Randomized Algorithms, Volume II

(much less polished than Volume I)

Nick Harvey
University of British Columbia

February 18, 2022

Contents

Part I

Introduction

Chapter 19

A Warmup: Set Cover

19.1 Definition and Background

Suppose we have a family of sets

$$S_1, \ldots, S_m \qquad \text{where} \qquad \bigcup_{j=1}^{m} S_j = [n].$$

The set $[n]$ is called the ***ground set***; recall that $[n]$ is standard notation for $\{1, \ldots, n\}$. Clearly each set in the family satisfies $S_j \subseteq [n]$.

We would like to choose as few of those sets as possible, while preserving the property that their union equals $[n]$. In mathematical notation, the Set Cover problem is

$$\min \left\{ |C| \,:\, C \subseteq [m] \text{ and } \bigcup_{j \in C} S_j = [n] \right\}.$$

Let OPT to refer to this minimum value.

An algorithm for this problem is called an α-approximation if it always outputs a cover C satisfying $|C| \leq \alpha \cdot \text{OPT}$. Clearly $\alpha \geq 1$. Here α could be a constant or it could depend on other parameters, like n or m.

Some known facts about the Set Cover problem.

- It is NP-hard, as was shown by Karp in 1972. Thus, it is unlikely that any polynomial time algorithm can guarantee to produce an exact solution.
- The natural *greedy algorithm* is a $\ln n$-approximation algorithm.
- For every constant $c < 1$, it is very unlikely that there is a $(c \ln n)$-approximation algorithm. (If such an algorithm existed, this would have similar consequences to P=NP.) Thus, the greedy algorithm is the best possible.

Instead of discussing the greedy algorithm, we will discuss this problem with a viewpoint of mathematical optimization.

Integer program. The first step is to write the problem as an integer program (IP). For each $j \in [m]$, we create a Boolean variable x_j that indicates whether we select the set S_j. Using these variables, we can write the problem as

$$\begin{aligned} \min \quad & \sum_{j=1}^{m} x_j & & \\ \text{s.t.} \quad & \sum_{j \,:\, i \in S_j} x_j & \geq 1 \qquad & \forall i \in [n] \\ & x_j \in \{0,1\} & & \forall j \in [m] \end{aligned}$$

The objective function $\sum_{j=1}^{m} x_j$ counts the number of chosen sets. The i^{th} inequality constraint ensures that at least one of the chosen sets contains element i. Thus, this integer program captures the Set Cover problem exactly. It follows that its minimum value is also OPT, and it is also NP-hard to solve exactly.

Linear program. The next idea is to relax the integer program to a ***linear program*** (LP), by allowing the variables x_j to lie in the interval $[0,1]$ rather than the discrete set $\{0,1\}$. The resulting LP can be written

$$\begin{aligned} \min \quad & \sum_{j=1}^{m} x_j & & \\ \text{s.t.} \quad & \sum_{j \,:\, i \in S_j} x_j & \geq 1 \qquad & \forall i \in [n] \\ & 0 \leq x_j \leq 1 & & \forall j \in [m] \end{aligned}$$

We will use LPOPT to denote the minimum value of this LP.

Question 19.1.1. Does the value of the LP always equal the value of the IP? That is, does LPOPT always equal OPT?

Answer.

No. If it were true, then we'd have proven P=NP, because LPs can be solved in polynomial time. That is not an airtight argument, but we'll see an example below where they differ.

Question 19.1.2. Do we always have LPOPT $\leq$ OPT or LPOPT $\geq$ OPT?

Answer.

We always have LPOPT $\leq$ OPT. To see this, note that every feasible solution x to the IP is also a feasible solution to the LP. Thus, the LP is minimizing over a larger set of feasible solutions, and therefore its value can only be smaller.

Example 1. Consider the following example with $n = m = 3$.

$$S_1 = \{1,2\} \qquad S_2 = \{2,3\} \qquad S_3 = \{1,3\}\,.$$

It is clear that we must choose at least two sets in order to cover all three elements of the ground set. However, if we set $x_1 = x_2 = x_3 = 0.5$, then one can verify that this is a feasible solution to the LP with objective value 1.5.

Keener Kwestion 19.1.3. In this example, what is LPOPT, the optimal value of the LP?

The term ***integrality gap*** is used to refer to the ratio of the optimum values of an IP and its corresponding LP. Example 1 shows that our LP for Set Cover has an integrality gap greater than 1. Nevertheless, the LP is still useful. We will show two results.

1. The integrality gap of this LP is at most $\approx \ln n$.
2. Given any feasible solution x to the LP, there is a randomized algorithm that, with constant probability, can produce a valid set cover C whose size is at most $\approx \ln(n) \cdot \sum_j x_j$.

Question 19.1.4. Do you see why the second assertion implies the first?

Answer.

gap.

Consider an optimum solution to the LP. It has objective value at most LPOPT. Applying the randomized algorithm, we obtain a set cover satisfying $|C| \lessapprox \ln(u)\text{LPOPT}$. Since the best Set Cover solution is no larger than C, it follows that $\text{OPT} \lessapprox \ln(u)\text{LPOPT}$, which gives the desired bound on the integrality

19.2 Randomized Rounding

In this section we design an algorithm that will prove the second assertion above. This is called a ***rounding algorithm***, because it converts a fractional solution into an integral solution.

Algorithm 19.1 Rounding the Set Cover LP. Assume that the input x is a feasible solution to the LP.

```
1: function SetCoverRound(x)
2:     Let C ← ∅                                  ▷ The indices of the chosen sets
3:     Let L ← ln(4n)                             ▷ The rounding has L phases
4:     for t = 1, ..., L
5:         for j = 1, ..., m
6:             Add j to C independently with probability x_j
7:     return C
8: end function
```

Lemma 19.2.1. The output C fails to cover all elements in $[n]$ (i.e., $\bigcup_{j \in C} S_j \neq [n]$) with probability at most $1/4$.

Claim 19.2.2. Consider the t^{th} phase of the algorithm. Each element $i \in [n]$ fails to be covered by the sets chosen in this phase with probability at most $1/e$.

Proof. Element i is not covered if every set containing i fails to be chosen. The probability of this is

$$\begin{aligned}
\Pr\left[\,(j \text{ not chosen}) \ \forall j \text{ s.t. } i \in S_j\,\right] &= \prod_{j\,:\,i \in S_j} \Pr\left[\,j \text{ not chosen}\,\right] \qquad \text{(by independence)} \\
&= \prod_{j\,:\,i \in S_j} (1 - x_j) \\
&< \prod_{j\,:\,i \in S_j} \exp(-x_j) \qquad \text{(by Fact A.2.5)} \\
&= \exp\Big(-\sum_{j\,:\,i \in S_j} x_j\Big).
\end{aligned}$$

This is at most $1/e$ since x is feasible for the LP. □

Proof of Lemma 19.2.1. First let us analyze the probability that a particular element is not covered by C. For any $i \in [n]$,

$$\begin{aligned}
\Pr[\text{element } i \text{ not covered}] &= \prod_{t=1}^{L} \Pr[\text{element } i \text{ not covered in phase } t] \\
&\le \prod_{t=1}^{L} (1/e) \qquad \text{(by Claim 19.2.2)} \\
&= \exp(-L) = \frac{1}{4n} \qquad \text{(since } L = \ln(4n)\text{)}.
\end{aligned}$$

By a union bound (Fact A.3.5), the probability that *any* element fails to be covered by C is

$$\Pr[\textit{any} \text{ element not covered by } C] \le \sum_{i=1}^{n} \Pr[\text{element } i \text{ not covered by } C] \le \sum_{i=1}^{n} \frac{1}{4n} = \frac{1}{4}. \quad \square$$

Lemma 19.2.3. The output C has $|C| > 4L \cdot \sum_j x_j$ with probability at most $1/4$.

Proof. In each phase, set j is added to C with probability x_j, so the expected number of sets added is $\sum_j x_j$. The expected number of edges added in *all* iterations is at most L times larger. That is, $\mathrm{E}[|C|] \le L \cdot \sum_j x_j$. It follows that

$$\begin{aligned}
\Pr\left[|C| \ge 4L \cdot \textstyle\sum_j x_j\right] &\le \frac{\mathrm{E}[|C|]}{4L \cdot \sum_j x_j} \qquad \text{(by Markov's inequality, Fact A.3.16)} \\
&\le \frac{L \cdot \sum_j x_j}{4L \cdot \sum_j x_j} = \frac{1}{4}.
\end{aligned}$$ □

We conclude with the following theorem.

Theorem 19.2.4. The output C covers all elements and has size at most $4L \cdot \sum_j x_j$ with probability at least $1/2$.

References: (Shmoys and Shmoys, 2010, Section 1.7).

Proof. By Lemma 19.2.1, C fails to cover all elements with probability at most $1/4$. By Lemma 19.2.3, C has size exceeding $4L \cdot \sum_j x_j$ with probability at most $1/4$. By a union bound, the probability that either of these occurs is at most $1/2$. □

Corollary 19.2.5. There is a randomized, polynomial-time algorithm that gives a $4\ln(4n)$-approximation to the Set Cover problem.

Proof. Compute an optimum solution x to the linear program; it satisfies $\sum_j x_j = \mathrm{LPOPT}$. This can be done in polynomial time, for example by interior point methods. By Theorem 19.2.4, the rounding algorithm above has probability $1/2$ of producing a Set Cover solution C with

$$|C| \le 4L \cdot \sum_j x_j = 4L \cdot LPOPT \le 4L \cdot OPT. \quad \square$$

Keener Kwestion 19.2.6. By adjusting the parameters, show that the integrality gap is at most $(1+2\epsilon)\ln(n/\epsilon)$ for any $\epsilon > 0$.

Part II

Concentration

Chapter 20

Concentration Bounds, with details

20.1 Chernoff bound, in detail

The Chernoff bound was presented in a simplified form in Theorem 9.2.1. Let us now present it in a more elaborate form.

Let $X_1, \dots, X_n$ be independent random variables such that X_i always lies in the interval $[0, 1]$. Define $X = \sum_{i=1}^{n} X_i$. The expectation $\mathrm{E}[X]$ need not be exactly known, but we assume that $\check{\mu} \le \mathrm{E}[X] \le \hat{\mu}$. If it is exactly known, we may define $\check{\mu} = \hat{\mu} = \mathrm{E}[X]$.

Theorem 20.1.1. For all $\delta > 0$,

$$\text{Left tail:}\quad \Pr[X \ge (1+\delta)\hat{\mu}] \overset{(a)}{\le} \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\hat{\mu}} \overset{(b)}{\le} \begin{cases} e^{-\delta^2\hat{\mu}/3} & (\text{if } \delta \le 1) \quad \text{“Gaussian tail”} \\ e^{-(1+\delta)\ln(1+\delta)\hat{\mu}/4} & (\text{if } \delta \ge 1) \quad \text{“Poisson tail”} \\ e^{-\delta\hat{\mu}/3} & (\text{if } \delta \ge 1) \quad \text{“Exponential tail”} \end{cases}$$

$$\text{Right tail:}\quad \Pr[X \le (1-\delta)\check{\mu}] \overset{(c)}{\le} \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\check{\mu}} \overset{(d)}{\le} e^{-\delta^2\check{\mu}/2} \quad \text{“Gaussian tail”}.$$

Inequalities (c) and (d) are only valid for $\delta < 1$, but $\Pr[X \le (1-\delta)\check{\mu}] = 0$ if $\delta > 1$.

References: (McDiarmid, 1998, Theorem 2.3), (Lehman et al., 2018, Theorem 20.5.1), (Motwani and Raghavan, 1995, Section 4.1), (Mitzenmacher and Upfal, 2005, equations (4.2) and (4.5)), (Klenke, 2008, Exercise 5.2.1), Wikipedia.

The tails have a qualitative difference in their dependence on δ.

- The “Gaussian tails” depend on δ^2. This resembles a Gaussian distribution (see Appendix B.5.2), whose tails look like $\exp(-\delta^2/2)$ (see Fact B.5.6).
- The “Poisson tail” depends on $\delta \ln \delta$. This resembles a Poisson distribution, whose mass function function looks like[1] $1/\delta! \approx \exp(-\delta \ln \delta)$.
- The “Exponential tail” depends on δ. This resembles the exponential distribution, whose tail looks like $e^{-\delta}$. This is weaker, but more convenient, than the “Poisson tail”.

[1] See, e.g., (Vershynin, 2018, Theorem 1.3.4) or these lecture notes.

20.2 Proofs for Chernoff Bound

20.2.1 Proof of inequality (a)

The Chernoff bounds would not be true without the assumption that $X_1, \ldots, X_n$ are independent. What special properties do independent random variables have? One basic property is that

$$\mathrm{E}[A \cdot B] \;=\; \mathrm{E}[A] \cdot \mathrm{E}[B] \tag{20.2.1}$$

for any independent random variables A and B.

References: (Lehman et al., 2018, Theorem 19.5.6), (Cormen et al., 2001, Exercise C.3-5), (Motwani and Raghavan, 1995, Proposition C.6), (Mitzenmacher and Upfal, 2005, Theorem 3.3), (Grimmett and Stirzaker, 2001, Lemma 3.3.9), (Durrett, 2019, Theorem 2.1.13), (Klenke, 2008, Theorem 5.4).

But the Chernoff bound has nothing to do with *products* of random variables, it is about *sums* of random variables. So one trick we could try is to convert sums into products using the exponential function. Fix some parameter $\theta > 0$ whose value we will choose later. We will look at the random variables

$$\begin{aligned} & \exp(\theta X_i) \qquad \forall i \in [n] \\ \text{and} \qquad & \exp(\theta X) \;=\; \exp\Big(\theta \sum_{i=1}^{n} X_i\Big) \;=\; \prod_{i=1}^{n} \exp(\theta X_i) \end{aligned}$$

Since $X_1, \ldots, X_n$ are independent, it follows[2] that $e^{\theta X_1}, \ldots, e^{\theta X_n}$ are also independent. Therefore, using (20.2.1) repeatedly,

$$\mathrm{E}\left[\, e^{\theta X} \,\right] \;=\; \prod_{i=1}^{n} \mathrm{E}\left[\, e^{\theta X_i} \,\right]. \tag{20.2.2}$$

So far this all seems promising. We want to prove that X is small, which is equivalent to proving that $e^{\theta X}$ is small. Using (20.2.2), we can do this by showing that each $\mathrm{E}\left[\, e^{\theta X_i} \,\right]$ is small. Perhaps we can somehow show $\mathrm{E}\left[\, e^{\theta X_i} \,\right]$ is small by comparing it to $\mathrm{E}[\, X_i \,]$?

If we were forgetting the rules of probability, we might be tempted to say that $\mathrm{E}\left[\, e^{\theta X_i} \,\right]$ equals $e^{\theta\, \mathrm{E}[\, X_i \,]}$, but that is false. We might remember one useful probability trick called Jensen's inequality[3] that says $f(\mathrm{E}[\, A \,]) \le \mathrm{E}[\, f(A) \,]$ for any random variable A and any convex function f. Applying this with $f(x) = e^{\theta x}$, we see that

$$e^{\theta\, \mathrm{E}[\, X_i \,]} \;\le\; \mathrm{E}\left[\, e^{\theta X_i} \,\right]. \tag{20.2.3}$$

So we get a *lower bound* on $\mathrm{E}\left[\, e^{\theta X_i} \,\right]$ in terms of $\mathrm{E}[\, X_i \,]$, but we actually wanted an *upper bound*.

Claim 20.2.2 gives the desired upper bound; it shows that the inequality in (20.2.3) can almost be reversed. The proof is easy once we have the following convexity fact.

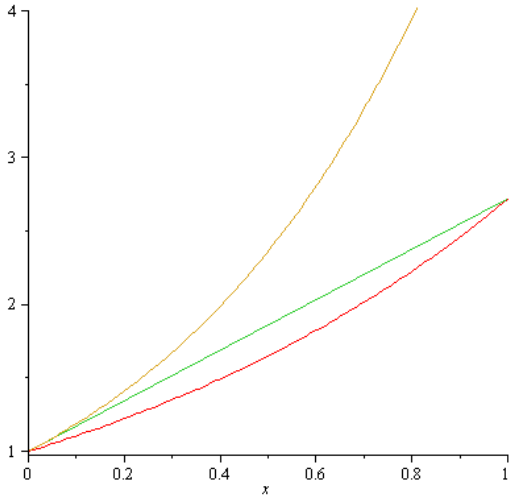
Claim 20.2.1. For all $\theta \in \mathbb{R}$ and all $x \in [0, 1]$,

$$\exp(\theta x) \;\le\; 1 + (e^{\theta} - 1)x \;\le\; \exp\big((e^{\theta} - 1)x\big).$$

The inequalities are illustrated by this plot.

