a, b)$, current and voltage obey Ohm's law,

$$i(e) = \frac{\psi(a) - \psi(b)}{r_e}, \tag{1.10.1}$$

where a positive $i(e)$ means that current flows from a to b. The network has *internal nodes* and *boundary nodes*. At internal nodes, the total current in equals the total current out. At boundary nodes, current enters or leaves the network from outside, for example at the two terminals of a battery connected to the network (Fig. 1.37).



red: potentials, blue: currents; every resistor is 1

Figure 1.37: An electrical network driven by a battery.

Effective resistance

Treat each edge as a resistor of resistance 1. (If an edge had capacity c, we would use resistance $1/c$.) The *effective resistance* $R_{\text{eff}}(e)$ of an edge e is the resistance between the endpoints of e in the whole network.

Example 1.10.1 (A triangle)**.** Take the triangle on the vertices u, v, a with unit resistors, and let $e = (u, v)$. Between u and v there are two routes: the edge (u, v), of resistance 1, and the path through a, whose resistance is 2. The resistance from u to v is therefore

$$R_{\text{eff}}(u, v) = \frac{1}{1/2 + 1/1} = \frac{2}{3}.$$

Equivalently, $R_{\text{eff}}(e)$ is the potential difference between the endpoints of e when one unit of current flows from one endpoint to the other. In the triangle, send one unit of current from u to v: two thirds of it go along the edge (u, v) and one third along the path through a. Ohm's law in the form $\psi = i\,r$ (potential difference equals current times resistance) gives, with the potential of a set to 0,

$$\psi(u) = \tfrac{1}{3}, \qquad \psi(a) = 0, \qquad \psi(v) = -\tfrac{1}{3}$$

(Fig. 1.38), and the potential difference is $\psi(u) - \psi(v) = \frac{2}{3}$, as before.

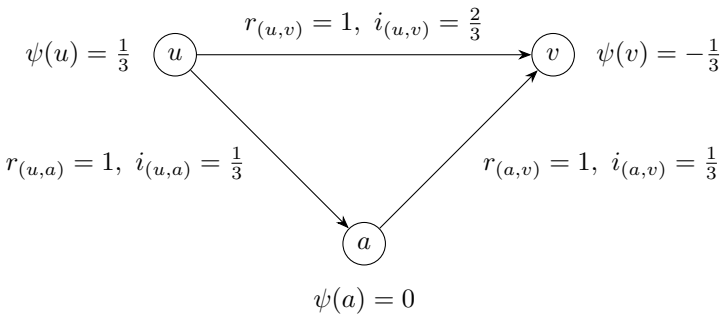


Figure 1.38: One unit of current from u to v in the triangle of Example 1.10.1: resistances, currents (arrows give the direction of flow) and potentials.

1.10.2 Laplacians and electrical flow

Now let G be a connected graph whose edges have conductances $c_e > 0$ (an unweighted graph has $c_e = 1$), resistances $r_e = 1/c_e$, and let $C = \operatorname{diag}(c)$. Orient each edge arbitrarily, and let $B \in \mathbb{R}^{m \times n}$ be the signed edge-vertex incidence matrix:

$$B(e,x) = \begin{cases} 1 & \text{if } x \text{ is the head of } e, \\ -1 & \text{if } x \text{ is the tail of } e, \\ 0 & \text{otherwise.} \end{cases}$$

For an edge $e = (u, v)$ with head u and tail v, the row of B indexed by e is $b_e^\top = (\chi_u - \chi_v)^\top$, and

$$L = B^\top C B = \sum_{e \in E} c_e b_e b_e^\top. \tag{1.10.2}$$

Let $\psi \in \mathbb{R}^V$ be the potentials and $i \in \mathbb{R}^E$ the currents, where a positive $i(e)$ means current flowing from the head of e to its tail. Since $(B\psi)(e) = \psi(u) - \psi(v)$, Ohm's law (1.10.1) for all edges at once reads

$$i = CB\psi, \qquad i \in \mathbb{R}^E,\ \psi \in \mathbb{R}^V. \tag{1.10.3}$$

Let $i_{\text{ext}} \in \mathbb{R}^V$ record the current that enters the network from outside at each vertex; it is zero at internal nodes. Conservation of current says

$$i_{\text{ext}} = B^\top i, \qquad i_{\text{ext}} \in \mathbb{R}^V, \tag{1.10.4}$$

since $(B^\top i)(x) = \sum_{e:\, x \text{ head of } e} i(e) - \sum_{e:\, x \text{ tail of } e} i(e)$ is the net current flowing out of x along the edges, and this has to equal the current entering x from outside. Substituting (1.10.3) into (1.10.4) and using (1.10.2),

$$i_{\text{ext}} = L\psi, \qquad \psi = L^\dagger i_{\text{ext}}. \tag{1.10.5}$$

For the second formula, recall that $\ker L = \operatorname{span}(\mathbf{1})$, since G is connected. The external currents add up to zero, $\mathbf{1}^\top i_{\text{ext}} = (B\mathbf{1})^\top i = 0$, so i_{ext} lies in $\operatorname{im}(L) = \mathbf{1}^\perp$ and $LL^\dagger i_{\text{ext}} = i_{\text{ext}}$. Every solution of $L\psi = i_{\text{ext}}$ differs from $L^\dagger i_{\text{ext}}$ by a multiple of $\mathbf{1}$, which changes neither the potential differences nor the currents.

Definition 1.10.2 (Effective resistance)**.** Let G be connected, let s, t be vertices, and let $b_{st} = \chi_s - \chi_t$. The *effective resistance* $R_{\text{eff}}(s,t)$ is the potential difference $\psi(s) - \psi(t)$ when one unit of current flows from s to t, that is, when $i_{\text{ext}} = b_{st}$. By (1.10.5) the potentials are then $\psi = L^\dagger b_{st}$, so

$$R_{\text{eff}}(s,t) = b_{st}^\top \psi = b_{st}^\top L^\dagger b_{st}. \tag{1.10.6}$$

For an edge $e = (u,v)$ we write $R_{\text{eff}}(e) = R_{\text{eff}}(u,v) = b_e^\top L^\dagger b_e$.

Electrical flows and energy

Another way to understand effective resistance is through energy. In this subsection $\mathcal{E}(f)$ denotes the *energy* of a flow $f \in \mathbb{R}^E$,

$$\mathcal{E}(f) = \sum_{e \in E} f(e)^2 r_e = f^\top C^{-1} f. \tag{1.10.7}$$

The flow coming from potentials ψ is $f = CB\psi$, by Ohm's law (1.10.3), and its energy is

$$f^\top C^{-1} f = (\psi^\top B^\top C) C^{-1} (CB\psi) = \psi^\top B^\top CB\psi = \psi^\top L \psi.$$

For the electrical potentials of one unit of current from s to t, $\psi = L^\dagger b_{st}$, this gives

$$(L^\dagger b_{st})^\top L (L^\dagger b_{st}) = b_{st}^\top L^\dagger L L^\dagger b_{st} = b_{st}^\top L^\dagger b_{st} = R_{\text{eff}}(s,t), \tag{1.10.8}$$

where $L^\dagger L L^\dagger = L^\dagger$ since $L^\dagger L L^\dagger = \sum_{i:\lambda_i \neq 0} \frac{1}{\lambda_i} \cdot \lambda_i \cdot \frac{1}{\lambda_i} v_i v_i^\top$. So the effective resistance between s and t is the energy of the unit electrical flow from s to t.

1.10.3 Sparsification by effective resistances

Comparing (1.10.6) with (1.9.9) answers the question left open in Section 1.9.8: the numbers $\|q_e\|^2$ are effective resistances.

Corollary 1.10.3. *Let G be a connected unweighted graph. The sampling probabilities* (1.9.10) *of the spectral sparsification algorithm are*

$$p_e = \frac{\|q_e\|^2}{n-1} = \frac{R_{\text{eff}}(e)}{n-1}.$$

In particular, $\sum_{e \in E} R_{\text{eff}}(e) = n - 1$ (this identity is known as Foster's theorem).

Proof. By (1.9.9) and (1.10.6), $\|q_e\|^2 = b_e^\top L_G^\dagger b_e = R_{\text{eff}}(e)$. The sum is $\sum_e \|q_e\|^2 = n - 1$ (Section 1.9.8). □

So the spectral analogue of the importance sampling of Benczúr–Karger [BK96] is sampling by effective resistance. Edges that are the only connection between two parts of the graph get sampled often. If e is a bridge, all of any current between its endpoints crosses e, so $R_{\text{eff}}(e) = 1$. Edges in dense, well-connected parts are sampled rarely, and get large weights when they are sampled. In the complete graph K_n, $L^\dagger = \frac{1}{n}(I - \frac{1}{n}J)$, so every edge has $R_{\text{eff}}(e) = 2/n$.

Figure 1.39 is a picture of a graph that illustrates effective resistances. Thicker edges have larger effective resistance: the edges inside the dense clusters are thin, and the edges that are among few connections between parts of the graph are thick.

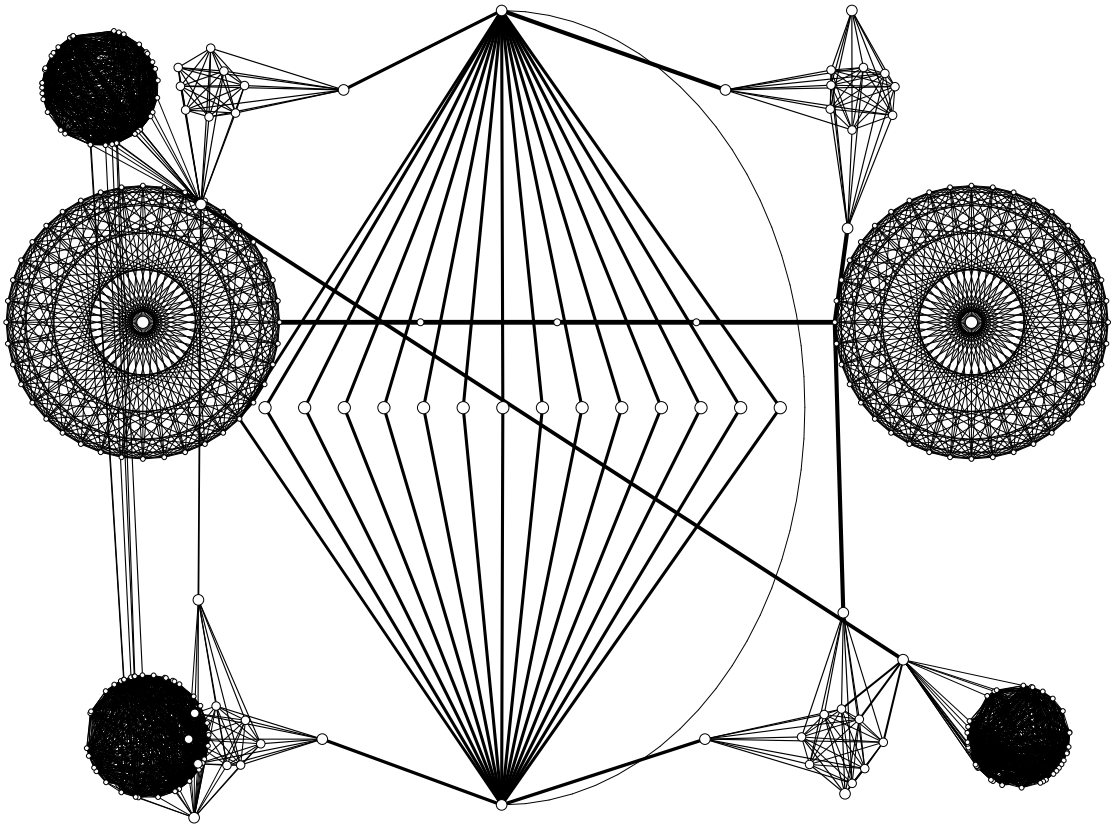


Figure 1.39: Effective resistance examples. The dense pieces are random graphs, and each edge is drawn with line width proportional to its effective resistance.

1.10.4 Random walks and electrical potentials

Motivating questions

A classic problem in probability is the *drunkard's walk*. The vertices $0, 1, \ldots, 8$ lie on a path (Fig. 1.40); vertex 0 is the bar and vertex 8 is the house. A drunkard goes left or right with probability $1/2$ at each step, and stops on reaching either the bar or the house.

- If the drunkard starts at node i, what is the probability of getting home before reaching the bar?
- What if the drunkard sort of knows which way the house is, and goes right with probability 0.6 and left with probability 0.4?
- One could ask the same kind of question in other ways. Go to a casino with \$100, and repeatedly gamble \$1 per hand on blackjack. What is the probability that you triple your money before you go broke? How does this probability change if you count cards?

We can modify the question, to make it a bit harder, and so as not to leave the impression that the theory is just about drinking and gambling. After finishing a problem set, a sleep-deprived student wanders the MIT tunnels, starting under building 2. The student wants to go home to building 62, and leaves the tunnels at building 54. If the student reaches the Stata Center, they

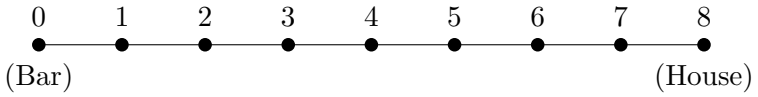


Figure 1.40: The drunkard's walk on the path $0, 1, \ldots, 8$.

get disoriented and pass out. What is the probability that the student makes it home before passing out? We model the tunnels as a graph whose *absorbing states* are the exit at 54 and the vertices at the Stata Center (Fig. 1.41).

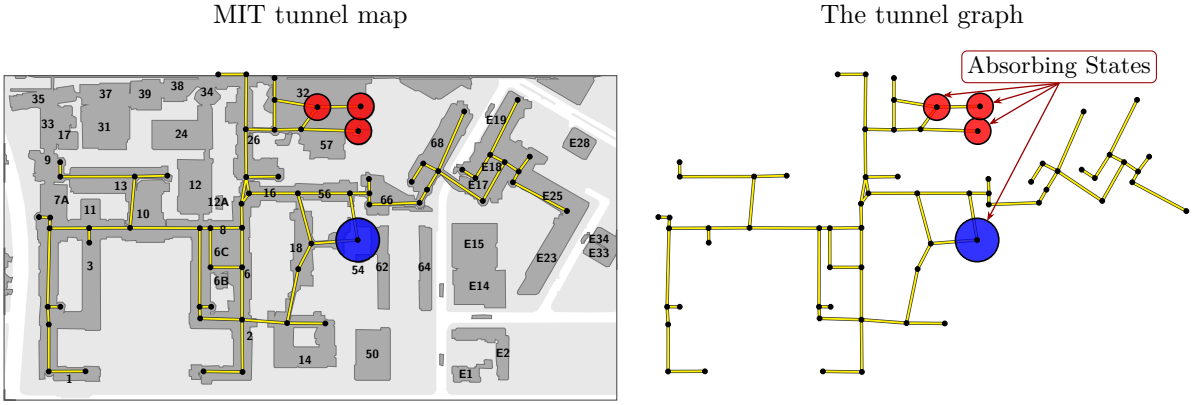


Figure 1.41: The MIT tunnels as a graph with absorbing states (the map is schematic, after the MIT Department of Facilities tunnel map).

Electrical potentials and walk probabilities

Here is the general question. We have a graph G and vertices x, s, and t, all distinct. Start a random walk at x, and stop when it hits s or t. Let $F(x)$ be the probability that the walk hits s before t. How can we compute F? We assume G is connected, so the walk hits $\{s, t\}$ with probability 1.

Clearly $F(s) = 1$ and $F(t) = 0$. At a vertex $x \notin \{s, t\}$, the walk goes to each neighbor with probability $\frac{1}{d(x)}$. Conditioning on the first step,

$$\begin{aligned} F(x) &= \sum_{y:\,(x,y)\in E} \Pr[\text{go from } x \text{ to } y]\ \Pr[\text{hit } s \text{ before } t \text{ if we start at } y] \\ &= \sum_{y:\,(x,y)\in E} \frac{1}{d(x)} F(y), \qquad \text{so} \qquad d(x)F(x) - \sum_{y:\,(x,y)\in E} F(y) = 0. \end{aligned} \tag{1.10.9}$$

The left side of the last equation is $(LF)(x)$, so if we treat F as a vector, (1.10.9) says that LF vanishes outside $\{s, t\}$. So $LF = b_{st}$, up to a scalar factor. The factor is $1/R_{\text{eff}}(s, t)$, which need not equal 1. Since $\mathbf{1}^\top L = 0$, we have $(LF)(t) = -(LF)(s)$, so $LF = c\, b_{st}$ with $c = (LF)(s)$. Let $\psi = L^\dagger b_{st}$ be the potentials of one unit of current from s to t. Then $L(F - c\psi) = 0$, so $F - c\psi$ is a multiple of $\mathbf{1}$. Taking the difference of the values at s and t gives

$1 = F(s) - F(t) = c(\psi(s) - \psi(t)) = cR_{\text{eff}}(s,t)$. So

$$F = \frac{\psi - \psi(t)\mathbf{1}}{R_{\text{eff}}(s,t)}, \tag{1.10.10}$$

the potentials when a current of $1/R_{\text{eff}}(s,t)$ flows from s to t, shifted so that t has potential 0 (and then s has potential 1). In particular, F is determined uniquely by (1.10.9) and the values $F(s) = 1$, $F(t) = 0$.

So the answer is given by electrical potentials! This can be generalized, and there is another way to think about it: *discrete harmonic functions*. By (1.10.9), F is harmonic at every vertex other than s and t: its value at x is the average of its values at the neighbors of x.

Example 1.10.4 (The drunkard's walk)**.** On the path of Fig. 1.40, take $s = 8$ (the house) and $t = 0$ (the bar). Harmonicity says $F(i) = \frac{1}{2}(F(i-1) + F(i+1))$ for $1 \le i \le 7$, so F is linear along the path. With $F(8) = 1$ and $F(0) = 0$, this gives $F(i) = i/8$. In electrical terms, the eight unit resistors are in series, so $R_{\text{eff}}(8,0) = 8$, and a current of $1/8$ drops the potential by $1/8$ across each edge. The tunnel question has the same form: s is the exit at building 54, and the vertices at the Stata Center together play the role of t (set $F = 0$ at all of them).

1.11 Spanning Trees and the Matrix-Tree Theorem

Kirchhoff's matrix-tree theorem expresses the number of spanning trees of a graph through its Laplacian. After some linear algebra facts, this section proves the theorem, first for trees and then for general graphs. It then gives one more application of the electrical view of Section 1.10: the probability that an edge belongs to a uniformly random spanning tree is exactly the edge's effective resistance.

1.11.1 Some useful linear algebra facts

In this subsection only, A and B denote arbitrary matrices; from Section 1.11.2 on, B is again the incidence matrix of a graph. Let A be $n \times m$ and let B be $m \times n$.

Claim 1.11.1. $\det(I_m + BA) = \det(I_n + AB)$.

(Proof omitted.)

A rescaled version is sometimes useful: for $z \neq 0$,

$$\det(zI_m + BA) = z^{m-n} \det(zI_n + AB). \tag{1.11.1}$$

To see this, apply Claim 1.11.1 to A and B/z: $\det(zI_m + BA) = z^m \det(I_m + (B/z)A) = z^m \det(I_n + A(B/z)) = z^{m-n} \det(zI_n + AB)$.

Lemma 1.11.2 (Matrix determinant lemma)**.** *If A is an invertible $n \times n$ matrix and $v \in \mathbb{R}^n$, then*

$$\det(A + vv^\top) = (1 + v^\top A^{-1} v) \det(A). \tag{1.11.2}$$

Proof. Write $\det(A + vv^\top) = \det(A) \det(I_n + (A^{-1}v)v^\top)$. By Claim 1.11.1 with the $n \times 1$ matrix $A^{-1}v$ and the $1 \times n$ matrix $v^\top$, the last determinant is $\det(I_1 + v^\top A^{-1} v) = 1 + v^\top A^{-1} v$. □

Theorem 1.11.3 (Cauchy–Binet)**.** *Let A be $n \times m$ and B be $m \times n$ with $m \geq n$. Then*

$$\det(AB) = \sum_{S \subseteq [m],\ |S|=n} \det(A_S)\det(B_S), \tag{1.11.3}$$

where A_S is the $n \times n$ submatrix of A formed by the columns in S, and B_S the submatrix of B formed by the rows in S.

(Proof omitted.)

The special case we need is $B = A^\top$:

$$\det(AA^\top) = \sum_{S \subseteq [m],\ |S|=n} \det(A_S)^2. \tag{1.11.4}$$

1.11.2 The matrix-tree theorem

How many spanning trees does a graph G have? Figure 1.42 shows a graph on six vertices together with all of its spanning trees; there are eleven of them. Throughout this section $\kappa(G)$ denotes the number of spanning trees of G.

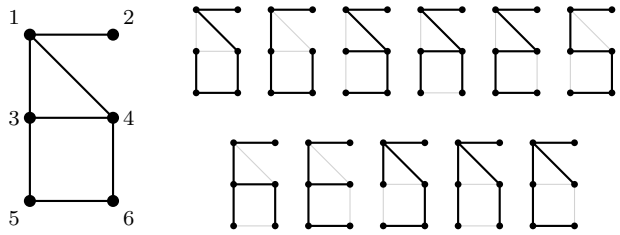


Figure 1.42: A graph on six vertices (left) and its eleven spanning trees. In each small copy the tree edges are black and the other two edges are light gray.

Theorem 1.11.4 (Kirchhoff's matrix-tree theorem)**.** *Let G be a graph with Laplacian eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. The number of spanning trees of G is*

$$\kappa(G) = \frac{1}{n}\lambda_2 \cdots \lambda_n. \tag{1.11.5}$$

For $i \in V$, let $(L_G)_{ii}$ denote the $(n-1) \times (n-1)$ matrix obtained from L_G by removing the ith row and the ith column. We will see that one can equivalently state the theorem as

$$\kappa(G) = \det\big((L_G)_{ii}\big) \qquad \text{for every } i. \tag{1.11.6}$$

This is the form we prove (Sections 1.11.3 and 1.11.4). It implies (1.11.5). Look at the coefficient of x in the characteristic polynomial $\det(xI - L_G)$. It is $(-1)^{n-1}$ times the sum of the n principal minors $\det((L_G)_{ii})$. It is also $(-1)^{n-1}$ times the sum of all products of $n-1$ of the eigenvalues, which is $\lambda_2 \cdots \lambda_n$ since $\lambda_1 = 0$. By (1.11.6) every minor equals $\kappa(G)$, so $n\kappa(G) = \lambda_2 \cdots \lambda_n$. For the graph of Fig. 1.42, each of the six minors $\det((L_G)_{ii})$ equals 11, matching the eleven trees in the figure.

Exercise 1.11.5 (warmup questions)**.**

(a) Why is $\kappa(G)$ not given by $\det(L_G)$?

(b) Why should $\det((L_G)_{ii}) = \det((L_G)_{jj})$?

(c) Is the theorem obvious for a tree?

1.11.3 Proof for trees

We first prove (1.11.6) when G is itself a tree, so that $\kappa(G) = 1$. Write $L = B^\top B$, where B is the signed edge-vertex incidence matrix of Section 1.10.2. For a matrix B, write $B_{:,2:}$ for B with its first column removed. Removing the first row and the first column of $L = B^\top B$ is the same as removing the first column of B on both sides, so

$$\det(L_{11}) = \det(B_{:,2:}^\top B_{:,2:}) = \det(B_{:,2:})^2, \tag{1.11.7}$$

where $B_{:,2:}$ is square, of size $(n-1) \times (n-1)$, because a tree has $n-1$ edges.

Example 1.11.6 (Warmup: the path P_5)**.** For the path on five vertices $1, 2, 3, 4, 5$, with each edge $(j, j+1)$ having head $j+1$ and tail j,

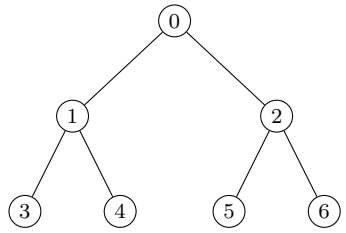
$$L = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}.$$

Removing the first column of B leaves a lower triangular matrix with 1s on the diagonal, so $\det(L_{11}) = \det(B_{:,2:})^2 = 1$.

General trees. For a general tree, order the vertices so that, after the first column is eliminated, the matrix has 1s on the diagonal. Root the tree at the first vertex. Orient each edge so that its head is the child (entry $+1$) and its tail is the parent (entry -1). Number the vertices so that every parent comes before its children, and list each edge in the row of its child. After the root's column is removed, the row of the edge into vertex j has its $+1$ in column j and its -1 in an earlier column (the parent's). So the matrix is lower triangular with 1s on the diagonal. Permuting the vertices might change the sign of $\det(B_{:,2:})$, but we square it, so this does not matter, and again

$$\det(L_{11}) = \det(B_{:,2:})^2 = 1.$$

Figure 1.43 shows an example. Rooting the tree at any vertex i gives $\det((L_G)_{ii}) = 1$ in the same way.



$$B = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 1.43: A tree with root 0 and its incidence matrix (columns indexed by the vertices $0, \ldots, 6$). Without its first column, B is lower triangular with 1s on the diagonal.

1.11.4 Proof for general graphs

The general matrix-tree theorem now follows by applying the special case (1.11.4) of the Cauchy–Binet theorem to the submatrices of B of size $n-1$. Let G be connected (otherwise both sides of (1.11.6) are 0), and let $B_{:,-i}$ be B with column i removed, an $m \times (n-1)$ matrix. Then $(L_G)_{ii} = B_{:,-i}^\top B_{:,-i}$ as in (1.11.7), so by (1.11.4)

$$\det((L_G)_{ii}) = \sum_{S \subseteq E,\ |S|=n-1} \det(B_{S,-i})^2,$$

where $B_{S,-i}$ consists of the rows of $B_{:,-i}$ indexed by the edges in S. If the $n-1$ edges in S form a spanning tree, $B_{S,-i}$ is the matrix of Section 1.11.3 for that tree, and $\det(B_{S,-i})^2 = 1$. If they do not, the graph (V, S) is disconnected, since a connected graph on n vertices with $n-1$ edges is a tree. Let W be the vertex set of a component that does not contain i. Every edge of S has either both endpoints in W or none, so the columns of $B_{S,-i}$ indexed by W add up to zero, and $\det(B_{S,-i}) = 0$. So the sum just counts the spanning trees of G.

A self-contained proof. One can also give a self-contained proof by induction. For weighted graphs, count each tree as the product of its edge weights, so that the claim becomes $\sum_T \prod_{e \in T} w_e = \det((L_G)_{ii})$, the sum over spanning trees T. Break the trees into those containing a fixed edge e and those not containing e, and check what happens to both sides of the equation. *(Proof omitted; only this outline was given.)*

1.11.5 Random spanning trees and effective resistance

Theorem 1.11.7. *Let G be a connected unweighted graph, let T be a uniformly random spanning tree of G, and let $e = (s, t)$ be an edge. Then*

$$\Pr[e \in T] = R_{\text{eff}}(e).$$

So the probability that a given edge belongs to a random spanning tree is its effective resistance.

Proof. The spanning trees of G that do not contain e are exactly the spanning trees of $G \setminus e$, so

$$\Pr[e \in T] = 1 - \frac{\kappa(G \setminus e)}{\kappa(G)}.$$

If $G \setminus e$ is disconnected, then e lies in every spanning tree and all of any current from s to t must cross e, so both sides equal 1. Assume from now on that $G \setminus e$ is connected. Fix a vertex $i \notin \{s, t\}$ (there is one, since $G \setminus e$ is connected and has no multiple edges). Since $L_G = L_{G \setminus e} + b_{st} b_{st}^\top$, removing row and column i gives

$$(L_G)_{ii} = (L_{G \setminus e})_{ii} + b_{st} b_{st}^\top,$$

where now $b_{st} = \chi_s - \chi_t$ stands for the vector with the ith coordinate removed. By the matrix-tree theorem in the form (1.11.6), and the matrix determinant lemma (Lemma 1.11.2) applied with $A = (L_{G \setminus e})_{ii}$ (invertible, because its determinant is $\kappa(G \setminus e) > 0$),

$$\frac{\kappa(G \setminus e)}{\kappa(G)} = \frac{\det((L_{G \setminus e})_{ii})}{\det((L_G)_{ii})} = \frac{1}{1 + b_{st}^\top (L_{G \setminus e})_{ii}^{-1} b_{st}}.$$

Let $R = b_{st}^\top (L_{G\setminus e})_{ii}^{-1} b_{st}$. Then

$$\Pr[e \in T] = 1 - \frac{1}{1+R} = \frac{R}{1+R} = \frac{1}{1/R + 1/1} = R_{\text{eff}}(e).$$

The last step uses two facts. First, R is the effective resistance between s and t in $G \setminus e$. To see this, let ψ be potentials for one unit of current from s to t in $G \setminus e$, shifted so that $\psi(i) = 0$. Then the terms with $\psi(i)$ drop out of the equations $L_{G\setminus e}\psi = b_{st}$ at the vertices other than i. So these equations say $(L_{G\setminus e})_{ii}\psi' = b_{st}$, where ψ' is ψ without its ith coordinate, and $R = b_{st}^\top \psi' = \psi(s) - \psi(t)$ is the effective resistance. Second, G is just $G \setminus e$ in parallel with the unit resistor e between s and t. So its effective resistance between s and t is $1/(1/R + 1/1)$, as in the triangle example, Example 1.10.1. □

Example 1.11.8. In the graph of Fig. 1.42, the edge $(3, 4)$ lies in 6 of the 11 spanning trees. Its effective resistance is also $6/11$: the edge is a unit resistor in parallel with the path $3, 1, 4$ (resistance 2) and the path $3, 5, 6, 4$ (resistance 3), giving $1/(1 + \frac{1}{2} + \frac{1}{3}) = 6/11$. The pendant edge $(1, 2)$ is a bridge; it lies in all 11 trees and has effective resistance 1.

Chapter 2

Convex Geometry

Chapter 1 studied graphs through the eigenvalues of matrices. This chapter turns to the geometry of convex bodies in high dimension, where intuition from two and three dimensions is often wrong. We start with the basic objects and the odd behavior of high-dimensional volume, then polarity and norms, and John's theorem on how well an ellipsoid approximates a convex body. The Brunn–Minkowski inequality and its consequences lead to a randomized algorithm for the volume of a convex body, which brings back random walks and conductance from Chapter 1. We end with concentration of measure and one of its algorithmic uses, the Johnson–Lindenstrauss theorem.

2.1 Convex Bodies and High-Dimensional Volume

This section fixes the basic objects: convex sets, convex functions, convex bodies, and a small gallery of examples that covers most of our intuition. Then we ask where the volume of a high-dimensional convex body sits. The answer (near the boundary, and also near every hyperplane through the center of a ball) is a first warning that low-dimensional intuition can mislead us.

2.1.1 Definitions and examples

In this chapter, inner products are written $\langle x, y\rangle$, and convex sets are called C (later sections also use K).

Definition 2.1.1 (Convex sets and functions)**.**

- A set $C \subseteq \mathbb{R}^n$ is *convex* if, for any $x, y \in C$ and any $t \in [0,1]$, the point $tx + (1-t)y$ is in C.
- A function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* if the region above its graph (in $\mathbb{R}^{n+1}$) is convex.
- A function f is *concave* if $-f$ is convex, so that the region below its graph is convex.
- A *convex body* is a convex set that is compact (closed and bounded) and has nonempty interior (is "full-dimensional").

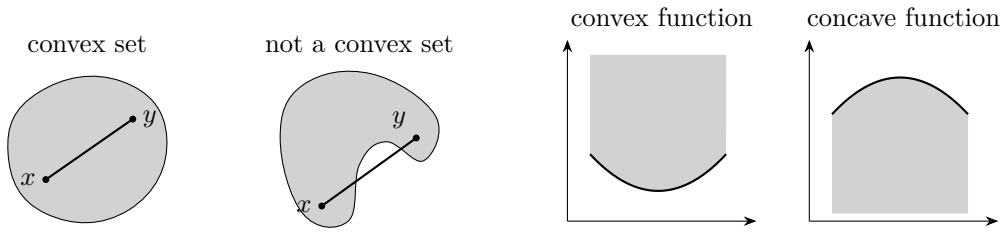


Figure 2.1: A convex set, a set that is not convex, the (convex) region above the graph of a convex function, and the (convex) region below the graph of a concave function.

A gallery of convex bodies

In Keith Ball's survey of convex geometry [Bal97], he writes: "All convex bodies behave a bit like Euclidean balls." Intuition from this sometimes works and sometimes fails. If we add a few more shapes, though, our intuition will very rarely be wrong. They are the following (Fig. 2.2).

- The *Euclidean ball* $B_2^n = \{x \in \mathbb{R}^n \mid \sum_i x_i^2 \leq 1\}$.
- An *ellipsoid* $E = \{x \in \mathbb{R}^n \mid x^\top Ax \leq 1\}$, for A a positive definite symmetric matrix (in this section A is not an adjacency matrix). We get the sphere if $A = I$. In the notation of later sections, $E = M^{1/2}B_2^n$ with $M = A^{-1}$.
- The *cube* $B_\infty^n = [-1, 1]^n$.
- The *regular simplex*: the convex hull of $n + 1$ equally spaced points in $\mathbb{R}^n$.
- The *cross-polytope*

$$B_1^n = \left\{x \in \mathbb{R}^n \;\middle|\; \sum_i |x_i| \leq 1\right\}$$
$$= \text{conv}(\text{all points of the form } (0, \ldots, 0, \pm 1, 0, \ldots, 0)) = \text{conv}(\pm\chi_1, \ldots, \pm\chi_n),$$

  which is the octahedron in $\mathbb{R}^3$.
- The *spherical cone*. It is the convex hull of an $(n-1)$-dimensional ball and a point outside its hyperplane, on the line through its center perpendicular to that hyperplane.
- Any of the above under a linear transformation.

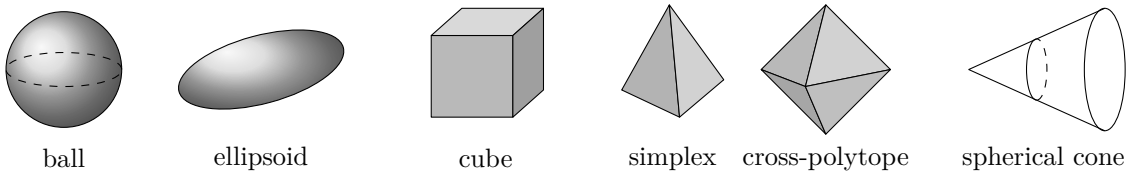


Figure 2.2: The gallery of convex bodies.

2.1.2 Where is high-dimensional volume concentrated?

One of the more counterintuitive properties of convex bodies is where their volume is concentrated. Let us first look at the square $[-1,1]^2$. How big an inscribed (concentric) cube do we need to get, say, 1/100 of the volume? We need the subcube $[-0.1,0.1]^2$. What about the 100-dimensional cube $[-1,1]^{100}$? There we need $[-0.955,0.955]^{100}$ (since $0.955^{100} = e^{100\ln 0.955} \approx e^{-4.6} \approx 0.010$). So almost all the volume is concentrated near the boundary. We get the same thing for Euclidean balls, of course. Scaling the ball by $1-\epsilon$ multiplies its volume by $(1-\epsilon)^n \le e^{-\epsilon n}$, so all but a fraction $e^{-\epsilon n}$ of the volume of B_2^n lies within distance ϵ of its boundary.

Balls versus cubes

How big a cube do we need to get total volume 1? The cube $[0,1]^n$. How big a ball do we need? One can compute by basic calculus that the volume of the n-dimensional ball of radius r is

$$\mathrm{vol}(rB_2^n) = \frac{r^n \pi^{n/2}}{\Gamma(\frac{n}{2}+1)}, \qquad \text{and} \qquad \mathrm{vol}(B_2^n) \approx \Big(\sqrt{\frac{2\pi e}{n}}\Big)^n. \tag{2.1.1}$$

(This is a good exercise if it is new to you.)[1] So we need a radius of about

$$\sqrt{\frac{n}{2\pi e}}$$

to get volume 1, and balls are much smaller than cubes! Why is this happening?

Slices of the ball

How big is an $(n-1)$-dimensional slice of the n-dimensional ball? So how does the volume of the slice vary as a function of its distance from the "equator" (Fig. 2.3)? The slice $\{x \in B_2^n : x_n = h\}$ at height $h \in [-1,1]$ is an $(n-1)$-dimensional ball of radius $\sqrt{1-h^2}$, so its $(n-1)$-dimensional volume is

$$(1-h^2)^{(n-1)/2}\,\mathrm{vol}_{n-1}(B_2^{n-1}) \le e^{-(n-1)h^2/2}\,\mathrm{vol}_{n-1}(B_2^{n-1}),$$

using $1-h^2 \le e^{-h^2}$. So, as a function of h, the volume of the slice falls off like a Gaussian with standard deviation about $1/\sqrt{n-1}$. Most of the volume of the ball lies within distance $O(1/\sqrt{n})$ of the equator.

So most of the volume lies near the hyperplane through the origin $\{x_n = 0\}$. But we could do the same with any hyperplane, so most of the volume lies near any subspace of dimension $n-1$. So most of the volume should lie near their intersection, that is, near the center of the ball, right? No, since most of the volume lies near the boundary... We have to be careful about low-dimensional intuition. The problem is applying "most of the volume" to n hyperplanes at once. Each slab around a hyperplane misses only a small fraction of the volume, but these fractions, added up over n hyperplanes, need not be small.

[1] By Stirling's formula, the approximation holds up to a factor polynomial in n: $\mathrm{vol}(B_2^n) = (1 + o(1))\,(2\pi e/n)^{n/2}/\sqrt{\pi n}$.

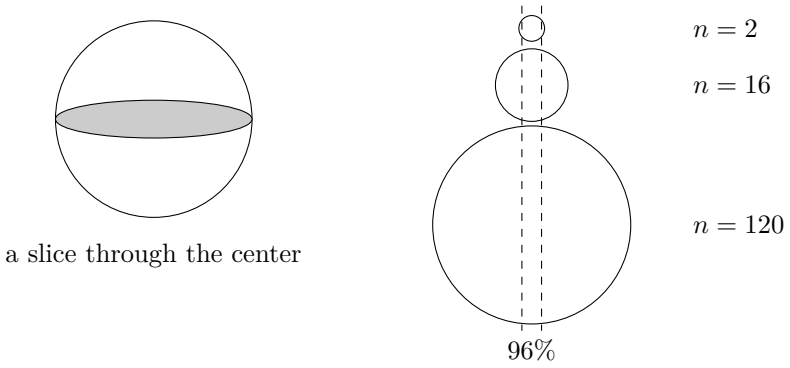


Figure 2.3: Left: a ball and its central slice. Right: three disks labeled $n = 2$, $n = 16$, $n = 120$, of increasing size, crossed by a vertical strip of fixed width labeled 96%. As in Keith Ball's survey [Bal97], these are balls of volume one[?] in these dimensions, of radius about $\sqrt{n/(2\pi e)}$, and the strip is the slab of width 1 about the equator (not drawn to scale). In high dimension this slab contains about 96% of the volume of a ball of volume one (for $n = 120$, 95.8%), since the slice volumes are close to a Gaussian density with standard deviation $1/\sqrt{2\pi e}$. So the ball grows with n, but its volume stays near the equator.

2.2 Polarity and Norms

Polarity is a way of turning a convex body inside out: the facets of a polytope become the vertices of its polar, and the other way around. We first define the polar of a polytope, then extend the definition to arbitrary convex bodies and check that the two definitions agree. With the separating hyperplane theorem we then prove that taking the polar twice gives back the body we started with. Finally, we see that polarity is the geometric face of duality for norms: the unit ball of the dual norm is the polar of the unit ball.

2.2.1 The polar of a polytope

Suppose we have a bounded polytope $C \subseteq \mathbb{R}^n$ that contains the origin in its interior.[2] We can write it as the set of points satisfying some inequalities,

$$C = \{x \mid \langle a_i, x\rangle \leq b_i,\ i = 1, \ldots, k\},$$

and we can scale the inequalities so that, without loss of generality, all $b_i = 1$:

$$C = \{x \mid \langle a_i, x\rangle \leq 1,\ i = 1, \ldots, k\}. \tag{2.2.1}$$

We can then form a new polytope, called the *polar* of C:

$$C^* = \operatorname{conv}(a_1, \ldots, a_k). \tag{2.2.2}$$

Exercise 2.2.1 (properties of the polar)**.** Let C and C^* be as in (2.2.1) and (2.2.2).

(a) If $A \subseteq B$, what does this mean about A^* and B^*?

[2]The rescaling to $b_i = 1$ below needs every $b_i > 0$, which holds when the origin is an interior point.

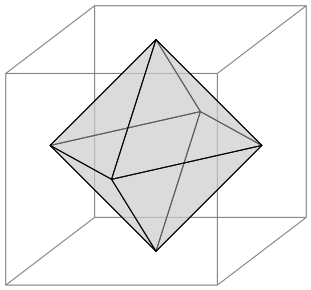


Figure 2.4: The cube $[-1,1]^3$ and, inside it, the octahedron whose vertices are the centers of the faces of the cube, $\mathrm{conv}(\pm\chi_1, \pm\chi_2, \pm\chi_3)$. Since $[-1,1]^3 = \{x : \langle \pm\chi_j, x\rangle \le 1,\ j = 1,2,3\}$, the octahedron is the polar of the cube by (2.2.2).

(b) What is $(C^*)^*$? We claim that $(C^*)^* = C$, to be proved a little later, when we have more machinery (Theorem 2.2.7).

(c) How does the polar behave under scaling? What about under a change of coordinates?

(d) If C is origin-symmetric, what does this say about C^*?

(e) What does it mean if the polar is lower-dimensional? What about unbounded?

2.2.2 The polar of a convex body

It would be nice if our definition worked for any convex body, not just for polytopes.

Exercise 2.2.2. What should the polar of the unit sphere be? What about the polar of a sphere of radius r? An ellipse? An ellipsoid?

The idea is that any convex body can be thought of as the intersection of a (possibly infinite) set of half-spaces, bounded by its *supporting hyperplanes*. So we can think of the polar as the convex hull of a possibly infinite set of points, coming from all of the supporting hyperplanes.

Definition 2.2.3 (Polar)**.** The *polar* of a convex body C is

$$C^* = \{x \in \mathbb{R}^n \mid \langle x, c\rangle \le 1 \quad \forall c \in C\}. \tag{2.2.3}$$

2.2.3 Do the two definitions match?

Why does Definition 2.2.3 agree with the earlier definition in the case of polytopes? Let $C = \{x \mid \langle a_i, x\rangle \le 1,\ i = 1,\dots,k\}$, so that by (2.2.2) the old definition gives $C^* = \mathrm{conv}(a_1,\dots,a_k)$. By the new definition it is clear that $a_i \in C^*$ (since $\langle a_i, c\rangle \le 1$ for every $c \in C$). We need to check two things:

- the polar is convex (good to know in general too!); it is an intersection of the half-spaces $\{x : \langle x, c\rangle \le 1\}$, and an intersection of convex sets is convex. With $a_i \in C^*$, this gives $\mathrm{conv}(a_1,\dots,a_k) \subseteq C^*$;
- the new definition does not add anything that is not in $\mathrm{conv}(a_1,\dots,a_k)$.

For the second point it is easier to start from the other end, with the polytope given by its vertices. In the claim below the roles of C and C^* are therefore switched relative to (2.2.1) and (2.2.2): $C = \operatorname{conv}(a_1, \dots, a_k)$, and the original definition says that $C^* = \{x \mid \langle a_i, x\rangle \le 1,\ i = 1, \dots, k\}$. Taking polars once more then gives the second point; this needs $(C^*)^* = C$, and so one more theorem first, the separating hyperplane theorem (Section 2.2.4).

Claim 2.2.4. *Definition 2.2.3 agrees with the original definition for polytopes: if C is the polytope $\operatorname{conv}(a_1, \dots, a_k)$, then*

$$\{x \in \mathbb{R}^n \mid \langle x, c\rangle \le 1 \ \forall c \in C\} = \{x \mid \langle a_i, x\rangle \le 1,\ i = 1, \dots, k\}.$$

Proof. Write

$$C^*_{\text{old}} = \{x \mid \langle a_i, x\rangle \le 1,\ i = 1, \dots, k\}, \qquad C^*_{\text{new}} = \{x \in \mathbb{R}^n \mid \langle x, c\rangle \le 1 \ \forall c \in \operatorname{conv}(a_1, \dots, a_k)\}.$$

Since every a_i lies in $\operatorname{conv}(a_1, \dots, a_k)$, a point of C^*_{new} satisfies in particular $\langle x, a_i\rangle \le 1$ for all i, so $C^*_{\text{new}} \subseteq C^*_{\text{old}}$.

Conversely, let $x \in C^*_{\text{old}}$ and $c \in \operatorname{conv}(a_1, \dots, a_k)$. Write $c = \sum_i \lambda_i a_i$ with $\lambda_i \ge 0$ for all i and $\sum_i \lambda_i = 1$. Then

$$\langle c, x\rangle = \sum_i \lambda_i \langle a_i, x\rangle \le \sum_i \lambda_i = 1,$$

so $x \in C^*_{\text{new}}$. Hence $C^*_{\text{old}} \subseteq C^*_{\text{new}}$. □

2.2.4 Separating hyperplanes and the polar of the polar

Definition 2.2.5 (separating hyperplane)**.** Given a convex body $K \subseteq \mathbb{R}^n$ and a point p, a *separating hyperplane* is a hyperplane that has K on one side and p on the other. That is, there is a vector η such that

- $\langle \eta, x\rangle \le 1$ for all $x \in K$, and
- $\langle \eta, p\rangle \ge 1$.

The right-hand side could equally be 0, or any other constant. The hyperplane is *strongly separating* if the second inequality is strict (Fig. 2.5).

Theorem 2.2.6 (separating hyperplane theorem)**.** *If K is a convex body and $p \notin K$, there is a hyperplane that strongly separates them.*

(Proof omitted.)

With the separating hyperplane theorem in hand we can prove the claim $(C^*)^* = C$ from Exercise 2.2.1.

Theorem 2.2.7. *If K is a convex body containing the origin, then $K^{**} = K$.*

Proof sketch. By definition,

$$K^* = \{x \mid \langle k, x\rangle \le 1 \ \forall k \in K\}, \qquad K^{**} = \{y \mid \langle p, y\rangle \le 1 \ \forall p \in K^*\}.$$

If $y \in K$ and $p \in K^*$, then $\langle y, p\rangle \le 1$ by the definition of K^*, for any such p. So $y \in K^{**}$, and thus $K \subseteq K^{**}$.

For the other inclusion, suppose $y \in K^{**}$, that is, $\langle p, y\rangle \le 1$ for all $p \in K^*$, but $y \notin K$. Let $\{x : \langle v, x\rangle = 1\}$ be a strongly separating hyperplane for y and K (Theorem 2.2.6). Then $\langle v, k\rangle \le 1$ for all $k \in K$, so $v \in K^*$. But $\langle v, y\rangle > 1$ with $v \in K^*$, which is a contradiction. □

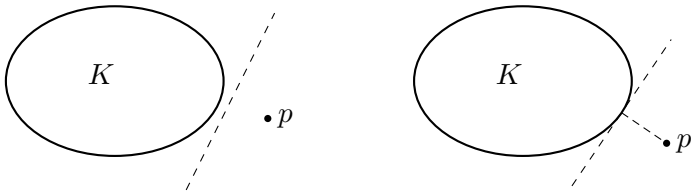


Figure 2.5: Left: a hyperplane with K on one side and p on the other. Right: the hyperplane through the point of K nearest to p, perpendicular to the segment joining them; it touches K, and p lies strictly on the other side.

Remark 2.2.8. The proof uses a strongly separating hyperplane whose right-hand side is exactly 1, and this is where the origin must lie in K. Theorem 2.2.6 gives η and c with $\langle \eta, x\rangle \le c$ for all $x \in K$ and $\langle \eta, y\rangle > c$. Since $0 \in K$, we have $c \ge 0$. Pick any c' with $c < c' < \langle \eta, y\rangle$ (so $c' > 0$) and set $v = \eta/c'$. Then $\langle v, x\rangle \le c/c' \le 1$ on K and $\langle v, y\rangle > 1$.

2.2.5 Norms and symmetric convex bodies

Many of the convex bodies we will talk about are *origin-symmetric*: $x \in C \Rightarrow -x \in C$.

Recall that a *norm* on $\mathbb{R}^n$ is a map $q : \mathbb{R}^n \to \mathbb{R}$ such that

1. $q(ax) = |a|\, q(x)$ for $a \in \mathbb{R}$ (homogeneity);
2. $q(x + y) \le q(x) + q(y)$ (triangle inequality);
3. $q(x) \ge 0$ for all x (nonnegativity; implied by 1 and 2);
4. $q(x) = 0$ if and only if $x = 0$ (positivity).

There is a deep connection between norms and convex bodies, which makes convex geometry a powerful tool for functional analysis. Given a norm q, let

$$B_q = \{x \in \mathbb{R}^n \mid q(x) \le 1\}$$

be the *unit ball* of q.

Claim 2.2.9. *B_q is a convex body.*

Exercise 2.2.10. Why?

Can the correspondence go the other way? That is, given a convex body C, can we produce a norm in which C is the unit ball? There is an obvious restriction: C must be origin-symmetric, because every unit ball is.

Definition 2.2.11 (Minkowski functional)**.** The *Minkowski functional* of an origin-symmetric convex body C is the map $p_C : \mathbb{R}^n \to \mathbb{R}$ given by

$$p_C(x) = \inf\{\lambda > 0 \mid x \in \lambda C\}.$$

We sometimes write $\|x\|_C$ for $p_C(x)$.

The Minkowski functional is a norm.

Exercise 2.2.12. Check homogeneity, the triangle inequality and nonnegativity for p_C.

Duals and the polar

From the definitions,

$$C = \{x \mid \|x\|_C \le 1\}. \tag{2.2.4}$$

For any norm q we can define its *dual norm* by

$$q^*(x) = \sup_{v \neq 0} \frac{\langle x, v\rangle}{q(v)}.$$

Duality of norms corresponds to polarity of convex bodies. If $C = B_q$ is the unit ball of q, then

$$q^*(x) \le 1 \iff \langle x, v\rangle \le 1 \ \forall v \text{ such that } q(v) \le 1 \iff x \in C^*, \tag{2.2.5}$$

so the unit ball of the dual norm is the polar of the unit ball: $B_{q^*} = (B_q)^*$.

The important examples are the ℓ_p norms for $p \ge 1$,

$$\|x\|_p := \Big(\sum_i |x_i|^p\Big)^{1/p}.$$

Let B_p^n be the unit ball in ℓ_p. Then

$$(B_p^n)^* = B_q^n, \qquad \text{where } \frac{1}{p} + \frac{1}{q} = 1. \tag{2.2.6}$$

Exercise 2.2.13. Check (2.2.6). ("I encourage you to check this!")

Figures 2.6 and 2.7 show some of these unit balls. For $p < 1$ the formula above does not define a norm, and its "unit ball" is not convex.

2.3 John's Theorem

How well can an arbitrary convex body be approximated by an ellipsoid? After finding the right way to ask this (the Banach–Mazur distance), Fritz John's theorem gives the answer: within a factor $\sqrt{n}$ for origin-symmetric bodies and n in general, and both bounds are tight. The ellipsoid that achieves this is the one of maximum volume inside the body. We write that ellipsoid as the solution of a convex program and show that the conditions of John's theorem are the first-order optimality conditions of that program. From them we derive the "rounding" statement $E \subseteq K \subseteq \sqrt{n}\,E$ for symmetric bodies.

2.3.1 How far can a convex body be from a ball?

The high-level question is: how well can we approximate an arbitrary convex body C by a Euclidean ball? What is the right way to ask this?

Version 1. What is the least t such that $aB_2^n \subseteq C \subseteq taB_2^n$ for some $a > 0$? This is not a great definition: there are convex bodies, even in $\mathbb{R}^2$, for which t has to be arbitrarily large.

Exercise 2.3.1. Give an example.

Better version. What is the least t such that $E \subseteq C \subseteq tE$ for some ellipsoid E? This motivates the following general definition.

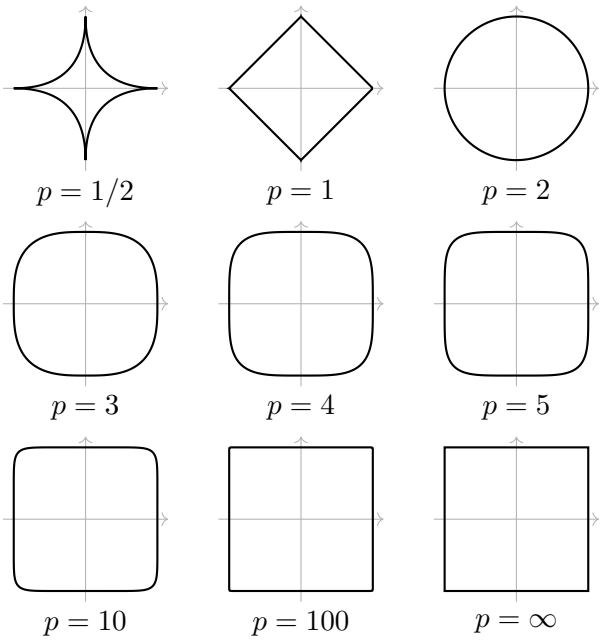


Figure 2.6: Unit balls $\{x \in \mathbb{R}^2 : \|x\|_p \le 1\}$. For $p = 1/2$ the set is not convex and $\|\cdot\|_{1/2}$ is not a norm. As p grows, the ball fills out toward the square B_∞^2.

Definition 2.3.2 (Banach–Mazur distance)**.** The *Banach–Mazur distance* $d(K, L)$ between convex bodies K and L is the least positive d for which there is a linear image $\tilde{L}$ of L such that $\tilde{L} \subseteq K \subseteq d\tilde{L}$.

In this section L denotes a convex body, not a Laplacian. It is easy to see that d is symmetric. If $\tilde{L} = TL \subseteq K \subseteq dTL$, then $\frac{1}{d}T^{-1}K \subseteq L \subseteq T^{-1}K$, so the linear image $\tilde{K} = \frac{1}{d}T^{-1}K$ of K satisfies $\tilde{K} \subseteq L \subseteq d\tilde{K}$. Note that the distance is multiplicative, not additive: for example $d(K, K) = 1$, not 0. We would need to take a logarithm to get a metric.

Remark 2.3.3. For bodies that are not origin-symmetric, such as the triangle in Fig. 2.8, the definition is meant with affine images (a linear map followed by a translation). Then $d\tilde{L}$ is the dilation of $\tilde{L}$ by the factor d about its center. Otherwise the answer would depend on where the origin sits inside K. For origin-symmetric bodies, linear images suffice.

2.3.2 Fritz John's theorem

Theorem 2.3.4 (John)**.** *For any n-dimensional, origin-symmetric convex body K,*

$$d(K, B_2^n) \le \sqrt{n}.$$

For a general (not necessarily origin-symmetric) convex body K,

$$d(K, B_2^n) \le n.$$

We will prove the symmetric case (Sections 2.3.4 and 2.3.5). The proof of the general case is similar but slightly messier, and we will not give it.

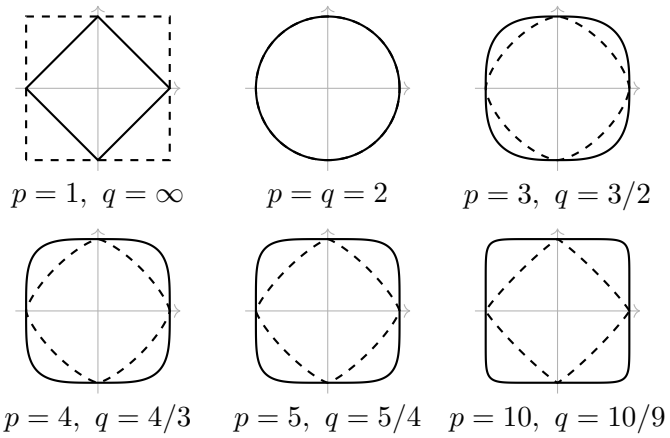


Figure 2.7: Dual pairs: B_p^2 (solid) and its polar B_q^2 (dashed), with $1/p + 1/q = 1$. (For $p = q = 2$ the two curves coincide.)

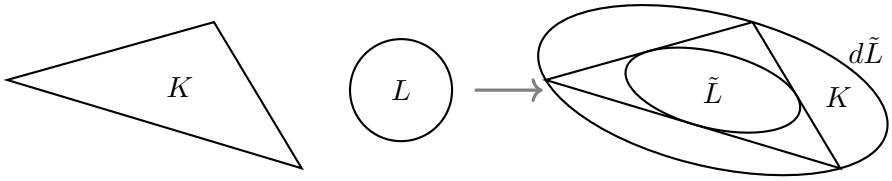


Figure 2.8: The Banach–Mazur distance between a triangle K and a disk L: an ellipse $\tilde{L}$ (a linear image of L, centered at the centroid of K) with $\tilde{L} \subseteq K \subseteq d\tilde{L}$. Here $d = 2$.

Phrased another way, the theorem says that we can find an ellipsoid E with

$$E \subseteq K \subseteq \sqrt{n}\, E.$$

Which ellipsoid should we use to get this bound? The answer used by the proof: the ellipsoid of maximal volume inside K.

The bounds are tight

The extremal examples are the cube and the simplex; Fig. 2.9 shows them in two dimensions.

Example 2.3.5 (the cube; $\sqrt{n}$ is tight)**.** Let $K = [-1, 1]^n$. One can check that the best ellipsoids are balls centered at the origin. The largest inscribed ball has radius 1, since the centers of the facets are at $\pm\chi_i$. To contain $[-1, 1]^n$, a ball centered at the origin needs radius $\sqrt{n}$, since the corners $(\pm 1, \ldots, \pm 1)$ have norm $\sqrt{n}$. So the ratio is exactly $\sqrt{n}$.

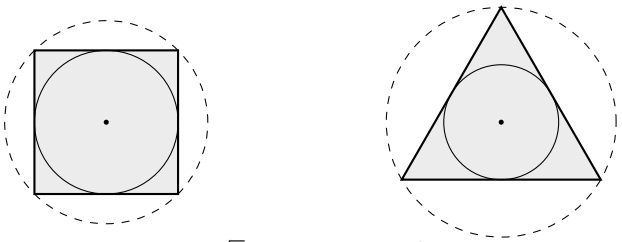
Example 2.3.6 (the simplex; n is tight)**.** Represent the regular n-dimensional simplex as $\operatorname{conv}(\chi_1, \ldots, \chi_{n+1})$, which lies in $\mathbb{R}^{n+1} \cap \{\sum_i x_i = 1\}$. One can check that the best ellipsoids are balls centered at $p = (\frac{1}{n+1}, \ldots, \frac{1}{n+1})$. The middles of the facets, for example $q = (\frac{1}{n}, \ldots, \frac{1}{n}, 0)$, have

$$\|q - p\| = \sqrt{\frac{1}{n(n+1)}},$$

while the corners have

$$\|\chi_i - p\| = \sqrt{\frac{n}{n+1}}.$$

The ratio of the two radii is $\sqrt{n/(n+1)}/\sqrt{1/(n(n+1))} = \sqrt{n^2} = n$.



cube, $n = 2$: ratio $\sqrt{2}$ simplex, $n = 2$: ratio 2

Figure 2.9: Two-dimensional versions of the extremal examples: the largest inscribed ball (solid) and the smallest enclosing ball with the same center (dashed).

Characterizing the maximum-volume ellipsoid

We will give the following characterization of the maximum-volume inscribed ellipsoid.

Theorem 2.3.7 (John)**.** *K contains a unique ellipsoid of maximal volume. It is B_2^n if and only if:*

1. *$B_2^n \subseteq K$, and*
2. *there are unit vectors $u_1, \dots, u_m$ on the boundary of K and positive numbers $c_1, \dots, c_m$ such that*

$$\sum_{i=1}^m c_i u_i = 0,$$
$$\sum_{i=1}^m c_i \langle x, u_i\rangle^2 = \|x\|^2 \ \ \forall x \iff \sum_{i=1}^m c_i u_i u_i^\top = I_n. \tag{2.3.1}$$

The two forms of the second equation in (2.3.1) are equivalent, since a symmetric matrix is determined by its quadratic form and $\sum_i c_i \langle x, u_i\rangle^2 = x^\top (\sum_i c_i u_i u_i^\top) x$.

We will show (Section 2.3.5) that these conditions imply $E \subseteq K \subseteq nE$, and $E \subseteq K \subseteq \sqrt{n}\, E$ if K is origin-symmetric, where $E = B_2^n$ is the maximum-volume ellipsoid.

What do the conditions mean?

- The sphere touches the boundary of K in a lot of places.
- The touching points can be weighted so that their center of mass is the origin and their inertia tensor is I_n.
- The touching points do not all lie "on one side of the sphere" or "near a proper vector subspace".

Figure 2.10 shows some examples. Let $U_i = u_i u_i^\top$. If we think geometrically about the space of ellipsoids as a space of matrices, $\mathbb{R}^{n^2}$, the second condition says that I_n/n is in the convex hull of the U_i. Taking the trace of $\sum_i c_i u_i u_i^\top = I_n$ gives $\sum_i c_i \|u_i\|^2 = \sum_i c_i = n$, so $I_n/n = \sum_i (c_i/n)\, U_i$ is a convex combination.

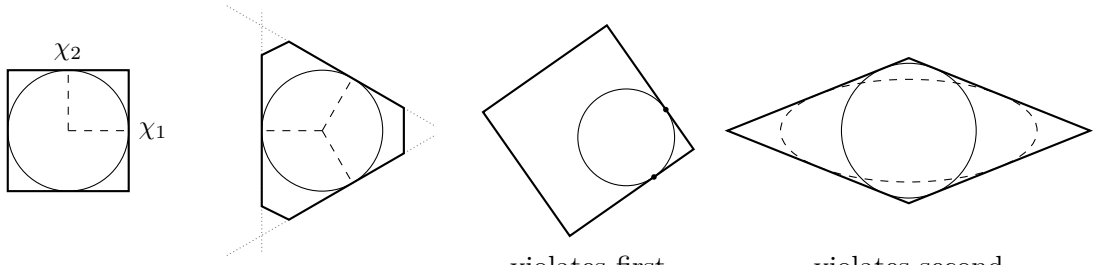


Figure 2.10: The conditions of Theorem 2.3.7. Left two: the unit circle is the maximum-area ellipse, with the touching points spread around it. Third: the circle lies inside K but touches ∂K only on one side (marked points). Right: the circle touches the rhombus only near the vertical axis, and the dashed ellipse inside the rhombus has larger area. Here "first condition" and "second condition" mean the two equations of condition 2, since the circle in the third picture does lie inside K. In the third picture no positive weights make $\sum_i c_i u_i = 0$, and a larger circle fits in K. In the fourth, $\sum_i c_i u_i = 0$ can hold but $\sum_i c_i u_i u_i^\top = I_2$ cannot.

2.3.3 The maximum-volume ellipsoid as an optimization problem

The idea is to write the maximum-volume ellipsoid as the solution of an optimization problem; the theorem is then just the first-order optimality conditions obeyed by the solution. For simplicity, assume that K is origin-symmetric.

It is easy to see that then the maximum-volume ellipsoid E is centered at the origin. If $E = c + E_0$, with E_0 centered at the origin, lies in K, then so does $-E = -c + E_0$, by symmetry. By convexity so does $\frac{1}{2}E + \frac{1}{2}(-E) = E_0$, which has the same volume. So a maximum-volume ellipsoid centered at the origin exists, and by the uniqueness part of Theorem 2.3.7, $E = E_0$. So for some positive definite M we can write

$$E = \{x \mid x^\top M^{-1} x \le 1\}.$$

Then $E = M^{1/2} B_2^n$, and

$$\operatorname{vol}(E) = \det(M)^{1/2} \operatorname{vol}(B_2^n).$$

So we want the M with maximal $\det(M)$ subject to $E \subseteq K$.

- To get a convex program, we will minimize $-\log\det(M)$.
- $E \subseteq K \iff K^* \subseteq E^* = \{x \mid x^\top M x \le 1\}$.

The second point combines three facts. Polarity reverses inclusion: if $A \subseteq B$ then $B^* \subseteq A^*$, since the conditions defining A^* are among those defining B^*. Applying this to $K^* \subseteq E^*$ and using $K^{**} = K$, $E^{**} = E$ (Theorem 2.2.7) gives the converse. Finally, for $E = M^{1/2} B_2^n$ we have $\sup_{y \in E} \langle x, y\rangle = \sup_{\|z\| \le 1} \langle M^{1/2} x, z\rangle = \|M^{1/2} x\|$, so $E^* = \{x : x^\top M x \le 1\}$.