[2] See (Lehman et al., 2018, Lemma 19.2.2), (Cormen et al., 2001, Equation (C.24)), (Grimmett and Stirzaker, 2001, Theorem 3.2.3), (Klenke, 2008, Remark 2.15(iii)).

[3] See (Cormen et al., 2001, Equation (C.26)), (Mitzenmacher and Upfal, 2005, Theorem 2.4), (Vershynin, 2018, Section 1.2), (Grimmett and Stirzaker, 2001, Exercise 5.6.1), (Durrett, 2019, Theorem 1.6.2), (Klenke, 2008, Theorem 7.9).



Proof of Claim 20.2.1. Consider the first inequality $e^{\theta x} \le 1 + (e^\theta - 1)x$ for all $x \in [0,1]$. This follows from Fact B.4.2 by setting $c = e^\theta$. The second inequality $1 + (e^\theta - 1)x \le \exp((e^\theta - 1)x)$ follows from the familiar Fact A.2.5. □

Claim 20.2.2. Let $\theta \in \mathbb{R}$ be arbitrary. Then $\mathrm{E}\left[\, e^{\theta X_i} \,\right] \le \exp\left((e^\theta - 1)\,\mathrm{E}\left[\, X_i \,\right]\right)$.

Proof. The main idea is as follows. Although we cannot "pull the expectation inside the exponential", we can use Claim 20.2.1 to approximate the exponential by a linear function, then "pull the expectation inside" via linearity of expectation, then finally switch back to an exponential function.

The formal argument is

$$\mathrm{E}\left[\, e^{\theta X_i} \,\right] \;\le\; \mathrm{E}\left[\, 1 + (e^\theta - 1)X_i \,\right] \;=\; 1 + (e^\theta - 1)\,\mathrm{E}\left[\, X_i \,\right] \;\le\; \exp((e^\theta - 1)\,\mathrm{E}\left[\, X_i \,\right]),$$

where both inequalities follow from Claim 20.2.1. □

Now we are ready to prove the inequality (a) of the Chernoff bound.

$$\begin{aligned}
\Pr\left[\, X \ge (1+\delta)\hat{\mu} \,\right] &= \Pr\left[\, \exp(\theta X) \ge \exp\left(\theta(1+\delta)\hat{\mu}\right) \,\right] && \text{(by monotonicity)} \\
&\le \frac{\mathrm{E}\left[\, \exp(\theta X) \,\right]}{\exp(\theta(1+\delta)\hat{\mu})} && \text{(by Markov's inequality)} \\
&= \frac{\prod_{i=1}^n \mathrm{E}\left[\, e^{\theta X_i} \,\right]}{\exp(\theta(1-\delta)\check{\mu})} && \text{(by (20.2.2))} \\
&\le \frac{\prod_{i=1}^n \exp((e^\theta - 1)\,\mathrm{E}\left[\, X_i \,\right])}{\exp(\theta(1+\delta)\hat{\mu})} && \text{(by Claim 20.2.2).}
\end{aligned}$$

Gathering everything inside one exponential we get

$$\Pr\left[\, X \ge (1+\delta)\hat{\mu} \,\right] \;\le\; \exp\left((e^\theta - 1)\sum_i \mathrm{E}\left[\, X_i \,\right] - \theta(1+\delta)\hat{\mu}\right).$$

Finally, substituting $\theta = \ln(1+\delta)$ and using $\sum_i \mathrm{E}\left[\, X_i \,\right] = \mathrm{E}\left[\, X \,\right] \le \hat{\mu}$ proves inequality (a).

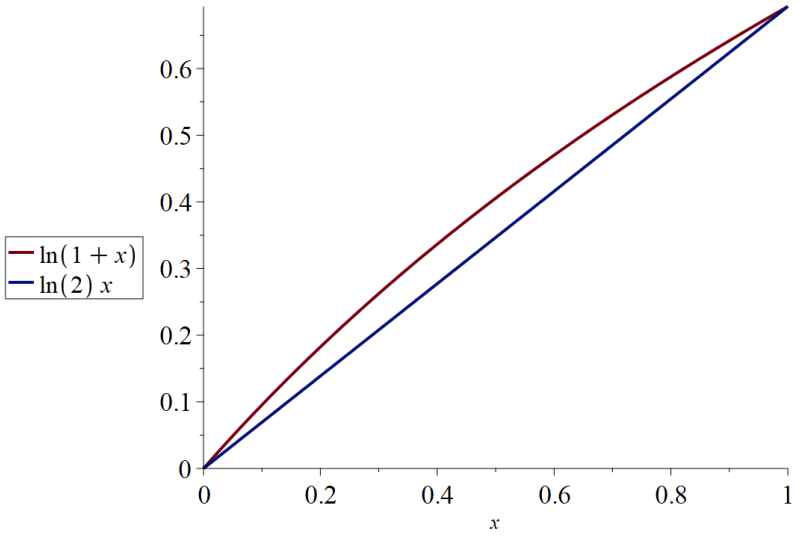
20.2.2 Proof of inequality (b), when $\delta \in [0,1]$

Claim 20.2.3. Then $(1+x)\ln(1+x) - x \geq \frac{\ln(2)}{2} \cdot x^2$ for all $x \in [0,1]$.

Proof. Note that the LHS and RHS both vanish at $x = 0$. So the claim holds if the derivative of the LHS is at least the derivative of the RHS on the interval $[0,1]$. By simple calculus,

$$\frac{d}{dx}\big[(1+x)\ln(1+x) - x\big] = \ln(1+x) \qquad \text{and} \qquad \frac{d}{dx}\frac{\ln(2)}{2}x^2 = \ln(2)x.$$

These derivatives are illustrated in the following figure.



They agree when $x = 0$ (both equal zero) and also agree when $x = 1$ (both equal $\ln 2$). Thus we have $\ln(1+x) \geq \ln(2)x$ for all $x \in [0,1]$, since the LHS is concave and the RHS is linear. □

Inequality (b) now follows straightforwardly because

$$\begin{aligned}
\frac{e^{\delta}}{(1+\delta)^{1+\delta}} &= \exp\Big(-\big((1+\delta)\ln(1+\delta) - \delta\big)\Big) \\
&\leq \exp\Big(-\frac{\ln 2}{2}\delta^2\Big) \qquad \text{(by Claim 20.2.3)} \\
&\leq e^{-\delta^2/3},
\end{aligned}$$

since $\ln(2) > 0.69 > 2/3$.

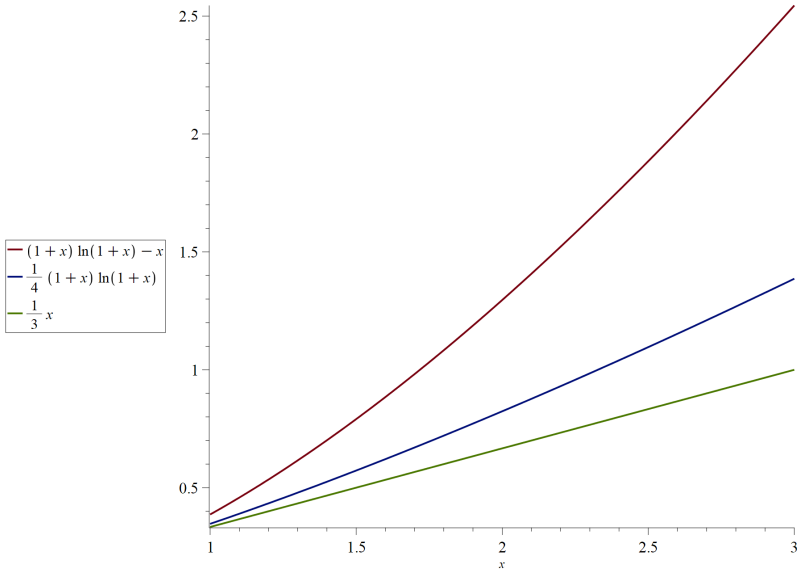
20.2.3 Proof of inequality (b), $\delta \geq 1$

Claim 20.2.4. Define

$$\begin{aligned}
f(x) &= (1+x)\ln(1+x) - x \\
g(x) &= (1+x)\ln(1+x)/4 \\
h(x) &= x/3
\end{aligned}$$

Then $f(x) > g(x) > h(x)$ for $x \geq 1$.

This claim is illustrated by the following plot.



Proof. First we observe that the claimed inequality holds at the point $x = 1$. We have

$$\begin{aligned} f(1) &= 2\ln(2) - 1 > 0.38 \\ g(1) &= \ln(2)/4 \approx 0.346 \\ h(1) &= 1/3 < 0.34. \end{aligned}$$

We will show that their derivatives satisfy

$$\frac{d}{dx}f(x) \overset{(1)}{\geq} \frac{d}{dx}g(x) \overset{(2)}{\geq} \frac{d}{dx}h(x) \qquad \forall x \geq 1,$$

from which the claim follows by integration. These derivatives have the following expressions.

$$\frac{d}{dx}f(x) = \ln(1+x) \qquad \frac{d}{dx}g(x) = (1+\ln(1+x))/4 \qquad \text{and} \qquad \frac{d}{dx}h(x) = 1/3.$$

Inequality (2) is straightforward. For $x \geq 1$, we have $(1+\ln(1+x))/4 \geq (1+\ln(2))/4 \approx 0.423$, which is greater than $1/3$.

For inequality (1), first observe that $y \geq (1+y)/4$ for $y \geq 1/3$. Substituting $y \leftarrow \ln(1+x)$ gives $\ln(1+x) \geq (1+\ln(1+x))/4$ for $x \geq e^{1/3} - 1 \approx 0.395$. This implies (1). □

We can now show inequality (b) of Theorem 20.1.1 in the case $\delta \geq 1$. Using the notation of Claim 20.2.4,

$$\begin{aligned} \frac{e^\delta}{(1+\delta)^{1+\delta}} &= \exp\Big(-\big((1+\delta)\ln(1+\delta) - \delta\big)\Big) \\ &= \exp\big(-f(\delta)\big) < \exp\big(-g(\delta)\big) < \exp\big(-h(\delta)\big). \end{aligned}$$

Substitituting the definition of g and h, we obtain

$$\frac{e^\delta}{(1+\delta)^{1+\delta}} < \exp\big(-(1+\delta)\ln(1+\delta)\hat{\mu}/4\big) < \exp\big(-\delta\hat{\mu}/3\big).$$

20.2.4 Proof of inequality (c)

The argument is very similar to the proof of inequality (a). We will choose θ to be negative.

$$\begin{aligned}
\Pr[\,X \le (1-\delta)\check{\mu}\,] &= \Pr\left[\,\exp(\theta X) \ge \exp\left(\theta(1-\delta)\check{\mu}\right)\,\right] && \text{(by monotonicity and } \theta < 0) \\
&\le \frac{\mathrm{E}\,[\,\exp(\theta X)\,]}{\exp(\theta(1-\delta)\check{\mu})} && \text{(by Markov's inequality)} \\
&\le \frac{\prod_{i=1}^{n} \exp((e^{\theta}-1)\,\mathrm{E}\,[\,X_i\,])}{\exp(\theta(1-\delta)\check{\mu})} && \text{(by Claim 20.2.2)} \\
&= \exp\left((e^{\theta}-1)\sum_{i=1}^{n}\mathrm{E}\,[\,X_i\,] - \theta(1-\delta)\check{\mu}\right) \\
&\le \exp\left((e^{\theta}-1)\check{\mu} - \theta(1-\delta)\check{\mu}\right).
\end{aligned}$$

This last inequality holds because $\theta < 0$ so $e^{\theta} - 1 < 0$, and because $\check{\mu} \le \sum_{i=1}^{n} \mathrm{E}\,[\,X_i\,]$. Plugging in $\theta = \ln(1-\delta)$, which is negative, we obtain

$$\Pr[\,X \le (1-\delta)\check{\mu}\,] \;=\; \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\check{\mu}},$$

which is inequality (c).

20.2.5 Proof of inequality (d)

The statement and the proof are similar to Claim 20.2.3.

Claim 20.2.5. Then $(1-x)\ln(1-x) + x \ge \frac{1}{2}\cdot x^2$ for all $x \in [0,1)$.

Proof. Note that the LHS and RHS both vanish at $x = 0$. So the claim holds if the derivative of the LHS is at least the derivative of the RHS on the interval $[0,1)$. By simple calculus,

$$\frac{d}{dx}\big[(1-x)\ln(1-x) + x\big] = -\ln(1-x) \qquad \text{and} \qquad \frac{d}{dx}x^2/2 = x.$$

The linear approximation of $-\ln(1-x)$ at $x = 0$ is

$$x \cdot \tfrac{d}{dx}\big(-\ln(1-x)\big)\Big|_{x=0} = x \cdot \big(\tfrac{1}{1-x}\big)\Big|_{x=0} = x.$$

Furthermore, $-\ln(1-x)$ is convex on $[0,1)$ because its second derivative is $1/(1-x)^2 \ge 0$. Thus $-\ln(1-x) \ge x$ on $[0,1)$. □

Inequality (d) now follows straightforwardly because, using Claim 20.2.5,

$$\frac{e^{-\delta}}{(1-\delta)^{1-\delta}} \;=\; \exp\Big(-\big((1-\delta)\ln(1-\delta)+\delta\big)\Big) \;\le\; \exp\Big(-\delta^2/2\Big).$$

20.3 Proof of the Hoeffding Bound

The Hoeffding Bound was introduced in Section 9.3. We will prove the following slightly weaker result.

Theorem 20.3.1. Let $X_1, \ldots, X_n$ be independent random variables such that X_i always lies in the interval $[0,1]$. Define $X = \sum_{i=1}^{n} X_i$. Then

$$\Pr[\,|X - \mathrm{E}\,[\,X\,]| \ge t\,] \;\le\; 2\exp(-t^2/2n) \qquad \forall t \ge 0.$$

Simplifications. First of all, we will "center" the random variables, which cleans up the inequality by eliminating the expectation. Define $\hat{X}_i = X_i - \mathrm{E}[X_i]$ and $\hat{X} = \sum_{i=1}^n \hat{X}_i$. Note that[4] $\hat{X}_i \in [-1, 1]$. Our main argument is to prove that

$$\Pr\left[\hat{X} \geq t\right] \leq \exp(-t^2/2n). \tag{20.3.1}$$

The same argument also applies to $-\hat{X}$, so we get that

$$\Pr\left[-\hat{X} \geq t\right] = \Pr\left[\hat{X} \leq -t\right] \leq \exp(-t^2/2n).$$

Combining them with a union bound, we get

$$\Pr[\,|X - \mathrm{E}[X]| \geq t\,] = \Pr\left[|\hat{X}| \geq t\right] \leq \Pr\left[\hat{X} \geq t\right] + \Pr\left[-\hat{X} \geq t\right] \leq 2\exp(-t^2/2n).$$

This proves the theorem (with the weaker exponent).

Proof of (20.3.1). As in the proof of the Chernoff Bound (see (20.2.1)), we will use the fact that $\mathrm{E}[A \cdot B] = \mathrm{E}[A] \cdot \mathrm{E}[B]$ for any independent random variables A and B.

Key Idea #1: As in the proof of the Chernoff Bound, we will convert sums into products using the exponential function. Fix some parameter $\lambda > 0$ whose value we will choose later. Define

$$\begin{aligned} Y_i &= \exp(\lambda \hat{X}_i) \\ Y &= \exp(\lambda \hat{X}) = \exp\left(\lambda \sum_{i=1}^n \hat{X}_i\right) = \prod_{i=1}^n \exp(\lambda \hat{X}_i) = \prod_{i=1}^n Y_i. \end{aligned}$$

It is easy to check that, since $\{X_1, \ldots, X_n\}$ is mutually independent, so is $\left\{\hat{X}_1, \ldots, \hat{X}_n\right\}$ and $\{Y_1, \ldots, Y_n\}$. Therefore, by (**??**),

$$\mathrm{E}[Y] = \prod_{i=1}^n \mathrm{E}[Y_i]. \tag{20.3.2}$$

So far this all seems quite good. We want to prove that $\hat{X}$ is small, which is equivalent to proving Y is small. Using (20.3.2), we can do this by showing that the $\mathrm{E}[Y_i]$ terms are small. Doing so involves an extremely useful tool.

Key Idea #2: The second main idea is a clever trick to bound terms of the form $\mathrm{E}[\exp(\lambda A)]$, where A is a mean-zero random variable. We discuss this idea in more detail in the next subsection. We will use[5] Claim 20.3.2 to show

$$\mathrm{E}[Y_i] = \mathrm{E}\left[\exp(\lambda \hat{X}_i)\right] \leq \exp(\lambda^2/2). \tag{20.3.3}$$

Thus, combining this with (20.3.2),

$$\mathrm{E}[Y] \leq \prod_{i=1}^n \exp(\lambda^2/2) = \exp(\lambda^2 n/2). \tag{20.3.4}$$

[4] This step is where the argument is not careful enough to obtain the optimal exponent: $\hat{X}_i$ is actually supported on an interval of length 1, although our argument only assumes that it is supported on an interval of length 2.