We will prove the theorem when K is a polytope. It is not too hard to show that this implies the general case, but we will not do that step. Let

$$K = \{x \in \mathbb{R}^n \mid a_i^\top x \le 1,\ i = 1, \dots, k\}.$$

Then $K^* = \operatorname{conv}(a_i)$, so

$$K^* \subseteq E^* \iff a_i \in E^*\ \forall i \iff a_i^\top M a_i \le 1\ \forall i.$$

By Claim 2.2.4, $\operatorname{conv}(a_i)^* = K$. Taking polars and using Theorem 2.2.7 gives $K^* = \operatorname{conv}(a_i)^{**} = \operatorname{conv}(a_i)$. The second equivalence holds since E^* is convex: it contains $\operatorname{conv}(a_i)$ exactly when it contains every a_i. So we can write the optimization problem as

$$\min_{M \succ 0} \; -\log\det(M) \qquad \text{s.t.} \quad a_i^\top M a_i \le 1 \;\; \forall i. \tag{2.3.2}$$

(Allowing $M \succeq 0$ gives the same program, since $-\log\det M = +\infty$ for singular M.)

2.3.4 The optimality conditions

It remains to find the optimality conditions of (2.3.2). Think of this program as a convex optimization problem in $\mathbb{R}^{n^2}$, the space of $n \times n$ matrices. The dot product in $\mathbb{R}^{n^2}$ is

$$A \bullet B = \sum_{i,j} A_{ij} B_{ij} = \operatorname{tr}(A^\top B).$$

Let $A_i = a_i a_i^\top$. Then

$$a_i^\top M a_i = \operatorname{tr}(a_i^\top M a_i) = \operatorname{tr}(a_i a_i^\top M) = A_i \bullet M,$$

so the constraints of (2.3.2) are linear in M, and the problem becomes

$$\min_{M \succ 0} \; g(M) \coloneqq -\log\det(M) \qquad \text{s.t.} \quad A_i \bullet M \le 1 \;\; \forall i. \tag{2.3.3}$$

At the optimum, $-\nabla g \in \operatorname{pos}(A_i)$, the cone generated by the A_i. That is,

$$-\nabla g = \sum_i c_i A_i, \qquad c_i \ge 0, \tag{2.3.4}$$

with c_i nonzero only at tight constraints, that is, for $i \in S \coloneqq \{i \mid A_i \bullet M = 1\}$. These are the first-order (Karush–Kuhn–Tucker) optimality conditions for minimizing a differentiable convex function subject to linear inequality constraints. At the optimum, no direction that keeps the tight constraints satisfied can decrease g. By Farkas' lemma, this means that $-\nabla g$ is a nonnegative combination of the normals A_i of the tight constraints.

Lemma 2.3.8. *For invertible M, $\nabla \det(M) = \det(M)\, M^{-\top}$, and hence $\nabla(-\log\det M) = -M^{-\top}$.*

Proof. Write E_{ij} for the matrix with a single 1 in position (i,j). Then $\det(I + \epsilon E_{ij}) = 1 + \epsilon\delta_{ij}$, and therefore

$$\det(I + \epsilon B) \approx 1 + \epsilon \operatorname{tr}(B)$$

up to terms of order ϵ^2. For $i \neq j$ the matrix $I + \epsilon E_{ij}$ is triangular with ones on the diagonal. For general B, expand $\det(I + \epsilon \sum_{i,j} B_{ij} E_{ij})$ by multilinearity in the rows. The terms of first order in ϵ come from changing a single row, so they add up to $\epsilon \sum_{i,j} B_{ij}\delta_{ij} = \epsilon \operatorname{tr}(B)$. Hence, up to terms of order ϵ^2,

$$\begin{aligned} \det(M + \epsilon B) &= \det(M)\det(I + \epsilon M^{-1} B) \approx \det(M)(1 + \epsilon \operatorname{tr}(M^{-1}B)) \\ &= \det(M) + \epsilon \det(M) \operatorname{tr}(M^{-1}B) = \det(M) + \epsilon \det(M)\, M^{-\top} \bullet B. \end{aligned}$$

So $\nabla \det(M) = \det(M)\, M^{-\top}$, and

$$\nabla(-\log\det M) = -\frac{1}{\det M} \det(M)\, M^{-\top} = -M^{-\top}. \qquad \square$$

Combining (2.3.4) with Lemma 2.3.8, the optimal M satisfies

$$M^{-1} = M^{-\top} = \sum_{i \in S} c_i A_i = \sum_{i \in S} c_i a_i a_i^\top. \tag{2.3.5}$$

The optimum is the unique such M, because $-\log\det$ is strictly convex.

The ellipse is B_2^n exactly when the optimum is $M = I$. In that case

$$S = \{i \mid A_i \bullet I = 1\} = \{i \mid \operatorname{tr}(a_i a_i^\top) = 1\} = \{i \mid a_i^\top a_i = 1\}.$$

So the a_i for $i \in S$ are unit vectors on the boundary of K, and by (2.3.5),

$$\sum_{i \in S} c_i a_i a_i^\top = I,$$

as claimed, with $u_i = a_i$ for $i \in S$. Each such a_i lies in $B_2^n \subseteq K$ and satisfies the ith constraint with equality, so it is on the boundary of K.

Remark 2.3.9. Two more parts of Theorem 2.3.7 need checking. First, the condition $\sum_i c_i u_i = 0$. For an origin-symmetric polytope the constraints come in pairs $\pm a_i$, and $a_i a_i^\top = (-a_i)(-a_i)^\top$, so we may split each weight c_i equally between u_i and $-u_i$. The weighted vectors then cancel in pairs, and the matrix identity is unchanged. Second, the "if" direction. For a convex program the first-order conditions are also sufficient for optimality, so if $M = I$ is feasible and satisfies (2.3.4), it is the optimum.

2.3.5 Proof of the rounding result

Theorem 2.3.10. *Let K be an origin-symmetric convex body. Then there is an ellipsoid E with*

$$E \subseteq K \subseteq \sqrt{n}\, E.$$

Proof. We show that the version of John's theorem in Theorem 2.3.7, with

$$\sum_{i=1}^m c_i u_i = 0, \qquad \sum_{i=1}^m c_i u_i u_i^\top = I,$$

implies the claim. Change coordinates so that the John ellipsoid (the maximum-volume ellipsoid in K) is B_2^n. We will show that $K \subseteq \sqrt{n}\, B_2^n$.

First, $\operatorname{tr}(u_i u_i^\top) = \|u_i\|^2 = 1$, so taking the trace of $\sum_i c_i u_i u_i^\top = I$ gives

$$\sum_{i=1}^m c_i = \operatorname{tr}(I) = n. \tag{2.3.6}$$

If I give you unit vectors u_i on ∂K, what is the largest convex body consistent with this fact? It is the polytope cut out by the tangent hyperplanes of the ball at the u_i (Fig. 2.11):

$$x \in K \;\Rightarrow\; u_i^\top x \le 1 \quad \text{for } i = 1, \dots, m.$$

Since $B_2^n \subseteq K$ and $u_i \in \partial K \cap \partial B_2^n$, a supporting hyperplane of K at u_i is also one of B_2^n at u_i, and the only one is $\{x : u_i^\top x = 1\}$. Since K is origin-symmetric, $-x \in K$ as well, so in fact $|u_i^\top x| \le 1$. Therefore, for $x \in K$,

$$\|x\|^2 = x^\top \Big(\sum_{i=1}^m c_i u_i u_i^\top\Big) x = \sum_{i=1}^m c_i (u_i^\top x)^2 \le \sum_{i=1}^m c_i = n,$$

using (2.3.6). So $K \subseteq \sqrt{n}\, B_2^n$. □

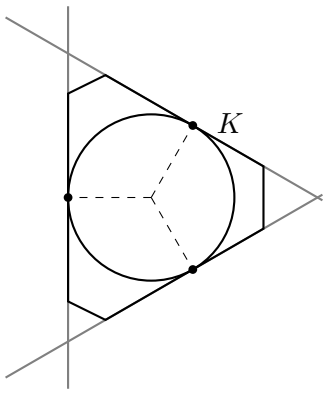


Figure 2.11: Given that the unit ball lies in K and touches ∂K at the marked points u_i, the body K must lie inside the polytope bounded by the tangent lines $\{u_i^\top x = 1\}$.

2.3.6 An alternative proof of the rounding result

As a side note, if you just want to prove the rounding result and do not also need to characterize E, there is another, possibly simpler, proof (Fig. 2.12).

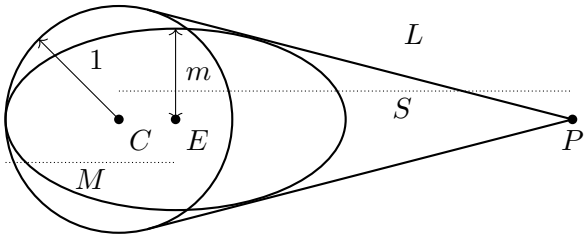


Figure 2.12: The picture for the alternative proof: the unit ball centered at C, a far point P at distance S, the lines L from P tangent to the ball, and an ellipsoid centered at E, with semi-axis M along CP and m across it, inside the convex hull of the ball and P.

The idea is the following. Suppose the unit ball centered at C is the maximum-volume ellipsoid in K, and K contains a point P at distance S from C. Then K contains the convex hull of the ball and P (for symmetric K, also of $-P$). If S is too large, this hull contains an ellipsoid of larger volume $Mm^{n-1} > 1$, like the one centered at E, which is a contradiction. Making "too large" precise ($S > n$ in general, $S > \sqrt{n}$ for symmetric K) is a calculation with M, m and S. *(Proof omitted.)*

2.4 The Brunn–Minkowski Inequality and Its Applications

The Brunn–Minkowski inequality bounds the volume of a Minkowski sum $A + B$ from below in terms of the volumes of A and B. It says that $\mathrm{vol}^{1/n}$ is concave with respect to Minkowski sums. We define Minkowski sums, prove the inequality, and then draw three consequences. Brunn's theorem: the $(n-1)$-dimensional volumes of the parallel slices of a convex body, raised to the power $1/(n-1)$, form a concave function. The isoperimetric inequality: among bodies of a given volume, Euclidean balls have the smallest surface area. Grünbaum's theorem: every hyperplane through the centroid of a convex body leaves at least a $1/e$ fraction of the volume on each side.

2.4.1 Minkowski sums

Definition 2.4.1 (Minkowski sum)**.** For $A, B \subseteq \mathbb{R}^n$, the *Minkowski sum* $A + B$ is

$$A + B = \{a + b \mid a \in A,\ b \in B\}.$$

We also write $\lambda A = \{\lambda a \mid a \in A\}$. Figure 2.13 shows two examples.

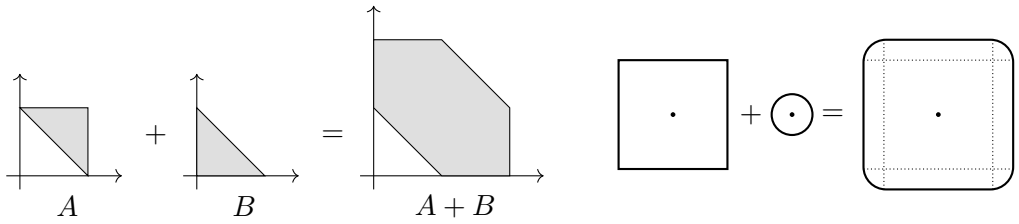


Figure 2.13: Minkowski sums. Left: two triangles. Right: a square plus a small disk centered at the origin is the square with its sides pushed out by the radius and its corners rounded.

2.4.2 The Brunn–Minkowski inequality

The Brunn–Minkowski inequality seems simple, but it implies a lot of important things; some regard it as the beginning of modern convex geometry.

The question is: can we bound $\mathrm{vol}(A + B)$ in terms of $\mathrm{vol}(A)$ and $\mathrm{vol}(B)$?

- Easy: $\mathrm{vol}(A + B) \geq \max\big(\mathrm{vol}(A), \mathrm{vol}(B)\big)$. (Any translate $A + b$ with $b \in B$ lies in $A + B$ and has the volume of A; similarly for B.)
- Can we do better? Can we say $\mathrm{vol}(A + B) \geq \mathrm{vol}(A) + \mathrm{vol}(B)$? This seems reasonable, even pretty weak: consider $A + A$. For convex A, $A + A = 2A$ has volume $2^n \,\mathrm{vol}(A)$, much more than $2\,\mathrm{vol}(A)$. Is there a quick proof?
- What is the right stronger statement? It should be possible to take n-th roots and still have a true statement.

Let us check this for boxes.

Claim 2.4.2 (Brunn–Minkowski for boxes)**.** *If A and B are boxes in $\mathbb{R}^n$ with sides parallel to the coordinate axes, then*

$$\mathrm{vol}(A + B)^{1/n} \geq \mathrm{vol}(A)^{1/n} + \mathrm{vol}(B)^{1/n}.$$

Proof. Let the side lengths be $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ respectively. The side lengths of $A + B$ are $a_1 + b_1, \ldots, a_n + b_n$, so we want to show that

$$\Big(\prod_{i=1}^{n}(a_i + b_i)\Big)^{1/n} \geq \Big(\prod_{i=1}^{n} a_i\Big)^{1/n} + \Big(\prod_{i=1}^{n} b_i\Big)^{1/n}.$$

Dividing,

$$\begin{aligned}\frac{\left(\prod_{i=1}^n a_i\right)^{1/n} + \left(\prod_{i=1}^n b_i\right)^{1/n}}{\left(\prod_{i=1}^n (a_i+b_i)\right)^{1/n}} &= \frac{\left(\prod_{i=1}^n a_i\right)^{1/n}}{\left(\prod_{i=1}^n (a_i+b_i)\right)^{1/n}} + \frac{\left(\prod_{i=1}^n b_i\right)^{1/n}}{\left(\prod_{i=1}^n (a_i+b_i)\right)^{1/n}} \\ &= \prod_{i=1}^n \Big(\frac{a_i}{a_i+b_i}\Big)^{1/n} + \prod_{i=1}^n \Big(\frac{b_i}{a_i+b_i}\Big)^{1/n} \\ &\le \frac{1}{n}\sum_{i=1}^n \frac{a_i}{a_i+b_i} + \frac{1}{n}\sum_{i=1}^n \frac{b_i}{a_i+b_i} = 1,\end{aligned}$$

where the inequality is the inequality of arithmetic and geometric means, applied to each of the two products. □

Theorem 2.4.3 (Brunn–Minkowski)**.** *For any (sufficiently nice) $A, B \subseteq \mathbb{R}^n$,*

$$\operatorname{vol}(A+B)^{1/n} \ge \operatorname{vol}(A)^{1/n} + \operatorname{vol}(B)^{1/n}. \tag{2.4.1}$$

- Equality holds (more or less) only when A is a translation of a dilation of B.
- An equivalent version: for $\lambda \in [0,1]$,

$$\operatorname{vol}(\lambda A + (1-\lambda)B)^{1/n} \ge \lambda \operatorname{vol}(A)^{1/n} + (1-\lambda)\operatorname{vol}(B)^{1/n}. \tag{2.4.2}$$

- This says that "$\operatorname{vol}^{1/n}$ is a concave function with respect to Minkowski sum".
- We have already proved it for boxes.

Exercise 2.4.4. Why are (2.4.1) and (2.4.2) equivalent? One direction is immediate: applying (2.4.1) to λA and $(1-\lambda)B$ and using $\operatorname{vol}(\lambda A)^{1/n} = \lambda \operatorname{vol}(A)^{1/n}$ gives (2.4.2).

2.4.3 Proof of the Brunn–Minkowski inequality

Proof of Theorem 2.4.3. We have already proved the inequality for boxes. So think of our body as a lot of little boxes, and prove the theorem by induction on the number of boxes in $A \cup B$. Formally, prove it for any finite collection of boxes (with sides parallel to the axes and disjoint interiors), and then approximate any measurable set by a sequence of finite collections of boxes.

Let

$$A^+ = A \cap \{x_n \ge 0\}, \quad A^- = A \cap \{x_n \le 0\}, \quad B^+ = B \cap \{x_n \ge 0\}, \quad B^- = B \cap \{x_n \le 0\}.$$

Translate A and B so that (Fig. 2.14)

1. A has some pair of boxes separated by the hyperplane $\{x_n = 0\}$, and
2. $\dfrac{\operatorname{vol}(A^+)}{\operatorname{vol}(A)} = \dfrac{\operatorname{vol}(B^+)}{\operatorname{vol}(B)}$.

Such translations exist. Two axis-parallel boxes with disjoint interiors have projections with disjoint interiors onto some coordinate axis. So after renaming coordinates they are separated by a hyperplane $\{x_n = c\}$, and we translate A to make $c = 0$. If A is a single box, exchange the roles of A and B; if both are single boxes, Claim 2.4.2 applies. Then translate B in the x_n direction. The fraction $\operatorname{vol}(B \cap \{x_n \geq 0\})/\operatorname{vol}(B)$ varies continuously from 0 to 1, so by the intermediate value theorem it takes the value $\operatorname{vol}(A^+)/\operatorname{vol}(A)$.

Now $A^+ \cup B^+$ and $A^- \cup B^-$ both have fewer boxes than $A \cup B$, so we can assume the theorem for them. A box cut by $\{x_n = 0\}$ contributes one box to each side, and a box that is not cut contributes to only one side. So neither side has more boxes than before, and since a pair of boxes of A is separated, each side has lost at least one. Also, $A^+ + B^+$ and $A^- + B^-$ are disjoint. The first lies in $\{x_n \geq 0\}$ and the second in $\{x_n \leq 0\}$, so they meet only in the hyperplane $\{x_n = 0\}$, which has volume zero. So

$$\operatorname{vol}(A + B) \geq \operatorname{vol}(A^+ + B^+) + \operatorname{vol}(A^- + B^-).$$

The rest is just a simple calculation. By induction, and then by condition 2, which gives $\frac{\operatorname{vol}(B^+)}{\operatorname{vol}(A^+)} = \frac{\operatorname{vol}(B)}{\operatorname{vol}(A)} = \frac{\operatorname{vol}(B^-)}{\operatorname{vol}(A^-)}$ (the last equality since $\operatorname{vol}(A^-) = \operatorname{vol}(A) - \operatorname{vol}(A^+)$, and similarly for B),

$$\begin{aligned}
\operatorname{vol}(A + B) &\geq \operatorname{vol}(A^+ + B^+) + \operatorname{vol}(A^- + B^-) \\
&\geq \big(\operatorname{vol}(A^+)^{1/n} + \operatorname{vol}(B^+)^{1/n}\big)^n + \big(\operatorname{vol}(A^-)^{1/n} + \operatorname{vol}(B^-)^{1/n}\big)^n \\
&= \operatorname{vol}(A^+)\left(1 + \left(\frac{\operatorname{vol}(B^+)}{\operatorname{vol}(A^+)}\right)^{1/n}\right)^n + \operatorname{vol}(A^-)\left(1 + \left(\frac{\operatorname{vol}(B^-)}{\operatorname{vol}(A^-)}\right)^{1/n}\right)^n \\
&= \big(\operatorname{vol}(A^+) + \operatorname{vol}(A^-)\big)\left(1 + \left(\frac{\operatorname{vol}(B)}{\operatorname{vol}(A)}\right)^{1/n}\right)^n \\
&= \big(\operatorname{vol}(A)^{1/n} + \operatorname{vol}(B)^{1/n}\big)^n.
\end{aligned}$$

Taking n-th roots gives (2.4.1). □

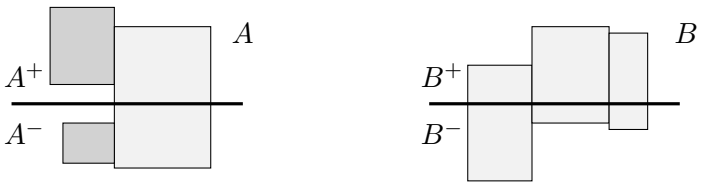


Figure 2.14: The inductive step in a small example: A and B are finite unions of boxes, translated so that the hyperplane $\{x_n = 0\}$ (thick line) separates two boxes of A (darker) and splits A and B in the same volume ratio (here 3 : 2 for both).

We now turn to the applications, starting with the volumes of the parallel slices of a convex body.

2.4.4 Volumes of parallel slices

Let $K \subseteq \mathbb{R}^n$ be a convex body. Let

$$K_t = K \cap \{x_1 = t\}, \qquad v_K(t) = \operatorname{vol}_{n-1}(K_t).$$

Figure 2.15 shows an example. What can you say about $v_K(t)$? It is concave in the example. Is it always? That seems reasonable: a ball in $\mathbb{R}^n$ has a lot more volume near the equator than near the poles.

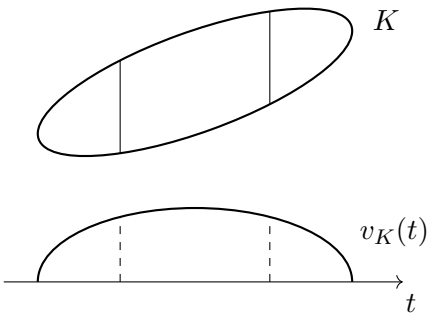


Figure 2.15: A convex body K in the plane (top) and its slice function $v_K(t)$, the length of the chord $K \cap \{x_1 = t\}$ (bottom). The two marked chords correspond to the dashed values.

What is the right theorem?

For a circular cone in $\mathbb{R}^3$ with its apex at the origin (Fig. 2.16), the slice at distance x from the apex is a disk of radius proportional to x, so $v(x)$ is proportional to x^2. So v is not concave.

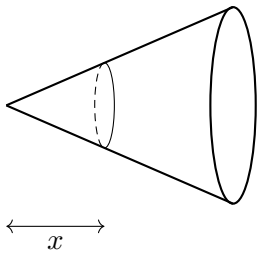


Figure 2.16: A circular cone in $\mathbb{R}^3$: the slice at distance x from the apex has area proportional to x^2.

What about a Euclidean ball in $\mathbb{R}^n$? Figure 2.17 plots the volume of the slices of the unit ball. The slice at height t is an $(n-1)$-dimensional ball of radius $\sqrt{1-t^2}$, so $v(t) = \mathrm{vol}_{n-1}(B_2^{n-1})(1-t^2)^{(n-1)/2}$. For a cone in $\mathbb{R}^n$, likewise, $v(x)$ is proportional to x^{n-1}. In both cases $v^{1/(n-1)}$ is concave. Any guesses at the right theorem?

Brunn's theorem

Theorem 2.4.5 (Brunn)**.** *The function $v(t)^{1/(n-1)}$ is concave on the interval of t for which K_t is nonempty.*

Proof. If $s = (1-\lambda)r + \lambda t$ for $\lambda \in [0,1]$, then (Fig. 2.18)

$$(1-\lambda)K_r + \lambda K_t \subseteq K_s. \tag{2.4.3}$$

A point of $(1-\lambda)K_r + \lambda K_t$ has the form $(1-\lambda)(r,x) + \lambda(t,y) = (s, (1-\lambda)x + \lambda y)$ with $(r,x), (t,y) \in K$. It lies in K by convexity, so it lies in K_s. So, by the Brunn–Minkowski

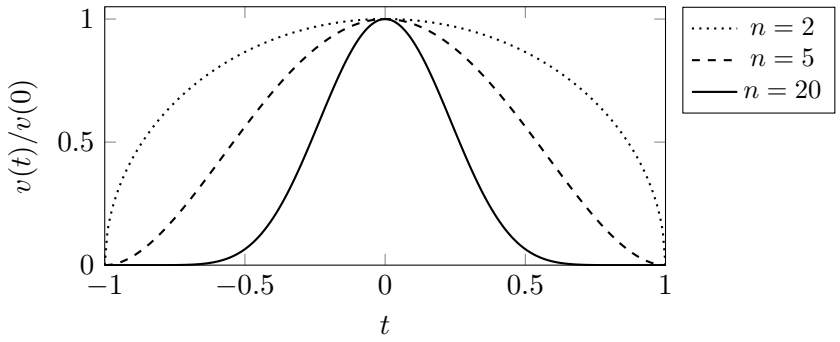


Figure 2.17: Volume of the sections $\{x_1 = t\}$ of the n-dimensional unit ball, normalized to 1 at $t = 0$. In high dimension almost all the volume is near the equator.

inequality (2.4.2) in dimension $n - 1$,

$$\operatorname{vol}(K_s)^{1/(n-1)} \geq (1-\lambda)\operatorname{vol}(K_r)^{1/(n-1)} + \lambda \operatorname{vol}(K_t)^{1/(n-1)}. \qquad \square$$

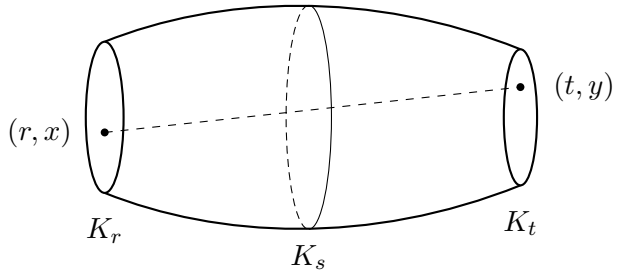


Figure 2.18: Brunn's theorem: the segment from a point of K_r to a point of K_t crosses K_s at the corresponding convex combination.

2.4.5 The isoperimetric inequality

One of the most useful questions in convex geometry is the following: what is the most volume you can enclose in a convex body with a given surface area? In two dimensions: I give you a piece of string that is one mile long; how much area can you rope off? The question is very old and has an interesting history.

Theorem 2.4.6 (isoperimetric inequality)**.** *Among bodies of a given volume, Euclidean balls have the smallest surface area. In symbols: for a convex body K, let $S(K)$ be its surface area (its $(n-1)$-dimensional volume) and $V(K)$ its (n-dimensional) volume. Then*

$$\Big(\frac{V(K)}{V(B_2^n)}\Big)^{1/n} \leq \Big(\frac{S(K)}{S(B_2^n)}\Big)^{1/(n-1)}. \tag{2.4.4}$$

The exponents are just there to get the dimensions right.

We have inequalities for volumes, but nothing yet for surface areas. How can we get a handle on surface areas? Hint: can we do something with Minkowski sums?

Proof. The surface area is the rate at which volume grows when we add a small ball (Fig. 2.19):

$$S(K) = V(\partial K) = \lim_{\epsilon \to 0} \frac{V(K + \epsilon B_2^n) - V(K)}{\epsilon}.$$

By the Brunn–Minkowski inequality,

$$\begin{aligned} V(K + \epsilon B_2^n) &\geq \left[V(K)^{1/n} + \epsilon V(B_2^n)^{1/n}\right]^n = V(K)\left[1 + \epsilon\Big(\frac{V(B_2^n)}{V(K)}\Big)^{1/n}\right]^n \\ &\geq V(K)\left[1 + n\epsilon\Big(\frac{V(B_2^n)}{V(K)}\Big)^{1/n}\right], \end{aligned}$$

using $(1+y)^n \geq 1 + ny$ for $y \geq 0$. Subtracting $V(K)$, dividing by ϵ and letting $\epsilon \to 0$,

$$S(K) \geq nV(K)\Big(\frac{V(B_2^n)}{V(K)}\Big)^{1/n} = nV(K)^{(n-1)/n} V(B_2^n)^{1/n}.$$

For the n-dimensional ball, $S(B_2^n) = nV(B_2^n)$, so

$$\frac{S(K)}{S(B_2^n)} \geq \Big(\frac{V(K)}{V(B_2^n)}\Big)^{(n-1)/n} \quad \Longrightarrow \quad \Big(\frac{S(K)}{S(B_2^n)}\Big)^{1/(n-1)} \geq \Big(\frac{V(K)}{V(B_2^n)}\Big)^{1/n}. \qquad \square$$

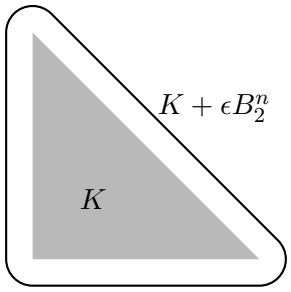


Figure 2.19: Adding ϵB_2^n to K adds a layer of thickness ϵ along the boundary (plus rounded corners, whose volume is of order ϵ^2), so the added volume is about $\epsilon\, S(K)$.

2.4.6 Grünbaum's theorem

Suppose you are sharing a (high-dimensional convex) cake. Your friend gets to cut it with a hyperplane, and you get the piece containing some point x that you choose. Which x do you pick? One reasonable choice is the *centroid*

$$x = \frac{1}{\mathrm{vol}(K)} \int_{y \in K} y \, dy.$$

You are guaranteed to get half for the sphere or the cube, or any other origin-symmetric body. How much cake are you guaranteed to get for a general convex body? A lot or a little? Does it grow or shrink with the dimension? Any guesses at the answer?

Example 2.4.7 (cones)**.** Let C be a cone with apex at the origin, axis along x_1, height h and base radius R, so that its slice at $x_1 = t$ is an $(n-1)$-dimensional ball of radius tR/h. Its centroid has first coordinate

$$\bar{x}_1 = \frac{1}{\mathrm{vol}(C)} \int_{t=0}^{h} t \,\mathrm{vol}_{n-1}(B_2^{n-1}) \Big(\frac{tR}{h}\Big)^{n-1} dt = \frac{n}{n+1}\, h,$$

since $\mathrm{vol}(C) = \int_0^h \mathrm{vol}_{n-1}(B_2^{n-1})(tR/h)^{n-1}\, dt$ and $\int_0^h t^n\, dt / \int_0^h t^{n-1}\, dt = \frac{n}{n+1} h$. The cut through the centroid perpendicular to the axis leaves on the apex side the cone scaled by $\frac{n}{n+1}$, so the fraction of the volume there is

$$\frac{V(\frac{n}{n+1}C)}{V(C)} = \Big(\frac{n}{n+1}\Big)^n \in \Big(\frac{1}{e}, \frac{1}{2}\Big].$$

Here $(\frac{n}{n+1})^n = (1+\frac{1}{n})^{-n}$ equals $\frac{1}{2}$ at $n = 1$ and decreases to $\frac{1}{e}$.
Since spheres and cubes are good, cones are probably the worst case.

Theorem 2.4.8 (Grünbaum)**.** *Let K be a convex body, and divide it into K_1 and K_2 using a hyperplane. If K_1 contains the centroid of K, then*

$$\frac{\mathrm{vol}(K_1)}{\mathrm{vol}(K)} \geq \frac{1}{e}.$$

In particular, a hyperplane through the centroid divides the volume into almost equal pieces: the worst-case ratio is $(1-\frac{1}{e}) : \frac{1}{e} \approx 63\% : 37\%$.

Proof idea

We give the idea of the proof, not every step. The proof is by *symmetrization*: transform the body into a nicer one while only making the ratio worse, and then prove the theorem for the nicer body. We already saw something like this when we first talked about the isoperimetric inequality.

Without loss of generality, put the centroid at 0 and let the cut be the hyperplane $\{x_1 = 0\}$.

Symmetrization step 1. Replace every $(n-1)$-dimensional slice K_t with an $(n-1)$-dimensional ball of the same volume, centered on the x_1-axis. Call the new body K' (Fig. 2.20).

- We have not changed the volume on either side of the cut, so the ratio in the theorem is the same for K' as for K, and the theorem for K' implies it for K. The centroid also stays on the cut. Its first coordinate is $\int t\, v(t)\, dt / \int v(t)\, dt$, which depends only on the slice volumes $v(t)$. By symmetry, the other coordinates of the centroid of K' are 0.
- K' is still convex: the radius of its slice at t is proportional to $\mathrm{vol}(K_t)^{1/(n-1)}$, which is concave by Brunn's theorem (Theorem 2.4.5). A body of revolution $\{(t, y) : \|y\| \leq \rho(t)\}$ with concave radius ρ is convex, since $\|(1-\lambda)y + \lambda y'\| \leq (1-\lambda)\rho(t) + \lambda\rho(t') \leq \rho((1-\lambda)t + \lambda t')$.

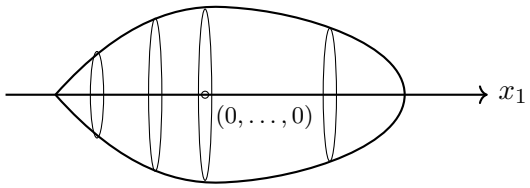


Figure 2.20: Symmetrization step 1: each slice of K is replaced by a ball of the same $(n-1)$-dimensional volume centered on the x_1-axis, giving a body of revolution K' with the centroid at the origin.

Symmetrization step 2. Turn K' into a cone (Fig. 2.21). Let

$$K'_+ = K' \cap \{x_1 \geq 0\}, \qquad K'_- = K' \cap \{x_1 \leq 0\}.$$

Make a cone with the same base (the slice of K' at $x_1 = 0$) and the same volume as K'_+, and then extend it to have the same volume on the left as K'_-. The claim is that the centroid can only move to the right. *(Proof omitted.)* This means that cones are the worst case, and we have already handled them (Example 2.4.7).