[5] If we were more careful here and instead used Lemma 20.3.4, we could improve the constant in the exponent in (20.3.3) from 1/2 to 1/8. This would improve the constant in the exponent in (20.3.1) from 1/2 to 2.

Now we are ready to prove Hoeffding's inequality:

$$\begin{aligned}
\Pr\left[\hat{X} \geq t\right] &= \Pr\left[\exp(\lambda\hat{X}) \geq \exp\left(\lambda t\right)\right] \quad \text{(by monotonicity of } e^x\text{)} \\
&\leq \frac{\mathrm{E}\left[\exp(\lambda\hat{X})\right]}{\exp(\lambda t)} \quad \text{(by Markov's inequality (Theorem ??))} \\
&= \mathrm{E}\left[Y\right] \cdot \exp(-\lambda t) \\
&\leq \exp(\lambda^2 n/2 - \lambda t) \quad \text{(by (20.3.4))} \\
&= \exp(-t^2/2n),
\end{aligned}$$

by optimizing to get $\lambda = t/n$. □

20.3.1 Exponentiated Mean-Zero RVs

The second main idea of Hoeffding's inequality is the following claim.

Claim 20.3.2. Let A be a random variable such that $|A| \leq 1$ with probability 1 and $\mathrm{E}\left[A\right] = 0$. Then for any $\lambda > 0$, we have $\mathrm{E}\left[\exp(\lambda A)\right] \leq \exp(\lambda^2/2)$.

Intuitively, the expectation should be maximized by the random variable A that is uniform on $\{-1, +1\}$. In this case,

$$\mathrm{E}\left[\exp(\lambda A)\right] = \frac{1}{2}e^{\lambda} - \frac{1}{2}e^{-\lambda} \leq e^{\lambda^2/2}.$$

This inequality is a nice bound on the hyperbolic cosine function (Claim 20.3.3). The full proof of Claim 20.3.2 basically reduces to the case of $A \in \{-1, 1\}$ using convexity of e^x.

Proof. Define $p = (1 + A)/2$ and $q = (1 - A)/2$. Observe that $p, q \geq 0$, $p + q = 1$, and $p - q = A$. By convexity,

$$\exp(\lambda A) = \exp\left(\lambda(p - q)\right) = \exp\left(\lambda p + (-\lambda)q\right) \leq p \cdot \exp(\lambda) + q \cdot \exp(-\lambda) = \frac{e^{\lambda} + e^{-\lambda}}{2} + \frac{A}{2}(e^{\lambda} - e^{-\lambda}).$$

Thus,

$$\mathrm{E}\left[\exp(\lambda A)\right] \leq \mathrm{E}\left[\frac{e^{\lambda} + e^{-\lambda}}{2} + \frac{A}{2}(e^{\lambda} - e^{-\lambda})\right] = \frac{e^{\lambda} + e^{-\lambda}}{2},$$

since $\mathrm{E}\left[A\right] = 0$. This last quantity is bounded by the following technical claim. □

Claim 20.3.3 (Approximation of Cosh). For any real x, we have $(e^x + e^{-x})/2 \leq \exp(x^2/2)$.

References: A more general result can be found in Alon & Spencer Lemma A.1.5.

Proof. First observe that the product of all the even numbers at most $2n$ does not exceed the product of all numbers at most $2n$. In symbols,

$$2^n(n!) = \prod_{i=1}^{n}(2i) \leq \prod_{i=1}^{2n} i = (2n)!$$

Now to bound $(e^x + e^{-x})/2$, we write it as a Taylor series and observe that the odd terms cancel.

$$\frac{e^x + e^{-x}}{2} = \sum_{n\geq 0}\frac{x^n}{n!} + \sum_{n\geq 0}\frac{(-x)^n}{n!} = \sum_{n\geq 0}\frac{x^{2n}}{(2n)!} \leq \sum_{n\geq 0}\frac{x^{2n}}{2^n(n!)} = \sum_{n\geq 0}\frac{(x^2/2)^n}{n!} = \exp(x^2/2).$$

□

A common scenario is that A is mean-zero, but lies in an "asymmetric" interval $[a, b]$, where $a < 0 < b$. A slightly tighter version of these MGF bounds can be derived for this scenario.

Lemma 20.3.4 (Hoeffding's Lemma). Let A be a random variable such that $A \in [a, b]$ with probability 1 and $\mathrm{E}[A] = 0$. Then for any $\lambda > 0$, we have $\mathrm{E}[\exp(\lambda A)] \leq \exp\left(\lambda^2 (b-a)^2/8\right)$.

References: Wikipedia, (Shalev-Shwartz and Ben-David, 2014, Lemma B.7).

The proof uses ideas similar to the proof of Claim 20.3.2, except we cannot use Claim 20.3.3 and must instead use an ad-hoc calculus argument.

Chapter 21

More Applications of Concentration

21.1 Balls and Bins: The Heaviest Bin

Let us return to the topic of balls and bins, which was first introduced in Chapter 6. Consider throwing n balls into n bins, uniformly and independently. Let B_i be the number of balls in bin i. In Section 6.5 we used an ad hoc argument to analyze the load on the heaviest bin, namely $\max_i B_i$. Now we will give an alternative analysis using the Chernoff bound.

Theorem 21.1.1. Assume $n \geq 3$. Define $k = 16 \ln n / \ln \ln n$. With probability at least $1 - 1/n$, the heaviest bin has at most k balls. That is,

$$\Pr\left[\max_i B_i \leq k\right] \;\geq\; 1 - 1/n.$$

This theorem is optimal up to constant factors. It is known that $\max_i B_i \geq \ln n / \ln \ln n$ with probability at least $1 - 1/n$. See, e.g., (Mitzenmacher and Upfal, 2005, Lemma 5.12).

Proof. The first step is to give tail bounds on B_1. We decompose B_1 into indicator random variables as

$$B_1 \;=\; X_1 + X_2 + \cdots + X_n,$$

where X_j is 1 if the j^{th} ball lands in the first bin. For each $j \in [n]$ we have $\mathrm{E}[X_j] = 1/n$, so $\mathrm{E}[B_1] = \sum_j \mathrm{E}[X_j] = 1$. What is the probability that this first bin has more than k balls?

We will analyze this event using inequality (b) of the Chernoff bound. Specifically, letting $X = B_1$ and $\delta = k - 1$, we obtain

$$\begin{aligned}
\Pr[B_1 \geq k] &= \Pr[B_1 \geq k\,\mathrm{E}[B_1]] && \text{(since } \mathrm{E}[B_1] = 1\text{)} \\
&\leq \exp\big(-\mathrm{E}[B_1] \cdot k \ln(k)/4\big) && \text{(by the Chernoff bound)} \\
&= \exp\big(-k \ln(k)/4\big) && \text{(since } \mathrm{E}[B_1] = 1\text{).}
\end{aligned}$$

These inequalities require that $\delta \geq 1$, i.e., $k \geq 2$.

To proceed any further, we must understand $k \ln k$. A quick calculation gives

$$\begin{aligned} k \cdot \ln k &= 16\frac{\ln n}{\ln \ln n} \cdot \ln\left(\frac{16 \ln n}{\ln \ln n}\right) \\ &> 16\frac{\ln n}{\ln \ln n} \cdot \ln\left(\frac{\ln n}{\sqrt{\ln n}}\right) \qquad (\text{since } \sqrt{x} > \ln x \text{ for all } x > 0) \\ &= 16\frac{\ln n}{\ln \ln n} \cdot \ln\left(\sqrt{\ln n}\right) \\ &= 8\frac{\ln n}{\ln \ln n} \cdot \ln \ln n \;=\; 8 \ln n. \end{aligned}$$

Plugging that in,

$$\Pr[B_1 \geq k] \;\leq\; \exp\big(-k \ln(k)/4\big) \;\leq\; \exp\big(-2 \ln n\big) \;=\; n^{-2}. \tag{21.1.1}$$

So bin 1 is unlikely to have more than α balls.

Here we have analyzed bin 1, but the bins are all equivalent so the same analysis actually holds for all bins. The remainder of the argument is just the union bound.

$$\begin{aligned} \Pr[\,\textit{any} \text{ bin has} \geq k \text{ balls}\,] &= \Pr[B_1 \geq k \;\vee\; B_2 \geq k \;\vee\; \cdots \;\vee\; B_n \geq k] \\ &\leq \sum_{i=1}^{n} \Pr[B_i \geq k] && (\text{Fact A.3.5}) \\ &\leq \sum_{i=1}^{n} \frac{1}{n^2} && (\text{by (21.1.1)}) \\ &= \frac{1}{n}. \end{aligned}$$

Thus, with probability at least $1 - 1/n$, all bins have at most k balls. □

21.2 Congestion Minimization

One of the classically important areas in algorithm design and combinatorial optimization is *network flows*. A central problem in that area is the maximum flow problem. We now look at a generalization of this problem.

An instance of the problem consists of a directed graph $G = (V, A)$ and a sequence $(s_1, t_1), \ldots, (s_k, t_k)$ of pairs of vertices. Let $n = |V|$. (It is not crucial that the graph be directed; the problem is equally interesting in undirected graphs. However in network flow problems it is often more convenient to look at directed graphs. Feel free to think about whatever variant you find easier.)

A natural question to ask is: do there exist paths P_i from s_i to t_i for every i such that these paths share no arcs? This is called the **edge-disjoint paths** problem. Quite remarkably, it is NP-hard even in the case $k = 2$, assuming the graph is directed. For undirected graphs, it is polynomial time solvable if k is a fixed constant, but NP-hard if k is a sufficiently large function of n.

We will look at a variant of this problem called the **congestion minimization** problem. The idea is to allow each arc to be used in multiple paths, but not too many. The number of paths using a given arc is the "congestion" of that arc. We say that a solution has congestion C if it is a collection of paths P_i from s_i to t_i, where each arc is contained in at most C of the paths. The problem is to find the

minimum value of C such that there is a solution of congestion C. This problem is still NP-hard, since determining if $C = 1$ is the edge-disjoint paths problem.

We will look at the congestion minimization problem from the point of view of approximation algorithms. Let OPT be the minimum congestion of any solution. We would like to give an algorithm which can produce a solution with congestion at most $\alpha \cdot OPT$ for some $\alpha \geq 1$. This factor α is the called the approximation factor of the algorithm.

Theorem 21.2.1. There is an algorithm for the congestion minimization problem with approximation factor $O(\log n / \log \log n)$.

To design such an algorithm we will use linear programming. We write down an integer program (IP) which captures the problem exactly, relax that to a linear program (LP), then design a method for "rounding" solutions of the LP into solutions for the IP.

The Integer Program. Writing an IP formulation of an optimization problem is usually quite simple. That is indeed true for the congestion minimization problem. However, we will use an IP which you might find rather odd: our IP will have exponentially many variables. This will simplify our explanation of the rounding.

Let $\mathcal{P}_i$ be the set of *all* paths in G from s_i to t_i. (Note that $|\mathcal{P}_i|$ may be exponential in n.) For every path $P \in \mathcal{P}_i$, we create a variable x^i_P. This variable will take values only in $\{0, 1\}$, and setting it to 1 corresponds to including the path P in our solution.

The integer program is as follows

$$\begin{array}{lll}
\min & C & \\
\text{s.t.} & \displaystyle\sum_{P \in \mathcal{P}_i} x^i_P = 1 & \forall i = 1, \ldots, k \\
& \displaystyle\sum_i \sum_{P \in \mathcal{P}_i \text{ with } a \in P} x^i_P \leq C & \forall a \in A \\
& x^i_P \in \{0, 1\} & \forall i = 1, \ldots, k \text{ and } P \in \mathcal{P}_i
\end{array}$$

The last constraint says that we must decide for every path whether or not to include it in the solution. The first constraint says that the solution must choose exactly one path between each pair s_i and t_i. The second constraint ensures that the number of paths using each arc is at most C. The optimization objective is to find the minimum value of C such that a solution exists.

Every solution to the IP corresponds to a solution for the congestion minimization problem with congestion C, and vice-versa. Thus the optimum value of the IP is OPT, which we previously defined to be the minimum congestion of any solution to the original problem.

This IP is NP-hard to solve, so we relax it into a linear program, by replacing the integrality constraints with non-negativity constraints. It turns out to be convenient also to add the constraint $C \geq 1$. The resulting linear program is:

$$\begin{array}{lll}
\min & C & \\
\text{s.t.} & \displaystyle\sum_{P \in \mathcal{P}_i} x^i_P = 1 & \forall i = 1, \ldots, k \\
& \displaystyle\sum_i \sum_{P \in \mathcal{P}_i \text{ with } a \in P} x^i_P \leq C & \forall a \in A \\
& C \geq 1 & \\
& x^i_P \geq 0 & \forall i = 1, \ldots, k \text{ and } P \in \mathcal{P}_i
\end{array}$$

Remarkably, this LP can be solved in time polynomial in n (the number of nodes of G), even though its

number of variables could be exponential in n. The details are best left for a course on optimization[1]. Our algorithm will solve this LP and obtain a solution where the number of non-zero x_P^i variables is only polynomial in n. Let C^* be the optimum value of the LP.

Claim 21.2.2. $C^* \leq OPT$.

Proof. The LP was obtained from the IP by *removing* constraints. Therefore any feasible solution for the IP is also feasible for the LP. In particular, the optimal solution for the IP is feasible for the LP. So the LP has a solution with objective value equal to OPT. □

The Rounding. Our algorithm will solve the LP and most likely obtain a "fractional" solution — a solution with some non-integral variables, which is therefore not feasible for the IP. The next step of the algorithm is to "round" that fractional solution into a solution which is is feasible for the IP. In doing so, the congestion might increase, but we will ensure that it does not increase too much.

The technique we will use is called **randomized rounding**. For each each $i = 1, \ldots, k$, we randomly choose *exactly one path* P_i by setting $P_i = P$ with probability x_P^i. (The LP's constraints ensure that these are indeed probabilities: they are non-negative and sum up to 1.) The algorithm outputs the chosen paths $P_1, \ldots, P_k$.

Analysis. All that remains is to analyze the congestion of these paths. Let Y_i^a be the indicator random variable that is 1 if $a \in P_i$ and 0 otherwise. Let $Y^a = \sum_i Y_i^a$ be the congestion on arc a. The expected value of Y^a is easy to analyze:

$$\mathrm{E}[\,Y^a\,] \;=\; \sum_i \mathrm{E}[\,Y_i^a\,] \;=\; \sum_i \sum_{P \in \mathcal{P}_i \text{ with } a \in P} x_P^i \;\leq\; C^*,$$

where the inequality comes from the LP's second constraint. (Recall we assume that the fractional solution is optimal for the LP, and therefore $C = C^*$.)

The Chernoff bound says, if X is a sum of independent random variables each of which take values in $[0, 1]$, and μ is an *upper bound* on $\mathrm{E}[\,X\,]$, then

$$\Pr[\,X \geq (1+\delta)\mu\,] \;\leq\; \exp\Big(-\mu \cdot \big((1+\delta)\ln(1+\delta) - \delta\big)\Big) \qquad \forall \delta > 0.$$

We apply this to Y^a, taking $\mu = C^*$ and $\alpha = 1 + \delta = 6 \log n / \log\log n$. Following our balls-and-bins argument from last time,

$$\begin{aligned} \Pr[\,Y^a \geq \alpha C^*\,] \quad &\leq \quad \exp\Big(-C^*\big(\alpha \ln \alpha - (\alpha - 1)\big)\Big) \\ &\leq \quad \exp\big(-\alpha \ln \alpha + \alpha - 1\big) \\ &\leq \quad \exp\big(-(6/2)\ln n\big) \;=\; 1/n^3. \end{aligned}$$

We now use a union bound to analyze the probability of *any* arc having congestion greater than αC^*.

$$\Pr[\,\text{any } a \text{ has } Y^a \geq \alpha C^*\,] \;\leq\; \sum_{a \in A} \Pr[\,Y^a \geq \alpha C^*\,] \;\leq\; \sum_{a \in A} 1/n^3 \;\leq\; 1/n,$$

[1] There are two ways to do this. The first way is to solve the dual LP using the ellipsoid method. This can be done in poly(n) time even though it can have exponentially many constraints. The second way is to find a "compact formulation" of the LP which uses fewer variables, much like the usual LP that you may have seen for the ordinary maximum flow problem.

since the graph has at most n^2 arcs. So, with probability at least $1 - 1/n$, the algorithm produces a solution for which every arc has congestion at most αC^*, which is at most $\alpha \cdot OPT$ by Claim 21.2.2. So our algorithm has approximation factor $\alpha = O(\log n / \log\log n)$.