Why is this enough? The new cone has its apex on the right, the same volume as K'_+ in $\{x_1 \geq 0\}$, and its centroid at some $\bar{x} \geq 0$. The part of the cone in $\{x_1 \geq 0\}$ contains the part in $\{x_1 \geq \bar{x}\}$. That is the apex side of the cut through the centroid, so by Example 2.4.7 it has at least a $(\frac{n}{n+1})^n > \frac{1}{e}$ fraction of the volume. So $\mathrm{vol}(K'_+) \geq \frac{1}{e}\,\mathrm{vol}(K')$. The same argument with x_1 replaced by $-x_1$ gives $\mathrm{vol}(K'_-) \geq \frac{1}{e}\,\mathrm{vol}(K')$.

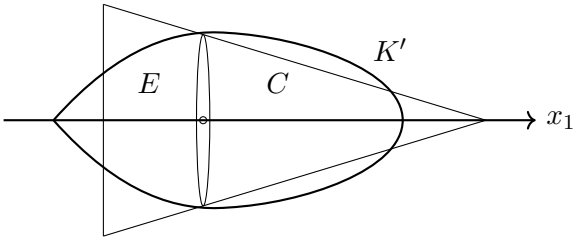


Figure 2.21: Symmetrization step 2: the part C of the cone to the right of the cut has the same base and volume as K'_+; the part E to the left has the volume of K'_-.

2.5 Approximating the Volume of a Convex Body

Given a convex body $K \subseteq \mathbb{R}^n$ through a membership oracle (a black box that answers whether a query point lies in K), how well can we compute its volume? We first show that no deterministic polynomial-time algorithm can get within better than an exponential factor. Randomness changes the picture: as with the permanent (Section 1.6.4), we write the volume as a product of ratios, each of which can be estimated by sampling uniform random points. The sampling is done with a random walk on a grid inside the body, and the main task is to show that this walk mixes in polynomial time. Bounding its conductance turns out to be a kind of isoperimetric inequality, and the *relative isoperimetric inequality* (Theorem 2.5.7) answers it even for regions that touch the boundary of the body.

For the grid walk we use the symbols of Chapter 1: conductance Φ, stationary distribution π, and $e(S)$ for the number of edges leaving a set S of vertices.

2.5.1 Deterministic algorithms cannot approximate the volume

Theorem 2.5.1. *There is no deterministic polynomial-time algorithm that, given a membership oracle for K, computes $\mathrm{vol}(K)$ within a better than exponential factor.*

How might we prove this? Impossibility results are hard: what if we choose really good points, or use some smart theorem about convex bodies?

Scale so that the body is contained in the unit ball (and contains $1/n$ times the unit ball). To get a version of the theorem for bodies that are promised to be well-rounded, replace K with $\mathrm{conv}(K \cup \{\pm\chi_1, \ldots, \pm\chi_n\})$. Now suppose we have picked points $p_1, \ldots, p_m$ by any method, and they all lie in K. Then the body could still be either $C = \mathrm{conv}(p_i)$ or B_2^n. Answer every query as if $K = B_2^n$. Then a deterministic algorithm asks the same questions and gets the same answers whether K is B_2^n or C. The points it is told are inside are the p_i, which lie in C, and the points it is told are outside lie outside $B_2^n \supseteq C$. So Theorem 2.5.1 follows from the next theorem, because for m polynomial in n the two candidate volumes differ by an exponential factor.

Theorem 2.5.2. *If $p_1, \ldots, p_m \in B_2^n$ and $C = \mathrm{conv}(p_1, \ldots, p_m)$, then*

$$\frac{\mathrm{vol}(C)}{\mathrm{vol}(B_2^n)} \le \frac{m}{2^n}.$$

For a point p, let B_p be the ball around $\frac{p}{2}$ of radius $\|p\|/2$.

Claim 2.5.3. *If $v = \lambda p + (1-\lambda)q$ with $\lambda \in [0,1]$, then $B_v \subseteq B_p \cup B_q$.*

Proof. (See Fig. 2.22.) Expanding the square,

$$\begin{aligned} B_p &= \left\{x \;\middle|\; \left\|x - \frac{p}{2}\right\|^2 \le \left\|\frac{p}{2}\right\|^2\right\} = \left\{x \;\middle|\; \|x\|^2 + \left\|\frac{p}{2}\right\|^2 - 2\left\langle x, \frac{p}{2}\right\rangle \le \left\|\frac{p}{2}\right\|^2\right\} \\ &= \{x \mid \|x\|^2 \le \langle x, p\rangle\}, \end{aligned}$$

and likewise $B_q = \{x \mid \|x\|^2 \le \langle x, q\rangle\}$. If $x \in B_v$, then

$$\|x\|^2 \le \langle x, v\rangle = \lambda\langle x, p\rangle + (1-\lambda)\langle x, q\rangle \le \max(\langle x, p\rangle, \langle x, q\rangle),$$

so $x \in B_p$ or $x \in B_q$. □

Proof of Theorem 2.5.2. The claim implies $C \subseteq \bigcup_{i=1}^m B_{p_i}$, so

$$\mathrm{vol}(C) \le \sum_{i=1}^m \mathrm{vol}(B_{p_i}) \le \frac{m}{2^n}\,\mathrm{vol}(B_2^n).$$

For the inclusion, note that each $c \in C$ lies in its own ball B_c, since $\|c\|^2 \le \langle c, c\rangle$. The computation in the proof of the claim works just as well for a convex combination $c = \sum_i \lambda_i p_i$ of any number of points, so $B_c \subseteq \bigcup_i B_{p_i}$. For the last inequality, $\|p_i\| \le 1$, so B_{p_i} has radius at most $\frac{1}{2}$ and volume at most $2^{-n}\,\mathrm{vol}(B_2^n)$. □

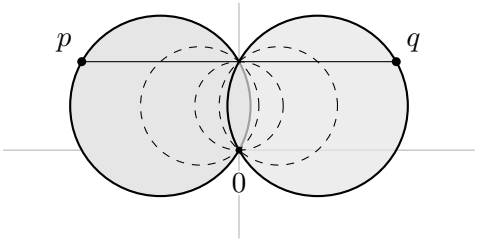


Figure 2.22: The balls B_p and B_q (shaded) have diameters $[0, p]$ and $[0, q]$. For v on the segment from p to q, the ball B_v (dashed) lies in $B_p \cup B_q$.

2.5.2 Game over?

So does this mean we cannot hope to get a polynomial approximation of the volume? The theorem used the word "deterministic", so we have to use randomness.

Maybe a Monte Carlo algorithm: pick random points in the ball and see how many land in K. This will not work, because we can make bodies where almost no points end up in K.

Does this situation sound familiar? It looks a lot like the problems we had approximating the permanent (Section 1.6.4). How did we get around them? The approach below follows the same pattern.

2.5.3 Comments on randomized volume algorithms

- *Self-indulgent philosophizing.* If we can do this (and we can), it is pretty impressive: this is one of the biggest known gaps between deterministic and randomized computation. Almost certainly BPP = P (Kelner: "I can defend this if you disagree"). So, somehow, any body whose separation oracle can be created in polynomial time can have its volume approximated deterministically, even though this is impossible for general bodies.
- Everything that follows aims for a polynomial-time algorithm; we will make no effort to get a good polynomial.
- We roughly follow the original paper of Dyer, Frieze and Kannan [DFK91] on approximating the volume.
- Much lower polynomials now exist. The current best computes the volume in $O(n^4)$.

2.5.4 Sketch of the approach

The main technical tool is a method for sampling from a convex body. Let $B = B_2^n$.

1. Change coordinates so that K is well-rounded: $B \subseteq K \subseteq n \cdot B$.
2. Let $\rho = 1 + 1/n$ and $K_i = K \cap (\rho^i \cdot B)$. Compute
$$\gamma_i = \frac{\mathrm{vol}(K_{i-1})}{\mathrm{vol}(K_i)}.$$
3. Return $\mathrm{vol}(B)/\prod_i \gamma_i$.

The last part works because $K_0 = K \cap B = B$. If t is the least integer with $\rho^t \geq n$, then $K_t = K$ (since $K \subseteq nB$), and the product telescopes: $\prod_{i=1}^{t} \gamma_i = \text{vol}(K_0)/\text{vol}(K_t) = \text{vol}(B)/\text{vol}(K)$. Here $t = \lceil \log n / \log(1 + 1/n) \rceil = O(n \log n)$.

The first part is not too hard using the ellipsoid algorithm (Section 2.5.9 sketches the idea). So we just need to do the second part, which is not hard if we can sample.

Each ratio is bounded below by a constant. If $x \in K_i$, then $x/\rho \in K$ (since $0 \in K$ and K is convex) and $\|x/\rho\| \leq \rho^{i-1}$. So $K_i \subseteq \rho K_{i-1}$, and $\gamma_i \geq \rho^{-n} = (1 + 1/n)^{-n} \geq 1/e$. So if we can draw nearly uniform samples from K_i, the fraction of them that land in K_{i-1} estimates γ_i. By a Chernoff bound, polynomially many samples give each γ_i within a factor $1 \pm \epsilon/t$, and so the product within a factor of about $1 \pm \epsilon$. This is the same pattern as for the permanent in Section 1.6.4.

2.5.5 Sampling from convex bodies

So we need a way of sampling from a convex body K. As you may have guessed, we will use a random walk.

Exercise 2.5.4 (warm-up)**.** How would you sample uniformly from a cube? A ball? An ellipse? A spherical cone? A polytope (given by polynomially many constraints)?

What might a good random walk be? A few walks have been used for this. We will study the *grid walk*.

2.5.6 The grid walk

Put a grid on $\mathbb{R}^n$ with width δ. This gives a graph H:

- the vertices are the points of the grid, that is, the points of $\delta \cdot \mathbb{Z}^n$;
- edges connect p to $p \pm \delta\chi_i$ for each i.

Let $G = H \cap K$: keep the vertices in K and the edges between them. Given a starting point, we can simulate a walk on this graph using a membership oracle. If δ is small enough, a random vertex of G is pretty much a random point in K. The graph has exponentially many vertices, so we need to show that the walk mixes really quickly; but convex bodies are nice. So our main goal will be to show that this walk mixes in polynomial time.

Exercise 2.5.5. How do we simulate the walk with a membership oracle? What happens if K is a cube?

First, there are a couple of problems to deal with.

Sampling only lattice points

First problem: we are only sampling lattice points, and never get points with coordinates that are not multiples of δ. How could we fix this? Let p be a random vertex of G, and just add a random vector v from the cube $\left[-\frac{\delta}{2}, \frac{\delta}{2}\right]^n$. Does this work? What if $v + p$ is not in K? We could keep trying v's until we get a point in K.

Exercise 2.5.6. What is wrong with retrying v until $p + v \in K$?

A (somewhat unsatisfying) fix: if we get a point not in K, start the whole random walk over.

Some not-too-bad problems

- The graph might be bipartite, or not actually bipartite but look bipartite until we take a lot of steps. So we just take a lazy random walk (Section 1.5.2).
- The graph we are walking on does not have constant degree. Most vertices have degree $2n$, but vertices near the boundary have lower degree. So the stationary distribution is not uniform. How do we fix this? Just add self-loops to equalize the degrees. Then every vertex has degree $2n$, and the stationary distribution $\pi(u) = d(u)/d(V)$ is uniform on the vertices of G.

We can restate the walk cleanly as follows. From the current vertex p:

- pick a random vector v from $\{\pm\delta\chi_i\}$;
- if $p + v \in K$, go to it;
- otherwise, stay where you are.

For the lazy walk, first stay put with probability $\frac{1}{2}$.

A worse problem

Our graph G might not be connected, and then clearly the walk will not mix quickly (Fig. 2.23). You might say "choose a finer grid", but this does not help: the problem depends on the angle, not the size. Intuitively, we need to get rid of "sharp corners". It is a plausible guess that somehow well-roundedness rules this out, but it does not. Any guesses at a fix?

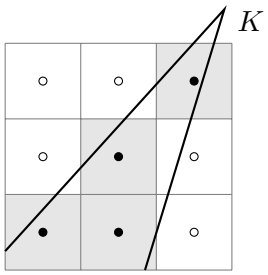


Figure 2.23: A thin wedge of K through the grid. The grid points inside K are filled and their cells shaded. The point near the tip has no grid neighbor in K (its diagonal neighbor is not adjacent in the grid graph), so G is not connected. Refining the grid does not help.

Smoothing out K

- We could try throwing in every grid point whose cube intersects K. But it is not obvious how to use a membership oracle to test this.
- Even better would be to run the walk on $K(\alpha)$, defined for any $\alpha > 0$ by

$$K(\alpha) = \{x \in \mathbb{R}^n \mid \text{dist}(x, K) \leq \alpha\} = K + \alpha \cdot B_2^n.$$

  $K(\alpha)$ has no sharp corners, as long as the grid is small enough relative to α.

- There is still a tricky question of oracles and such. It can be resolved, but it is better to avoid it. This body does provide the right intuition for what we are actually going to do.
- Instead, let $K' = (1+\alpha)K$, and walk on K'. (In this section K' denotes this scaled body.) We need to be a little careful about what we ask for. One reason walking on K' helps: since $B_2^n \subseteq K$, we have $K(\alpha) = K + \alpha B_2^n \subseteq K + \alpha K = (1+\alpha)K$, so K' contains the smoothed body. And a membership oracle for K' is just a membership oracle for K applied to $x/(1+\alpha)$.

2.5.7 Mixing and isoperimetry

Now we have a reasonable walk that converges to the distribution we are trying to sample from, and we just need to show that it mixes. For now, let us try to get an intuition. What does a body look like on which the grid walk mixes slowly? We would like to bound the conductance of the graph G (Section 1.5.6). What does this mean geometrically?

What we need to show

We need to show that the conductance of our graph is not too small. Let S be a set of vertices of G with $|S| \le |V(G)|/2$, and let $e(S)$ be the number of edges leaving S. We need to show that

$$\frac{e(S)}{|S|}$$

cannot be too small. Since every vertex has degree $2n$ after adding self-loops, the conductance of S is $\Phi(S) = e(S)/(2n|S|)$.

Further, let $Q = Q(S)$ be the union of the cubes of the grid around the vertices of S, and let $\partial Q = \partial Q(S)$ be its $(n-1)$-dimensional boundary. Suppose for now that none of the vertices of S hit the boundary of the body. Then

- $|S|$ is proportional to $\mathrm{vol}_n(Q)$, and
- $e(S)$ is proportional to $\mathrm{vol}_{n-1}(\partial Q)$.

Each cube has volume δ^n, so $\mathrm{vol}_n(Q) = \delta^n|S|$. Each facet of ∂Q separates a cube of S from the cube of a neighboring grid point outside S, which is in K by assumption. So $\mathrm{vol}_{n-1}(\partial Q) = \delta^{n-1}e(S)$, and $e(S)/|S| = \delta\ \mathrm{vol}_{n-1}(\partial Q)/\mathrm{vol}_n(Q)$. So bounding the following two quantities is equivalent:

$$\underbrace{\frac{e(S)}{|S|}}_{\text{conductance for } S} \quad \longleftrightarrow \quad \underbrace{\frac{\mathrm{vol}_{n-1}(\partial Q)}{\mathrm{vol}_n(Q)}}_{\text{isoperimetric inequality!}}$$

More precisely, what we need

So there is a strong link between the geometry and the conductance. We will need to do two things.

1. Formulate a version of the isoperimetric inequality that deals with sets hitting the boundary of the body.

2. Show that this geometric theorem properly captures the graph's properties. This is a little subtler than in Section 2.5.7, since we have to deal with surface areas near the boundary, corners, and so on.

We do (1) first, in Section 2.5.8. Part (2) is not too hard, since we have already talked about all the complexities that arise (Section 2.5.9).

2.5.8 The relative isoperimetric inequality

To incorporate the boundary we use the following theorem. From here on, d denotes the diameter of the convex body K.

Theorem 2.5.7 (Relative isoperimetric inequality)**.** *Let $K \subseteq \mathbb{R}^n$ be a convex body with diameter d. Let S be an $(n-1)$-dimensional surface that cuts K into two pieces, A and B. Then*

$$\min\{\mathrm{vol}_n(A), \mathrm{vol}_n(B)\} \leq d \cdot \mathrm{vol}_{n-1}(S). \tag{2.5.1}$$

If A does not hit the boundary of K, and K is reasonably round, this is basically the standard isoperimetric inequality (Theorem 2.4.6). The new feature is that the cut surface S only counts inside K: the part of the boundary of A that lies on the boundary of K is free.

One can ask the same question for any set X in place of the convex body K: what ϕ makes

$$\min\{\mathrm{vol}_n(A), \mathrm{vol}_n(B)\} \leq \phi \cdot \mathrm{vol}_{n-1}(S)$$

true for every partition of X into two pieces A and B by a surface S? Up to some constants and normalizations (depending on whom you ask), ϕ is called the *isoperimetric constant* or *Cheeger constant* of X. In the most common normalization the Cheeger constant is $h(X) = \inf_S \mathrm{vol}_{n-1}(S)/\min\{\mathrm{vol}_n(A), \mathrm{vol}_n(B)\}$, so the smallest valid ϕ is $1/h(X)$. Yes, this is the same Cheeger as in Cheeger's inequality (Theorem 1.4.5).

Isoperimetry for graphs and bodies

We have seen that we can relate the mixing time of a walk to geometric isoperimetry. This is why the graph quantity of Definition 1.4.1 is called an "isoperimetric" number; the connection is very deep.

- One can define a "diffusion" process on many types of sets: manifolds, graphs, convex bodies. One can then define its eigenvalues and its mixing time.
- Cheeger's inequality in its original form said that, for manifolds, the isoperimetric constant is related to the mixing time, with the same basic formula as for graphs. This is where the graph version came from. In graphs, vertices play the role of "volume" and edges play the role of "$(n-1)$-dimensional volume".
- If you ignore real-analysis issues, you can essentially prove the manifold version from the graph version, by running an appropriately defined grid walk and sending the grid size to zero.

This is a general tool and a deep theory: if you want to prove something about mixing, there is a geometric isoperimetric inequality somewhere.

Sketch of the proof

The $(n-1)$-dimensional volume in Theorem 2.5.7 is annoying to work with, so we expand S a bit and prove the following version instead.

Theorem 2.5.8. *Let $K \subseteq \mathbb{R}^n$ be a convex body with diameter d. Decompose K as $K = K_1 \cup K_2 \cup B$, where $\operatorname{dist}(K_1, K_2) \geq t$. Then*

$$\min\{\operatorname{vol}_n(K_1), \operatorname{vol}_n(K_2)\} \leq \frac{d}{t} \cdot \operatorname{vol}_n(B). \tag{2.5.2}$$

We get Theorem 2.5.7 from Theorem 2.5.8 by sending $t \to 0$. Suppose the surface S cuts K into two pieces P_1 and P_2 (called A and B in Theorem 2.5.7). Take the set B of Theorem 2.5.8 to be the points of K within distance $t/2$ of S, and let $K_i = P_i \setminus B$. Then $\operatorname{dist}(K_1, K_2) \geq t$ and $\operatorname{vol}_n(B) \approx t \cdot \operatorname{vol}_{n-1}(S)$ for small t, so (2.5.2) becomes (2.5.1) in the limit.

Proof sketch of Theorem 2.5.8. Fix a small $\epsilon > 0$. Call a body *needle-like* if its John ellipsoid (Theorem 2.3.7) has at most one axis of length at least ϵt. The theorem is fairly easy when K is needle-like (we skip this case).

Now suppose that $K = K_1 \cup K_2 \cup B$ is not needle-like and gives a counterexample. Let a hyperplane H split K_1 into $K_1' \cup K_1''$, K_2 into $K_2' \cup K_2''$, and B into $B' \cup B''$.

Claim: there is a hyperplane H such that

- H is orthogonal to the span of the two biggest axes of the John ellipsoid of K (that is, its normal vector lies in this span), and
- $\operatorname{vol}(K_1') = \operatorname{vol}(K_1'')$ and $\operatorname{vol}(K_2') = \operatorname{vol}(K_2'')$, that is, H divides both sets equally.

This follows from the *ham sandwich theorem*. Project K_1 and K_2 onto the plane P spanned by the two biggest axes. By the two-dimensional ham sandwich theorem, some line in P bisects both projected volumes. Take H to be that line times the orthogonal complement of P.

For such an H, either (K_1', K_2', B') or (K_1'', K_2'', B'') is another counterexample, and it is closer to needle-like. Each half is a convex body of diameter at most d, and its two pieces are still at distance at least t. If both halves satisfied (2.5.2), then, since $\operatorname{vol}(K_i') = \operatorname{vol}(K_i'') = \operatorname{vol}(K_i)/2$,

$$\begin{aligned}\min\{\operatorname{vol}(K_1), \operatorname{vol}(K_2)\} &= \min\{\operatorname{vol}(K_1'), \operatorname{vol}(K_2')\} + \min\{\operatorname{vol}(K_1''), \operatorname{vol}(K_2'')\} \\ &\leq \frac{d}{t}(\operatorname{vol}(B') + \operatorname{vol}(B'')) = \frac{d}{t}\operatorname{vol}(B),\end{aligned}$$

so K would not have been a counterexample. Repeating this until the body is needle-like gives a contradiction. □

2.5.9 The remaining details

Here is what remains to turn the relative isoperimetric inequality into a polynomial-time sampling algorithm, and hence (Section 2.5.4) a volume algorithm.

- We need to check that we can arrange for our body K to be polynomially well-rounded, so that its diameter is not too big. We leave this as an exercise. The rough idea: given a separation oracle, we can optimize a linear function over the body (for example with the ellipsoid algorithm). If the body is far from isotropic, we can find a point far from the origin and use it to construct a better John ellipsoid.

- We need to show that the isoperimetry of our graph is properly related to the isoperimetry of the body near the boundary. This is where we use the rounding of the corners discussed in Section 2.5.6. There are a couple of choices of how to proceed.

- The walk samples from the smoothed body $K(\alpha)$, but we need samples from K. What do we do? And we need to show that we do not reject too many samples. How? One answer: draw a sample from the smoothed body and keep it only if it lies in K (one membership query); otherwise draw again. A kept sample is uniform on K. The expected number of draws per kept sample is the volume ratio of the smoothed body to K. If K contains the unit ball B_2^n (well-roundedness), convexity gives $K(\alpha) = K + \alpha B_2^n \subseteq K + \alpha K = (1+\alpha)K$. So the ratio is at most $(1+\alpha)^n \le e^{\alpha n}$, which is at most e when $\alpha \le 1/n$. The same bound works for the body $(1+\alpha)K$ of Section 2.5.6.

- All together, we get an algorithm for sampling, and thus for volume.

2.6 Concentration of Measure

Here we see how to think of convex-body theory in terms of probability theory. We sort of did this already, by relating isoperimetry to mixing time (Section 2.5.8); here the relation will be more direct. We will be able to prove interesting theorems about probability using convex bodies, and vice versa. **The main point:** our theorems about where volume is concentrated in convex bodies have very strong probabilistic consequences.

We first rephrase the Chernoff bound as a statement about the volume of the cube near a hyperplane. We then state the isoperimetric inequality on the sphere and derive from it that every 1-Lipschitz function on a high-dimensional sphere is almost constant. Finally, we prove an approximate isoperimetric inequality on the sphere, which is a short application of the Brunn–Minkowski inequality (Theorem 2.4.3).

2.6.1 A first concentration theorem: the Chernoff bound

We already have one concentration theorem, the Chernoff bound (Theorem 1.6.1). Here it is, slightly rephrased.

Theorem 2.6.1 (Chernoff (Hoeffding, Azuma, Bernstein, ...) bound)**.** *Let $x \in \{\pm 1\}^n$ be a random vector with independent coordinates, $\Pr[x_i = 1] = 1/2$ for all i. Let $a_1, \dots, a_n \in \mathbb{R}$ satisfy $\sum_i a_i^2 = 1$ (some of them could be negative). Then for every $t > 0$,*

$$\Pr\Big[\Big|\sum_{i=1}^n a_i x_i\Big| > t\Big] \le 2e^{-t^2/2}. \tag{2.6.1}$$

(Proof omitted.)

Let us change the distribution so that x is uniform in the cube $[-\frac{1}{2}, \frac{1}{2}]^n$. The bound is still true, with slightly different constants:

$$\Pr\Big[\Big|\sum_{i=1}^n a_i x_i\Big| > t\Big] \le 2e^{-6t^2}, \qquad x \text{ uniform in } [-\tfrac{1}{2}, \tfrac{1}{2}]^n, \quad \sum_i a_i^2 = 1. \tag{2.6.2}$$

To see where the 6 comes from, take y uniform in $[-\frac{1}{2}, \frac{1}{2}]$. Then $\mathbb{E}[e^{sy}] = \frac{\sinh(s/2)}{s/2} \le e^{s^2/24}$ (compare Taylor series term by term). So $\mathbb{E}[e^{\lambda \sum_i a_i x_i}] \le e^{\lambda^2/24}$, and Markov's inequality gives $\Pr[\sum_i a_i x_i > t] \le e^{-\lambda t + \lambda^2/24}$. Taking $\lambda = 12t$ gives e^{-6t^2}, and the lower tail is the same.

What does this say? Write $a = (a_1, \dots, a_n)$. Since $\sum_i a_i^2 = 1$, the number $|\sum_i a_i x_i| = |\langle a, x \rangle|$ is the distance of x from the hyperplane $H_a = \{x : \langle a, x \rangle = 0\}$. So the theorem says that *most of the mass of the cube lies near any hyperplane through the origin.* We can also phrase it as follows: let S be the set of all points within distance t of H_a; then

$$\frac{\mathrm{vol}(S \cap [-\frac{1}{2}, \frac{1}{2}]^n)}{\mathrm{vol}([-\frac{1}{2}, \frac{1}{2}]^n)} \ge 1 - 2e^{-6t^2}. \tag{2.6.3}$$

This feels like our statement that most of the volume of the n-dimensional ball lies near the equator (Section 2.1.2; see also Fig. 2.17). It also feels a bit like the relative isoperimetric inequality, although it deals with big sets rather than small ones, so it is quantitatively different. Below we show how all these are related.

2.6.2 Isoperimetry on the sphere

Let us ask the isoperimetric question on the sphere $S^{n-1} \subseteq \mathbb{R}^n$.

- For distance we use the distance in $\mathbb{R}^n$ (the chord), not the geodesic distance on the sphere.
- Volumes are with respect to the unique rotationally invariant measure on the sphere, normalized to have total volume 1.
- For any $A \subseteq S^{n-1}$, let A_ϵ be the set of points of the sphere within distance ϵ of A: $A_\epsilon = \{x \in S^{n-1} : \mathrm{dist}(x, A) < \epsilon\}$.

Question: out of all sets A with $\mathrm{vol}(A) = q$, which ones give the minimum possible $\mathrm{vol}(A_\epsilon)$?
The answer: spherical caps.

Definition 2.6.2 (Spherical cap)**.** The *spherical cap* $C(r, v)$, for $v \in S^{n-1}$ and $r > 0$, is

$$C(r, v) = \{x \in S^{n-1} : \|x - v\| \le r\}.$$

A cap is just a ball in the metric on the sphere, so this answer is like the Euclidean isoperimetric inequality, where balls are optimal. We need a slightly more complicated formulation than in the Euclidean case, because the volume ratio depends on the radius; this is why the question fixes both the volume q and the distance ϵ.

It is sometimes easier to talk about the *cap at height t*:

$$c_t = c(t, v) = \{x \in S^{n-1} : \langle x, v \rangle \ge t\}.$$

The two descriptions agree: for unit vectors, $\|x - v\|^2 = 2 - 2\langle x, v \rangle$, so $C(r, v) = c(1 - r^2/2, v)$. The cap c_0 is a hemisphere, and

$$\mathrm{vol}(c_t) \approx e^{-nt^2/2}. \tag{2.6.4}$$

This is the same computation as "most of the volume of the ball is near the equator" (Section 2.1.2). In fact $\mathrm{vol}(c_t) \le e^{-nt^2/2}$ for $0 \le t < 1$, and the exponent is right up to lower-order terms. This upper bound is what we use below.

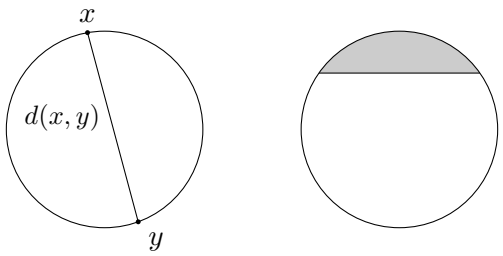


Figure 2.24: Left: the Euclidean (chordal) metric on the sphere. Right: a spherical cap is a ball for this metric.

Theorem 2.6.3 (Isoperimetric inequality on the sphere)**.** *Among all measurable $A \subseteq S^{n-1}$ with* $\mathrm{vol}(A) = q$*, the minimum of* $\mathrm{vol}(A_\epsilon)$ *is attained when A is a spherical cap.*

(Proof omitted; the actual proof is somewhat tricky; an approximate version is proved in Section 2.6.6.)

What does this imply?

We will prove an approximate version of Theorem 2.6.3 in Section 2.6.5. First let us see why it is interesting. The actual proof is tricky, but we can get the asymptotics right very easily with what we already know.

Corollary 2.6.4. *For any $A \subseteq S^{n-1}$ with* $\mathrm{vol}(A) = 1/2$*,*

$$\mathrm{vol}(A_\epsilon) \geq 1 - e^{-n\epsilon^2/2}. \tag{2.6.5}$$

Why? By Theorem 2.6.3, the worst case is a cap of volume $1/2$, that is, a hemisphere $c_0 = \{x : \langle x, v\rangle \geq 0\}$. A point at height $-t$ is at distance $\sqrt{2 - 2\sqrt{1-t^2}} \approx t$ from the hemisphere. So the complement of $(c_0)_\epsilon$ is a cap $c(t, -v)$ at height $t = \epsilon\sqrt{1 - \epsilon^2/4}$, just below ϵ. By (2.6.4) its volume is about $e^{-n\epsilon^2/2}$, which gets the exponent right to leading order. The bound $\mathrm{vol}(c_t) \leq e^{-nt^2/2}$ alone only gives $e^{-n\epsilon^2(1-\epsilon^2/4)/2}$, but a numerical check with the exact cap volumes shows that (2.6.5) holds as stated. See Fig. 2.25.

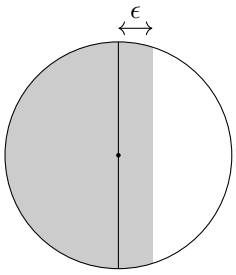


Figure 2.25: An ϵ-neighborhood of a hemisphere (shaded). What is left over is a cap at height about ϵ.

So almost all of the sphere is near any set of volume $1/2$, even though there may be many points far away from it. We could have used $1/4$ or $1/8$ instead of $1/2$ and gotten the same thing, with different constants.

2.6.3 Lipschitz functions are almost constant

Definition 2.6.5. A function $f : S^{n-1} \to \mathbb{R}$ is 1-*Lipschitz* if for all $a, b \in S^{n-1}$,

$$|f(a) - f(b)| \leq \|a - b\|_2.$$

Let M be a *median* of a 1-Lipschitz function f, so that

$$\text{vol}(\{x : f(x) \leq M\}) \geq 1/2 \quad \text{and} \quad \text{vol}(\{x : f(x) \geq M\}) \geq 1/2.$$

How many points have $f(x) \leq M + \epsilon$?

Theorem 2.6.6 (Concentration of Lipschitz functions)**.** *If $f : S^{n-1} \to \mathbb{R}$ is 1-Lipschitz with median M, then for every $\epsilon > 0$,*

$$\text{vol}(\{x : |f(x) - M| > \epsilon\}) \leq 2e^{-n\epsilon^2/2}. \tag{2.6.6}$$

Proof. Let $A = \{x : f(x) \leq M\}$, so $\text{vol}(A) \geq 1/2$. If $y \in A_\epsilon$, there is an $x \in A$ with $\|x - y\| < \epsilon$, so $f(y) < f(x) + \epsilon \leq M + \epsilon$ by the Lipschitz condition. So $A_\epsilon \subseteq \{f < M + \epsilon\}$. Now apply Corollary 2.6.4 to a subset of A of volume exactly $1/2$ (enlarging A only enlarges A_ϵ). This gives $\text{vol}(\{f \geq M + \epsilon\}) \leq 1 - \text{vol}(A_\epsilon) \leq e^{-n\epsilon^2/2}$. The same argument for $\{f \geq M\}$ gives $\text{vol}(\{f \leq M - \epsilon\}) \leq e^{-n\epsilon^2/2}$, and adding the two bounds gives (2.6.6). □

So even though f can vary by up to 2 on the sphere (its diameter), it is almost equal to M almost everywhere: "1-Lipschitz functions are almost constant". This is a little weird. It is what we call *concentration of measure*. Note that for this we only need the right asymptotics, not the exact isoperimetric inequality.

2.6.4 Equivalence of concentration and isoperimetry

We have two types of properties.

- **Isoperimetric inequalities:** every set A with $\text{vol}(A) \geq \frac{1}{2}$ has $\text{vol}(A_t) \geq 1 - e^{-ct^2}$.
- **Concentration theorems:** for every 1-Lipschitz function f with median M, $\text{vol}(\{x : |f(x) - M| > t\}) \leq 2e^{-ct^2/2}$.

(On S^{n-1} the constant c is proportional to n.) We showed in Section 2.6.3 that isoperimetric inequalities imply concentration theorems. One can go the other way as well. This works whenever we have a notion of volume and distance.

Exercise 2.6.7. Show that a concentration theorem implies an isoperimetric inequality, by applying the concentration theorem to the function $f(x) = \text{dist}(x, A)$.

2.6.5 An approximate isoperimetric inequality for the sphere

We will prove the following theorem, which is qualitatively the same as the isoperimetric inequality but has different constants.

Theorem 2.6.8 (Approximate isoperimetric inequality)**.** *For any $A \subseteq S^{n-1}$ and any $\epsilon > 0$,*

$$\text{vol}(A_\epsilon) > 1 - \frac{2e^{-n\epsilon^2/16}}{\text{vol}(A)}. \tag{2.6.7}$$

Note the dependence on $\mathrm{vol}(A)$. The proof (Section 2.6.6) uses the following definition. In the rest of this section, δ denotes the modulus of convexity.

Definition 2.6.9 (Modulus of convexity)**.** The *modulus of convexity* of the sphere is

$$\delta(\epsilon) = \inf\Big\{1 - \Big\|\frac{x+y}{2}\Big\| \;\Big|\; \|x\| = \|y\| = 1,\ \|x-y\| \geq \epsilon\Big\}.$$

It is easy to check that

$$\delta(\epsilon) \geq 1 - \sqrt{1 - \frac{\epsilon^2}{4}} \geq \frac{\epsilon^2}{8}. \tag{2.6.8}$$

By the parallelogram law, $\|x+y\|^2 = 2\|x\|^2 + 2\|y\|^2 - \|x-y\|^2 \leq 4 - \epsilon^2$, so $\|(x+y)/2\| \leq \sqrt{1-\epsilon^2/4}$. The second inequality is just $\sqrt{1-u} \leq 1 - u/2$ for $u \in [0,1]$.

2.6.6 Proof of the approximate isoperimetric inequality

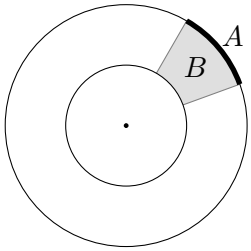


Figure 2.26: The set A on the sphere (thick arc) and the set $B = [\frac{1}{2}, 1]A$ inside the ball (shaded).

Proof of Theorem 2.6.8. In this proof, the volume of a subset of the unit ball B_2^n is its Lebesgue measure normalized so that $\mathrm{vol}(B_2^n) = 1$, and the volume of a subset of the sphere is normalized as before. Let

$$B = [\tfrac{1}{2}, 1]A = \{sa : a \in A,\ \tfrac{1}{2} \leq s \leq 1\}$$

(see Fig. 2.26), and let $B_\epsilon = \{x \in \mathbb{R}^n : \mathrm{dist}(x, B) < \epsilon\}$. We will show

$$\mathrm{vol}(B_\epsilon \cap B_2^n) \geq 1 - \frac{e^{-2n\delta(\epsilon)}}{\mathrm{vol}(B)}. \tag{2.6.9}$$

This implies the theorem:

$$\mathrm{vol}(A_{2\epsilon}) > \mathrm{vol}(B_\epsilon \cap B_2^n) \geq 1 - \frac{e^{-2n\delta(\epsilon)}}{\mathrm{vol}(B)} \geq 1 - \frac{2e^{-n\epsilon^2/4}}{\mathrm{vol}(A)}. \tag{2.6.10}$$

The first inequality comes from projecting radially onto the sphere. A point within ϵ of B projects to a point within about 2ϵ of A, since the points of B have norm at least $\frac{1}{2}$ (Remark 2.6.10 has the precise constant). The last inequality uses $\delta(\epsilon) \geq \epsilon^2/8$ and $\mathrm{vol}(B) = (1 - 2^{-n})\,\mathrm{vol}(A) \geq \mathrm{vol}(A)/2$, since B is the part of the cone over A between radii $\frac{1}{2}$ and 1. Replacing ϵ by $\epsilon/2$ in (2.6.10) gives (2.6.7).

It remains to prove (2.6.9). Let

$$C = \{x \in B_2^n : \mathrm{dist}(x, B) \geq \epsilon\},$$

so that $B_\epsilon \cap B_2^n = B_2^n \setminus C$. If $b \in B$ and $c \in C$, then $\|b\| \le 1$, $\|c\| \le 1$ and $\|b - c\| \ge \epsilon$, so

$$\Big\| \frac{b+c}{2} \Big\| \le 1 - \delta(\epsilon).$$

The definition of δ uses unit vectors, but the parallelogram-law computation after (2.6.8) works just as well for vectors in the ball: $\|b+c\|^2 = 2\|b\|^2 + 2\|c\|^2 - \|b-c\|^2 \le 4 - \epsilon^2$, and $\delta(\epsilon) = 1 - \sqrt{1 - \epsilon^2/4}$ for $\epsilon \le 2$. Hence

$$\tfrac{1}{2}(B + C) \subseteq (1 - \delta(\epsilon))B_2^n \quad \Longrightarrow \quad \operatorname{vol}\big(\tfrac{1}{2}(B+C)\big)^{1/n} \le 1 - \delta(\epsilon).$$

By the Brunn–Minkowski inequality and then the inequality between the arithmetic and geometric means,

$$1 - \delta(\epsilon) \ge \operatorname{vol}\big(\tfrac{1}{2}(B+C)\big)^{1/n} \ge \tfrac{1}{2}\operatorname{vol}(B)^{1/n} + \tfrac{1}{2}\operatorname{vol}(C)^{1/n} \ge \big(\operatorname{vol}(B)\operatorname{vol}(C)\big)^{1/(2n)}.$$

Raising to the power $2n$,

$$\operatorname{vol}(C) \le \frac{(1 - \delta(\epsilon))^{2n}}{\operatorname{vol}(B)} \le \frac{e^{-2n\delta(\epsilon)}}{\operatorname{vol}(B)},$$

using $1 - u \le e^{-u}$. Finally,

$$\operatorname{vol}(B_\epsilon \cap B_2^n) = 1 - \operatorname{vol}(C) \ge 1 - \frac{e^{-2n\delta(\epsilon)}}{\operatorname{vol}(B)},$$

which is (2.6.9). □

Remark 2.6.10. The first inequality in (2.6.10) uses two facts. The normalized volume of a cone $\{sa' : a' \in A',\ 0 < s \le 1\}$ equals $\operatorname{vol}(A')$, and every point $x \in B_\epsilon \cap B_2^n$ projects radially into a neighborhood of A. How big a neighborhood? Let $\epsilon < \frac{1}{2}$ and $\|x - sa\| < \epsilon$ with $a \in A$, $s \ge \frac{1}{2}$. Then $x \ne 0$, and the angle θ between x and a satisfies $s \sin\theta < \epsilon$, so $\sin\theta < 2\epsilon$. The chordal distance from $x/\|x\|$ to a is $2\sin(\theta/2) = \sin\theta/\cos(\theta/2) < 2\epsilon/\cos(\theta/2)$. This is $2\epsilon(1 + O(\epsilon^2))$, and at most $2\sqrt{2}\,\epsilon$, but not exactly 2ϵ. So this route, done exactly, gives a slightly worse constant than 16.

The theorem as stated is still true. Applying the same Brunn–Minkowski argument to full cones avoids the radial step. Let $\widetilde{A} = \{sa : a \in A,\ 0 \le s \le 1\}$ and $\widetilde{D} = \{rb : b \in S^{n-1} \setminus A_\epsilon,\ 0 \le r \le 1\}$, so $\operatorname{vol}(\widetilde{A}) = \operatorname{vol}(A)$ and $\operatorname{vol}(\widetilde{D}) = 1 - \operatorname{vol}(A_\epsilon)$. For $a \in A$ and $b \notin A_\epsilon$ we have $\langle a, b\rangle = 1 - \|a-b\|^2/2 \le 1 - \epsilon^2/2$. The function $(s,r) \mapsto \|sa + rb\|^2 = s^2 + r^2 + 2sr\langle a,b\rangle$ is convex on $[0,1]^2$, so it is at most its largest value at a corner, $\max(1, 4 - \epsilon^2)$. So for $0 < \epsilon \le 2$ we get $\frac{1}{2}(\widetilde{A} + \widetilde{D}) \subseteq (1 - \epsilon^2/8)B_2^n$. The computation in the proof then gives $\operatorname{vol}(A)(1 - \operatorname{vol}(A_\epsilon)) \le (1 - \epsilon^2/8)^{2n} \le e^{-n\epsilon^2/4}$, that is, $\operatorname{vol}(A_\epsilon) \ge 1 - e^{-n\epsilon^2/4}/\operatorname{vol}(A)$. This is stronger than (2.6.7). For $\epsilon > 2$ and nonempty A, A_ϵ is the whole sphere.

2.7 Metric Embeddings and the Johnson–Lindenstrauss Theorem

Concentration on the sphere has a striking algorithmic consequence. Any n points in Euclidean space, of any dimension, can be mapped into $O(\epsilon^{-2}\log n)$ dimensions while all pairwise distances change by a factor of at most $1 + \epsilon$. This is the Johnson–Lindenstrauss theorem. The map is a scaled projection onto a random subspace, so we also discuss how to generate random subspaces with Gaussians. We end with a survey of algorithmic applications of dimension reduction.

2.7.1 Metric embeddings

We will get a very nice theorem, the Johnson–Lindenstrauss theorem, as a simple corollary of measure concentration. It is a first example of a very powerful algorithmic tool: metric embeddings.

Let d be any metric on a finite set $X = \{x_1, \dots, x_n\}$, and let $f : X \to \mathbb{R}^k$ be a map such that $\|f(x_i) - f(x_j)\| \le d(x_i, x_j)$ for all i, j.

Definition 2.7.1 (Distortion)**.** The *distortion* of f is the minimum D for which

$$\|f(x_i) - f(x_j)\| \le d(x_i, x_j) \le D\|f(x_i) - f(x_j)\| \quad \text{for all } i, j.$$

Claim 2.7.2 (Johnson–Lindenstrauss)**.** *The Euclidean metric on any finite set X of n points (in a space of any dimension) can be embedded with distortion $1 + \epsilon$ in $\mathbb{R}^k$ for $k = O(\epsilon^{-2} \log n)$.*

- If we lose the ϵ (that is, ask for distortion exactly 1), this becomes impossible for $k < n - 1$ in general. Any n points lie in an affine subspace of dimension $n - 1$, and the vertices of a regular simplex need all $n - 1$ dimensions.
- We need the assumption that the metric is Euclidean: not all metrics can be embedded as distances in some Euclidean space.
- The embedding will just be: let y be the projection of x onto a random k-dimensional subspace, and return cy for some fixed constant c.

2.7.2 The Johnson–Lindenstrauss theorem

Here is the precise statement.

Theorem 2.7.3 (Johnson–Lindenstrauss)**.** *Let $X = \{x_1, \dots, x_n\} \subseteq \mathbb{R}^m$, let $k = O(\log n/\epsilon^2)$, let $L \subseteq \mathbb{R}^m$ be a uniformly random k-dimensional subspace, let $y_1, \dots, y_n$ be the projections of the x_i onto L, and let $y_i' = cy_i$ for a fixed constant $c = \Theta(\sqrt{m/k})$. Then, with high probability,*

$$\|x_i - x_j\| \le \|y_i' - y_j'\| \le (1 + \epsilon)\|x_i - x_j\| \quad \textit{for all } i, j. \tag{2.7.1}$$

Projecting a unit vector onto a random k-dimensional subspace of $\mathbb{R}^m$ leaves a vector of length about $\sqrt{k/m}$, so the rescaling factor that restores lengths is $\sqrt{m/k}$. Note that the theorem normalizes the map so that it does not shrink distances, while Definition 2.7.1 asks that it not stretch them. Dividing the y_i' by $1 + \epsilon$ converts one into the other. So (2.7.1) gives an embedding of distortion at most $1 + \epsilon$, which is Claim 2.7.2.

Proof idea. Let

$$f(x) = \sqrt{x_1^2 + \cdots + x_k^2},$$

the length of the projection of x onto a fixed k-dimensional subspace (the span of the first k coordinates). Let M be the median of f over the sphere. The function f is Lipschitz, so it is tightly concentrated (Theorem 2.6.6): a random x on the sphere has $f(x)$ very close to M. Instead of choosing a random x and a fixed k-dimensional subspace, take a fixed x and a random subspace; this is the same thing. So projection onto a random subspace is very likely to send a unit vector to something of length about M. Apply this to the vectors $x_i - x_j$ for all i, j.

Proof sketch of Theorem 2.7.3. The function f is 1-Lipschitz, since $|f(x) - f(x')| \le f(x - x') \le \|x - x'\|$. So the concentration theorem of Theorem 2.6.6 on S^{m-1} gives $\Pr[|f(x) - M| > t] \le 2e^{-mt^2/2}$ for x uniform on S^{m-1}. Also $M = \Theta(\sqrt{k/m})$, since $\mathbb{E}[f(x)^2] = k/m$ by symmetry and f is concentrated around M (once k is at least a large constant). For a fixed unit vector u, the length of its projection onto L has the same distribution as $f(x)$. Taking $t = \epsilon M$,

$$\Pr\big[\,|\|P_L u\| - M| > \epsilon M\,\big] \le 2e^{-m\epsilon^2 M^2/2} = e^{-\Omega(\epsilon^2 k)},$$

where P_L is the orthogonal projection onto L. With $k = C\epsilon^{-2}\log n$ for a large constant C this is less than $1/n^2$. There are fewer than $n^2/2$ unit vectors $u = (x_i - x_j)/\|x_i - x_j\|$, so by a union bound, with probability at least $1/2$ every pair satisfies $(1-\epsilon)M\|x_i - x_j\| \le \|y_i - y_j\| \le (1+\epsilon)M\|x_i - x_j\|$. Setting $c = 1/((1-\epsilon)M) = \Theta(\sqrt{m/k})$ gives (2.7.1) with $(1+\epsilon)/(1-\epsilon) = 1 + O(\epsilon)$ in place of $1 + \epsilon$. Now rescale ϵ. □

2.7.3 Where do we get a random subspace?

The algorithms for the applications below all say that we should embed our high-dimensional points into a low-dimensional space. This means that we actually need the embedding, not just the theorem that one exists. The proof was constructive(-ish): it said that a random subspace works. So how do we get a random subspace? First, how do we get a random point on the sphere? Answer: Gaussians.

Quick comments on Gaussians

Gaussians are really nice. A multivariate Gaussian has density

$$f_X(x_1, \ldots, x_n) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\Big(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu)\Big)$$

for a nonsingular matrix Σ (the covariance matrix) and a vector μ (the mean); here $|\Sigma|$ is the determinant of Σ. We write $\mathcal{N}(\mu, \Sigma)$ for this distribution.

- The following operations on Gaussian variables yield other Gaussian variables: projecting onto a lower-dimensional subspace; restricting to a lower-dimensional subspace (conditional probabilities); applying any linear map.
- In particular, the coordinates of a Gaussian with $\Sigma = I$ are independent one-dimensional Gaussians. So we can sample from $\mathcal{N}(0, I)$ by choosing each coordinate independently from a one-dimensional Gaussian, and we can get other Gaussians by applying linear maps.

More on Gaussians

Gaussians let us get a random vector on the sphere. How?

- When each coordinate is drawn from a Gaussian with mean 0 and variance 1, most n-vectors have norm about $\sqrt{n}$.
- So a Gaussian variable looks a lot like a random vector on the (scaled) sphere. Other Gaussians give ellipsoids.

- The central limit theorem just says that product distributions of (nice enough) one-dimensional distributions look like Gaussians if we take enough products. So cubes look like spheres too, and the Chernoff bound is a concentration of measure theorem about discrete cubes (Theorem 2.6.1).

Generating a random subspace

So we can get a random vector on the sphere by generating n one-dimensional Gaussians $g_1, \ldots, g_n$ and returning $g/\|g\|$. This is uniform on the sphere since the density of $\mathcal{N}(0, I)$ depends only on $\|x\|$. So the density is invariant under rotations, and so is the distribution of $g/\|g\|$.

We can get a random k-dimensional subspace by generating kn one-dimensional Gaussians. Hint: projections of Gaussians are Gaussians, so this gives a random projection map.

Exercise 2.7.4. How, and why, do kn independent one-dimensional Gaussians give a uniformly random k-dimensional subspace of $\mathbb{R}^n$ (and a random projection map onto it)?

2.7.4 Applications of Johnson–Lindenstrauss

There are tons of algorithmic applications of this theorem. It says that we can approximately answer questions about m-dimensional objects in a much lower-dimensional space. High-dimensional geometry questions often have complexity growing very quickly in the dimension, so "dimension reduction" can be used to speed up almost any such algorithm. We give a few examples: proximity problems, sublinear storage and streaming algorithms, and online algorithms. See the survey by Piotr Indyk [Ind01] and its references for more on applications of Johnson–Lindenstrauss.

Fast spectral sparsification. Johnson–Lindenstrauss is an important part of the fast spectral sparsification algorithm (Sections 1.9.9 and 1.10.3): we can compute *all* effective resistances by computing norms of logarithmically many vectors. The reason: with $L = B^\top C B$, we have $R_{\text{eff}}(u, v) = b_{uv}^\top L^\dagger b_{uv} = \|C^{1/2} B L^\dagger b_{uv}\|^2$, since $L^\dagger L L^\dagger = L^\dagger$. So every effective resistance is a squared distance between two columns of the $m \times n$ matrix $C^{1/2} B L^\dagger$. A random projection Q with $O(\log n/\epsilon^2)$ rows preserves all these distances up to $1 \pm \epsilon$, and the rows of $QC^{1/2}BL^\dagger$ can be computed with one Laplacian solve each.

Proximity problems. The input is a set P of points in $\mathbb{R}^d$, and we want to compute some property defined only in terms of the distances between points in P: the closest pair, the furthest pair, a minimum spanning tree, a minimum-cost matching, many clustering problems, and so on. We can solve the problem approximately in the lower-dimensional space, much faster.

Online problems. Here we get the points as we go, and we have to create a data structure that answers some query quickly, is not too hard to construct, and answers queries quickly. Such data structures tend to have exponential dependence on the dimension, such as $(1/\epsilon)^d$, so dimension reduction *really* helps.

Sublinear storage and streaming. For example, we get data in a stream and cannot store it all, but we want to be able to answer questions about it later: are there a lot of repeated entries? How are the counts distributed? The idea is to use "sketches" based on Johnson–Lindenstrauss and other such probabilistic techniques. Piotr Indyk has taught a class on this; again see his survey [Ind01] and its references.

Chapter 3

Iterative Methods and Optimization

This chapter is about iterative methods: algorithms that improve a current guess step by step, using only cheap operations such as multiplying a vector by a matrix or evaluating a gradient. Sections 3.1 to 3.3 solve linear systems this way. Sections 3.4 to 3.8 minimize general convex functions, given an oracle for their gradients or subgradients. In both settings the number of iterations is governed by a condition number κ, and the best methods need about $\sqrt{\kappa}$ iterations rather than κ.

3.1 Iterative Methods for Linear Systems

The problem is to solve a linear system $Ax = b$. For the large sparse systems that arise in practice, inverting A or factoring it is too slow and uses too much memory, while multiplying a vector by A is cheap. Iterative methods use only such multiplications. We first turn the linear problem into the problem of minimizing a quadratic function, then study steepest descent and its simpler cousin, Richardson iteration, whose number of iterations grows linearly with the condition number κ of A. Viewing each iterate as a polynomial in A applied to b shows that Chebyshev polynomials do much better, with about $\sqrt{\kappa}$ iterations, and suggests looking for the best iterate in each *Krylov subspace*. This leads to the conjugate gradient method of Section 3.2.

In this section and the next two, x_* is the solution of $Ax = b$, the iterates are x_k, the residual is $r_k = b - Ax_k$, and the error is $e_k = x_k - x_*$.

3.1.1 Why iterative methods

How do you solve the matrix equation $Ax = b$?

- **Standard answer:** multiply both sides by A^{-1}. This can be slow and memory-intensive and can have precision issues. Real problems tend to be sparse, but inverses are not.
- **A better answer, sometimes:** Gaussian elimination (LU factorization) and back-substitution. This is a good idea if your problem is dense and unstructured, but problems are rarely dense and unstructured.
- **Often a better answer for sparse or structured systems:** iterative methods.

In this section and in Section 3.2, all matrices are symmetric $n \times n$ matrices that are positive definite (and thus nondegenerate). We can reduce any nondegenerate square linear system $Ax = b$

to this case by solving $A^\top Ax = A^\top b$. If A is sparse, $A^\top A$ may not be, but we can compute $A^\top Ax$ with two sparse matrix-vector multiplications.

3.1.2 Turning a linear problem into a quadratic one

Let

$$f(x) = \frac{1}{2}x^\top Ax - b^\top x + c \tag{3.1.1}$$

At what vector x is f minimized? The gradient is

$$\nabla f(x) = \frac{1}{2}(A + A^\top)x - b.$$

If A is symmetric this equals $Ax - b$, so $\nabla f(x) = 0$ exactly when $Ax = b$. So minimizing f gives us the solution if A is *symmetric and positive definite*. For symmetric A with $Ax_* = b$, expanding f around x_* gives

$$f(x) = f(x_*) + \frac{1}{2}(x - x_*)^\top A(x - x_*), \tag{3.1.2}$$

and for positive definite A the last term is positive unless $x = x_*$.

What happens if A is unsymmetric? Symmetric and negative definite? Symmetric and neither positive nor negative definite? Let us look at some pictures, for the example

$$A = \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}, \qquad b = \begin{pmatrix} 2 \\ -8 \end{pmatrix}, \qquad c = 0, \qquad x_* = \begin{pmatrix} 2 \\ -2 \end{pmatrix}. \tag{3.1.3}$$

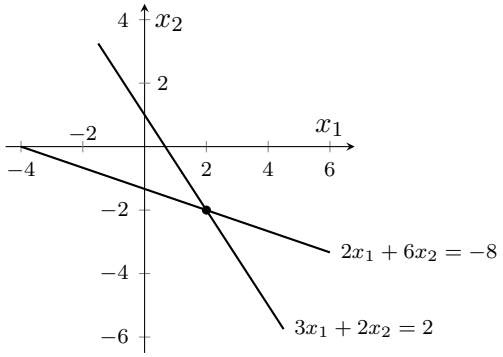


Figure 3.1: The two-dimensional system (3.1.3): each equation is a line, and the solution $x_* = (2, -2)$ is their intersection.

What are the eigenvectors of A? By (3.1.2), each contour of f is an ellipse centered at x_*, and its axes point along the eigenvectors of A. Here the eigenvalues are 2 and 7, with eigenvectors $(2, -1)$ and $(1, 2)$. The ellipse is longer along the eigenvector of the smaller eigenvalue.

Figure 3.5 answers the questions above for symmetric matrices. If A is positive definite, f is a bowl with a unique minimum. If A is negative definite, the bowl is upside down, and the solution of $Ax = b$ is the maximum of f. If A is positive semidefinite and singular, f has a whole line of minima (when $Ax = b$ is solvable). If A is indefinite, the solution is a saddle point, so minimizing f does not find it. If A is not symmetric, the formula for the gradient shows that f only sees the symmetric part $\frac{1}{2}(A + A^\top)$, so minimizing f solves $\frac{1}{2}(A + A^\top)x = b$ rather than $Ax = b$.

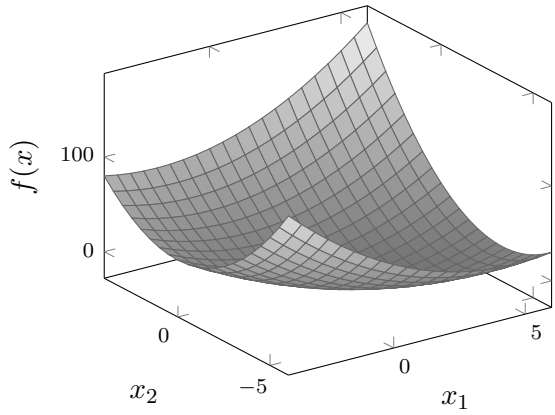


Figure 3.2: Graph of the quadratic form $f(x) = \frac{1}{2}x^\top Ax - b^\top x + c$ for (3.1.3). The minimum point of this surface is the solution to $Ax = b$.

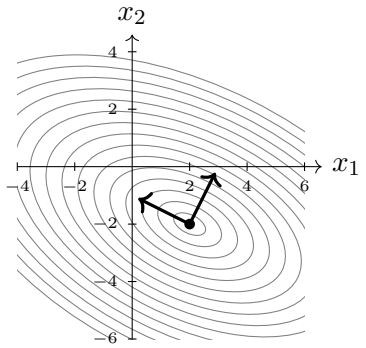


Figure 3.3: Contours of the quadratic form; each ellipse has constant $f(x)$. The arrows point along the eigenvectors of A.

3.1.3 Steepest descent

How should we minimize the quadratic form f? A very natural approach is *steepest descent*: find the direction of steepest decrease and take a step in that direction. Two obvious questions:

- What direction is it? The negative gradient.
- How big a step should we take? Use a line search.

From now on A is symmetric positive definite and $c = 0$: $f(x) = \frac{1}{2}x^\top Ax - b^\top x$. With the notation fixed at the start of the section, the residual is

$$r_k = b - Ax_k = -Ae_k = -\nabla f(x_k),$$

so the residual is the direction of steepest descent. The steepest descent step is

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) = x_k + \alpha_k r_k. \tag{3.1.4}$$

We can solve for the optimal α_k and get

$$\alpha_k = \frac{r_k^\top r_k}{r_k^\top A r_k}. \tag{3.1.5}$$

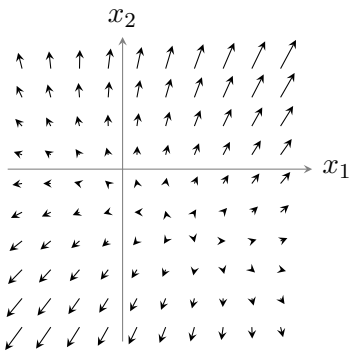


Figure 3.4: The gradient $\nabla f(x) = Ax - b$ of the quadratic form. At every x, the gradient points in the direction of steepest increase of $f(x)$, and is orthogonal to the contour lines.

To see this, note that $\frac{d}{d\alpha} f(x_k + \alpha r_k) = \nabla f(x_k + \alpha r_k)^\top r_k = -(r_k - \alpha A r_k)^\top r_k$, which vanishes exactly at (3.1.5). The same computation shows that at the chosen point the new gradient is orthogonal to the search direction: $r_{k+1}^\top r_k = 0$. This is the orthogonality visible in Fig. 3.6.

Algorithm 3.1.1 (Steepest descent, implementation note)**.** Each iteration computes

$$r_k = b - Ax_k, \qquad \alpha_k = \frac{r_k^\top r_k}{r_k^\top A r_k}, \qquad x_{k+1} = x_k + \alpha_k r_k.$$

The time is dominated by two matrix-vector multiplications, Ax_k and Ar_k. We can cut this to one (after the first iteration) by using

$$r_{k+1} = r_k - \alpha_k A r_k$$

(expand $r_{k+1} = b - A(x_k + \alpha_k r_k)$), since Ar_k is needed for α_k anyway. This can make roundoff errors accumulate, so we should periodically use the top equation $r_k = b - Ax_k$ too.

The running time is dictated by the condition number

$$\kappa = \kappa(A) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

(see Fig. 3.8).

3.1.4 Analyzing steepest descent: Richardson iteration

We analyze a slightly simpler algorithm with the same asymptotic convergence rate (we do not prove this), *first-order Richardson iteration*. Instead of doing a line search, it uses a fixed step length α:

$$x_{k+1} = x_k + \alpha r_k = x_k + \alpha(b - Ax_k) = (I - \alpha A)x_k + \alpha b. \tag{3.1.6}$$

If x_* is the actual solution and $e_k = x_k - x_*$, then $\alpha A x_* = \alpha b$, so

$$e_{k+1} = (I - \alpha A)x_k + \alpha b - x_* = (I - \alpha A)x_k + \alpha A x_* - x_* = (I - \alpha A)(x_k - x_*) = (I - \alpha A)e_k.$$

So

$$e_k = (I - \alpha A)^k e_0 \qquad \text{and} \qquad r_k = -Ae_k = (I - \alpha A)^k r_0. \tag{3.1.7}$$

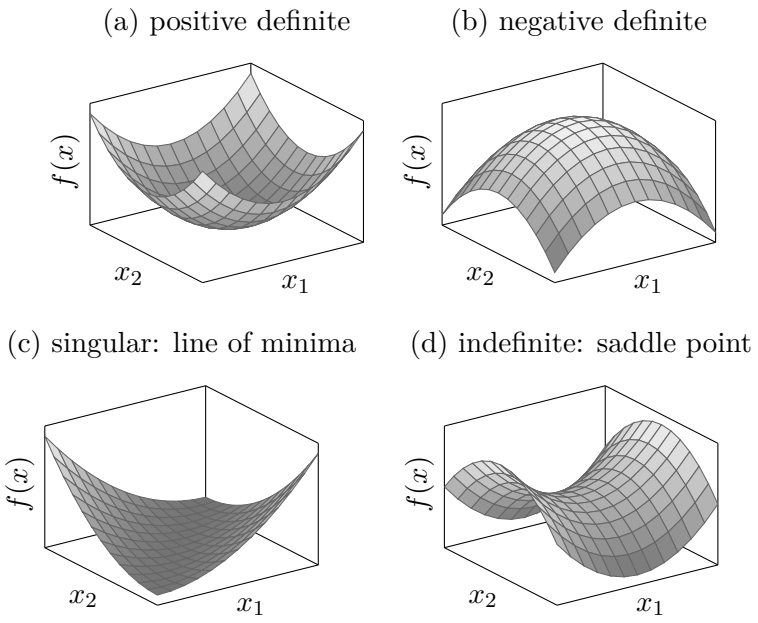


Figure 3.5: Quadratic forms of symmetric matrices that are (a) positive definite, (b) negative definite, (c) positive semidefinite and singular (Shewchuk [She94] calls this "positive-indefinite"), with a line of minima, and (d) indefinite, with a saddle point. The example functions are chosen for illustration.

If we use $x_0 = 0$, then $r_0 = b$, and

$$\|r_k\| \le \|(I - \alpha A)^k\| \, \|b\|.$$

If A has eigenvalues $\lambda_1, \ldots, \lambda_n \in [\lambda_{\min}, \lambda_{\max}]$, this gives

$$\|r_k\| \le \max_j |1 - \alpha\lambda_j|^k \, \|b\|. \tag{3.1.8}$$

This is best when $|1 - \alpha\lambda_{\min}| = |1 - \alpha\lambda_{\max}|$, that is, when $\alpha = 2/(\lambda_{\min} + \lambda_{\max})$, which gives

$$\max_j |1 - \alpha\lambda_j| = 1 - \frac{2\lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} = 1 - \frac{2}{\kappa + 1}, \qquad \kappa = \frac{\lambda_{\max}}{\lambda_{\min}}. \tag{3.1.9}$$

So the number of iterations needed grows linearly in the *condition number* κ. We have $(1 - \frac{2}{\kappa+1})^k \le e^{-2k/(\kappa+1)}$, so $k \ge \frac{\kappa+1}{2} \ln \frac{1}{\epsilon}$ iterations give $\|r_k\| \le \epsilon \|b\|$.