Further Remarks. The *rounding* algorithm that we presented is actually optimal: there are graphs for which $OPT/C^* = \Omega(\log n / \log\log n)$. Consequently, every rounding algorithm which converts a fractional solution of LP to an integral solution of IP must necessarily incur an increase of $\Omega(\log n / \log\log n)$ in the congestion.

That statement does not rule out the possibility that there is a better algorithm which behaves completely differently (i.e., one which does not use IP or LP at all). But sadly it turns out that there is no better algorithm (for the case of directed graphs). It is known that every efficient algorithm must have approximation factor $\alpha = \Omega(\log n / \log\log n)$, assuming a reasonable complexity theoretic conjecture (NP $\not\subseteq$ BPTIME($n^{O(\log\log n)}$)). So the algorithm that we presented is optimal, up to constant factors.

21.3 Error-correcting codes

Suppose a person (the sender) wants to transmit a message to another person (the receiver). This message might somehow be corrupted during transmission. The goal is for the receiver to determine the original message, even if these corruptions occur.

There are various ways to model corruptions, such as stochastic noise (or Shannon model), stochastic erasures, stochastic deletions, adversarial erasures, etc.

We will consider the ***adversarial noise model*** (or Hamming model), in which it is assumed that at most t symbols of the message are be corrupted. The values and locations of those corruptions can be arbitrary.

The sender and receiver will first agree upon a ***code***, or ***binary code***, which we define to be a set

$$\text{Code:} \qquad \mathcal{C} \subseteq \{0,1\}^n ,$$

whose entries are called ***codewords***. Since the codewords are binary strings, we sometimes call $\mathcal{C}$ a ***binary code***. The set $\mathcal{C}$ can be ordered, for example by viewing the codewords as binary numbers. Thus we may write $\mathcal{C} = \{C_1, \ldots, C_M\}$ for some value M. Note that $|\mathcal{C}| \le 2^n$, so

$$2^n \;\ge\; M \tag{21.3.1}$$

The communication scheme is as follows. Suppose that there are M different messages that the sender could send. We can think of a message simply as being an integer $m \in [M]$. (For example, the messages might be arbitrary binary strings of length $\lg M$.) Each message m can be identified with a codeword in a canonical way; for example, message m corresponds to codeword C_m.

Transmitting m directly would take $\lceil \lg M \rceil$ bits. Instead, the sender transmits C_m, which takes n bits. This requires more bits to be transmitted:

$$n \;\ge\; \lceil \lg M \rceil \tag{21.3.2}$$

because of (21.3.1). This value n is called the ***block length*** of the code. The efficiency of the code is measured by how tight the inequality (21.3.2) is. Quantatively, the ***rate*** of $\mathcal{C}$ is defined to be

$$R \;=\; \frac{\lg|\mathcal{C}|}{n} \;=\; \frac{\lg M}{n}.$$

The main question: if the codeword is corrupted during transmission, can the receiver still determine the original message m?

21.3.1 Decoding

Let $\oplus$ denote Binary Xor. The ***Hamming distance*** between $x, y \in \{0,1\}^n$ is

$$\Delta(x,y) \;=\; \sum_{i=1}^{n}(x_i \oplus y_i) \;=\; \sum_{i=1}^{n}|x_i - y_i| \;=\; \|x-y\|_1 .$$

Observe that Δ satisfies the triangle inequality

$$\Delta(x,z) \;\le\; \Delta(x,y) + \Delta(y,z) \tag{21.3.3}$$

because $\|\cdot\|_1$ is a norm. (This also follows by applying Fact A.2.4 component-wise.)

The ***minimum distance*** (or simply ***distance***) of $\mathcal{C}$ is

$$d \;=\; \min_{x \neq y \in \mathcal{C}} \Delta(x,y).$$

The ***relative distance*** is d/n.

Claim 21.3.1 (Unique decoding). Let $x \in \mathcal{C}$ be arbitrary. Produce y from x by flipping at most $\lfloor \frac{d-1}{2} \rfloor$ bits. Then, given y, one can determine x.

Proof. By construction of y, the Hamming distance $\Delta(x,y)$ satisfies

$$\Delta(x,y) \;\le\; \left\lfloor \frac{d-1}{2} \right\rfloor .$$

However, for any other codeword $z \in \mathcal{C}$, where $z \neq x$, we have

$$\begin{aligned}
\Delta(z,y) &\ge \Delta(x,z) - \Delta(x,y) \qquad \text{(by the triangle inequality (21.3.3))} \\
&\ge d - \left\lfloor \frac{d-1}{2} \right\rfloor \\
&\ge d/2 + 1/2 \;>\; \left\lfloor \frac{d-1}{2} \right\rfloor .
\end{aligned}$$

We have shown that y's Hamming distance to x is smaller than to any other z. □

21.3.2 A random code construction

Theorem 21.3.2. For any $\epsilon > 0$, there exists a binary code of relative distance $1/2 - \epsilon$ and rate $\Omega(\epsilon^2)$.

References: MIT course notes.

Remark 21.3.3.

- You might wonder why we aim for relative distance nearly $1/2$ instead of relative distance nearly 1. This is because binary codes with relative distance nearly 1 have rate going to zero with n. Specifically, if $\delta > 1/2 + \epsilon$ then $M \le 1/\epsilon$ so the rate is $R \le \lg(1/\epsilon)/n$, which vanishes as $n \to \infty$. See (Guruswami et al., 2019, Theorem 4.4.1).

- The Gilbert-Varshamov greedy code construction also achieves relative distance $1/2 - \epsilon$ and rate $\Omega(\epsilon^2)$. See (Guruswami et al., 2019, Theorem 4.2.1 and Proposition 3.3.5).
- For codes with relative distance $1/2 - \epsilon$, the rate of $\Omega(\epsilon^2)$ is nearly optimal. The so-called Linear Programming bound implies a $O(\epsilon^2 \log(1/\epsilon))$ upper bound on the rate. See (Guruswami et al., 2019, Section 8.2). For the special case of "balanced codes", the $O(\epsilon^2 \log(1/\epsilon))$ bound follows from an algebraic argument due to Alon.

Claim 21.3.4. Let y, z be independent and uniformly random points in $\{0,1\}^n$. Then

$$\Pr\left[\,\Delta(y,z)/n < 1/2 - \epsilon\,\right] \;\leq\; \exp(-\epsilon^2 n).$$

Proof. The Hamming distance $\Delta(y,z)$ can be decomposed into indicators as

$$\Delta(y,z) \;=\; \sum_{i=1}^{n} X_i$$

where $X_i = y_i \oplus z_i$. Note that X_i is an unbiased Bernoulli RV — it is 0 or 1 with probability 1/2. (This actually follows from Corollary A.3.20.) The expected distance is $\mu = \mathrm{E}\left[\,\Delta(y,z)\,\right] = n/2$. Then, by the Chernoff bound,

$$\begin{aligned}
\Pr\left[\,\Delta(y,z)/n < 1/2 - \epsilon\,\right] &= \Pr\left[\,\Delta(y,z) < (1-2\epsilon)\mu\,\right] \\
&< \exp\left(-(2\epsilon)^2\mu/2\right) \\
&= \exp(-\epsilon^2 n).
\end{aligned}$$

□

Proof of Theorem 21.3.2. Let $M = \exp(\epsilon^2 n/2)$ and pick $C_1, ..., C_M$ in $\{0,1\}^n$ independently and uniformly at random. The previous claim show that, for any pair of these points, they are unlikely to have small Hamming distance. This is extended to all pairs of points by the union bound.

$$\begin{aligned}
\Pr\left[\,\exists i,j \text{ s.t. } \Delta(C_i,C_j)/n < 1/2 - \epsilon\,\right] &\leq \binom{M}{2} \cdot \exp(-\epsilon^2 n) && \text{(union bound)} \\
&< \frac{M^2}{2} \cdot \exp(-\epsilon^2 n) && \text{(by Claim 21.3.4)} \\
&= \frac{1}{2} \cdot \Big(\underbrace{\exp(\epsilon^2 n/2)}_{=M}\Big)^2 \cdot \exp(-\epsilon^2 n) \\
&= 1/2.
\end{aligned}$$

We define the code $\mathcal{C} = \{C_1, \ldots, C_M\}$. It definitely has rate

$$\frac{\lg M}{n} \;=\; \frac{\epsilon^2 n/2}{\ln(2) n} \;=\; \Omega(\epsilon^2).$$

Furthermore, with probability at least 1/2, we have $\Delta(C_i, C_j)/n \geq 1/2 - \epsilon$ for all $i \neq j$, which implies that $\mathcal{C}$ has relative distance at least $1/2 - \epsilon$.

Since this construction has probability at least 1/2 of producing a code with the desired rate and distance, it follows that such a code must exist. This style of argument is an example of the probabilistic method. □

Research Questions. There are several open questions relating to this topic. See (Guruswami et al., 2019, Section 8.3).

- Is the $\Omega(\epsilon^2)$ rate optimal?
- Can one explicitly describe a code achieving rate $\Omega(\epsilon^2)$?
- Is there a code with rate $\Omega(\epsilon^2)$ that can be efficiently decoded?

Recent research. Very recently, startling progress has been made on the second and third questions. A breakthrough of Ta-Shma in STOC 2017 has shown that there are explicit codes with distance $1/2-\epsilon$ and rate $\Omega(\epsilon^{2+\alpha})$ where $\alpha \to 0$ as $\epsilon \to 0$. Moreover, in STOC 2021, Jeronimo et al. showed that these codes can be decoded in nearly-linear time.

Chapter 22

Dimensionality Reduction

Dimensionality reduction is the process of mapping a high dimensional dataset to a lower dimensional space, while preserving much of the important structure. In statistics and machine learning, this often refers to the process of finding a few directions in which a high dimensional random vector has maximum variance. Principal component analysis is a standard technique for that purpose.

In this lecture, we consider a different sort of dimensionality reduction. Given a set of high-dimensional points, the goal is to find lower-dimensional points whose *pairwise distances* approximately match the original points. We present a technique, known as the ***random projection method*** or ***Johnson-Lindenstrauss method***, for solving this problem.

In the previous chapters, our main tool has been the Chernoff bound. Here we will not directly use the Chernoff bound, but the main proof uses very similar ideas.

22.1 Intuition

Let us begin with some three-dimensional examples. We will measure lengths using the Euclidean norm. Our notation for the length of a vector v is $\|v\| = \sqrt{\sum_i v_i^2}$.

In our first example, the dimension can be reduced while exactly preserving pairwise distances.

Example 2. Consider the points

$$x_1 = \begin{pmatrix} 0.3237 \\ -2.1870 \\ -0.3354 \end{pmatrix} \qquad x_2 = \begin{pmatrix} 0.1327 \\ -3.5215 \\ -0.7627 \end{pmatrix} \qquad x_3 = \begin{pmatrix} -1.6022 \\ -2.2986 \\ -1.4660 \end{pmatrix}$$

They have pairwise distances

$$\|x_1 - x_2\| \approx 1.4142 \qquad \|x_1 - x_3\| \approx 2.2361 \qquad \|x_2 - x_3\| \approx 2.2361$$

We now define a linear map

$$L = \begin{pmatrix} -0.7056 & -0.4820 & -0.5193 \\ 0.5146 & -0.8525 & 0.0920 \end{pmatrix}$$

Then define two-dimensional points y_i by $y_i = Lx_i$. This yields the points

$$y_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \qquad y_2 = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \qquad y_3 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}.$$

They have pairwise distances

$$\|y_1 - y_2\| \approx \|x_1 - x_2\| \approx 1.4142 \quad \|y_1 - y_3\| \approx \|x_1 - x_3\| \approx 2.2361 \quad \|y_2 - y_3\| \approx \|x_2 - x_3\| \approx 2.2361.$$

This is not so impressive because the points x live in a two-dimensional subspace, so we have simply performed an orthogonal change of coordinates to obtain their two-dimensional representation.

In our second example, it is not possible to reduce the dimension while exactly preserving pairwise distances.

Example 3. Define the points

$$x_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \qquad x_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \qquad x_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \qquad x_4 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

They have pairwise distances

$$\|x_i - x_j\| \;=\; \sqrt{2} \qquad \forall i, j.$$

Thus, they form a regular 3-simplex (or tetrahedron) in $\mathbb{R}^3$. It can be shown that there is no representation for the 3-simplex in $\mathbb{R}^1$ or $\mathbb{R}^2$. So there is no way to reduce the dimension of these points while exactly preserving the pairwise distances.

Nevertheless, in the next section we will show that we can reduce the dimension of *any* high-dimensional point set if we are allowed to *approximate* the pairwise distance.

22.2 The Johnson-Lindenstrauss Theorem

Suppose we have n points $x_1, \ldots, x_n \in \mathbb{R}^d$. We would like to find n points $y_1, \ldots, y_n \in \mathbb{R}^t$, where $t \ll d$, such that the lengths and pairwise distances of the y vectors are approximately the same as for the x vectors. We will show that this can be accomplished while taking t to be surprisingly small.

Each vector y_i will be obtained from x_i by a linear map. Let R be a random $t \times d$ ***Gaussian matrix***. This means that each entry of R is an independent standard Gaussian random variable. Gaussians are defined in Appendix B.5.2.

Theorem 22.2.1 (Johnson-Lindenstrauss 1984). Let $x_1, \ldots, x_n \in \mathbb{R}^d$ be arbitrary. Pick any $\epsilon = (0, 1)$. There exists $t = O(\log(n)/\epsilon^2)$ such that, if R is a $t \times d$ Gaussian matrix, and $y_i = Rx_i/\sqrt{t}$ then, with probability at least $1 - 1/n$,

$$\begin{array}{rcccll} (1-\epsilon)\,\|x_j\| & \leq & \|y_j\| & \leq & (1+\epsilon)\,\|x_j\| & \forall j \\ (1-\epsilon)\,\|x_j - x_{j'}\| & \leq & \|y_j - y_{j'}\| & \leq & (1+\epsilon)\,\|x_j - x_{j'}\| & \forall j, j'. \end{array} \tag{22.2.1}$$

References: (Shalev-Shwartz and Ben-David, 2014, Section 23.2), (Dubhashi and Panconesi, 2009, Section 2.5), (Vershynin, 2018, Theorem 5.3.1), (Boucheron et al., 2012, Theorem 2.13), (Roch, 2020, Section 2.4.6), (Blum et al., 2018, Section 2.7), (Wainwright, 2019, Example 2.12).

Observations.

- The linear map R is *oblivious*: it does not depend on the points $x_1, \ldots, x_n$ at all.

- A standard Gaussian is also called a $N(0,1)$ random variable, meaning that has zero mean and variable 1. Instead of defining $y_i = Rx_i/\sqrt{t}$, we could equivalently let the entries of R have distribution $N(0, 1/t)$ and define $y_i = Rx_i$. (This follows from Fact B.5.4 with $d = 1$ and $\sigma_1 = 1/\sqrt{t}$.)

Whereas principal component analysis is only useful when the original data points $\{x_1, \ldots, x_n\}$ are nearly low dimensional, this theorem requires *absolutely no assumption* on the original data. Also, note that the final data points $\{y_1, \ldots, y_n\}$ have no dependence on d. The original data could live in an arbitrarily high dimension. (Although one can always assume $d \leq n$ since $x_1, \ldots, x_n$ lie in their linear span, which is a Euclidean space of dimension at most n.)

22.2.1 Overview of proof

To prove the theorem, let us define the random linear map that is used to construct the points y_j. The parameter t will be chosen appropriately below. Let R is a $t \times d$ matrix whose entries are independently drawn from $N(0,1)$, i.e., Gaussians with mean 0 and variance 1. The point y_j in Theorem 22.2.1 is simply by multiplication with R and then rescaling

$$y_j \leftarrow \frac{R \cdot x_j}{\sqrt{t}}.$$

In pseudocode, we can write the algorithm as follows.

Algorithm 22.1 The Johnson-Lindenstrauss algorithm. Each input vector x_i is in $\mathbb{R}^d$.

```
function JL(vector x_1, ..., x_n, int d, float ε)
    Let t ← O(ln(n)/ε^2)
    Let R be a random Gaussian matrix of size t × d
    for i = 1, ..., n
        Let y_i ← R · x_i/√t
    return y_1, ..., y_n
end function
```

Our proof of Theorem 22.2.1 will show that this algorithm produces vectors $y_1, \ldots, y_n$ satisfying (22.2.1) with probability at least $1 - 1/n$. (There is a caveat: the definition of t involves a constant that we have not specified.)

The key to proving Theorem 22.2.1 is to prove the following lemma.

Lemma 22.2.2 (Distributional JL). Let R be a $t \times d$ Gaussian matrix. Let $\delta \in (0, 1]$ be arbitrary. Let $t = 8\ln(2/\delta)/\epsilon^2$. Then for all vectors $v \in \mathbb{R}^d$ with $\|v\| = 1$,

$$\Pr\left[\frac{\|Rv\|}{\sqrt{t}} \notin (1-\epsilon, 1+\epsilon) \right] \leq \delta. \tag{22.2.2}$$

Proof of Theorem 22.2.1. Set $\delta = 1/n^3$. Consider the set of vectors

$$W = \{ x_i : i = 1, \ldots, n \} \cup \{ x_i - x_j : i \neq j \}.$$

There are at most n^2 vectors in W.

For any non-zero $w \in W$, we may apply the DJL lemma to $v = w/\|w\|$. The event that w's norm is *not* adequately preserved is

$$\mathcal{E}_w \;=\; \left\{ \frac{\|Rw\|}{\sqrt{t}} \notin [1-\epsilon, 1+\epsilon] \cdot \|w\| \right\} \;=\; \left\{ \frac{\|Rv\|}{\sqrt{t}} \notin [1-\epsilon, 1+\epsilon] \right\},$$

because $\|Rw\| = \|Rv \cdot \|w\|\| = \|w\| \cdot \|Rv\|$. The condition (22.2.1) holds if and only if no event $\mathcal{E}_w$ occurs. Thus

$$\begin{aligned}
\Pr[\text{ condition (22.2.1) fails to hold }] \;&=\; \Pr\left[\bigcup_{w \in W} \mathcal{E}_w \right] \\
&\le\; \sum_{w \in W} \Pr[\,\mathcal{E}_w\,] \qquad \text{(the union bound)} \\
&\le\; |W| \cdot \delta \qquad \text{(by Lemma 22.2.2)} \\
&\le\; 1/n. \quad \square
\end{aligned}$$

22.2.2 Random Dot Products

The idea of taking a random dot product is crucial to our discussion. Consider an arbitrary vector $v \in \mathbb{R}^d$. Let $g \in \mathbb{R}^d$ be a random vector where $\mathrm{E}[\,g_i\,] = 0$ for each i. We are interested in the analyzing the dot product $g^\mathsf{T} v$. This quantity may not seem so interesting at first glance because, by linearity of expectation,

$$\mathrm{E}\left[\, g^\mathsf{T} v \,\right] \;=\; \sum_{i=1}^{d} \mathrm{E}[\,g_i\,]\, v_i \;=\; 0. \tag{22.2.3}$$

So let us also introduce the notion of ***variance***, which is defined in Definition B.5.1. We require the following fact.

Fact B.5.3. Let $g_1, \ldots, g_d$ be independent random variables with finite variance. Let $\sigma_1, \ldots, \sigma_d \in \mathbb{R}$ be arbitrary. Then $\mathrm{Var}\left[\sum_{i=1}^d \sigma_i g_i\right] = \sum_{i=1}^d \sigma_i^2 \,\mathrm{Var}[\,g_i\,]$.

We will additionally assume that $\mathrm{Var}[\,g_i\,] = 1$ for each i. Then we have

$$\begin{aligned}
\mathrm{E}\left[\, (g^\mathsf{T} v)^2 \,\right] \;&=\; \mathrm{Var}\left[\, g^\mathsf{T} v \,\right] && \text{(by (22.2.3) and (B.5.2))} \\
&=\; \mathrm{Var}\left[\sum_{i=1}^{d} v_i g_i \right] \;=\; \sum_{i=1}^{d} v_i^2 \,\mathrm{Var}[\,g_i\,] && \text{(by Fact B.5.3)} \\
&=\; \|v\|^2 .
\end{aligned}$$

This shows that $(g^\mathsf{T} v)^2$ is an unbiased estimator for $\|v\|^2$. Thus, we can estimate $\|v\|^2$ by estimating $\mathrm{E}\left[\,(g^\mathsf{T} v)^2\,\right]$. If the estimate is sufficiently good, we can then take the square root and estimate $\|v\|$.

Which random vector? So far, the discussion is valid for *any* independent RVs g_i with mean 0 and variance 1. For example, we could take g_i to be uniform on $\{-1, 1\}$.

Our subsequent analysis will be made easier by choosing g_i to have the Gaussian distribution $N(0,1)$. This is due to a key property: *the sum of Gaussians is also Gaussian.*

Fact B.5.4. Let $g_1, \ldots, g_d$ be independent random variables where g_i has distribution $N(0,1)$. Then, for any scalars $\sigma_1, \ldots, \sigma_d$, the sum $\sum_{i=1}^d \sigma_i g_i$ has distribution $N(0, \sum_{i=1}^d \sigma_i^2) = N(0, \|\sigma\|_2^2)$.

Consequently, our random dot product $g^\mathsf{T} v$ has the distribution

$$\text{Distribution of } g^\mathsf{T} v: \quad N(0, \sum_{i=1}^d v_i^2) = N(0, \|v\|^2). \tag{22.2.4}$$

Improved estimates by averaging. A common idea in randomized algorithms is to average multiple independent estimates in order to get a higher quality estimate. (See, e.g., Section 12.2.2.) Perhaps we can get a high-quality estimate of $\|v\|^2$ by picking several random vectors g, then averaging the values of $(g^\mathsf{T} v)^2$?

This idea is exactly what the Johnson-Lindenstrauss lemma does. Our matrix R is a $t \times d$ Gaussian matrix. Let r_i refer to the i^{th} row of R. Then $r_1, \ldots, r_t$ are independent Gaussian vectors, so each component of Rv is a random dot product:

$$(Rv)_i \;=\; r_i^\mathsf{T} v \qquad \forall i \in [t].$$

The squared norm of $Rv/\sqrt{t}$ is then

$$\left\| \frac{Rv}{\sqrt{t}} \right\|^2 \;=\; \sum_{i=1}^t \frac{(Rv)_i^2}{t} \;=\; \sum_{i=1}^t \frac{(r_i^\mathsf{T} v)^2}{t}, \tag{22.2.5}$$

which is the average of t independent random dot products.

22.2.3 Proof of Lemma 22.2.2

Let us rewrite (22.2.2) as follows.

$$\begin{aligned}
\Pr\left[\frac{\|Rv\|}{\sqrt{t}} \notin (1-\epsilon, 1+\epsilon) \right] &= \Pr\left[\|Rv\|^2 \notin \left((1-\epsilon)^2, (1+\epsilon)^2\right) \cdot t \right] && \text{(squaring both sides)} \\
&= \Pr\left[\sum_{i=1}^t (r_i^\mathsf{T} v)^2 \notin \left((1-\epsilon)^2, (1+\epsilon)^2\right) \cdot t \right] && \text{(by (22.2.5))} \\
&\le \Pr\left[\sum_{i=1}^t (r_i^\mathsf{T} v)^2 \notin (1-\epsilon, 1+\epsilon) \cdot t \right]
\end{aligned}$$

Here we have used the simple bounds

$$(1-\epsilon)^2 \le 1-\epsilon \quad \text{and} \quad (1+\epsilon)^2 \ge 1+\epsilon.$$

We have shown in (22.2.4) that $r_i^\mathsf{T} v$ has the distribution $N(0, \|v\|^2)$, which is $N(0,1)$ since we assume that v is a unit vector. It now turns out that $\sum_{i=1}^t (r_i^\mathsf{T} v)^2$ has a well-known distribution too. It is the sum of the *squares* of t independent standard normal random variables, which is called the chi-squared distribution with parameter t.

Question 22.2.3. What is $\mathrm{E}\left[\|Rv\|^2 \right]$?

Answer.

We have

$$\mathrm{E}\left[\,\|Rv\|^2\,\right] \;=\; \sum_{i=1}^{t} \mathrm{E}\left[\,(r_i^{\mathsf{T}} v)^2\,\right] \;=\; \sum_{i=1}^{t} \mathrm{Var}\left[\,r_i^{\mathsf{T}} v\,\right] \;=\; t,$$

since $r_i^{\mathsf{T}} v$ is $N(0,1)$.

To summarize, our desired inequality is

$$\Pr\left[\;\underbrace{\sum_{i=1}^{t} (r_i^{\mathsf{T}} v)^2}_{X} \notin (1-\epsilon,\, 1+\epsilon) \cdot \underbrace{t}_{\mathrm{E}[\,X\,]}\;\right] \;\le\; \delta \tag{22.2.6}$$

where X is a chi-squared RV with parameter t. Fortunately, tail bounds for these RVs are known.

Lemma 22.2.4 (Tail bound for chi-squared). Let X have the chi-squared distribution with parameter t. Then

$$\Pr\left[\,|X - t| \ge \epsilon t\,\right] \;\le\; 2\exp(-\epsilon^2 t/8) \qquad \forall \epsilon \in (0,1).$$

References: (Shalev-Shwartz and Ben-David, 2014, Lemma B.12), (Wainwright, 2019, Example 2.11 and Example 2.28).

This lemma easily allows us to prove (22.2.6), which proves (22.2.2).

$$\Pr\left[\,X \notin (1-\epsilon,\, 1+\epsilon)\cdot t\,\right] \;=\; \Pr\left[\,|X - t| \ge \epsilon t\,\right] \;\le\; 2\exp(-\epsilon^2 t/8) \;=\; \delta, \tag{22.2.7}$$

since $t = 8\ln(2/\delta)/\epsilon^2$.

22.2.4 Example of a chi-squared tail bound

To illustrate one strategy for proving Lemma 22.2.4, let us prove the following bound on the right tail. This even yields the better constant in the exponent of $1/2$ rather than $1/8$.

Claim 22.2.5. Let X have the chi-squared distribution with parameter t. Then $\Pr\left[\,X \ge t(1+\epsilon)^2\,\right] \le \exp(-t\epsilon^2/2)$.

Proof. Our proof will follow the Chernoff bound strategy. For any $\alpha \in \mathbb{R}$ and $\theta \ge 0$, we have

$$\Pr\left[\,X \ge \alpha\,\right] \;=\; \Pr\left[\,e^{\theta X} \ge e^{\theta\alpha}\,\right] \;\le\; e^{-\theta\alpha}\,\mathrm{E}\left[\,e^{\theta X}\,\right]. \tag{22.2.8}$$

The quantity $\mathrm{E}\left[\,e^{\theta X}\,\right]$ is called the moment generating function, and for many standard distributions it has a known closed form. We now cheat by referring to Wikipedia, where we find that the moment generating function for the chi squared distribution is $\mathrm{E}\left[\,e^{\theta X}\,\right] = (1-2\theta)^{-t/2}$ (for $\theta < 1/2$), so

$$\Pr\left[\,X > \alpha\,\right] \;\le\; e^{-\theta\alpha}(1-2\theta)^{-t/2} \qquad (\text{if } \theta \in [0, 1/2)).$$

The next step is to plug in an appropriate choice of θ. We set $\theta = (1 - t/\alpha)/2$, giving

$$\begin{aligned}
\Pr[\, X \geq \alpha \,] &\leq e^{(t-\alpha)/2}(t/\alpha)^{-t/2} \\
&= \exp\left(\frac{t}{2}\Big(1-(1+\epsilon)^2\Big) - \frac{t}{2}\ln\left(\frac{1}{(1+\epsilon)^2}\right)\right) \qquad (\text{setting } \alpha = t(1+\epsilon)^2) \\
&= \exp\Big(-t\big(\epsilon + \epsilon^2/2 - \ln(1+\epsilon)\big)\Big) \\
&\leq \exp\Big(-t\big(\epsilon + \epsilon^2/2 - \epsilon\big)\Big) \qquad (\text{using } \ln(1+x) \leq x; \text{ see Exercise B.1}) \\
&\leq \exp(-t\epsilon^2/2). \qquad \square
\end{aligned}$$

Question 22.2.6. Improve the bound $\log(1+\epsilon) \leq \epsilon$ to $\log(1+\epsilon) \leq \epsilon - \epsilon^2/4$ for $\epsilon \in [0,1]$.

Answer.

It follows from Fact B.4.1 by replacing $z \to x/2$ and integration.

22.2.5 Broader context

In this section we have switched from the world of discrete probability to continuous probability. This is to make our lives easier. The same theorem would be true if we picked the coordinates of r_i to be uniform in $\{+1, -1\}$ rather than Gaussian. But the analysis of the $\{+1, -1\}$ case is trickier, and most proofs analyze that case by showing that its failure probability is not much worse than in the Gaussian case. So the Gaussian case is really the central problem.

Second of all, you might be wondering where the name ***random projection method*** comes from. Earlier versions of the Johnson-Lindenstrauss theorem used a slightly different linear map. Specifically, they used the map Rv where $R^T R$ is a *projection* onto a uniformly random subspace of dimension t. (Recall that an orthogonal projection matrix is any symmetric, positive semidefinite matrix whose eigenvalues are either 0 or 1.) One advantage of that setup is its symmetry: one can argue that the failure probability in Lemma 22.2.2 would be the same if one instead chose a *fixed* subspace of dimension t and a *random* unit vector v. The latter problem can be analyzed by choosing the subspace to be the most convenient one of all: the span of the first t vectors in the standard basis.

So how is our map $R/\sqrt{t}$ different? It is almost a projection, but not quite. If we chose R to be a matrix of independent Gaussians, it turns out that the range of $R^T R/t$ is indeed a uniformly random subspace, but its eigenvalues are not necessarily in $\{0, 1\}$. If we had insisted that the random vectors r_i that we choose were *orthonormal*, then we would have obtained a projection matrix. We could explicitly orthonormalize them by the Gram-Schmidt method, but fortunately that turns out to be unnecessary: the Johnson-Lindenstrauss theorem is true, even if we ignore orthonormality of the r_i's.

Our linear map $R/\sqrt{t}$ turns out to be a bit more convenient in some algorithmic applications, because we avoid the awkward Gram-Schmidt step.

Optimality. Recently there has been much progress on understanding the optimality of these results. The DJL lemma is actually optimal, up to constant factors.

Theorem 22.2.7 (Jayram-Woodruff 2013, Kane-Meka-Nelson 2011). Any f satisfying the DJL lemma must satisfy $t = \Omega(\log(1/\delta)/\epsilon^2)$.

But, this does not necessarily imply that Theorem 22.2.1 is optimal; perhaps the theorem can be proven without using the DJL lemma. Alon proved the following lower bound. (See Theorem 9.3 of this paper.)

Theorem 22.2.8 (Alon). Let $x_1, \ldots, x_{n+1} \in \mathbb{R}^n$ be the vertices of a simplex, i.e., $\|x_i - x_j\| = 1$ for all $i \neq j$. If $y_1, \ldots, y_{n+1} \in \mathbb{R}^t$ satisfy (22.2.1), then $t = \Omega(\frac{\log(n)}{\epsilon^2 \log(1/\epsilon)})$.

This shows that Theorem 22.2.1 is almost optimal, up to the factor $\log(1/\epsilon)$ in the denominator. Actually, for this particular set of points (the vertices of a simplex), Theorem 22.2.1 is *not* optimal and Alon's bound is the right one. However, there is a different point set showing that Theorem 22.2.1 is in fact optimal.

Theorem 22.2.9 (Larsen-Nelson FOCS 2017). There exist points $x_1, \ldots, x_n \in \mathbb{R}^d$ such that the following is true. Consider any map $L : \mathbb{R}^d \to \mathbb{R}^t$, let $y_j = L(x_j)$, and suppose that (22.2.1) is satisfied. Then $t = \Omega(\log(n)/\epsilon^2)$.

Other norms and metrics. The Johnson-Lindenstrauss lemma very strongly depends on properties of the Euclidean norm. For other norms, this remarkable dimensionality reduction is not necessarily possible. For example, for the ℓ_1 norm $\|x\|_1 := \sum_i |x_i|$, it is known that any map into $\mathbb{R}^d$ that preserves pairwise ℓ_1-distances between n points up to a factor $c \geq 1$ must have $d = \Omega(n^{1/c^2})$. If $c = 1 + \epsilon$, then there are upper bounds of $d = O(n \log n/\epsilon^2)$ and $d = O(n/\epsilon^2)$.

References: Talagrand Proc. AMS 1990, Brinkman-Charikar FOCS 2003, Lee-Naor 2004, Newman-Rabinovich SODA 12.

For more on this subject, see the survey of Indyk and Matousek or the tutorial of Indyk.