3.1.5 Why you already did this in calculus class

Nothing depends on the basis, so let our matrix be diagonal. For simplicity we show the 2×2 case, with eigenvalues λ_1, λ_2. With $x_0 = 0$, the algorithm (3.1.6) gives

$$x_k = \Big[\sum_{i=0}^{k-1} \alpha(I - \alpha A)^i\Big] b, \qquad r_k = (I - \alpha A)^k b$$

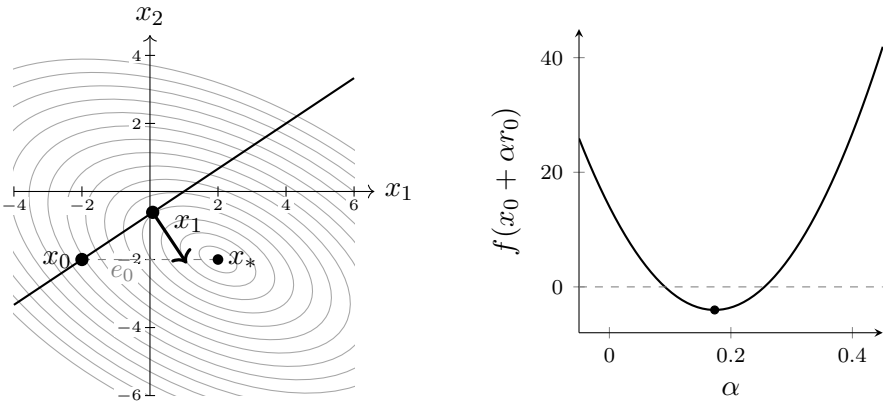


Figure 3.6: One step of steepest descent on (3.1.3) from $x_0 = (-2, -2)$. Left: the search line through x_0 in the direction of steepest descent $r_0 = (12, 8)$, and the error e_0. The line search stops at $x_1 \approx (0.08, -0.61)$, where the line is tangent to a contour, so the new gradient direction r_1 (arrow) is orthogonal to the old one. Right: f along the search line is a parabola in α, minimized at $\alpha_0 = 208/1200 \approx 0.173$. The parabola is where the surface of Fig. 3.2 meets the vertical plane through the search line.

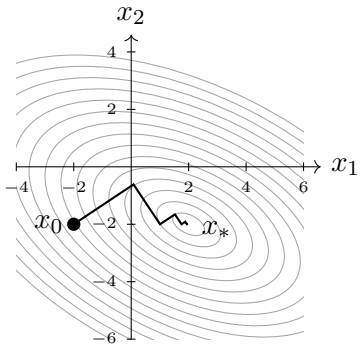


Figure 3.7: The method of steepest descent on (3.1.3): the iterates zigzag toward x_*, each step orthogonal to the previous one.

(unroll the recursion $x_{k+1} = (I - \alpha A)x_k + \alpha b$). Let

$$p_k^{(\alpha)}(y) = \sum_{i=0}^{k-1} \alpha(1-\alpha y)^i = \sum_{i=0}^{k-1} \alpha^{i+1}\Big(\frac{1}{\alpha} - y\Big)^i, \qquad \text{so that} \qquad x_k = p_k^{(\alpha)}(A)\, b. \tag{3.1.10}$$

In the diagonal case,

$$x_k = \begin{pmatrix} p_k^{(\alpha)}(\lambda_1) & 0 \\ 0 & p_k^{(\alpha)}(\lambda_2) \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} p_k^{(\alpha)}(\lambda_1) b_1 \\ p_k^{(\alpha)}(\lambda_2) b_2 \end{pmatrix}, \qquad r_k = \begin{pmatrix} (1-\alpha\lambda_1)^k b_1 \\ (1-\alpha\lambda_2)^k b_2 \end{pmatrix}.$$

So $p_k^{(\alpha)}$ is the first k terms of the Taylor series for $1/y$ at $1/\alpha$, and the entries of r_k are the error terms. To see this, note that $\frac{1}{y} = \frac{\alpha}{1-(1-\alpha y)} = \sum_{i\geq 0} \alpha(1-\alpha y)^i$ is the Taylor expansion of $1/y$ around $y = 1/\alpha$, valid for $|1 - \alpha y| < 1$. The exact solution has entries b_j/λ_j, and the truncation leaves $b_j/\lambda_j - p_k^{(\alpha)}(\lambda_j) b_j = (1-\alpha\lambda_j)^k b_j/\lambda_j$. Multiplying by λ_j gives the residual entries.

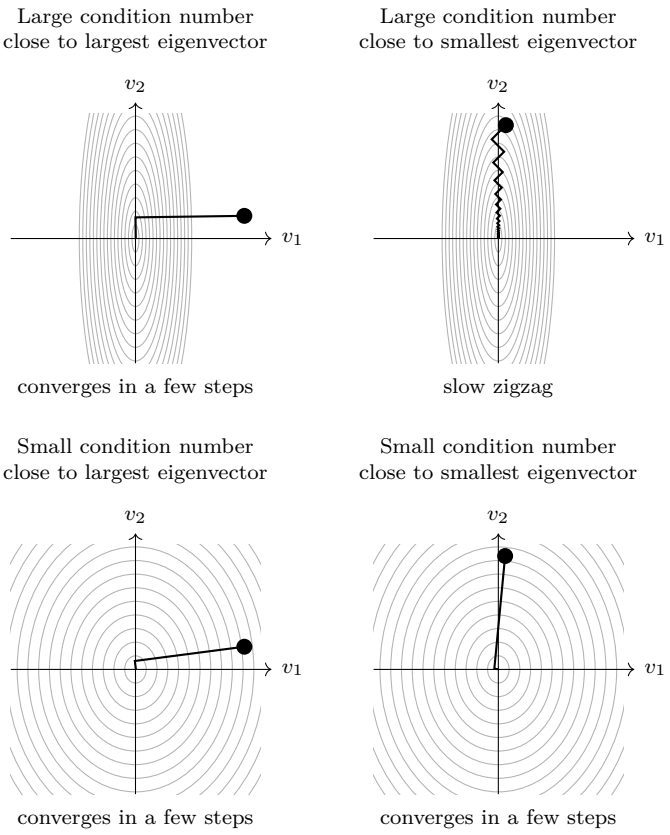


Figure 3.8: Running time is dictated by the condition number κ. Steepest descent for $f(x) = \frac{1}{2}x^\top Ax$ with A diagonal, drawn in the coordinates of the eigenvectors v_1 (largest eigenvalue) and v_2 (smallest eigenvalue); the dot is the starting point. Top row: large condition number (here $\kappa = 15$, very elongated contours). Bottom row: small condition number (here $\kappa = 1.6$). Left column: the starting error is close to the eigenvector of the largest eigenvalue; right column: close to the eigenvector of the smallest eigenvalue. Only the top right case zigzags for a long time. The values of κ and the starting points are chosen for illustration.

3.1.6 Better polynomials: Chebyshev iteration

Suppose all eigenvalues of A are in the interval $S = [\lambda_{\min}, \lambda_{\max}]$. We can replace the Taylor series with any polynomial:

$$x_k = p_k(A)\, b, \qquad x_* - x_k = x_* - p_k(A)b = x_* - p_k(A)Ax_* = q_k(A)\, x_*,$$

where $q_k(y) = 1 - y\, p_k(y)$. We want a polynomial q of degree k with $q(0) = 1$ that is small on S; equivalently, $p_k(y) \approx 1/y$ on S. We already found such polynomials (more or less) in Section 1.7.4: variants of Chebyshev polynomials, which can do much better than Taylor series. They give the error bound

$$\|e_k\|_A \le 2\Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^k \|e_0\|_A = 2\Big(1 - \frac{2}{\sqrt{\kappa}+1}\Big)^k \|e_0\|_A, \tag{3.1.11}$$

where $\|x\|_A = \sqrt{x^\top Ax}$. Compare Richardson iteration: with the step size of (3.1.9), (3.1.7) gives $\|e_k\|_A \le (1 - \frac{2}{\kappa+1})^k \|e_0\|_A$ in the same norm.

To get (3.1.11) from Section 1.7.6, start with $x_0 = 0$, so $e_0 = -x_*$ and $e_k = q_k(A)e_0$. Write $e_0 = \sum_j \xi_j v_j$ in an orthonormal eigenbasis of A. Then

$$\|q_k(A)e_0\|_A^2 = \sum_j \lambda_j q_k(\lambda_j)^2 \xi_j^2 \le \max_{\lambda \in S} q_k(\lambda)^2 \, \|e_0\|_A^2.$$

Now take

$$q_k(y) = \frac{T_k(\ell(y))}{T_k(\ell(0))}, \qquad \ell(y) = \frac{\lambda_{\max} + \lambda_{\min} - 2y}{\lambda_{\max} - \lambda_{\min}},$$

where T_k is the Chebyshev polynomial of degree k. Then $q_k(0) = 1$, and ℓ maps S onto $[-1, 1]$, where $|T_k| \le 1$. Also $\ell(0) = \frac{\kappa+1}{\kappa-1}$, so

$$\ell(0) + \sqrt{\ell(0)^2 - 1} = \frac{\kappa + 1 + 2\sqrt{\kappa}}{\kappa - 1} = \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}.$$

Since $T_k(x) = \frac{1}{2}\left[(x + \sqrt{x^2-1})^k + (x - \sqrt{x^2-1})^k\right]$, this gives $T_k(\ell(0)) \ge \frac{1}{2}\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^k$, and so $\max_S |q_k| \le 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k$.

So the number of iterations now scales with the square root of κ, which is way better than steepest descent. Is this the best we can do?

3.1.7 Krylov subspace methods

Let $f(x) = \frac{1}{2}x^\top Ax - b^\top x$ for A symmetric and positive definite, so that $\nabla f(x) = Ax - b = A(x - x_*)$, where $Ax_* = b$. Suppose we look at algorithms of the following form:

- start with $x_0 = 0$;
- set x_k to be some linear combination of b, x_i, and Ax_i for $i < k$.

Note that this includes $r_i = b - Ax_i$ and $\nabla f(x_i) = -r_i$, so it captures gradient descent and the Chebyshev algorithms. Such an algorithm produces x_k in the kth *Krylov subspace*

$$\mathcal{K}_k = \operatorname{span}(b, Ab, A^2b, \ldots, A^{k-1}b) = \{p(A)\, b : \deg(p) \le k - 1\}. \tag{3.1.12}$$

This follows by induction on k, with $\mathcal{K}_0 = \{0\} \ni x_0$. If $x_i \in \mathcal{K}_i \subseteq \mathcal{K}_{k-1}$ for all $i < k$, then $Ax_i \in \mathcal{K}_k$. Also $b \in \mathcal{K}_1 \subseteq \mathcal{K}_k$, so $x_k \in \mathcal{K}_k$.

Our goal will be to find the "best" algorithm of this form with respect to f, that is, we want

$$x_k = \operatorname*{arg\,min}_{x \in \mathcal{K}_k} f(x). \tag{3.1.13}$$

Looking for the best x in each Krylov subspace leads to the conjugate gradient algorithm, the subject of Section 3.2.

3.2 The Conjugate Gradient Method

The conjugate gradient method (CG) finds, in each Krylov subspace, the point that minimizes the quadratic objective f of Section 3.1. Given a basis of the Krylov subspace that is orthogonal in the inner product defined by A, the minimizer has an explicit formula that does not require knowing the solution. Such a basis can be built by Gram–Schmidt, and for Krylov subspaces

Gram–Schmidt collapses to a three-term recurrence, so each step is cheap. The resulting method is at least as good as any polynomial method, so its error after k steps is bounded by the Chebyshev bound, about $(1 - 2/\sqrt{\kappa})^k$. Notation is as in Section 3.1: A is symmetric positive definite, x_* solves $Ax = b$, and the error of the kth iterate is $e_k = x_k - x_*$.

3.2.1 Overview: a three-step plan

We have $f(x) = \frac{1}{2}x^\top Ax - b^\top x$ and $\mathcal{K}_k = \operatorname{span}(b, Ab, \dots, A^{k-1}b)$, and the best polynomial corresponds to $x_k = \arg\min_{x \in \mathcal{K}_k} f(x)$. The plan has three steps.

Step 1: rewrite the objective. The objective is, up to a positive factor and an additive constant, $\|x - x_*\|_A^2$, so

$$x_k = \operatorname*{arg\,min}_{x \in \mathcal{K}_k} \|x - x_*\|_A^2. \tag{3.2.1}$$

Here $\langle x, y \rangle_A \coloneqq x^\top Ay$ and $\|x\|_A^2 = \langle x, x \rangle_A = x^\top Ax$. Indeed,

$$\begin{aligned} \|x - x_*\|_A^2 &= (x - x_*)^\top A(x - x_*) \\ &= x^\top Ax - 2x_*^\top Ax + x_*^\top Ax_* \\ &= x^\top Ax - 2b^\top x + x_*^\top Ax_* = 2f(x) + x_*^\top Ax_*. \end{aligned}$$

Since we are working with the A-norm, it is nicest to use an A-orthogonal basis: vectors $d_0, \dots, d_{n-1}$ with

$$d_i \in \mathcal{K}_{i+1}, \qquad \langle d_i, d_j \rangle_A = 0 \ \text{ for all } i \neq j \quad \Longrightarrow \quad \mathcal{K}_k = \operatorname{span}(d_0, \dots, d_{k-1}).$$

This uses that the d_i are nonzero and that $\dim \mathcal{K}_k = k$. Then $d_0, \dots, d_{k-1}$ are k nonzero A-orthogonal vectors, so they are linearly independent and span the k-dimensional space $\mathcal{K}_k$.

Step 2: show how to find x_k if we are given an A-orthogonal basis.

Step 3: give an efficient way to construct an A-orthogonal basis.

Steps 2 and 3 are carried out in Sections 3.2.2 and 3.2.3.

3.2.2 Finding x_k given an A-orthogonal basis

We want x_k as in (3.2.1), and we have $d_0, \dots, d_{k-1} \in \mathcal{K}_k$ with $\langle d_i, d_j \rangle_A = 0$ for all $i \neq j$.

Motivation. Local optimality says that $\nabla f(x_k) \perp \mathcal{K}_k$, that is, for any $s \in \mathcal{K}_k$,

$$0 = \nabla f(x_k)^\top s = (Ax_k - b)^\top s = (x_k - x_*)^\top As = \langle x_k - x_*, s \rangle_A.$$

So $x_k - x_*$ is *A-orthogonal* to $\mathcal{K}_k$, and we can think of the iterative step as a projection: x_k is the A-orthogonal projection of x_* onto $\mathcal{K}_k$.

Consequently, if $x_* = \sum_{i=0}^{n-1} c_i d_i$, then

$$x_k = \sum_{i=0}^{k-1} c_i d_i. \tag{3.2.2}$$

Here $d_0, \dots, d_{n-1}$ is an A-orthogonal basis of $\mathbb{R}^n$. The difference $x_* - \sum_{i<k} c_i d_i = \sum_{i \geq k} c_i d_i$ is A-orthogonal to $d_0, \dots, d_{k-1}$, so to $\mathcal{K}_k$, and the A-orthogonal projection onto $\mathcal{K}_k$ is unique.

We cannot compute dot products with $x_k - x_*$, since we do not know x_*. But we can compute them with respect to the A-norm, since $Ax_* = b$. Computing $\langle \cdot, \cdot \rangle_A$ and $\|\cdot\|_A$ is simple in an A-orthogonal basis:

$$\Big\langle \sum_i \alpha_i d_i, \sum_j \beta_j d_j \Big\rangle_A = \sum_i \sum_j \alpha_i \beta_j \langle d_i, d_j \rangle_A = \sum_i \alpha_i \beta_i \|d_i\|_A^2, \qquad \Big\| \sum_i \alpha_i d_i \Big\|_A^2 = \sum_i \alpha_i^2 \|d_i\|_A^2.$$

If $x_* = \sum_{i=0}^{n-1} c_i d_i$, then $\langle d_i, x_* \rangle_A = c_i \|d_i\|_A^2$, so

$$c_i = \frac{\langle d_i, x_* \rangle_A}{\|d_i\|_A^2} = \frac{d_i^\top A x_*}{\|d_i\|_A^2} = \frac{d_i^\top b}{d_i^\top A d_i}. \tag{3.2.3}$$

The last expression involves only b, A and d_i, so (3.2.2) computes x_k without knowing x_*. It also shows that $x_{k+1} = x_k + c_k d_k$: each iteration adds one term.

3.2.3 Getting an A-orthogonal basis

We want an A-orthogonal basis $d_0, d_1, \dots$ with $\operatorname{span}(d_0, \dots, d_{k-1}) = \mathcal{K}_k$. The standard answer is Gram–Schmidt. In general, Gram–Schmidt is too slow, but we will choose really good vectors to orthogonalize, which will let us do it faster.

Set $d_0 = b$, and get d_{i+1} by taking Ad_i and subtracting its A-projection onto $\mathcal{K}_{i+1} = \operatorname{span}(d_0, \dots, d_i)$:

$$d_{i+1} = Ad_i - \sum_{j=0}^{i} \frac{\langle d_j, Ad_i \rangle_A}{\|d_j\|_A^2} d_j. \tag{3.2.4}$$

By induction, $d_{i+1} \in \mathcal{K}_{i+2}$, and it is A-orthogonal to $d_0, \dots, d_i$. As long as the Krylov subspaces keep growing ($\dim \mathcal{K}_{i+2} = i + 2$), Ad_i has a nonzero coefficient on the new power $A^{i+1}b$, and so does d_{i+1}. So $d_{i+1} \neq 0$, and $d_0, \dots, d_{i+1}$ span $\mathcal{K}_{i+2}$.

Claim 3.2.1. *$\langle d_j, Ad_i \rangle_A = 0$ for all $j < i - 1$.*

So only two terms of the sum survive:

$$d_{i+1} = Ad_i - \frac{\langle d_{i-1}, Ad_i \rangle_A}{\|d_{i-1}\|_A^2} d_{i-1} - \frac{\langle d_i, Ad_i \rangle_A}{\|d_i\|_A^2} d_i. \tag{3.2.5}$$

Proof of Claim 3.2.1. The vector d_i is A-orthogonal to everything in $\mathcal{K}_i = \operatorname{span}(d_0, \dots, d_{i-1})$. For $j < i - 1$ we have $d_j \in \mathcal{K}_{j+1}$, so $Ad_j \in \mathcal{K}_{j+2} \subseteq \mathcal{K}_i$, and therefore

$$\langle d_j, Ad_i \rangle_A = \langle Ad_j, d_i \rangle_A = 0.$$

The first equality uses that A is symmetric: $d_j^\top A(Ad_i) = (Ad_j)^\top Ad_i$. □

Remark 3.2.2. If we combine the terms with d_i in (3.2.5), this corresponds to the fact that orthogonal polynomials obey three-term recurrence relations. Write $d_i = \pi_i(A)\, b$ with $\deg \pi_i = i$. Then the polynomials π_i are orthogonal for the inner product $(p, q) \mapsto b^\top p(A)\, A\, q(A)\, b$, and (3.2.5) says $\pi_{i+1}(y) = (y - \beta_i)\pi_i(y) - \gamma_i \pi_{i-1}(y)$ for constants β_i, γ_i.

Remark 3.2.3 (Implementation note). This was derived assuming exact arithmetic. If you actually implement it, you should write the same quantities in a slightly different way, or you will run into issues with numerical stability.

Algorithm 3.2.4 (Conjugate gradient, as derived above)**.** Start with $x_0 = 0$ and $d_0 = b$. For $k = 0, 1, 2, \ldots$:

1. compute Ad_k and set $c_k = \dfrac{d_k^\top b}{d_k^\top A d_k}$ and $x_{k+1} = x_k + c_k d_k$;
2. set d_{k+1} by (3.2.5) (without the d_{k-1} term when $k = 0$).

Since $\langle d_j, Ad_k\rangle_A = (Ad_j)^\top (Ad_k)$ and $\|d_k\|_A^2 = d_k^\top (Ad_k)$, keeping the vectors Ad_{k-1} and Ad_k makes each iteration cost one multiplication by A plus $O(n)$ further work. This puts Steps 2 and 3 together. The usual form of CG orthogonalizes residuals instead of the vectors Ad_k (see the implementation note). In exact arithmetic both give the same iterates, since both compute the minimizer (3.2.1).

3.2.4 Convergence of conjugate gradient

We showed that after k iterations, conjugate gradient finds the point of minimal A-norm in

$$e_0 + \operatorname{span}(Ae_0, A^2 e_0, \ldots, A^k e_0).$$

With $x_0 = 0$ we have $e_0 = -x_*$ and $Ae_0 = -b$, so this set is $\{x - x_* : x \in \mathcal{K}_k\}$, and minimizing $\|e\|_A$ over it is (3.2.1). So

$$e_k = \Big(I + \sum_{j=1}^{k} \psi_j A^j\Big) e_0,$$

and we know the algorithm finds the coefficients ψ_j that minimize the A-norm. Let $P_k(\lambda) = 1 + \sum_{j=1}^k \psi_j \lambda^j$ be this degree-k polynomial (with constant term 1), so that $e_k = P_k(A)e_0$. Writing $\Lambda(A)$ for the set of eigenvalues of A, we get

$$\|e_k\|_A^2 \le \min_{P_k} \max_{\lambda \in \Lambda(A)} [P_k(\lambda)]^2 \, \|e_0\|_A^2, \tag{3.2.6}$$

where the minimum is over polynomials P_k of degree k with $P_k(0) = 1$. To see this, expand $e_0 = \sum_j \xi_j v_j$ in an orthonormal eigenbasis of A. For any such P, $\|P(A)e_0\|_A^2 = \sum_j \lambda_j P(\lambda_j)^2 \xi_j^2 \le \max_{\lambda\in\Lambda(A)} P(\lambda)^2 \|e_0\|_A^2$, and CG does at least as well as every such P.

So if we can construct *any* polynomial P of degree k with $P(0) = 1$ that is small on all the eigenvalues of A, we have proven a bound on our error. How good a polynomial can we find? We can do at least as well as Chebyshev, so by (3.1.11),

$$\|e_k\|_A \le 2\Big(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\Big)^k \|e_0\|_A. \tag{3.2.7}$$

Figure 3.9 illustrates a general consequence of (3.2.6). If A has only s distinct eigenvalues $\lambda_1, \ldots, \lambda_s$, then $P(\lambda) = \prod_{j=1}^s (1 - \lambda/\lambda_j)$ has $P(0) = 1$ and vanishes on $\Lambda(A)$. So $e_s = 0$, and CG terminates after s steps (in exact arithmetic).

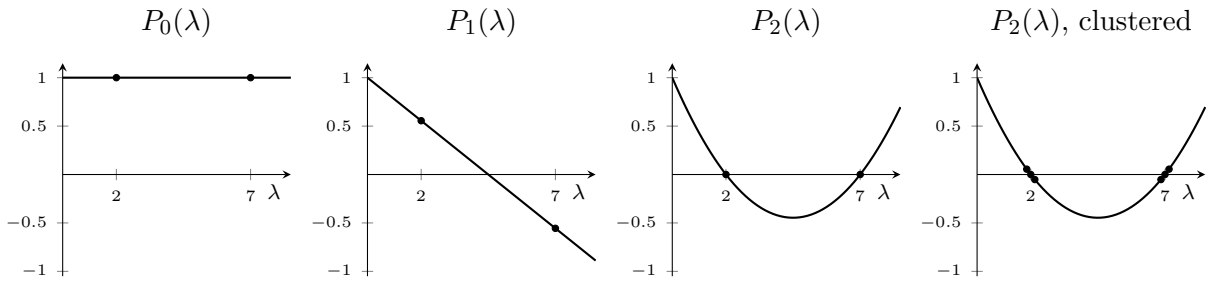


Figure 3.9: The convergence of CG after k iterations depends on how close a polynomial P_k of degree k can be to zero on each eigenvalue, given the constraint $P_k(0) = 1$. In the first three panels A has eigenvalues 2 and 7 (the example (3.1.3)). The constant P_0 is 1 on both; the line shown for P_1 is the one that makes $\max(|P_1(2)|, |P_1(7)|)$ smallest; and $P_2(\lambda) = (1 - \lambda/2)(1 - \lambda/7)$ vanishes on both eigenvalues, so CG finds the exact solution in two steps. The last panel shows the same P_2 with three eigenvalues close to each of 2 and 7 (at 1.85, 2, 2.15 and 6.85, 7, 7.15): with clustered eigenvalues a low-degree polynomial is already small on all of them.

End result

Define the condition number $\kappa = \lambda_{\max}/\lambda_{\min}$. The running time of CG is bounded by $\sqrt{\kappa}$: by (3.2.7), $O(\sqrt{\kappa}\log(1/\epsilon))$ iterations reduce the A-norm of the error by a factor ϵ.

- It can sometimes be better, for example when the eigenvalues are clustered.
- But our analysis is tight in the worst case (and, in fact, in most cases).

CG tends to be very slow when the matrix is *ill-conditioned*.

3.3 Preconditioning

By Section 3.2.4, conjugate gradient needs about $\sqrt{\kappa}$ iterations, so it is slow when A is ill-conditioned. When κ is large, one solves a *preconditioned* system instead: an equivalent system with a much smaller condition number. This section shows how to do this while keeping the system symmetric and positive definite, so that conjugate gradient still applies, and where preconditioners come from.

3.3.1 Preconditioned systems

What do we do if κ is really big? It is not totally hopeless. For example, take

$$A = \begin{pmatrix} 10000 & 777 & 123 \\ 0.1 & 1 & 0.2 \\ 0.002 & 0.001 & 0.01 \end{pmatrix}.$$

Here $\kappa \approx 1{,}000{,}000$, and this gets squared when you compute $A^\top A$. Since this A is not symmetric, κ here is the ratio of its largest to smallest singular value. Numerically, $\kappa(A) \approx 1.04 \times 10^6$ and

$\kappa(A^\top A) \approx 1.09 \times 10^{12}$. However, with D the diagonal part of A, $D = \mathrm{diag}(10000, 1, 0.01)$,

$$D^{-1}A = \begin{pmatrix} 1 & 0.0777 & 0.0123 \\ 0.1 & 1 & 0.2 \\ 0.2 & 0.1 & 1 \end{pmatrix}$$

is quite well-conditioned ($\kappa(D^{-1}A) \approx 1.45$). So we could just solve $D^{-1}Ax = D^{-1}b$ instead, which is much nicer.

More generally, suppose we want to solve $Ax = b$ where A is not well-conditioned (that is, κ is big). Suppose we know some positive definite M that approximates A but is easy to invert. We can instead solve

$$M^{-1}Ax = M^{-1}b,$$

which ought to have a much better condition number. There are complications:

- *Where do we get M?* This is a hard question, with no general answer.
- *It is expensive to compute $M^{-1}A$.* Don't! An iterative method only needs products $M^{-1}(Av)$, which cost one multiplication by A and one solve with M.
- *$M^{-1}A$ may not be symmetric or positive definite.* See Section 3.3.1.

Making the system symmetric and positive definite

We can use a trick similar to the one we used to make adjacency matrices symmetric (passing from $W = AD^{-1}$ to $D^{-1/2}AD^{-1/2}$ in Section 1.5.3). We can factor $M = EE^\top$, because M is positive definite (for example $E = M^{1/2}$, or a Cholesky factor). Then $M^{-1}A$ and $E^{-1}AE^{-\top}$ have the same eigenvalues: v is an eigenvector of $M^{-1}A$ if and only if $E^\top v$ is an eigenvector of $E^{-1}AE^{-\top}$. If $M^{-1}Av = \lambda v$, then $E^{-\top}E^{-1}Av = \lambda v$, so $E^{-1}AE^{-\top}(E^\top v) = \lambda E^\top v$, and the steps reverse. Also, $E^{-1}AE^{-\top}$ is symmetric and positive definite.

So we can solve

$$E^{-1}AE^{-\top}\hat{x} = E^{-1}b, \qquad \text{with } \hat{x} = E^\top x.$$

Use conjugate gradient to find $\hat{x}$, then compute x.

New problem: it might be hard to compute the factorization $M = EE^\top$. This is not a big deal: if you look at the formulas for conjugate gradient, the actual iterations only use M^{-1}, not E. The matrix E is only used when computing inner products, which give

$$(\text{something} \cdot E^{-\top})(E^{-1} \cdot \text{something}) = \text{something} \cdot M^{-1} \cdot \text{something}.$$

We can check this directly for the version of CG in Algorithm 3.2.4. Run it on $\hat{A} = E^{-1}AE^{-\top}$ and $\hat{b} = E^{-1}b$, and write each direction as $\hat{d}_i = E^\top u_i$. Then $u_0 = E^{-\top}E^{-1}b = M^{-1}b$. Since $\hat{A}\hat{d}_i = E^{-1}Au_i$, the step $\hat{d}_{i+1} = \hat{A}\hat{d}_i - (\cdots)$ becomes $u_{i+1} = M^{-1}Au_i - (\cdots)$. The coefficients are $\langle \hat{d}_j, \hat{A}\hat{d}_i \rangle_{\hat{A}} = (Au_j)^\top M^{-1}(Au_i)$, $\|\hat{d}_i\|^2_{\hat{A}} = u_i^\top Au_i$ and $\hat{d}_i^\top \hat{b} = u_i^\top b$. Finally $x = E^{-\top}\hat{x} = \sum_i c_i u_i$. Only A and M^{-1} appear.

Where do preconditioners come from?

Designing preconditioners is more art than science. There are different preconditioners for different classes of matrices. A very useful case is Laplacians of graphs, and here the spectral graph theory of Chapter 1 already gives us some preconditioners. For example, let H be an ϵ-spectral sparsifier of G (Section 1.9.4), so that $(1-\epsilon)L_G \preceq L_H \preceq (1+\epsilon)L_G$. Then, on the space orthogonal to $\mathbf{1}$, the preconditioned matrix with $M = L_H$ has condition number at most $(1+\epsilon)/(1-\epsilon)$.

3.4 Gradient Descent

We now move from linear systems to general convex optimization. This section asks what can be proved about gradient descent without the special structure of a quadratic, and which properties of the function control the number of iterations. It introduces the standard assumptions (Lipschitz continuity, smoothness, strong convexity) and analyzes gradient descent for smooth functions, with and without strong convexity.

Notation. In this section and the rest of the chapter, L denotes a Lipschitz or smoothness constant, not a graph Laplacian, and μ denotes a strong-convexity parameter, not an eigenvalue of a random-walk matrix.

3.4.1 Minimizing more general convex functions

Recall what we did for linear systems in Sections 3.1 and 3.2. To solve $Ax = b$ with A symmetric positive definite, we minimized

$$f(x) = \tfrac{1}{2}x^\top Ax - b^\top x.$$

The running time depended on the condition number $\kappa(A) = \lambda_{\max}(A)/\lambda_{\min}(A)$.

- *Gradient descent* needs $O(\kappa \log \frac{1}{\epsilon})$ iterations to multiply the error by ϵ.
- Better algorithms (Chebyshev iteration, conjugate gradient) improve this to $O(\sqrt{\kappa} \log \frac{1}{\epsilon})$.
- These analyses relied heavily on the precise form of f (its eigenvalues, and so on).

Gradient descent, however, makes sense for any differentiable convex function. This suggests several natural questions.

- Can we give a general analysis of gradient descent?
- If so, what properties of f determine the number of iterations needed? For instance, is there something corresponding to the condition number for a non-quadratic f?
- Is there a better algorithm, with a speedup analogous to the $\kappa \to \sqrt{\kappa}$ improvement that Chebyshev iteration and conjugate gradient gave?
- Not all convex functions are differentiable. Can we do gradient descent without a gradient?

In this section we focus on gradient descent for unconstrained optimization and see how different assumptions about f affect the convergence bounds. The faster algorithm is the subject of Section 3.8; the last question is taken up in Section 3.5.

3.4.2 Gradient descent

Setup. We are given a convex function $f : \mathbb{R}^n \to \mathbb{R}$ and want to solve

$$\min_{x \in \mathbb{R}^n} f(x). \tag{3.4.1}$$

We need to say what we mean by "given" and "solve". For now we assume that f is differentiable, and that f is given through an *oracle* that, for any point x, tells us $f(x)$ and $\nabla f(x)$. We can rarely solve (3.4.1) exactly, so we look for an almost optimal x. Let

$$f_* = \min_{x \in \mathbb{R}^n} f(x), \qquad X_* = \{x : f(x) = f_*\}.$$

We will study two different notions of "almost optimal":

- **function value near the optimal value:** $f(x) - f_* < \epsilon$;
- **point near the set of optima:** $\operatorname{dist}(x, X_*) < \epsilon$.

Algorithm 3.4.1 (Gradient descent)**.** Start at some initial point x_0. For a sequence of step lengths $\eta_k \in \mathbb{R}$, set

$$x_{k+1} = x_k - \eta_k \nabla f(x_k), \qquad k = 0, 1, 2, \ldots \tag{3.4.2}$$

It will be useful to generalize this to arbitrary norms; standard gradient descent corresponds to the ℓ_2 norm. We come back to this in Section 3.6.

3.4.3 Assumptions

For later use we state the assumptions with respect to an arbitrary norm $\|\cdot\|$. In that case the gradient should be measured with the dual norm $\|\cdot\|_*$ (defined in (3.6.2)). For now, think of the case $\|\cdot\| = \|\cdot\|_* = \|\cdot\|_2$.