22.3 Fast Johnson-Lindenstrauss

In the previous section we saw the Johnson-Lindenstrauss theorem on dimensionality reduction. Suppose we have n points in $\mathbb{R}^d$ and we map them to $\mathbb{R}^t$, where $t = O(\log(n)/\epsilon^2)$, simply by applying a $t \times d$ matrix of Gaussians (scaled appropriately). Then all lengths and pairwise distances are preserved up to a factor $1 + \epsilon$, with high probability. This is a very useful tool in algorithm design.

Let's consider the efficiency of such a mapping. Directly applying matrix-vector multiplication, the time to map a single point from $\mathbb{R}^d$ to $\mathbb{R}^t$ is $O(td)$. There has been much work on trying to make this faster.

One direction of research considered using a slightly *sparse* matrix instead of a dense matrix of Gaussians. The state of the art (Kane-Nelson JACM 2014) allows the matrix to have only an ϵ fraction of non-zero entries, so the time to map a single point becomes $O(\epsilon td)$. Their result is optimal to with a factor of $O(\log(1/\epsilon))$.

Today we discuss a different line of research. Instead of using sparse matrices, we will use *structured* matrices, for which multiplication can be done faster than the naive algorithm. (As a trivial example, consider the matrix of all ones. It is dense, but multiplying by it is very easy.) Such matrices are called "Fast Johnson-Lindenstrauss Transforms", and they have been used extensively in the algorithms, compressed sensing, machine learning and numerical linear algebra communities.

The first result of this type is stated below. Is is of the Distributional JL type, in that it preserves the length of any fixed vector with good probability.

Theorem 22.3.1 (Ailon-Chazelle STOC 2006). There is a $t \times d$ random matrix that satisfies the DJL lemma and for which matrix-vector multiplication takes time $O(d \log d + t^3)$.

To understand if this runtime is good, consider the scenario where we are applying dimensionality reduction to n data points. The Ailon-Chazelle result is only interesting for certain parameters n, d and t. First, suppose n is very small, say $d = n$, so $t \approx \log n = \log d$. Then the original JL theorem takes time roughly $O(td) = O(d \log d)$ per vector, which is the same as Ailon-Chazelle. Second, suppose n is

very large, say $n = 2^{\sqrt{d}}$, so $t \approx \log n = d^{1/2}$. Then the original JL theorem takes time $O(td) = O(d^{3/2})$ per vector and Ailon-Chazelle also takes time $O(d \log d + t^3) = O(d^{3/2})$. The Ailon-Chazelle result is interesting in the intermediate range, where $d \ll n \ll 2^{d^{1/2}}$

We will prove the following theorem, which is a slightly simplified form of the Ailon-Chazelle result.

Theorem 22.3.2. There is a random matrix R of size $t \times d$ with $t = O(\log(d/\delta)^2 \log(1/\delta)/\epsilon^2)$ such that, for each $x \in \mathbb{R}^d$,

$$\|Rx\| \in [1-\epsilon, 1+\epsilon] \cdot \|x\|$$

holds with probability at least $1 - \delta$. Matrix-vector multiplication with R takes time $O(d \log d + t)$.

22.3.1 A Simple Start: Super-Sparse Sampling

Let's start with simple idea: given a vector x, we will sample it using a very sparse sampling matrix. This matrix is denoted S and it has size $t \times d$. Each row of S has a single non-zero entry of value $\sqrt{d/t}$ in a uniformly random location.

For any vector $x \in \mathbb{R}^d$, we have

$$\mathrm{E}\left[\,(Sx)_i^2\,\right] \;=\; \sum_{j=1}^{d} \underbrace{\Pr\left[\,i^{\text{th}} \text{ row's nonzero entry is in location } j\,\right]}_{=1/d} \cdot (\sqrt{d/t})^2 \cdot x_j^2 \;=\; (1/t) \cdot \|x\|_2^2$$

$$\implies \quad \mathrm{E}\left[\,\|Sx\|^2\,\right] \;=\; \mathrm{E}\left[\,\sum_{i=1}^{t} (Sx)_i^2\,\right] \;=\; \|x\|_2^2 \tag{22.3.1}$$

So this works well in expectation, even if we have $t = 1$. Unfortunately the variance can be terrible.

Example 4 (Sparse case). Suppose x has just one non-zero coordinate, say $x = e_1$. The expected number of non-zero coordinates in Sx is t/d. So Sx is very likely to be 0 unless $t = \Omega(d)$. So there is very little dimensionality reduction.

Example 5 (Dense case). Suppose $x = [1, 1, \cdots, 1]/\sqrt{d}$, which has $\|x\|_2 = 1$. Then $Sx = \sqrt{1/t}x$, so $\|Sx\| = \sqrt{1/t} \cdot \sqrt{t} = 1$. Thus the norm of x is preserved exactly, regardless of the value of t.

Let us try to extract some general principles from these examples.

- *Sparse case.* If one coordinate is much larger than the others in absolute value (as in Example 4) then this approach seems unlikely to work.
- *Dense case.* If all coordinates have similar absolute value (as in Example 5, then the approach seems promising. In this case, we also say that x is "uncorrelated with the standard basis".

To make these principles mathematically precise, we turn to the language of norms. Let us recall the following standard inequality.

Fact B.2.1. For all $x \in \mathbb{R}^d$,

$$\|x\|_p \;\le\; \|x\|_r \;\le\; d^{1/r - 1/p} \cdot \|x\|_p \qquad \forall 1 \le r \le p \le \infty.$$

In particular, the most useful cases are

$$\begin{aligned} \|x\|_1 \;&\ge\; \|x\|_2 \;\ge\; \|x\|_\infty \\ \frac{1}{\sqrt{d}} \|x\|_1 \;&\le\; \|x\|_2 \;\le\; \sqrt{d} \cdot \|x\|_\infty . \end{aligned}$$

Let us now consider the norms in the preceding examples.

- *Sparse case.* In Example 4, we have $\|x\|_2 = \|x\|_\infty = 1$.
- *Dense case.* In Example 4, we have $\|x\|_2 = 1$ but $\|x\|_\infty = 1/\sqrt{d}$.

Inspired by the examples, we use the ratio $\|x\|_\infty / \|x\|_2$ as a measure of sparsity of the vector x.

$$\begin{aligned} x \text{ is sparse:} \quad & \frac{\|x\|_\infty}{\|x\|_2} \gtrapprox 1 \\ x \text{ is dense:} \quad & \frac{\|x\|_\infty}{\|x\|_2} \lessapprox \frac{1}{\sqrt{d}}. \end{aligned}$$

By Fact B.2.1, these are the most extreme values of this ratio.

The following claim formally proves that super-sparse sampling works when x is dense. Define

$$\lambda \;=\; \sqrt{\frac{2\ln(4d/\delta)}{d}}. \tag{22.3.2}$$

Claim 22.3.3. Let y be a fixed vector in $\mathbb{R}^d$ with $\|y\|_2 = 1$ and $\|y\|_\infty \le \lambda$. Let S be a $t \times d$ super-sparse sampling matrix with $t = 2\ln(4d/\delta)^2 \ln(4/\delta)/\epsilon^2$. Then

$$\Pr\Big[\, \|Sy\|_2^2 \notin (1-\epsilon, 1+\epsilon) \,\Big] \;\le\; \delta/2.$$

Proof. See Exercise 22.3. □

22.3.2 Idea: Rotating the basis

Stating the problematic scenario in these terms leads to a useful idea. We said that super-sparse sampling will require large t when $\|x\|_\infty / \|x\|_2 \approx 1$. Now we recall an important property of the Euclidean norm: it is invariant under rotations and reflections. Formally, $\|Mx\|_2 = \|x\|_2$ for any orthogonal matrix M. Another way to say this is that $\|\cdot\|_2$ is invariant under an orthogonal change of basis, and indeed $\|\cdot\|_2$ can be defined without reference to any basis (using an inner product). On the other hand, the infinity norm $\|\cdot\|_\infty$ is *heavily* dependent on the choice of basis. For example,

$$\begin{aligned} u = (1,0,0,\ldots,0) \in \mathbb{R}^d \qquad & \text{has} \qquad \|u\|_2 = \|u\|_\infty = 1 \\ v = (1,1,\ldots,1)/\sqrt{d} \in \mathbb{R}^d \qquad & \text{has} \qquad \|v\|_2 = 1 \text{ and } \|v\|_\infty = 1/\sqrt{d} \end{aligned}$$

even though v is a rotation of u. Intuitively, the quantity $\|x\|_\infty / \|x\|_2$ is telling us how well the vector x "aligns" with the standard basis.

In order for our super-sparse sampling to work we need x to be dense, which means that x is not aligned with the standard basis. However we have no control over x because our theorem needs to work for all x. The key idea is to control our basis instead. Why not choose a random basis that is unlikely to align with the given vector x? Indeed, that is basically what is accomplished by the dense matrix of Gaussians in the previous section.

The only issue with the dense matrix of Gaussians is that matrix-vector multiplications are too slow. So let's think if there is a quicker way to randomly rotate the basis. Whenever I think of vectors that "disagree" with the standard basis, the first object that comes to mind is a Hadamard matrix.

Definition 22.3.4. A ***Hadamard matrix*** is a $d \times d$ real matrix H that is orthogonal (meaning $H^\mathsf{T} H = I$) and has all entries in $\left\{\pm 1/\sqrt{d}\right\}$.

It is not a priori clear that Hadamard matrices exist, and indeed it is not fully known for what values of d they exist[1]. In the case $d = 2$, we have the Hadamard matrix

$$H_2 \;=\; \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} / \sqrt{2},$$

which amounts to a 45-degree rotation of the standard basis. In fact, we can build on this recursively to obtain a Hadamard matrix whenever d is a power of two.

$$H_d \;=\; \begin{pmatrix} H_{d/2} & H_{d/2} \\ H_{d/2} & -H_{d/2} \end{pmatrix} / \sqrt{2}. \tag{22.3.3}$$

This is called a Walsh-Hadamard matrix, and it has many nice properties.

First of all H_d is symmetric, so $H_d^\mathsf{T} = H_d$. To see that H_d is indeed a Hadamard matrix, we must make two observations. First, induction shows that every entry of H_d is $\pm(1/\sqrt{2})^{\log_2 d} = \pm 1/\sqrt{d}$. Second, we have

$$H_d^\mathsf{T} H_d \;=\; H_d H_d \;=\; \begin{pmatrix} H_{d/2} & H_{d/2} \\ H_{d/2} & -H_{d/2} \end{pmatrix} \cdot \begin{pmatrix} H_{d/2} & H_{d/2} \\ H_{d/2} & -H_{d/2} \end{pmatrix} /2 \;=\; \begin{pmatrix} 2H_{d/2}H_{d/2} & 0 \\ 0 & 2H_{d/2}H_{d/2} \end{pmatrix} /2 \;=\; I,$$

so H_d is orthogonal.

Another nice property of H_d is that matrix-vector multiplication can be performed in $O(d \log d)$ time. This follows from an easy divide-and-conquer algorithm.

22.3.3 Randomized Hadamard Matrix

Using H_d to change our basis will not guarantee that x is dense. It is obvious that, for *any* fixed orthogonal matrix M, there exist vectors x for which $\|Mx\|_\infty / \|Mx\|_2 = 1$: simply let x be any row of M. This is why the randomization is necessary.

Instead, we must pick a *random* change of basis M and argue that

$$\text{each } x \in \mathbb{R}^d \text{ satisfies} \qquad \left(\frac{\|Mx\|_\infty}{\|Mx\|_2} \approx \frac{1}{\sqrt{d}} \quad \text{with high probability} \right). \tag{22.3.4}$$

So far our Hadamard matrix $H = H_d$ has no randomness. How can we "randomize" it? Well, in the recursive definition (22.3.3) we quite arbitrarily put the minus sign in the lower-right quadrant. The construction works just as well if we put the minus sign in any quadrant, so this suggests that we should be able to randomize the construction using random signs.

We will introduce random signs in a slightly more convenient way. Let D be a diagonal matrix whose i^{th} diagonal entry is a random sign $\xi_i \in \{-1, 1\}$, and these are independent. Our random Hadamard matrix is the product $M = HD$. This is indeed a Hadamard matrix: its entries are still $\pm 1/\sqrt{d}$, and M is orthogonal because $M^\mathsf{T} M = DH^\mathsf{T} HD = D^2 = I$.

The following claim shows that, with this randomized Hadamard matrix M, every vector x satisfies (22.3.4).

[1] One open question is that a $d \times d$ Hadamard matrix exists whenever d is a multiple of 4.

Claim 22.3.5. Let $x \in \mathbb{R}^d$ be non-zero. Let $y = HDx$, where HD is the random Hadamard matrix. Then

$$\Pr_D \left[\frac{\|y\|_\infty}{\|y\|_2} \geq \underbrace{\sqrt{\frac{2\ln(4d/\delta)}{d}}}_{\lambda} \right] \leq \delta/2.$$

Proof. It suffices to consider the case $\|x\|_2 = 1$, because $\|HDx\|_\infty / \|HDx\|_2$ is invariant under rescaling x. Note that $\|HDx\|_2 = \|x\|_2 = 1$ as HD is orthogonal, so it suffices to show that all coordinates of HDx are likely small. We will follow our usual approach of showing that each coordinate is very likely to be small, then union bounding over all coordinates.

Consider y_1, the first coordinate of $y = HDx$. It is obtained by multiplying each coordinate of x by a random sign, then taking the dot-product with the first row of H. So

$$y_1 = \sum_j \xi_j H_{1,j} x_j, \tag{22.3.5}$$

Note that the terms of this sum are independent random variables. It is tempting to apply the Chernoff bound to analyze y_1, but the Chernoff bound that we have used so far is only valid for sums of non-negative random variables. Let us state the following generalized form of the Hoeffding bound, which is very useful in our scenario.

Theorem B.5.7 (Hoeffding's General Inequality). Let $X_1, \ldots, X_n$ be independent random variables where $X_i \in [a_i, b_i]$. Let $X = \sum_{i=1}^n X_i$. Then

$$\Pr\left[\left| \sum_{i=1}^n X_i - \mathrm{E}[X] \right| \geq s \right] \leq 2\exp\Big(-2\frac{s^2}{\sum_{i=1}^n (b_i - a_i)^2} \Big).$$

In particular, for any desired $q \in (0,1)$, setting $s = \sqrt{\ln(2/q) \sum_{i=1}^n (b_i - a_i)^2/2}$ gives

$$\Pr[\, |\textstyle\sum_i X_i - \mathrm{E}[X]| \geq s \,] \leq q.$$

We apply this with $X_j = \xi_j H_{1,j} x_j$. Since $\xi_j \in \{-1, +1\}$, we have $X_j = \pm H_{1,j} x_j$. So X_j lies in the interval $[a_j, b_j]$ where $-a_j = b_j = |H_{1,j} x_j|$. Note that $\mathrm{E}[X_j] = 0$ and

$$\sum_{j=1}^d (b_j - a_j)^2 = 4\sum_{j=1}^d H_{1,j}^2 x_j^2 = 4\sum_{j=1}^d (1/d) x_j^2 = 4\|x\|_2^2/d = 4/d.$$

Defining $q = \delta/2d$, the quantity s in Theorem B.5.7 becomes $s = \sqrt{2\ln(4d/\delta)/d}$, which equals the value of λ defined in (22.3.2). Consequently, (22.3.5) and Theorem B.5.7 yield

$$\Pr[\, |y_1| \geq \lambda \,] = \Pr\left[\Big| \textstyle\sum_j X_j - \underbrace{\mathrm{E}\Big[\sum_j X_j\Big]}_{=0} \Big| \geq s \right] \leq \delta/2d.$$

A union bound over all coordinates of y shows that

$$\Pr[\, \|y\|_\infty \geq \lambda \,] \leq \sum_{j=1}^d \Pr[\, |y_j| \geq \lambda \,] \leq d \cdot (\delta/2d) = \delta/2. \qquad \square$$

22.3.4 Putting it all together

Earlier we said that super-sparse sampling should work as long as x is dense holds. We have just shown x is likely to be dense after applying the randomized Hadamard matrix; more precisely, (22.3.4) holds with $M = HD$. The final step is to apply the super-sparse sampling matrix S to Mx. To summarize, the overall linear map is $R = SHD$ where S is a super-sparse sampling matrix of size $t \times d$, H is a $d \times d$ Hadamard matrix, and D is a diagonal matrix of random signs.

Proof of Theorem 22.3.2. Fix any vector x with $\|x\|_2 = 1$. Construct the random matrix $R = SHD$ as explained above and let $y = HDx$. Define the events

$$\begin{aligned} \mathcal{E}_1 &= \{ \|y\|_\infty \geq \lambda \} \\ \mathcal{E}_2 &= \{ \|Sy\|_2 \notin (1-\epsilon, 1+\epsilon) \} \end{aligned}$$

Claim 22.3.5 shows that $\Pr[\mathcal{E}_1] \leq \delta/2$. Claim 22.3.3 shows that $\Pr[\mathcal{E}_2 \mid \mathcal{E}_1^c] \leq \delta/2$. Combining those bounds we obtain

$$\begin{aligned} \Pr[\|Rx\| \notin (1-\epsilon, 1+\epsilon)] &= \Pr[\mathcal{E}_2] \\ &= \Pr[\mathcal{E}_1 \cap \mathcal{E}_2] + \Pr[\mathcal{E}_1^c \cap \mathcal{E}_2] && \text{(law of total probability, Fact A.3.3)} \\ &\leq \Pr[\mathcal{E}_1] + \Pr[\mathcal{E}_2 \mid \mathcal{E}_1^c] && \text{(by Fact A.3.1 and Exercise A.1)} \\ &\leq \delta/2 + \delta/2 = \delta. \end{aligned}$$

Finally, let us check that matrix-vector multiplication with $R = SHD$ is efficient.

- Multiplication by D is trivial and takes $O(d)$ time.
- Multiplication by H requires $O(d \log d)$ time as explained above.
- Multiplication by S is trivial and takes $O(t)$ time.

So the total time is $O(d \log d + t)$. □

22.4 Subspace Embeddings

The Johnson-Lindenstrauss Theorem shows how to reduce the dimension while preserving distances between a finite set of points. Now we will consider how to reduce the dimension for the *infinitely many* points in a subspace.

Let U be a subspace of $\mathbb{R}^d$ with dimension k. We would like to find a matrix R of size $t \times d$ such that $t \approx k$ and

$$\frac{\|Rx\|}{\|x\|} \in [1-\epsilon, 1+\epsilon] \qquad \forall x \in U,\ x \neq 0. \tag{22.4.1}$$

Question 22.4.1. Is this trivial?

Answer.

Yes. Let $t = k$ and let the rows of R be an orthonormal basis of U. This satisfies (22.4.1) with $\epsilon = 0$.

What makes the problem non-trivial is that we want R to be *oblivious to* U, just like Johnson-Lindestrauss is oblivious to the points whose distance are preserved.

Theorem 22.4.2. For any integer k and any $\epsilon \in (0,1)$, let $t = O(k \log(1/\epsilon)/\epsilon^2)$. Let R be a $t \times d$ matrix where the entries are independent with distribution $N(0, 1/t)$. Then, for any subspace U of dimension k, we have

$$\Pr\left[\frac{\|Rx\|}{\|x\|} \in [1-\epsilon, 1+\epsilon] \ \ \forall x \in U,\, x \neq 0 \right] \geq 1 - 2e^{-k}.$$

To prove the JL Theorem, we proved the DJL Lemma for individual vectors x, then extended that to a finite collection of vectors by taking a union bound. Now Theorem 22.4.2 requires that we approximate the norm for all vectors in U simultaneously. Since U is infinitely large, a naive union bound will not suffice.

A simplifying reduction. Instead of considering an arbitrary subspace U, it suffices to consider the case that $U = \operatorname{span}\{e_1, \ldots, e_k\}$. The reason stems from the following fact.

Fact 22.4.3. Let R be a $t \times d$ random Gaussian matrix and let A be any $d \times d$ orthogonal matrix. Then RA has the same distribution as R.

References: See (Vershynin, 2018, Proposition 3.3.2).

Imagine letting the first k columns of A be an orthonormal basis of U, then extending that to an orthonormal basis of $\mathbb{R}^d$. Then (22.4.1) is equivalent to

$$\frac{\|RAx\|}{\|x\|} \ \in \ [1-\epsilon, 1+\epsilon] \qquad \forall x \in \operatorname{span}\{e_1, \ldots, e_k\}. \tag{22.4.2}$$

Since RA and R have the same distribution (by Fact 22.4.3), our goal is equivalent to showing

$$\Pr\left[\frac{\|Rx\|}{\|x\|} \in [1-\epsilon, 1+\epsilon] \ \ \forall x \in \mathbb{R}^k,\, x \neq 0 \right] \ \geq \ 1 - e^{-k} \tag{22.4.3}$$

where R has now shrunk from size $t \times d$ to $t \times k$, since we only cared about $\operatorname{span}\{e_1, \ldots, e_k\}$.

22.4.1 First approach: Appling a standard theorem

The event in (22.4.3) uses important quantities from linear algebra. For any $t \times k$ matrix R, define

$$\text{Maximum singular value:} \quad s_1(R) \ = \ \max_{x \in \mathbb{R}^k,\, x \neq 0} \frac{\|Rx\|}{\|x\|}$$

$$\text{Minimum singular value:} \quad s_k(R) \ = \ \min_{x \in \mathbb{R}^k,\, x \neq 0} \frac{\|Rx\|}{\|x\|}.$$

References: Wikipedia.

Using these definitions, (22.4.3) is equivalent to showing that

$$\Pr[\ 1 - \epsilon \leq s_k(R) \ \wedge \ s_1(R) \leq 1 + \epsilon\ ] \ \geq \ 1 - 2e^{-k}. \tag{22.4.4}$$

The singular values of Gaussian matrices are very well-studied, and the following bound is known.

Theorem 22.4.4. If R has size $t \times k$ where the entries are independent $N(0, 1/t)$ then

$$\Pr\left[1 - \frac{\sqrt{k} + z}{\sqrt{t}} \ \leq \ s_k(R) \ \wedge \ s_1(R) \ \leq \ 1 + \frac{\sqrt{k} + z}{\sqrt{t}} \right] \ \geq \ 1 - 2\exp(-z^2/2).$$

References: See Theorem 2.13 in Davidson-Szarek or (Vershynin, 2018, Theorem 4.6.1).

Taking $t = k/\epsilon^2$ and $z = \sqrt{k}$ this proves (22.4.4) (up to constant factors) which proves Theorem 22.4.2.

22.4.2 Second approach: A direct argument

Next we observe that the function $x \mapsto \|Rx\| / \|x\|$ is a continuous function on non-zero x. Thus if $\|Rx\| / \|x\| \in [1-\epsilon, 1+\epsilon]$ holds for some x, then it approximately holds for all nearby x. Also, since $\|Rx\| / \|x\|$ only depends on the direction of x, not its norm, it suffices to consider vectors with $\|x\| = 1$.

Define the Euclidean sphere

$$S = \left\{ x \in \mathbb{R}^k \,:\, \|x\| = 1 \right\}.$$

Since we only need to consider points $y \in S$, it seems conceivable that we could find *finitely* many points $P = \{p_1, \ldots, p_\ell\}$ such that *every* point in S is "nearby" to some p_i.

Definition 22.4.5. An ***ϵ-net*** of a set S is a set $P \subseteq S$ satisfying $\min_{p \in P} \|p - x\| \le \epsilon$ for all $x \in S$.

References: (Vershynin, 2018, Definition 4.2.1), Wikipedia.

The following fact is known.

Fact 22.4.6 (ϵ-net of sphere). The sphere S in $\mathbb{R}^k$ has an ϵ-net of size $(3/\epsilon)^k$.

References: (Vershynin, 2018, Corollary 4.2.13).

Let P be an ϵ-net of size $(3/\epsilon)^k = \exp\big(k \ln(3/\epsilon)\big)$. Then we apply the Johnson-Lindenstrauss lemma with error parameter ϵ to the points in P. Note that $\|p\| = 1$ for all $p \in P$, since $P \subseteq S$. By (22.2.7) and a union bound, we have

$$\|Rp\| \;\in\; [1-\epsilon, 1+\epsilon] \qquad \forall p \in P \tag{22.4.5}$$

with probability at least

$$1 - |P| \cdot \exp(-\epsilon^2 t/8) \;\ge\; 1 - \exp\big(k \ln(3/\epsilon) - \epsilon^2 t/8\big).$$

This probability is at least $1 - e^{-k}$ if $t = 16k \ln(3/\epsilon)/\epsilon^2$.

Now we show that, if the norm is (approximately) preserved on the ϵ-net, then it is preserved on the entire sphere.

Claim 22.4.7. Suppose that (22.4.5) holds. Then $\|Rx\| \le 1 + O(\epsilon)$ for all $x \in S$.

References: (Vershynin, 2018, Lemma 4.4.1).

Proof. Consider any $x^* \in \arg\max_{x \in S} \|Rx\|$. The existence of x^* comes from the Weierstrass extreme value theorem, since $x \mapsto \|Rx\|$ is continuous and S is a closed, bounded set.

Since P is an ϵ-net, there exists $p^* \in P$ such that $\|x^* - p^*\| \le \epsilon$. By (22.4.5), we have $\|Rp^*\| \in [1-\epsilon, 1+\epsilon]$. We now bound $\|Rx^*\|$ by relating it to $\|Rp^*\|$.

$$\begin{aligned}
\|Rx^*\| &\le \|Rp^*\| + \|R(x^* - p^*)\| && \text{(by the triangle inequality)} \\
&\le (1+\epsilon) + \|R(x^* - p^*)\| && \text{(by (22.4.5))} \\
&= (1+\epsilon) + \left\| R \frac{(x^* - p^*)}{\|x^* - p^*\|} \right\| \cdot \|x^* - p^*\| && \text{(normalizing the vector)} \\
&\le (1+\epsilon) + \|Rx^*\| \cdot \epsilon && \text{(since } x^* \text{ is a maximizer).}
\end{aligned}$$

Rearranging, we get $\|Rx^*\| \le \frac{1+\epsilon}{1-\epsilon}$. Since $\frac{1}{1-\epsilon} \le 1 + 2\epsilon$ (see Fact B.4.1), the result follows. □

A similar argument yields the following claim, which proves Theorem 22.4.2.

Claim 22.4.8. Suppose that (22.4.5) holds. Then $\|Rx\| \ge 1 - O(\epsilon)$ for all $x \in S$.

22.4.3 Subspace Embeddings with Fast JL

In the preceding discussion we have assumed that R is a Gaussian matrix, which simplified matters due to the rotational invariance (Fact 22.4.3), but unfortunately multiplication by R is slow. We now discuss how to replace R with a Fast JL matrix.

The desired conclusion of Theorem 22.4.2 is that

$$\Pr\left[\frac{\|Rx\|}{\|x\|} \in [1-\epsilon, 1+\epsilon] \quad \forall x \in U,\, x \neq 0\right] \geq 1 - 2e^{-k}.$$

Now let A be a $d \times k$ matrix whose columns are an orthonormal basis of U. Recall that P is an ϵ-net for S, the sphere in $\mathbb{R}^k$. Since A is an ***isometry*** (a distance-preserving bijective map) between $\mathbb{R}^k$ and U,

- $A \cdot S = \{\, Ax \,:\, x \in S \,\}$ is the sphere in U.
- $A \cdot P = \{\, Ap \,:\, p \in P \,\}$ is an ϵ-net for $A \cdot S$.

Now apply Theorem 22.3.2 with $\delta = e^{-k}/|P| \geq (\epsilon/3e)^k$ and $t = O(\log(d/\delta)^2 \log(1/\delta)/\epsilon^2)$. It follows that

$$\Pr\left[\, \|Rx\| \in [1-\epsilon, 1+\epsilon] \;\forall x \in A \cdot P \,\right] \;\geq\; 1 - 2e^{-k}.$$

An argument similar to Claim 22.4.7 implies that

$$\Pr\left[\, \|Rx\| \in [1-O(\epsilon), 1+O(\epsilon)] \;\forall x \in U \,\right] \tag{22.4.6}$$

$$= \Pr\left[\, \|Rx\| \in [1-O(\epsilon), 1+O(\epsilon)] \;\forall x \in A \cdot S \,\right] \;\geq\; 1 - 2e^{-k}. \tag{22.4.7}$$

This proves Theorem 22.4.2 with the weaker dimension of $t = O\big((k \log(d)/\epsilon)^3\big)$, but with a matrix R supporting multiplication in time $O(d \log d + t)$.

Discussion. See Section 2.1 of this survey, Section 8.7 of this survey or Section 6 of this survey.

Exercises

Exercise 22.1. Let R be a JL matrix of size $t \times d$, where the entries are independent $N(0, 1/t)$. Prove that $\mathrm{E}\left[\, R^\mathsf{T} R \,\right] = I$. This is similar to the notion of being in isotropic position.

Exercise 22.2. Let $x_1, \ldots, x_n$ be given vectors in $\mathbb{R}^d$, and let $\epsilon \in (0,1)$ be arbitrary. For some $t = O(\log(n)/\epsilon^2)$, define R to be a $t \times d$ matrix whose entries are independent $N(0, 1/t)$.

1. Suppose that $\|x_i\| \leq 1$ for all $i \in [n]$. Prove that, with probability at least $1 - 1/n$,

$$|x_i^\mathsf{T} x_j - x_i R^\mathsf{T} R x_j| \;\leq\; \epsilon \qquad \forall i, j.$$

2. Now assume no constraint on the norm of $x_1, \ldots, x_n$. Prove that, with probability at least $1 - 1/n$,

$$|x_i^\mathsf{T} x_j - x_i R^\mathsf{T} R x_j| \;\leq\; \epsilon \, \|x_i\| \, \|x_j\| \qquad \forall i, j.$$

Exercise 22.3. Prove Claim 22.3.3.

Hints:

- $\|Sy\|_2^2 = \sum_i (Sy)_i^2$, and these summands are independent.
- The expectation was already analyzed above.
- The Generalized Hoeffding bound (Theorem B.5.7) is recommended.

Chapter 23

Applications of Johnson-Lindenstrauss

In this chapter we will see several applications of the Johnson-Lindenstrauss lemma. These examples illustrate two key reasons why Johnson-Lindenstrauss is so useful.

- First, it's *oblivious* to the points being transformed. We can pick the matrix ahead of time, and nevertheless it is likely to work well on whatever points it is applied.
- If there is an algorithm that involves the Euclidean norm, and whose performance is exponential in the dimension, dimensionality reduction will improve its performance to polynomial in the dimension!

23.1 Streaming algorithms for ℓ_2

Recall the streaming model from Chapter 13. We will change the notation slightly. The algorithm receives a sequence of items $(a_1, a_2, \ldots, a_n)$, where each $a_i \in [d]$. The frequency vector $x \in \mathbb{Z}^d$ is

$$x_j \;=\; |\{\; i \;:\; a_i = j \;\}| \;=\; \text{number of occurrences of } j \text{ in the stream.}$$

The objective is to estimate some properties of x while using $O(\log(dn))$ space. Properties of interest include $\|x\|_p$, or the number of non-zero entries, etc.

Today we will give a simple algorithm to estimate $\|x\|_2$. As an example of a scenario where this would be useful, consider a database table $((a_1, b_1), \ldots, (a_n, b_n))$. A self-join with the predicate $a = a$ would output all triples (a, b, b') where (a, b) and (a, b') belong to the table. What is the size of this self-join? It is simply $\|x\|^2$, where x is the frequency vector for the a values in the table. So a streaming algorithm for estimating $\|x\|$ could be quite useful in database query optimization.

The Algorithm. The idea is very simple: instead of storing x explicitly, we will store a *dimensionality reduced* form of x. Let R be a random Gaussian matrix of size $t \times d$, divided by $\sqrt{t}$. (In other words, each entry is drawn from the distribution $N(0, 1/t)$. This is a rescaling of the linear map R defined in Section 22.2.) Instead of explictly maintaining the vector x, the algorithm maintains the vector $y = R \cdot x$.

Algorithm 23.1 Estimating the ℓ_2 norm of the frequency vector.

```
function EstimateL2(int n)
    Set t = O(1/ε²)
    Let R be a t × d matrix with independent entries drawn from N(0, 1/t)
    Let y be the zero vector in ℝ^t
    for i = 1, ..., m do
        Receive item a_i from the stream
        Add column a_i of R to y
    end for
    return ‖y‖
end function
```

At time step i, the algorithm receives the index $j = a_i$. This implicitly causes x_j to increase by 1. Since $y = R \cdot x$, the corresponding change in y is to add the j^{th} column of R to y.

To analyze this algorithm, we use the Johnson-Lindenstrauss lemma. In Eq. (22.2.7), we proved that

$$\Pr\left[\,(1-\epsilon)\|x\| \;\le\; \|y\| \;\le\; (1+\epsilon)\|x\|\,\right] \;\ge\; 1 - \exp(-\epsilon^2 t/8).$$

Thus, setting $t = \Theta(1/\epsilon^2)$, we conclude that $\|y\|$ gives a $(1+\epsilon)$ approximation of $\|x\|$ with constant probability. Alternatively, if we want y to give an accurate estimate at *each* of the n time steps, we can take $t = \Theta(\log(n)/\epsilon^2)$.