Definition 3.4.2 (Convexity, Lipschitz continuity, smoothness, strong convexity)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$.

- **Convexity:** f is convex if for all x, y and all $t \in [0, 1]$,

$$f((1-t)x + ty) \le (1-t)f(x) + tf(y).$$

- **Lipschitz:** for $L \ge 0$, f is *L-Lipschitz* (with respect to $\|\cdot\|$) if

$$|f(x) - f(y)| \le L\|x - y\| \qquad \forall x, y \in \mathbb{R}^n.$$

- **Smoothness:** for $L \ge 0$, f is *L-smooth* (with respect to $\|\cdot\|$) if ∇f is L-Lipschitz:

$$\|\nabla f(x) - \nabla f(y)\|_* \le L\|x - y\| \qquad \forall x, y \in \mathbb{R}^n. \tag{3.4.3}$$

- **Strong convexity:** for $\mu \ge 0$, f is *μ-strongly convex* (with respect to $\|\cdot\|$) if for all x, y and all $t \in [0, 1]$,

$$f((1-t)x + ty) \le (1-t)f(x) + tf(y) - \frac{\mu}{2} t(1-t)\|x - y\|^2.$$

The following consequences are left as exercises. We use them freely below.

Exercise 3.4.3. Let f be differentiable. Show that for all x, y:

(a) if f is convex, then $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$;

(b) if f is L-smooth, then

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2; \tag{3.4.4}$$

(c) if f is μ-strongly convex, then

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2; \tag{3.4.5}$$

(d) if f is μ-strongly convex with $\mu > 0$, then f has a unique minimizer x_*, and

$$\frac{\mu}{2}\|x - x_*\|^2 \leq f(x) - f_* \leq \frac{1}{2\mu}\|\nabla f(x)\|_2^2. \tag{3.4.6}$$

3.4.4 Convergence of gradient descent

We use the constant step length $\eta_k = 1/L$ and prove two bounds.

Claim 3.4.4. *If f is L-smooth, then gradient descent with step length $\eta_k = \frac{1}{L}$ for all k gives x_k such that*

$$f(x_k) - f_* \leq O\Big(\frac{LR^2}{k}\Big), \qquad \text{where } R = \max_{x\,:\,f(x)\leq f(x_0)} \min_{x_* \in X_*} \|x - x_*\|.$$

In words, R is the largest distance from a point of the sublevel set $\{x : f(x) \leq f(x_0)\}$ to the set of minimizers.

Claim 3.4.5. *If f is L-smooth and μ-strongly convex, then gradient descent with step length $\eta_k = \frac{1}{L}$ for all k gives x_k such that*

$$f(x_k) - f_* \leq \Big(1 - \frac{\mu}{L}\Big)^k (f(x_0) - f_*).$$

Since $(1 - \frac{\mu}{L})^k \leq e^{-k\mu/L}$, in the strongly convex case $O(\frac{L}{\mu}\log\frac{1}{\epsilon})$ iterations reduce the initial error by a factor ϵ. So the ratio L/μ plays the role of the condition number. (From Section 3.5.3 on we write $\kappa = L/\mu$.)

Both proofs show two things:

- **Primal progress:** $f(x)$ decreases a lot in each iteration.
- **Dual bound:** a lower bound on f_* that lets us show that the primal progress is a nonnegligible fraction of $f(x_k) - f_*$.

Where did the step length come from? Intuitively, L-smoothness says $\|\nabla f(x)-\nabla f(y)\|_* \le L\|x-y\|$, so the function is still improving in the gradient direction as long as $\eta \le 1/L$. The choice $\eta = 1/L$ gives the best value in the smoothness bound (3.4.4): the right-hand side of (3.4.4) is minimized over y at $y = x - \frac{1}{L}\nabla f(x)$ (in ℓ_2), and there it gives the following bound.

Lemma 3.4.6 (Gradient step)**.** *If f is L-smooth with respect to $\|\cdot\|_2$ and $y = x - \frac{1}{L}\nabla f(x)$, then*

$$f(y) \le f(x) - \frac{1}{2L}\|\nabla f(x)\|_2^2. \tag{3.4.7}$$

Proof. Substitute $y - x = -\frac{1}{L}\nabla f(x)$ into (3.4.4): $f(y) \le f(x) - \frac{1}{L}\|\nabla f(x)\|_2^2 + \frac{L}{2}\cdot\frac{1}{L^2}\|\nabla f(x)\|_2^2 = f(x) - \frac{1}{2L}\|\nabla f(x)\|_2^2$. □

The strongly convex case is a little easier, so we do it first.

Smooth and strongly convex functions

Proof of Claim 3.4.5. Primal progress. By Lemma 3.4.6 with $x = x_k$,

$$f(x_{k+1}) \le f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|_2^2,$$

that is,

$$(f(x_k) - f_*) - (f(x_{k+1}) - f_*) = f(x_k) - f(x_{k+1}) \ge \frac{1}{2L}\|\nabla f(x_k)\|_2^2. \tag{3.4.8}$$

Dual bound. By (3.4.6),

$$f(x_k) - f_* \le \frac{1}{2\mu}\|\nabla f(x_k)\|_2^2. \tag{3.4.9}$$

Combining (3.4.8) and (3.4.9) gives

$$(f(x_k) - f_*) - (f(x_{k+1}) - f_*) \ge \frac{\mu}{L}(f(x_k) - f_*),$$

that is, $f(x_{k+1}) - f_* \le (1 - \frac{\mu}{L})(f(x_k) - f_*)$. Iterating this k times gives the claim. □

Smooth functions without strong convexity

Proof of Claim 3.4.4. Let $\epsilon_k := f(x_k) - f_*$.

Primal progress. Same as before: by Lemma 3.4.6,

$$\epsilon_k - \epsilon_{k+1} = (f(x_k) - f_*) - (f(x_{k+1}) - f_*) \ge \frac{1}{2L}\|\nabla f(x_k)\|_2^2. \tag{3.4.10}$$

In particular $f(x_{k+1}) \le f(x_k)$ for every k, so every iterate lies in the sublevel set $\{x : f(x) \le f(x_0)\}$, and $\min_{x_* \in X_*}\|x_k - x_*\| \le R$.

Dual bound. By convexity (Exercise 3.4.3(a) with $x = x_k$ and $y = x_*$) and the Cauchy–Schwarz inequality,

$$\epsilon_k = f(x_k) - f_* \le \min_{x_* \in X_*}\langle \nabla f(x_k), x_k - x_*\rangle \le \|\nabla f(x_k)\|_2 \min_{x_* \in X_*}\|x_k - x_*\|_2 \le \|\nabla f(x_k)\|_2 R. \tag{3.4.11}$$

Combining (3.4.10) and (3.4.11) gives

$$\epsilon_k - \epsilon_{k+1} \geq \frac{\epsilon_k^2}{2LR^2}. \tag{3.4.12}$$

An intuitive convergence bound, with slightly weak constants. Suppose the current error is δ. As long as the error is at least $\delta/2$, (3.4.12) says that each step decreases the error by at least $\frac{\delta^2}{8LR^2}$, so it takes at most $\frac{4LR^2}{\delta}$ steps to get the error down to $\delta/2$. To get error ϵ (which, for simplicity of notation, we suppose is of the form $\epsilon_0/2^t$ for some t), we therefore need

$$\#\text{steps} \leq 4LR^2\Big(\frac{1}{\epsilon_0} + \frac{1}{\epsilon_0/2} + \frac{1}{\epsilon_0/4} + \cdots + \frac{1}{\epsilon}\Big) = \frac{4LR^2}{\epsilon}\Big(\frac{1}{2^t} + \frac{1}{2^{t-1}} + \cdots + 1\Big) \leq \frac{8LR^2}{\epsilon}.$$

A slightly slicker bound with better constants. If some $\epsilon_{k+1} = 0$ we are done, so assume all $\epsilon_k > 0$. Dividing (3.4.12) by $\epsilon_k\epsilon_{k+1}$ and using $\epsilon_{k+1} \leq \epsilon_k$,

$$\frac{1}{\epsilon_{k+1}} - \frac{1}{\epsilon_k} = \frac{\epsilon_k - \epsilon_{k+1}}{\epsilon_k\epsilon_{k+1}} \geq \frac{\epsilon_k}{2LR^2\epsilon_{k+1}} \geq \frac{1}{2LR^2}.$$

Since $\epsilon_0 \leq \frac{LR^2}{2}$,[1] summing gives

$$\frac{1}{\epsilon_k} \geq \frac{1}{\epsilon_0} + \frac{k}{2LR^2} \geq \frac{k+4}{2LR^2} \qquad \Longrightarrow \qquad \epsilon_k \leq \frac{2LR^2}{k+4}. \qquad \square$$

3.5 Subgradient Descent

Not all convex functions are differentiable. This section replaces the gradient by a subgradient, analyzes the resulting subgradient descent for Lipschitz convex functions, compares the three convergence rates obtained so far, and sketches how matching lower bounds are proved.

3.5.1 Subgradients and the algorithm

What if f is convex (and, say, L-Lipschitz) but not differentiable? The main property of the gradient g that we used is that $f(y) \geq f(x) + \langle g, y - x\rangle$ for all y. So we simply define a *subgradient* to be a vector g with this property.

Definition 3.5.1 (Subgradient)**.** A vector g is a *subgradient* of f at x, written $g \in \partial f(x)$, if

$$f(y) \geq f(x) + \langle g, y - x\rangle \qquad \forall y. \tag{3.5.1}$$

Algorithm 3.5.2 (Subgradient descent)**.** Use any subgradient in place of the gradient:

$$x_{k+1} = x_k - \eta g_k, \qquad g_k \in \partial f(x_k). \tag{3.5.2}$$

This creates new difficulties.

- The (sub)gradient could change very quickly, and give a good approximation of f only in a very small neighborhood.

[1] Take $x_* \in X_*$ closest to x_0. Since $\nabla f(x_*) = 0$, (3.4.4) with $x = x_*$ and $y = x_0$ gives $\epsilon_0 \leq \frac{L}{2}\|x_0 - x_*\|^2 \leq \frac{LR^2}{2}$.

- If we fix η in advance (as opposed to, say, choosing it by a line search), we do not even know that $f(x_{k+1}) < f(x_k)$.
- So we need to change the algorithm slightly: return the best point encountered. That is, return x_i where $i = \arg\min_j f(x_j)$, and write $f_{\text{best}} = f(x_i)$ for the value at this point.
- But if f does not always improve, how should we keep track of progress?

The answer, and the analysis of subgradient descent, are in Section 3.5.2.

Remark 3.5.3 (Where the worse rate comes from). We will need to be more careful with step sizes, and we will get a worse convergence rate in this case. People sometimes attribute this to f not being differentiable. We think this is the wrong viewpoint.

- One can modify f infinitesimally to make it differentiable as many times as one wants, and then subgradients are just gradients at (arbitrarily) nearby points.
- So the convergence should be the same as for gradient descent on this smoothed f.
- The worse behavior comes from assuming nothing about how quickly the gradients can change: we assume that f is Lipschitz, but not that it is smooth (that is, that its gradient is Lipschitz).

3.5.2 Analysis of subgradient descent

Claim 3.5.4. *If f is L-Lipschitz, then running subgradient descent with step length $\eta_k = \eta = \frac{R}{L\sqrt{t}}$ for all k, for t steps, returns x such that*

$$f(x) - f_* = f_{\text{best}} - f_* \le \frac{RL}{\sqrt{t}}, \qquad \text{where } R = \min_{x_* \in X_*} \|x_0 - x_*\|.$$

Note that the definitions of R and L differ from the smooth case (Claim 3.4.4): here L bounds the function's rate of change rather than its gradient's, and R is the distance from the starting point to the set of minimizers rather than a maximum over a sublevel set.

Idea. For notational simplicity we assume that there is a unique optimum x_*. If there is a larger set X_* of optima, the analysis goes through essentially unchanged if we take x_* to be the closest point of X_* at each step. We track the distance $\|x_k - x_*\|$ instead of the function value. The subgradient condition (3.5.1) with $y = x_*$ says

$$f(x_k) - f_* \le \langle g_k, x_k - x_* \rangle. \tag{3.5.3}$$

So if $f(x_k) > f_*$, we at least know that an infinitesimal step in the direction of $-g_k$ decreases the distance to x_*. And since $\|g_k\| \le L$, a large $f(x_k) - f_*$ forces a large inner product in (3.5.3), which should (roughly) mean large progress in distance.

Proof of Claim 3.5.4. More precisely, by (3.5.3),

$$\begin{aligned}
\|x_{k+1} - x_*\|^2 &= \|x_k - \eta g_k - x_*\|^2 = \|(x_k - x_*) - \eta g_k\|^2 \\
&= \|x_k - x_*\|^2 - 2\eta\langle g_k, x_k - x_* \rangle + \eta^2 \|g_k\|^2 \\
&\le \|x_k - x_*\|^2 - 2\eta(f(x_k) - f_*) + \eta^2 \|g_k\|^2.
\end{aligned}$$

We can rewrite this as

$$\|x_k - x_*\|^2 - \|x_{k+1} - x_*\|^2 \geq 2\eta(f(x_k) - f_*) - \eta^2\|g_k\|^2. \tag{3.5.4}$$

To first order in η, the progress in $\|x_k - x_*\|^2$ is proportional to $f(x_k) - f_*$; the second-order term $\eta^2\|g_k\|^2$ penalizes us if we take too large a step. So as long as η is not too big, there cannot be too many steps where f is far from f_*. (A good way to get intuition is to pretend that we know f_* and to think of projecting onto the corresponding separating hyperplane for X_*.) Summing (3.5.4) over $k = 0, \ldots, t-1$, and using $f(x_k) \geq f_{\text{best}}$ and $\|g_k\| \leq L$ (Remark 3.5.5),

$$\begin{aligned}
\|x_0 - x_*\|^2 - \|x_t - x_*\|^2 &\geq \sum_{k=0}^{t-1} \Big(2\eta(f(x_k) - f_*) - \eta^2\|g_k\|^2\Big) \\
&= 2\eta \sum_{k=0}^{t-1} (f(x_k) - f_*) - \eta^2 \sum_{k=0}^{t-1} \|g_k\|^2 \\
&\geq 2\eta t(f_{\text{best}} - f_*) - \eta^2 t L^2.
\end{aligned}$$

Rearranging, and using $\|x_t - x_*\|^2 \geq 0$, gives

$$f_{\text{best}} - f_* \leq \frac{\|x_0 - x_*\|^2 + \eta^2 t L^2}{2\eta t} = \frac{\|x_0 - x_*\|^2}{2\eta t} + \frac{\eta L^2}{2} \leq \frac{1}{\eta}\frac{R^2}{2t} + \eta\frac{L^2}{2}.$$

Choosing $\eta = \frac{R}{L\sqrt{t}}$ balances the two terms and gives $f_{\text{best}} - f_* \leq LR/\sqrt{t}$, as claimed. □

Remark 3.5.5 (Lipschitz constants of convex functions). The bound $\|g\| \leq L$ for $g \in \partial f(x)$ follows from the Lipschitz condition: if $g \neq 0$, take $y = x + g/\|g\|$ in (3.5.1); then $\|g\| = \langle g, y - x\rangle \leq f(y) - f(x) \leq L\|y - x\| = L$ (in ℓ_2).

3.5.3 Interpreting the bounds, and lower bounds

If we want $f(x) - f_* \leq \epsilon$, how does the number of steps grow? Treating R (and, in the last case, the initial error) as fixed, the bounds of Claims 3.4.4, 3.4.5 and 3.5.4 give:

- **L-Lipschitz (non-smooth):** $\#\text{steps} \propto \dfrac{L^2}{\epsilon^2}$;
- **L-smooth (not strongly convex):** $\#\text{steps} \propto \dfrac{L}{\epsilon}$;
- **L-smooth, μ-strongly convex:** $\#\text{steps} \propto \dfrac{L}{\mu}\log\dfrac{1}{\epsilon} =: \kappa \log\dfrac{1}{\epsilon}$.

Idea for how to prove lower bounds.

- Suppose we have any algorithm such that the kth iterate x_k lies in the span of x_0 and the subgradients $g_0, \ldots, g_{k-1}$ returned by the oracle in the previous iterations.
- Set up examples in which you can control this span. The standard examples make $x_k \in \text{span}(\chi_1, \ldots, \chi_k)$, that is, one new coordinate per iteration.
- Show that any solution supported on this span has large f.

Example 3.5.6 (The Lipschitz bound is asymptotically tight)**.** The function

$$f(x) = \frac{\alpha}{2}\|x\|_2^2 + \beta \max_{i \le t} x_i, \qquad \alpha = \frac{L}{R}\frac{1}{1+\sqrt{t}}, \quad \beta = L\frac{\sqrt{t}}{1+\sqrt{t}},$$

shows that the L-Lipschitz bound of Claim 3.5.4 is asymptotically tight. *(Proof omitted.)* The example is used on the ball $\|x\|_2 \le R$, with $x_0 = 0$; f is L-Lipschitz on this ball but not on all of $\mathbb{R}^n$. Its minimizer has $x_*(i) = -R/\sqrt{t}$ for $i \le t$, and $f_* = -\frac{RL}{2(1+\sqrt{t})}$. Every point supported on the first $k < t$ coordinates has $\max_{i \le t} x_i \ge 0$, so $f(x) - f_* \ge \frac{RL}{2(1+\sqrt{t})}$. The upper bound $\frac{RL}{\sqrt{t}}$ of Claim 3.5.4 is larger only by the factor $2(1 + 1/\sqrt{t})$, which tends to 2.

One can also construct examples based on the Laplacian of the path graph (we omit the details) that give

- L-smooth functions that need $\Theta(\sqrt{L/\epsilon})$ iterations;
- L-smooth, μ-strongly convex functions that need $\Theta(\sqrt{\kappa}\log\frac{1}{\epsilon})$ iterations.

There is a gap between these lower bounds and the upper bounds above for smooth functions. Section 3.8 shows that the algorithms can be improved to match these lower bounds.

3.6 Descent in General Norms

Gradient descent, as defined in Algorithm 3.4.1, is tied to the ℓ_2 norm. This section generalizes it to an arbitrary norm, which changes the direction of the step, and checks that the analyses of Section 3.4.4 go through when the gradient is measured in the dual norm.

3.6.1 Steepest descent in an arbitrary norm

Goal. Minimize a convex function $f : \mathbb{R}^n \to \mathbb{R}$. Start at some x_0 and repeat $x_{k+1} = x_k + \delta_k$, where δ_k is some step that decreases f.

Standard gradient descent.

- The gradient $\nabla f(x)$ is the linear term when we approximate f locally around x:

$$f(y) = f(x) + \langle \nabla f(x), y - x\rangle + o(\|y - x\|).$$

- Look in a ball $B = \{y : \|y - x\| \le r\}$ around x, with r small enough that the linear approximation is pretty good.
- Find the point of B that minimizes the linear approximation:

$$\arg\min_{y\,:\,\|x-y\|\le r} \; f(x) + \langle \nabla f(x), y - x\rangle. \tag{3.6.1}$$

- In ℓ_2, this ends up being in the direction of $-\nabla f(x)$.

Gradient descent for general norms.

- Same thing, different norm: step in the direction of the minimizer in (3.6.1), where now $\|\cdot\|$ is an arbitrary norm.
- Note that the step is no longer necessarily in the direction of $-\nabla f$.
- The *dual norm* of $\|\cdot\|$ is
$$\|z\|_* = \max_{\|y\|\le 1} \langle z, y\rangle. \tag{3.6.2}$$
- The Lipschitz constant of the gradient is the smallest L such that
$$\|\nabla f(x) - \nabla f(y)\|_* \le L\|x - y\| \qquad \forall x, y.$$
- The update is
$$x_{k+1} = \arg\min_y \; f(x_k) + \langle \nabla f(x_k), y - x_k\rangle + \frac{L}{2}\|y - x_k\|^2. \tag{3.6.3}$$

Figure 3.10 compares the steps for the ℓ_2 norm,
$$x_{k+1} = \arg\min_{x\,:\,\|x - x_k\|_2 < r} \; f(x_k) + \langle \nabla f(x_k), x - x_k\rangle,$$
and for the ℓ_∞ norm, where the constraint is $\|x - x_k\|_\infty < r$. With ℓ_2 balls each step moves perpendicular to the level set. With ℓ_∞ balls (boxes) each step moves to a corner of the box: each coordinate changes by $\pm r$, following the sign of that coordinate of $-\nabla f(x_k)$ (when it is nonzero).

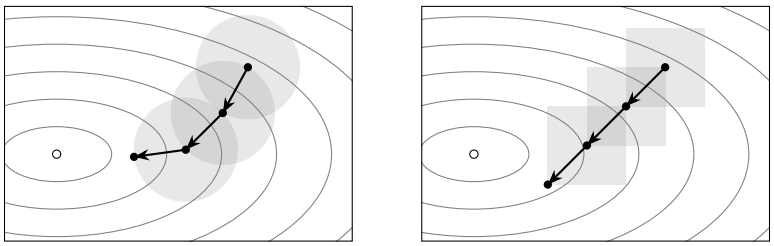


Figure 3.10: Steepest-descent steps on the level sets of a convex function (schematic). Left: each step minimizes the linear approximation over an ℓ_2 ball around the current point. Right: over an ℓ_∞ ball. The open circle is the minimizer.

3.6.2 The analyses in general norms

The two analyses of gradient descent for smooth functions from Section 3.4.4 (stated there for ℓ_2) carry over to general norms: in the primal-progress and dual-bound steps the ℓ_2 norm of the gradient is replaced by its dual norm $\|\nabla f(x_k)\|_*$. Here the gradient step is the update (3.6.3),

and smoothness and strong convexity are with respect to $\|\cdot\|$. Recall from Exercise 3.4.3 that these give, for all x, y,

$$f(y) \le f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|y - x\|^2 \qquad (L\text{-smooth}), \tag{3.6.4}$$

$$f(y) \ge f(x) + \langle \nabla f(x), y - x\rangle + \frac{\mu}{2}\|y - x\|^2 \qquad (\mu\text{-strongly convex}). \tag{3.6.5}$$

Claim 3.6.1. *If f is L-smooth and μ-strongly convex with respect to $\|\cdot\|$, then gradient descent (3.6.3) gives x_k such that*

$$f(x_k) - f_* \le \Big(1 - \frac{\mu}{L}\Big)^k (f(x_0) - f_*).$$

Proof. Primal progress.

$$f(x_{k+1}) \le f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|_*^2 \quad \Longrightarrow \quad (f(x_k) - f_*) - (f(x_{k+1}) - f_*) \ge \frac{1}{2L}\|\nabla f(x_k)\|_*^2. \tag{3.6.6}$$

Dual bound.

$$f(x_k) - f_* \le \frac{1}{2\mu}\|\nabla f(x_k)\|_*^2. \tag{3.6.7}$$

Combining these gives $(f(x_k) - f_*) - (f(x_{k+1}) - f_*) \ge \frac{\mu}{L}(f(x_k) - f_*)$, that is, $f(x_{k+1}) - f_* \le (1 - \frac{\mu}{L})(f(x_k) - f_*)$, which implies the claim.

Both (3.6.6) and (3.6.7) follow from one computation: for $g \in \mathbb{R}^n$ and $c > 0$,

$$\min_y \ \langle g, y - x\rangle + \frac{c}{2}\|y - x\|^2 = \min_{s \ge 0}\Big(-s\|g\|_* + \frac{c}{2}s^2\Big) = -\frac{1}{2c}\|g\|_*^2, \tag{3.6.8}$$

since by (3.6.2) the smallest value of $\langle g, y - x\rangle$ over $\|y - x\| = s$ is $-s\|g\|_*$. Since x_{k+1} minimizes the upper bound (3.6.4) at $x = x_k$, taking $c = L$ gives (3.6.6). Minimizing both sides of (3.6.5) over y and taking $c = \mu$ gives $f_* \ge f(x_k) - \frac{1}{2\mu}\|\nabla f(x_k)\|_*^2$, which is (3.6.7). □

Claim 3.6.2. *If f is L-smooth with respect to $\|\cdot\|$, then gradient descent (3.6.3) gives x_k such that*

$$f(x_k) - f_* \le O\Big(\frac{LR^2}{k}\Big), \qquad \text{where } R = \max_{x\,:\,f(x)\le f(x_0)} \ \min_{x_* \in X_*} \|x - x_*\|.$$

Proof. Let $\epsilon_k := f(x_k) - f_*$. *Primal progress:* same as before, $\epsilon_k - \epsilon_{k+1} \ge \frac{1}{2L}\|\nabla f(x_k)\|_*^2$ by (3.6.6). *Dual bound:*

$$\epsilon_k = f(x_k) - f_* \le \min_{x_* \in X_*} \langle \nabla f(x_k), x_k - x_*\rangle \le \|\nabla f(x_k)\|_* \min_{x_* \in X_*} \|x_k - x_*\| \le \|\nabla f(x_k)\|_* R,$$

where the middle inequality is $\langle z, y\rangle \le \|z\|_*\|y\|$, which follows from (3.6.2), and the last holds because the iterates stay in the sublevel set $\{x : f(x) \le f(x_0)\}$. Combining these gives $\epsilon_k - \epsilon_{k+1} \ge \epsilon_k^2/(2LR^2)$, and the rest of the proof is word for word the one in Section 3.4.4:

$$\frac{1}{\epsilon_{k+1}} - \frac{1}{\epsilon_k} = \frac{\epsilon_k - \epsilon_{k+1}}{\epsilon_k \epsilon_{k+1}} \ge \frac{\epsilon_k}{2LR^2\epsilon_{k+1}} \ge \frac{1}{2LR^2},$$

and since $\epsilon_0 \le \frac{LR^2}{2}$, $\frac{1}{\epsilon_k} \ge \frac{1}{\epsilon_0} + \frac{k}{2LR^2} \ge \frac{k+4}{2LR^2}$, so $\epsilon_k \le \frac{2LR^2}{k+4}$. □

3.7 Online Linear Optimization and Follow the Regularized Leader

We now generalize subgradient descent in several ways:

- to norms other than ℓ_2;
- to constrained optimization over a convex set (instead of all of $\mathbb{R}^n$);
- to online linear optimization.

Online linear optimization generalizes convex optimization with a subgradient oracle, and the problem of learning from experts is a special case of it. The natural strategy of playing the best point so far does not work; adding a strongly convex regularizer fixes it. The resulting algorithm, follow the regularized leader, has average regret $O(1/\sqrt{T})$, with a constant that depends on how the norm, the regularizer and the loss vectors fit together. For learning from experts, choosing the ℓ_1 norm and the entropy regularizer improves the number of rounds from n/ϵ^2 to $2\log n/\epsilon^2$.

3.7.1 Online linear optimization

Definition 3.7.1 (Online linear optimization)**.** Fix a convex set $S \subseteq \mathbb{R}^n$. In each round $t = 1, 2, \ldots$:

- choose $x_t \in S$;
- after you make your choice, find out a vector p_t, and incur the penalty $\langle p_t, x_t \rangle$.

The *regret* after T rounds is

$$\text{Regret}(T) := \sum_{t=1}^{T} \langle p_t, x_t \rangle - \min_{x \in S} \sum_{t=1}^{T} \langle p_t, x \rangle, \tag{3.7.1}$$

the difference between your total penalty and the penalty of the best *fixed* choice of x. The goal is to make the *average regret* $\text{Regret}(T)/T$ tend to 0 as $T \to \infty$.

Example application. The case

$$S = \Big\{ x \geq 0 : \sum_i x_i = 1 \Big\} =: \Delta_n, \qquad p_t(i) \in [-1, 1],$$

has a nice interpretation, given in Section 3.7.2.

It generalizes convex optimization with a subgradient oracle. Take the p_t to be the subgradients returned by the oracle at the points x_t, so that $f(x_t) - f(x_*) \leq \langle p_t, x_t - x_* \rangle$, where x_* is a minimizer of f over S. Then, by convexity of f,

$$\begin{aligned} f\Big(\frac{1}{T}\sum_{t=1}^{T} x_t\Big) - f_* &\leq \frac{1}{T}\sum_{t=1}^{T} f(x_t) - f_* \leq \frac{1}{T}\sum_{t=1}^{T} \langle p_t, x_t - x_* \rangle \\ &\leq \frac{1}{T}\sum_{t=1}^{T} \langle p_t, x_t \rangle - \frac{1}{T} \min_{x \in S} \sum_{t=1}^{T} \langle p_t, x \rangle = \text{Regret}(T)/T. \end{aligned}$$

So an online algorithm with average regret ϵ yields a point (the average of its choices) with error at most ϵ.

3.7.2 Learning from experts

Definition 3.7.2 (Learning from experts)**.** There are n "experts" who give you advice in each round t.

- You choose a probability distribution x_t and pick the ith expert with probability $x_t(i)$.
- The world supplies a penalty vector p_t, such that the penalty of following the ith expert is $p_t(i)$.
- So your expected penalty in round t equals

$$\sum_i p_t(i)\, x_t(i) = \langle p_t, x_t \rangle.$$

The regret measures your total penalty over T steps minus the total penalty of the best expert. Indeed, since a linear function on Δ_n is minimized at a vertex,

$$\min_{x \in \Delta_n} \sum_{t=1}^{T} \langle p_t, x \rangle = \min_{x \in \Delta_n} \sum_{i=1}^{n} x_i \Big(\sum_{t=1}^{T} p_t(i) \Big) = \min_i \sum_{t=1}^{T} p_t(i),$$

and therefore

$$\text{Regret}(T) = \sum_{t=1}^{T} \langle p_t, x_t \rangle - \min_{x \in \Delta_n} \sum_{t=1}^{T} \langle p_t, x \rangle = \sum_{t=1}^{T} \langle p_t, x_t \rangle - \min_i \sum_{t=1}^{T} p_t(i).$$

Remark 3.7.3.

- From the perspective of our formulation, the loss is not random: it equals $\langle p_t, x_t \rangle$. One could also think of investing a fraction $x_t(i)$ of your money with expert i, so that we measure an aggregate, not an expectation.
- A good example to keep in mind is the case where the penalties are always 0 or 1.
- For example, you want to bet on football games every week, and you can find a bunch of (alleged) experts online who predict each game. You want to use their advice to do (asymptotically) as well in the long run as the best expert.

We now give an algorithm for online linear optimization (Sections 3.7.3 and 3.7.4) and then apply it to learning from experts (Section 3.7.5).

3.7.3 Follow the leader, regularized

It is helpful to think about why a few simple strategies do not work. The most natural one is called *follow the leader*: at time $t+1$, choose the point that would have done best on the first t rounds, that is,

$$x_{t+1} = \arg\min_{x \in S} \sum_{i=1}^{t} \langle p_i, x \rangle. \tag{3.7.2}$$

Exercise 3.7.4. Why does follow the leader not work?

Instead, we will use a *regularized* version of this strategy.

Algorithm 3.7.5 (Follow the regularized leader)**.** Fix a *regularizer* $R : \mathbb{R}^n \to \mathbb{R}$ that is differentiable and μ-strongly convex with respect to some norm $\|\cdot\|$, and a step size $\eta > 0$. Define the regularized loss function

$$\Phi_t(x) \coloneqq \eta\Big(\sum_{i=1}^{t} \langle p_i, x\rangle\Big) + R(x), \tag{3.7.3}$$

and set

$$x_{t+1} = \arg\min_{x \in S} \Phi_t(x). \tag{3.7.4}$$

In particular $\Phi_0 = R$, so $x_1 = \arg\min_{x\in S} R(x)$.

Claim 3.7.6. *For all $y \in S$,*

$$\Phi_t(y) \geq \Phi_t(x_{t+1}) + \frac{\mu}{2}\|y - x_{t+1}\|^2.$$

Proof. Local optimality of x_{t+1} (the first-order optimality condition for minimizing the convex function Φ_t over the convex set S) says that

$$\langle \nabla\Phi_t(x_{t+1}), y - x_{t+1}\rangle \geq 0 \qquad \forall y \in S.$$

Strong convexity of R implies strong convexity of Φ_t, since Φ_t differs from R by a linear function. Hence

$$\Phi_t(y) \geq \Phi_t(x_{t+1}) + \langle \nabla\Phi_t(x_{t+1}), y - x_{t+1}\rangle + \frac{\mu}{2}\|y - x_{t+1}\|^2 \geq \Phi_t(x_{t+1}) + \frac{\mu}{2}\|y - x_{t+1}\|^2. \qquad \square$$

3.7.4 The regret of follow the regularized leader

Claim 3.7.7. *For any $p \in \mathbb{R}^n$ and $x, x' \in \mathbb{R}^n$,*

$$\eta\langle p, x' - x\rangle + \frac{\mu}{2}\|x' - x\|^2 \geq -\frac{\eta^2}{2\mu}\|p\|_*^2.$$

Proof. By the definition of the dual norm, $\langle p, x' - x\rangle \geq -\|p\|_*\|x' - x\|$. With $s = \|x' - x\|$ the left-hand side is therefore at least $-\eta\|p\|_* s + \frac{\mu}{2}s^2$, a quadratic in s whose minimum, at $s = \eta\|p\|_*/\mu$, is $-\frac{\eta^2}{2\mu}\|p\|_*^2$. $\square$

Lemma 3.7.8. *For every $t \geq 1$,*

$$\Phi_t(x_{t+1}) - \Phi_{t-1}(x_t) \geq \eta\langle p_t, x_t\rangle - \frac{\eta^2}{2\mu}\|p_t\|_*^2.$$

Proof. By the definition (3.7.3), and by Claim 3.7.6 applied to Φ_{t-1} (whose minimizer over S is x_t) with $y = x_{t+1}$,

$$\Phi_t(x_{t+1}) = \Phi_{t-1}(x_{t+1}) + \eta\langle p_t, x_{t+1}\rangle \geq \Phi_{t-1}(x_t) + \frac{\mu}{2}\|x_{t+1} - x_t\|^2 + \eta\langle p_t, x_{t+1}\rangle.$$

Therefore

$$\begin{aligned}
\Phi_t(x_{t+1}) - \Phi_{t-1}(x_t) &\geq \frac{\mu}{2}\|x_{t+1} - x_t\|^2 + \eta\langle p_t, x_{t+1}\rangle \\
&= \eta\langle p_t, x_t\rangle + \frac{\mu}{2}\|x_{t+1} - x_t\|^2 + \eta\langle p_t, x_{t+1} - x_t\rangle \\
&\geq \eta\langle p_t, x_t\rangle - \frac{\eta^2}{2\mu}\|p_t\|_*^2,
\end{aligned}$$

where the last step is Claim 3.7.7 with $x = x_t$ and $x' = x_{t+1}$. $\square$

Claim 3.7.9 (Convergence of follow the regularized leader)**.** *If $\|p_t\|_* \le L$ for all t, then for all $y \in S$,*

$$\sum_{t=1}^{T} \langle p_t, x_t - y\rangle \le \frac{\eta T L^2}{2\mu} + \frac{1}{\eta}\Big(R(y) - \min_{x\in S} R(x)\Big). \tag{3.7.5}$$

Let $D = \max_{x\in S} R(x) - \min_{x\in S} R(x)$. If we set $\eta = \sqrt{\frac{2\mu D}{L^2 T}}$, we get $\mathrm{Regret}(T) \le \sqrt{2DL^2T/\mu}$*, so*

$$\frac{\mathrm{Regret}(T)}{T} \le \sqrt{\frac{2DL^2}{T\mu}}. \tag{3.7.6}$$

Proof. Summing Lemma 3.7.8 over $t = 1, \dots, T$, the left-hand sides telescope:

$$\Phi_T(x_{T+1}) - \Phi_0(x_1) \ge \sum_{t=1}^{T}\Big(\eta\langle p_t, x_t\rangle - \frac{\eta^2}{2\mu}\|p_t\|_*^2\Big).$$

The claim follows by plugging in $\Phi_0(x_1) = \min_{x\in S} R(x)$ and $\|p_t\|_* \le L$, and by using $\Phi_T(x_{T+1}) \le \Phi_T(y) = \eta\sum_{t=1}^{T}\langle p_t, y\rangle + R(y)$, since x_{T+1} minimizes Φ_T over S. Rearranging and dividing by η gives (3.7.5). Taking the maximum over $y \in S$ turns the left-hand side into $\mathrm{Regret}(T)$ and bounds the last term by D/η. With the stated η, both terms equal $\sqrt{DL^2T/(2\mu)}$, so their sum is $\sqrt{2DL^2T/\mu}$. □

Remark 3.7.10 (Subgradient descent as a special case). If $S = \mathbb{R}^n$, $\|\cdot\| = \|\cdot\|_2$ and $R(x) = \frac{\|x\|^2}{2}$, we recover the subgradient descent algorithm (and essentially the same analysis). In this case $x_{t+1} = \arg\min_x \eta\sum_{i\le t}\langle p_i, x\rangle + \frac{1}{2}\|x\|^2 = -\eta\sum_{i\le t} p_i$, so $x_1 = 0$ and $x_{t+1} = x_t - \eta p_t$, which is the subgradient step when $p_t \in \partial f(x_t)$.

3.7.5 Solving learning from experts: the entropy regularizer

Take a norm $\|\cdot\|$ and a regularizer R that is 1-strongly convex with respect to $\|\cdot\|$. We showed (Claim 3.7.9 with $\mu = 1$) that if $\|p_t\|_* \le L$ and $D = \max_{x\in S} R(x) - \min_{x\in S} R(x)$, then

$$\frac{\mathrm{Regret}(T)}{T} \le \sqrt{\frac{2DL^2}{T}}.$$

So the average regret is at most ϵ when $T \ge \frac{2DL^2}{\epsilon^2}$. We want to apply this when

$$S = \Big\{x \ge 0 : \sum_i x_i = 1\Big\} =: \Delta_n, \qquad p_t(i) \in [0, 1].$$

The Euclidean regularizer. If we use $\|\cdot\|_2$ and $R(x) = \frac{\|x\|^2}{2}$, we get $D = \frac{1}{2} - \frac{1}{2n} \le 1/2$ and $L \le \max_t\|p_t\|_2 \le \sqrt{n}$, so we need $T \ge n/\epsilon^2$ to get average regret ϵ. Can we do better if we choose another norm?

Choosing the norm.

- Note that the simplex is part of the unit sphere of ℓ_1, and that $p_t \in [-1, 1]^n$ has $\|p_t\|_\infty \le 1$, so it is bounded in the dual norm.

- So measuring x in the 1-norm and $p = p_t$ in the ∞-norm has a chance to give a much smaller L, and hopefully does not blow up D. Strong convexity of the regularizer implies that a large diameter with respect to $\|\cdot\|$ forces a large D, so this is at least reasonable.
- We need a regularizer that is 1-strongly convex with respect to $\|\cdot\|_1$.

Claim 3.7.11. *The (negative) entropy regularizer*

$$R(x) = \sum_i x_i \log x_i \qquad (\textit{where } 0 \log 0 := 0)$$

is 1*-strongly convex with respect to* $\|\cdot\|_1$ *on* Δ_n.

Granting the claim:

- R is maximized when one coordinate is 1 and the rest are 0, where $R(x) = 0$;
- R is minimized at $x = \frac{\mathbf{1}}{n}$, where $R(x) = \sum_{i=1}^n \frac{1}{n} \log \frac{1}{n} = -\log n$.

The first holds since $x_i \log x_i \le 0$ for $x_i \in [0,1]$. The second is Jensen's inequality for the concave function log: $\sum_i x_i \log \frac{1}{x_i} \le \log \sum_{i: x_i > 0} x_i \cdot \frac{1}{x_i} \le \log n$. So $D = \log n$ and $L = 1$, and the average regret is at most ϵ when

$$T \ge \frac{2 \log n}{\epsilon^2}.$$

Compare $T \ge n/\epsilon^2$ for the Euclidean regularizer.

Proof of Claim 3.7.11. We have $\nabla R(x)_i = 1 + \log x_i$, so

$$\nabla^2 R(x) = \operatorname{diag}(x)^{-1} =: X^{-1}.$$

We need $v^\top (\nabla^2 R(x)) v \ge \|v\|_1^2$ for all x and v, which follows from a simple calculation using the Cauchy–Schwarz inequality:

$$\|v\|_1^2 = \Big(\sum_i |v_i|\Big)^2 = \Big(\sum_i \frac{|v_i|}{\sqrt{x_i}} \sqrt{x_i}\Big)^2 \le \Big(\sum_i \Big(\frac{|v_i|}{\sqrt{x_i}}\Big)^2\Big)\Big(\sum_i \sqrt{x_i}^2\Big) = v^\top X^{-1} v \cdot \|x\|_1,$$

and $\|x\|_1 = 1$ on Δ_n. Here x ranges over the points of Δ_n with all $x_i > 0$, where R is twice differentiable. To get strong convexity from the Hessian bound, take x, y in this set and integrate $\frac{d^2}{ds^2} R(x + s(y-x)) = (y-x)^\top \nabla^2 R(x + s(y-x))(y-x) \ge \|y - x\|_1^2$ twice over $s \in [0,1]$. This gives $R(y) \ge R(x) + \langle \nabla R(x), y - x \rangle + \frac{1}{2}\|y - x\|_1^2$. The boundary of Δ_n follows by continuity. $\square$

Remark 3.7.12 (The resulting algorithm). With the entropy regularizer on $S = \Delta_n$, the update (3.7.4) has a closed form. Write $c = \sum_{i \le t} p_i$. We minimize $\eta \langle c, x \rangle + \sum_j x_j \log x_j$ subject to $\sum_j x_j = 1$. The Lagrange condition $\eta c_j + 1 + \log x_j = \lambda$ gives

$$x_{t+1}(j) = \frac{e^{-\eta c_j}}{\sum_l e^{-\eta c_l}} = \frac{\exp\big(-\eta \sum_{i \le t} p_i(j)\big)}{\sum_l \exp\big(-\eta \sum_{i \le t} p_i(l)\big)}.$$

So in round t each expert's weight is multiplied by $e^{-\eta p_t(j)}$, and then the weights are renormalized. This is the multiplicative weights (or exponential weights, or Hedge) algorithm.

3.8 The Accelerated Gradient Method

We now return to (offline) convex optimization. Gradient descent on an L-smooth, μ-strongly convex function needs about $\kappa \log \frac{1}{\epsilon}$ steps, where $\kappa = L/\mu$, while the lower bound of Section 3.5.3 is $\sqrt{\kappa} \log \frac{1}{\epsilon}$. This section derives an algorithm that closes the gap. We first see heuristically why combining a gradient step with a subgradient step should help. We then derive the algorithm from a different viewpoint: every point gives a quadratic upper bound and a quadratic lower bound on f, and the algorithm maintains a point and a lower bound and shrinks the gap between them by a factor $1 - 1/\sqrt{\kappa}$ in each step. Throughout, we work in ℓ_2 to keep things a little simpler: $\|\cdot\| = \|\cdot\|_2$, L is the smoothness constant and μ the strong-convexity parameter.

3.8.1 Goal and motivation

Goal. Improve the running times of gradient descent to match the lower bounds of Section 3.5.3. To get error ϵ:

- **L-smooth (not strongly convex):** gradient descent needs #steps $\propto \frac{L}{\epsilon}$; we want $\sqrt{\frac{L}{\epsilon}}$.
- **L-smooth, μ-strongly convex:** gradient descent needs #steps $\propto \frac{L}{\mu} \log \frac{1}{\epsilon} =: \kappa \log \frac{1}{\epsilon}$; we want $\sqrt{\kappa} \log \frac{1}{\epsilon}$.

One can reduce the cases with and without strong convexity to each other, so it suffices to treat either one. *(Proof omitted.)*

Many people say that the accelerated algorithm is unmotivated and difficult to analyze, and that its analysis is basically an algebraic magic trick. We disagree, and to demonstrate this we give not just one but four different ways to motivate it. The first two are:

- **Motivation 1:** Chebyshev iteration and conjugate gradient for minimizing quadratics (Sections 3.1.6 and 3.2.4), which already achieve the $\kappa \to \sqrt{\kappa}$ speedup when f is quadratic;
- **Motivation 2:** the heavy ball method.

We do not go further into these two. The heavy ball method adds a "momentum" term to gradient descent: $x_{k+1} = x_k - a\nabla f(x_k) + b(x_k - x_{k-1})$ for constants $a, b > 0$. Section 3.8.2 gives Motivation 3, and Sections 3.8.3 to 3.8.6 derive the algorithm from upper and lower bounds.

3.8.2 Motivation 3: combining gradient descent and subgradient descent

Suppose f is L-smooth and $f(x_0) - f(x_*) \le \epsilon$. How many steps do we need to get the error down to $\epsilon/2$? For intuition, suppose that all gradients (while going from ϵ to $\epsilon/2$) have the same size, say $\|\nabla f\| = K$.

- The gradient descent analysis (Lemma 3.4.6) showed that f decreases by at least $\frac{\|\nabla f\|^2}{2L} = \frac{K^2}{2L}$ in each step. So we need $\frac{\epsilon L}{K^2}$ steps. Note that bigger gradients mean faster convergence.
- The subgradient analysis (Claim 3.5.4) showed that if $\|\nabla f\| \le K$, the number of steps is proportional to $\frac{K^2}{\epsilon^2}$ (for a fixed distance R to the optimum). Note that smaller gradients mean faster convergence.

So if we choose gradient descent when K is large and subgradient descent when K is small, we should hope to get

$$\#\text{steps} = O\Big(\min\Big(\frac{\epsilon L}{K^2}, \frac{K^2}{\epsilon^2}\Big)\Big).$$

The two terms balance when $K^2 = \sqrt{\epsilon^3 L}$, so the minimum is at most

$$O\Big(\frac{\epsilon L}{\sqrt{\epsilon^3 L}}\Big) = O\Big(\frac{\sqrt{\epsilon^3 L}}{\epsilon^2}\Big) = O\Big(\sqrt{\frac{L}{\epsilon}}\Big),$$

which matches our lower bound.

Issues.

- The two algorithms take different step sizes, and their analyses use different potential functions. And steps of one algorithm may hurt the other.
- What do we do if gradient descent tells us to go to y and subgradient descent tells us to go to z? *Answer:* just keep track of both, that is, keep two sequences of iterates, y_k and z_k.
- But where do we take the next gradient? *Answer:* somewhere in the middle. Fix some parameter τ and take the gradient at a convex combination of y_k and z_k.

Algorithm 3.8.1 (Coupling gradient and subgradient steps)**.** Given y_k and z_k, set

$$\begin{aligned} x_{k+1} &= \tau y_k + (1-\tau) z_k, \\ y_{k+1} &= [\text{gradient step from } x_{k+1}] = x_{k+1} - \frac{1}{L}\nabla f(x_{k+1}), \\ z_{k+1} &= [\text{subgradient step from } z_k] = z_k - \eta \nabla f(x_{k+1}). \end{aligned}$$

3.8.3 Upper and lower bounds from a single point

Suppose f is L-smooth and μ-strongly convex. Then each point x gives an upper and a lower bound on f (Exercise 3.4.3): for all $y \in \mathbb{R}^n$,

$$f(y) \le U_x(y) \coloneqq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|y-x\|^2, \tag{3.8.1}$$

$$f(y) \ge L_x(y) \coloneqq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\mu}{2}\|y-x\|^2. \tag{3.8.2}$$

We write L_x (and below L_k) for these lower-bound functions; they should not be confused with the constant L.

Gradient descent in this language. Given y_k:

- Use U_{y_k} to find a point with a smaller value of f:

$$y_{k+1} = \arg\min_y U_{y_k}(y) = y_k - \frac{1}{L}\nabla f(y_k), \qquad f(y_{k+1}) \le U_{y_k}(y_{k+1}) = f(y_k) - \frac{1}{2L}\|\nabla f(y_k)\|^2.$$

- Use L_{y_k} to find a lower bound on f_*. Let

$$v_k = \arg\min_y L_{y_k}(y) = y_k - \frac{1}{\mu}\nabla f(y_k).$$

  Then

$$f_* \geq \min_y L_{y_k}(y) = L_{y_k}(v_k) = f(y_k) - \frac{1}{2\mu}\|\nabla f(y_k)\|^2.$$

- Together, these imply $f(y_k) - f(y_{k+1}) \geq \frac{1}{\kappa}(f(y_k) - f_*)$, the rate of Claim 3.4.5.

This step optimizes the improvement of the upper bound. We would learn more about the lower bound if we took a longer step, but then we might hurt f. Analogously to what we did with subgradient descent, we need to average lower bounds.

3.8.4 The idea of the accelerated gradient method

- Keep track of points y_k and lower-bound functions $L_k(y)$, with $f \geq L_k$.
- Evaluate the gradient at some point x_{k+1}, and use it to get y_{k+1} and L_{k+1}.
- Instead of taking the gradient at y_k to optimize the improvement of f, look for a point x_{k+1} that optimizes the gap $f(y_{k+1}) - \min_y L_{k+1}(y)$.
- The lower bounds will be convex combinations of the L_x at various points x. Each L_x is a quadratic with leading term $\frac{\mu}{2}\|y\|^2$, so we can write them in the form

$$L_k(y) = \psi_k + \frac{\mu}{2}\|y - v_k\|^2. \tag{3.8.3}$$

  In this section ψ_k is a number (the minimum value of L_k) and v_k a point (its minimizer).
- We will take the gradient at points of the form $x_{k+1} = \alpha y_k + (1-\alpha)v_k$.
- We get y_{k+1} by optimizing $U_{x_{k+1}}$: $y_{k+1} = \arg\min_y U_{x_{k+1}}(y) = x_{k+1} - \frac{1}{L}\nabla f(x_{k+1})$. Note that $f(y_{k+1})$ may be worse than $f(y_k)$!
- We get L_{k+1} by combining L_k and $L_{x_{k+1}}$: $L_{k+1}(y) = \beta L_k(y) + (1-\beta)L_{x_{k+1}}(y) = \psi_{k+1} + \frac{\mu}{2}\|y - v_{k+1}\|^2$.
- We need to work out how to find ψ_{k+1} and v_{k+1}, and how to choose α and β.

To summarize, the update is

$$\begin{aligned} x_{k+1} &= \alpha y_k + (1-\alpha)v_k, \\ y_{k+1} &= x_{k+1} - \frac{1}{L}\nabla f(x_{k+1}), \\ L_{k+1}(y) &= \beta L_k(y) + (1-\beta)L_{x_{k+1}}(y) =: \psi_{k+1} + \frac{\mu}{2}\|y - v_{k+1}\|^2. \end{aligned} \tag{3.8.4}$$

Plan.

1. Work out the details of combining lower bounds, and use them to bound ψ_{k+1} (Section 3.8.5, Lemma 3.8.6).
2. Combine with the usual bound on the gradient step (from x_{k+1}) to bound $f(y_{k+1}) - \psi_{k+1}$ (Lemma 3.8.7).
3. Choose the parameters α, β to get a large improvement:
$$f(y_{k+1}) - \psi_{k+1} \le \Big(1 - \frac{1}{\sqrt{\kappa}}\Big)\big(f(y_k) - \psi_k\big)$$
(Theorem 3.8.8).
4. Show that we can get a reasonable value for the initial gap $f(y_0) - \psi_0$ (Section 3.8.6).

Remark 3.8.2. Once we have decided to optimize the gap, and to maintain a point and a single quadratic lower bound, the rest is pretty much forced, and we just need to do the appropriate calculations.

3.8.5 Combining lower bounds

Lemma 3.8.3. *Suppose we have two lower bounds on f:*
$$f(y) \ge L_A(y) = \psi_A + \frac{\mu}{2}\|y - v_A\|^2, \qquad f(y) \ge L_B(y) = \psi_B + \frac{\mu}{2}\|y - v_B\|^2.$$
For $\beta \in [0,1]$ we get the new lower bound
$$f(y) \ge \beta L_A(y) + (1-\beta)L_B(y) = \beta\Big(\psi_A + \frac{\mu}{2}\|y - v_A\|^2\Big) + (1-\beta)\Big(\psi_B + \frac{\mu}{2}\|y - v_B\|^2\Big),$$
and with a little algebra it can be written in the form $\psi + \frac{\mu}{2}\|y - v\|^2$, where
$$v = \beta v_A + (1-\beta)v_B, \qquad \psi = \beta\psi_A + (1-\beta)\psi_B + \frac{\mu}{2}\beta(1-\beta)\|v_A - v_B\|^2. \tag{3.8.5}$$

Proof. It suffices to check that $\beta\|y - v_A\|^2 + (1-\beta)\|y - v_B\|^2 = \|y - v\|^2 + \beta(1-\beta)\|v_A - v_B\|^2$. Write $a = y - v_A$ and $b = y - v_B$, so that $y - v = \beta a + (1-\beta)b$ and $v_A - v_B = b - a$. Expanding $\|\beta a + (1-\beta)b\|^2 + \beta(1-\beta)\|b - a\|^2$, the $\langle a, b\rangle$ terms cancel, and what is left is $\beta\|a\|^2 + (1-\beta)\|b\|^2$. □

In the algorithm. We have $L_{k+1}(y) = \psi_{k+1} + \frac{\mu}{2}\|y - v_{k+1}\|^2$, and, completing the square in (3.8.2),
$$L_{x_{k+1}}(y) = \Big(f(x_{k+1}) - \frac{1}{2\mu}\|\nabla f(x_{k+1})\|^2\Big) + \frac{\mu}{2}\Big\|y - \Big(x_{k+1} - \frac{1}{\mu}\nabla f(x_{k+1})\Big)\Big\|^2 =: \psi_{x_{k+1}} + \frac{\mu}{2}\|y - v_{x_{k+1}}\|^2. \tag{3.8.6}$$
So, by Lemma 3.8.3, $L_{k+1}(y) = \beta L_k(y) + (1-\beta)L_{x_{k+1}}(y) = \psi_{k+1} + \frac{\mu}{2}\|y - v_{k+1}\|^2$ has
$$v_{k+1} = \beta v_k + (1-\beta)v_{x_{k+1}}, \tag{3.8.7}$$
$$\begin{aligned}\psi_{k+1} &= \beta\psi_k + (1-\beta)\psi_{x_{k+1}} + \frac{\mu}{2}\beta(1-\beta)\|v_k - v_{x_{k+1}}\|^2 \\ &= \beta\psi_k + (1-\beta)\Big(f(x_{k+1}) - \frac{1}{2\mu}\|\nabla f(x_{k+1})\|^2\Big) + \frac{\mu}{2}\beta(1-\beta)\Big\|v_k - \Big(x_{k+1} - \frac{1}{\mu}\nabla f(x_{k+1})\Big)\Big\|^2.\end{aligned} \tag{3.8.8}$$

Putting (3.8.4) and (3.8.7) together, the algorithm only needs to track the points y_k and v_k:

Algorithm 3.8.4 (Accelerated gradient method)**.** Given y_k and v_k, set

$$\begin{aligned} x_{k+1} &= \alpha y_k + (1-\alpha)v_k, \\ y_{k+1} &= x_{k+1} - \frac{1}{L}\nabla f(x_{k+1}), \\ v_{k+1} &= \beta v_k + (1-\beta)\Big(x_{k+1} - \frac{1}{\mu}\nabla f(x_{k+1})\Big). \end{aligned}$$

The numbers ψ_k are used only in the analysis. The parameters are chosen in (3.8.10).

Remark 3.8.5. This has the shape of Algorithm 3.8.1: y_{k+1} is a gradient step from the query point x_{k+1}, and v_k plays the role of z_k. The step to v_{k+1} starts from the average $\beta v_k + (1-\beta)x_{k+1}$ and moves by $-\frac{1-\beta}{\mu}\nabla f(x_{k+1})$. This is a long step: with β as in (3.8.10), the step size is $\frac{1-\beta}{\mu} = \frac{1}{\sqrt{L\mu}} = \frac{\sqrt{\kappa}}{L}$, compared with $\frac{1}{L}$ for the gradient step.

3.8.6 Analysis of the gap

Write $g := \nabla f(x_{k+1})$ in this section.

Lemma 3.8.6. *If $0 \le \alpha < 1$ and $0 \le \beta \le 1$, then*

$$\psi_{k+1} \ge \beta\psi_k + (1-\beta)\Big(f(x_{k+1}) - \frac{1}{2\mu}\|g\|^2\Big) + \beta(1-\beta)\Big(\frac{\alpha}{1-\alpha}(f(x_{k+1}) - f(y_k)) + \frac{1}{2\mu}\|g\|^2\Big).$$

Proof. Expanding the square,

$$\frac{\mu}{2}\Big\|v_k - \Big(x_{k+1} - \frac{1}{\mu}g\Big)\Big\|^2 \ge \langle g, v_k - x_{k+1}\rangle + \frac{1}{2\mu}\|g\|^2, \tag{3.8.9}$$

just dropping the nonnegative term $\frac{\mu}{2}\|v_k - x_{k+1}\|^2$. Substituting into (3.8.8), which is fine since $\beta(1-\beta) \ge 0$,

$$\psi_{k+1} \ge \beta\psi_k + (1-\beta)\Big(f(x_{k+1}) - \frac{1}{2\mu}\|g\|^2\Big) + \beta(1-\beta)\Big(\langle g, v_k - x_{k+1}\rangle + \frac{1}{2\mu}\|g\|^2\Big).$$

Two facts handle the inner product. First, the definition $x_{k+1} = \alpha y_k + (1-\alpha)v_k$ gives

$$v_k - x_{k+1} = \frac{\alpha}{1-\alpha}(x_{k+1} - y_k).$$

Second, by convexity of f,

$$\langle \nabla f(x_{k+1}), x_{k+1} - y_k\rangle \ge f(x_{k+1}) - f(y_k)$$

Hence, since $\frac{\alpha}{1-\alpha} \ge 0$,

$$\begin{aligned} \psi_{k+1} &\ge \beta\psi_k + (1-\beta)\Big(f(x_{k+1}) - \frac{1}{2\mu}\|g\|^2\Big) + \beta(1-\beta)\Big(\Big\langle g, \frac{\alpha}{1-\alpha}(x_{k+1} - y_k)\Big\rangle + \frac{1}{2\mu}\|g\|^2\Big) \\ &\ge \beta\psi_k + (1-\beta)\Big(f(x_{k+1}) - \frac{1}{2\mu}\|g\|^2\Big) + \beta(1-\beta)\Big(\frac{\alpha}{1-\alpha}(f(x_{k+1}) - f(y_k)) + \frac{1}{2\mu}\|g\|^2\Big). \qquad \square \end{aligned}$$

Lemma 3.8.7. *Under the assumptions of Lemma 3.8.6,*

$$f(y_{k+1}) - \psi_{k+1} \le \beta(f(y_k) - \psi_k) + \beta\Big(1 - \frac{\alpha(1-\beta)}{1-\alpha}\Big)(f(x_{k+1}) - f(y_k)) + \Big(\frac{(1-\beta)^2}{2\mu} - \frac{1}{2L}\Big)\|g\|^2.$$

Proof. The gradient step satisfies

$$f(y_{k+1}) = f\Big(x_{k+1} - \frac{1}{L}g\Big) \le f(x_{k+1}) - \frac{1}{2L}\|g\|^2.$$

Subtracting the bound of Lemma 3.8.6,

$$\begin{aligned} f(y_{k+1}) - \psi_{k+1} &\le \Big(f(x_{k+1}) - \frac{1}{2L}\|g\|^2\Big) - \Big(\beta\psi_k + (1-\beta)\Big(f(x_{k+1}) - \frac{1}{2\mu}\|g\|^2\Big) \\ &\qquad + \beta(1-\beta)\Big(\frac{\alpha}{1-\alpha}(f(x_{k+1}) - f(y_k)) + \frac{1}{2\mu}\|g\|^2\Big)\Big) \\ &= \beta f(x_{k+1}) - \beta\psi_k + \Big(\frac{(1-\beta)^2}{2\mu} - \frac{1}{2L}\Big)\|g\|^2 - \beta(1-\beta)\frac{\alpha}{1-\alpha}(f(x_{k+1}) - f(y_k)) \\ &= \beta(f(y_k) - \psi_k) + \beta\Big(1 - \frac{\alpha(1-\beta)}{1-\alpha}\Big)(f(x_{k+1}) - f(y_k)) + \Big(\frac{(1-\beta)^2}{2\mu} - \frac{1}{2L}\Big)\|g\|^2. \end{aligned}$$

In the first equality the coefficient of $\|g\|^2$ is $-\frac{1}{2L} + \frac{1-\beta}{2\mu} - \frac{\beta(1-\beta)}{2\mu} = \frac{(1-\beta)^2}{2\mu} - \frac{1}{2L}$; in the second we wrote $\beta f(x_{k+1}) = \beta f(y_k) + \beta(f(x_{k+1}) - f(y_k))$. □

Choosing the parameters. Let $\kappa = L/\mu$ and set

$$\alpha = \frac{\sqrt{\kappa}}{1+\sqrt{\kappa}}, \qquad \beta = 1 - \sqrt{\frac{1}{\kappa}}. \tag{3.8.10}$$

These are chosen so that

$$(1-\beta)^2 = \frac{1}{\kappa} \qquad \text{and} \qquad \frac{\alpha(1-\beta)}{1-\alpha} = 1.$$

Note that $1-\beta = 1/\sqrt{\kappa}$ and $\alpha/(1-\alpha) = \sqrt{\kappa}$. With these choices the last two terms of Lemma 3.8.7 vanish: the middle coefficient is $1-1=0$, and $\frac{(1-\beta)^2}{2\mu} = \frac{1}{2\kappa\mu} = \frac{1}{2L}$. Since $\kappa \ge 1$, also $0 \le \alpha < 1$ and $0 \le \beta < 1$. This gives the contraction promised in the plan.

Theorem 3.8.8. *With the parameters (3.8.10), the accelerated gradient method (Algorithm 3.8.4) satisfies*

$$f(y_{k+1}) - \psi_{k+1} \le \Big(1 - \frac{1}{\sqrt{\kappa}}\Big)(f(y_k) - \psi_k).$$

Proof. Substitute (3.8.10) into Lemma 3.8.7. As just noted, the last two terms vanish, and $\beta = 1 - 1/\sqrt{\kappa}$. □

Completing the argument

It remains to bound the initial gap (the last step of the plan) and read off the resulting rate.

Corollary 3.8.9. *Start with any y_0 and the lower bound $L_0 = L_{y_0}$, that is, $v_0 = y_0 - \frac{1}{\mu}\nabla f(y_0)$ and $\psi_0 = f(y_0) - \frac{1}{2\mu}\|\nabla f(y_0)\|^2$. Then*

$$f(y_k) - f_* \le \kappa\Big(1 - \frac{1}{\sqrt{\kappa}}\Big)^k (f(y_0) - f_*) \le \kappa\, e^{-k/\sqrt{\kappa}}(f(y_0) - f_*),$$

so $O\big(\sqrt{\kappa}\log\frac{\kappa}{\epsilon}\big)$ iterations reduce the initial error by a factor ϵ.

Proof. Each L_k is a convex combination of lower bounds L_x, so $L_k \le f$ everywhere and $\psi_k = \min_y L_k(y) \le \min_y f(y) = f_*$. By Theorem 3.8.8, $f(y_k) - f_* \le f(y_k) - \psi_k \le (1 - 1/\sqrt{\kappa})^k (f(y_0) - \psi_0)$. Finally $f(y_0) - \psi_0 = \frac{1}{2\mu}\|\nabla f(y_0)\|^2 \le \frac{L}{\mu}(f(y_0) - f_*)$, since a gradient step from y_0 gives $f_* \le f(y_0) - \frac{1}{2L}\|\nabla f(y_0)\|^2$. □

This matches the lower bound $\Theta(\sqrt{\kappa}\log\frac{1}{\epsilon})$ of Section 3.5.3, up to an additive $O(\sqrt{\kappa}\log\kappa)$ from the initial gap. So we get the same $\kappa \to \sqrt{\kappa}$ speedup that Chebyshev iteration and conjugate gradient gave for quadratics (Sections 3.1.6 and 3.2.4), now for every smooth, strongly convex function.

References

[Bal97] Keith Ball. An elementary introduction to modern convex geometry. In Silvio Levy, editor, *Flavors of Geometry*, volume 31 of *MSRI Publications*, pages 1–58. Cambridge University Press, Cambridge, 1997.

[BK96] András A. Benczúr and David R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC 1996)*, pages 47–55, 1996.

[BSS12] Joshua Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-Ramanujan sparsifiers. *SIAM Journal on Computing*, 41(6):1704–1721, 2012.

[Chu97] Fan R. K. Chung. *Spectral Graph Theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, Providence, RI, 1997.

[CRVW02] Michael Capalbo, Omer Reingold, Salil Vadhan, and Avi Wigderson. Randomness conductors and constant-degree lossless expanders. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC 2002)*, pages 659–668, 2002.

[DFK91] Martin Dyer, Alan Frieze, and Ravi Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *Journal of the ACM*, 38(1):1–17, 1991.

[Ind01] Piotr Indyk. Algorithmic applications of low-distortion geometric embeddings. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS 2001)*, pages 10–33, 2001.

[JS89] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM Journal on Computing*, 18(6):1149–1178, 1989.

[JSV01] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC 2001)*, pages 712–721, 2001.

[RV07] Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM*, 54(4), 2007. Article 21.

[She94] Jonathan Richard Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical Report CMU-CS-94-125, School of Computer Science, Carnegie Mellon University, 1994.