How much space does this algorithm take? The algorithm stores two objects: the vector y and the matrix R. The vector y uses $t = O(1/\epsilon^2)$ words of space, which is consistent with our goal of using very little space.

Unfortunately, the matrix R uses td words of space. Storing R explicitly is worse than storing the frequency vector x. (There is also the issue of how many bits of accuracy are needed when generating the Gaussian random variables, but we will ignore that.) However, it seems that there may be some benefits to storing R instead of x, because R contains purely random numbers. The values of R do not depend on values appearing in the stream.

- At first glance, it may seem that R needn't be stored at all: every time an entry of R is accessed, a new independent random variable could be generated. But this does not work: each time an item j appears in the stream, we must add the j^{th} column of R to y, and it must be *the same column* each time.

- Does it help to use the Fast JL transform $R = SHD$ introduced in Section 22.3? Storing this matrix R requires only $O(d+t)$ words of space, which is an improvement over the ordinary JL matrix, but still too much.

- In a practical implementation, R will be generated by a pseudorandom generator initialized by some seed. This has the advantage that we can regenerate columns of R at will by resetting the seed. This is likely to work well in practice, but may not have provable guarantees.

- There is another approach that has been studied by theorists, and has provable guarantees but is probably too complicated to use in practice. Theorists have developed provably good pseudo-random generators, but only for algorithms *that use a small amount of space*. Since streaming

algorithms do indeed use little space, this approach can indeed be used to regenerate the matrix R as necessary.

References: See this paper of Nisan.

- Another approach with theoretical guarantees is to generate R using special low-independence hash functions. A basic form of these ideas was discussed in Chapter 14. We may return to this in future chapters. See (Blum et al., 2018, Section 6.2.3) or Chapter 6 of these notes.

23.2 Euclidean nearest neighbor

The nearest neighbor problem is a classic problem involving high-dimensional data. Given points $P = \{p_1, ..., p_n\} \in \mathbb{R}^d$, the goal is to build a (static) data structure so that, given a query point $q \in \mathbb{R}^d$, we can quickly find i minimizing $\|q - p_i\|$. We focus on the Euclidean norm $\|\cdot\| = \|\cdot\|_2$, but this problem is interesting for many norms.

Trivial solutions. This problem can trivially be solved in polynomial time. We could do no processing of P, then for each query find the closest point by exhaustive search. This requires time $O(nd)$ for each query. An alternative approach is to use a k-d tree, which is a well-known data structure for representing geometric points. Unfortunately this could take $O(dn^{1-1/d})$ time for each query, which is only a substantial improvement over exhaustive search when the dimension d is a constant. This phenomenon, the failure of low dimensional methods when applied in high dimensions, is known as the "curse of dimensionality".

Overcoming the curse. We will show that the curse can be overcome through the use of randomization and approximation. The ***ϵ-approximate nearest neighbor*** problem (ϵ-NN) is defined as follows. Given a query point $q \in \mathbb{R}^d$, define

$$\text{OPT} \;=\; \min_{p \in P} \|p - q\|$$

we must find a point $\hat{p} \in P$ such that

$$\|\hat{p} - q\| \;\le\; \big(1 + O(\epsilon)\big) \cdot \text{OPT}. \tag{23.2.1}$$

Our goal is to preprocess P and produce a data structure of size $\text{poly}(n)$. Given a query point q, we wish to spend time say $O(d \log(n)/\epsilon^2)$ finding a point $p \in P$ satisfying (23.2.1). Imagine a setting where $n \approx 10^{10}$ is the number of web pages, $d \approx 10^3$ is the number of features describing those pages, and $\epsilon = 10^{-1}$ is the desired approximation. Our approach can answer a query in time $d \log(n)/\epsilon^2 \approx 10^6$, whereas the trivial approach requires time $nd \approx 10^{13}$.

The high-level idea of our solution is very simple. First, design an exhaustive search algorithm that requires space roughly $2^{O(d)}$ to solve ϵ-NN for d-dimensional data. Then, apply dimensionality reduction to reduce the data set to dimension $t \approx \log n$. Running the exhaustive search algorithm on the t-dimensional data only requires space $2^{O(t)} = n^{O(1)}$.

23.2.1 Point Location in Equal Balls

The first step is to reduce our problem to a simpler one, in which a query only needs to determine whether the closest point is at distance less than or greater than roughly r. This simpler problem is called the ***ϵ-Point Location in Equal Balls*** problem (ϵ-PLEB). It is defined as follows.

The input data is a collection of n points $P = \{p_1, \ldots, p_n\} \subset \mathbb{R}^d$. The notation $B(p, r)$, defined in (B.2.2), denotes the Euclidean ball of radius r around p. We will call $B(p, r)$ a small ball, and $B(p, (1+\epsilon)r)$ a big ball.

Given a query point $q \in \mathbb{R}^d$, the requirements are as follows:

- *Case 1:* $(1+\epsilon)r < \text{OPT}$.
  In this case, q is not in any big ball, i.e., $\nexists p_i$ with $q \in B(p_i, (1+\epsilon)r)$. We must output No.

- *Case 2:* $r < \text{OPT} \le (1+\epsilon)r$.
  In this case, q is in a big ball but not any small ball. We can either output No, or output Yes together with a point p_j with $q \in B(p_j, (1+\epsilon)r)$.

- *Case 3:* $\text{OPT} \le r$.
  In this case, q is in a small ball (i.e., $\exists p_i \in P$ with $q \in B(p_i, r)$). We must output Yes and a point p_j with $q \in B(p_j, (1+\epsilon)r)$.

Although there is some uncertainty about what happens in Case 2, we can draw some conclusions from the output.

- If the output is Yes, then we are always provided with a point p_j such that $q \in B(p_j, (1+\epsilon)r)$.

- If the output is No, then either Case 1 or Case 2 could apply. In either case,

$$\text{there is no } p \in P \text{ s.t. } q \in B(p, r). \tag{23.2.2}$$

23.2.2 Reduction of ϵ-NN to ϵ-PLEB

We now explain how to solve the ϵ-NN problem using a solution to the ϵ-PLEB problem. First scale the point set P so that the minimum interpoint distance is at least 1, then let Δ be the maximum interpoint distance. So $1 \le \|p - p'\| \le \Delta$ for all $p, p' \in P$.

Initialization. To initialize the data structure, we create an instance of ϵ-PLEB(r) for every radius r in

$$\mathcal{R} \;=\; \left\{(1+\epsilon)^0, (1+\epsilon)^1, (1+\epsilon)^2, \ldots, \Delta\right\}. \tag{23.2.3}$$

Question 23.2.1. How many different radii are there?

Answer.

There are $\log_{1+\epsilon}(\Delta) = \frac{\ln(\Delta)}{\ln(1+\epsilon)} \le O(\log(\Delta)/\epsilon)$, by Exercise B.1.

Queries. Fix a query point $q \in \mathbb{R}^d$. Let us consider what the different outputs of ϵ-PLEB(r) might be for the radii in $\mathcal{R}$.

Example 1:

Radius	$(1+\epsilon)^0$	$(1+\epsilon)^1$	$(1+\epsilon)^2$	$(1+\epsilon)^3$	$(1+\epsilon)^4$	$\cdots$	Δ
Output	No	No	No	Yes	Yes	$\cdots$	Yes
Case	1	1	2	3	3	$\cdots$	3

Example 2:

Radius	$(1+\epsilon)^0$	$(1+\epsilon)^1$	$(1+\epsilon)^2$	$(1+\epsilon)^3$	$(1+\epsilon)^4$	$\cdots$	Δ
Output	No	No	No	Yes	Yes	$\cdots$	Yes
Case	1	1	1	2	3	$\cdots$	3

Figure 23.1: Some possible outcomes of the ϵ-PLEB subroutine. Note that there can be at most one radius that results in Case 2.

Claim 23.2.2. For the radii in $\mathcal{R}$, all No outputs occur for smaller radii than all Yes outputs.

Proof. If radius r falls into Case 1, then $\nexists p \in P$ such that $q \in B(p,(1+\epsilon)r)$. Every smaller radius also satisfies that condition, and therefore also falls into Case 1. So r and all smaller radii produce the output No.

If radius r falls into Case 3, then $\exists p \in P$ such that $q \in B(p,r)$. Every larger radius also satisfies that condition, and therefore also falls into Case 3. So r and all larger radii produce the output Yes.

It remains to consider Case 2. At most one radius in $\mathcal{R}$ can satisfy (**??**), and therefore fall into Case 2. With this radius, either Yes or No could be output, but both possibilities are consistent with the claim. □

Given any query point q, let $\hat{r}$ be the minimum radius r for which ϵ-PLEB(r) says Yes. Let $\hat{p} \in P$ be the returned point for which $q \in B(\hat{p},(1+\epsilon)\hat{r})$. By Claim 23.2.2, $\hat{r}$ can be found by binary search.

Claim 23.2.3. $\hat{p}$ satisfies (23.2.1), and therefore is a solution to the ϵ-NN problem.

Proof. The requirements of the ϵ-PLEB$(\hat{r})$ subroutine ensure that that $\|\hat{p}-q\| \le \hat{r}(1+\epsilon)$.

Recall that $\hat{r}$ is the *minimum* radius that said Yes. If $\hat{r}=1$ then clearly p satisfies (23.2.1). Otherwise, ϵ-PLEB$(\hat{r}/(1+\epsilon))$ output No. It follows (from (23.2.2)) that there is no point $p' \in P$ with $q \in B(p', r/(1+\epsilon))$. In other words,

$$\frac{r}{1+\epsilon} \;<\; \min_{p' \in P} \left\|p'-q\right\|.$$

Thus $\hat{p}$ satisfies

$$\|\hat{p}-q\| \;\le\; (1+\epsilon)\hat{r} \;<\; (1+\epsilon)^2 \cdot \min_{p' \in P} \left\|p'-q\right\|.$$

Since $(1+\epsilon)^2 \le 1+3\epsilon$ (see Fact B.1.2), this proves (23.2.1). □

Question 23.2.4. How many radii must the binary search consider in order to determine r^*?

Answer.

$$O\left(\log\left(\log(\Delta)/\epsilon\right)\log|\mathcal{R}|\right) \;=\; \log_2|\mathcal{R}| \cdot$$

23.2.3 Solving PLEB

The main idea here is quite simple. We discretize the space, then use a hash table to identify locations belonging to a ball.

Preprocessing. In more detail, the preprocessing step for ϵ-PLEB(r) proceeds as follows. We first partition the space into cuboids (d-dimensional cubes) of side length $\epsilon r/\sqrt{d}$. Note that[1] the Euclidean diameter of a cuboid is its side length times $\sqrt{d}$, which is ϵr. Each cuboid is identified by a canonical point, say the minimal point contained in the cuboid. We then create a hash table, initially empty. For each point p_i and each cuboid C that intersects $B(p_i, r)$, we insert the (key, value) pair (C, p_i) into the hash table.

Queries. Now consider how to perform a query for a point q. The first step is to determine the cuboid C that contains q, by simple arithmetic. Next, we look up C in the hash table. If there are no matches, that means that no ball $B(p_i, r)$ intersects C, and therefore q is not contained in any ball of radius r (a small ball). So, by the requirements of ϵ-PLEB(r), we can say No.

Suppose that C is in the hash table. Then the hash table can return an arbitrary pair (C, p_j), which tells us that $B(p_j, r)$ intersects C. By the triangle inequality, the distance from p_j to q is at most r plus the diameter of the cuboid, which is ϵr. So $\|p_j - q\| \le (1+\epsilon)r$, which means that q is contained in the big ball around p_j. By the requirements of ϵ-PLEB(r), we can say Yes and we can return the point p_j.

Time and Space Analysis. To analyze this algorithm, we first need to determine the number of cuboids that intersect a ball of radius r. From (B.2.3), the volume of $B(p, r)$ is $2^{O(d)} r^d / d^{d/2}$. On the other hand, the volume of a cuboid is $(\epsilon r/\sqrt{d})^d$. So the number of cuboids that intersect this ball is roughly

$$\frac{2^{O(d)} r^d / d^{d/2}}{(\epsilon r/\sqrt{d})^d} \;=\; O(1/\epsilon)^d.$$

Therefore the time and space used by the preprocessing step is roughly $O(1/\epsilon)^d$.

To perform a query, we just need to compute the cuboid containing q then look up that cuboid in the hash table. This takes $O(d)$ time if we use the perfect hashing of Section 15.4. This cannot be improved — $\Omega(d)$ time is clearly necessary, since we must examine most coordinates of the vector q.

Unfortunately the preprocessing time and space is exponential in d, which is another example of the curse of dimensionality. The next section addresses this via dimensionality reduction.

23.2.4 Applying Dimensionality Reduction

The next step is to apply the Johnson-Lindenstrauss lemma to map our points to a low-dimensional space. Let $t = O(\log(n)/\epsilon^2)$. Let R be a random $t \times d$ matrix whose entries have distribution $N(0, 1/t)$. For any fixed query point q, Theorem 22.2.1 says that the linear map R approximately preserves pairwise distances between all points in $P \cup \{q\}$ with probability $1 - 1/n$.

The analysis of ϵ-PLEB changes as follows. The preprocessing step must apply the matrix to all points in P, which takes time $O(dnt)$. The time to set up the hash table improves to $O(1/\epsilon)^t = n^{O(\log(1/\epsilon)/\epsilon^2)}$. So assuming ϵ is a constant, the preprocessing step runs in polynomial time. Each query must also apply the Johnson-Lindenstrauss matrix to the query point, which takes time $O(td) = O(d \log(n)/\epsilon^2)$.

Finally, we analyze the reduction which allowed us to solve ϵ-NN. Recall that the preprocessing step simply instantiates ϵ-PLEB(r) for all values of r in $\mathcal{R}$. As discussed in Question 23.2.1, the number of different instances is $|\mathcal{R}| = O\big(\log(\Delta)/\epsilon\big)$, so the total preprocessing time is

$$n^{O(\log(1/\epsilon)/\epsilon^2)} \;\cdot\; O\big(\log(\Delta)/\epsilon\big).$$

[1] This can be seen from Fact B.2.1, because the cuboid has ℓ_∞ diameter $\epsilon r/\sqrt{d}$.

This is polynomial time assuming $\Delta \le 2^{n^{\text{poly}(1/\epsilon)}}$ and ϵ is a constant. Each query must perform binary search over different radii to find $\hat{r}$. Each query to ϵ-PLEB now takes time $O(t)$. So the total query time is

$$O\big(dt\big) + O(\log|\mathcal{R}|) \cdot O(t) \;=\; O\Big(t \cdot \big(d + \log(\log(\Delta)/\epsilon)\big)\Big).$$

This is roughly $O(d \log(n)/\epsilon^2)$ so long as Δ is reasonable, say $\Delta \le 2^{2^d}$.

23.2.5 Discussion

Some of the earliest papers on approximate nearest neighbour are Kleinberg, Indyk and Motwani and Kushilevitz-Ostrovsky-Rabani. The algorithm we present is due to Indyk and Motwani. It seems quite inefficient, but of course there have been numerous improvements. The Indyk-Motwani paper itself also proposed the alternative approach of "locality sensitive hashing", which was touched upon in Section 12.3. Some modern techniques for nearest neighbor are described in the survey of Andoni and Indyk.

23.3 Fast Least-Squares Regression

Given matrix A of size $d \times m$ (with $d \ge m$) and a vector $b \in \mathbb{R}^d$, the goal is to find

$$x^* \in \operatorname*{argmin}_{x \in \mathbb{R}^m} \|Ax - b\| .$$

For some background motivation, see CPSC 340 Lecture 12 and Lecture 13.

Mathematically, this is quite a simple problem to solve: $\|Ax - b\|_2^2$ is a convex, differentiable quadratic function, so the minimizers are the points with zero gradient. After some simple vector calculus, this amounts to solving to the linear system

$$A^\mathsf{T} A x = A^\mathsf{T} b. \tag{23.3.1}$$

References: UBC MATH 307 notes, (Trefethen and Bau, 1997, Theorem 11.1).

The focus of this section is algorithms for computationally solving (23.3.1).

Conventional Solutions. There are several standard approaches to solving (23.3.1).

- Naively, one could compute $A^\mathsf{T} A$ and $A^\mathsf{T} b$ explicitly and use Gaussian elimination to solve the problem exactly.

  Question 23.3.1. How much time does this take?

  Answer.
  $O(dm^2)$ time to compute $A^\mathsf{T} A$, $O(dm)$ time to compute $A^\mathsf{T} b$, and $O(m^3)$ time to perform Gaussian elimination. Since $d \ge m$, this is $O(dm^2)$ time.
  This can be improved to $O(dm^{1.38})$ by a galactic algorithm.

- Since $A^\mathsf{T} A$ is symmetric, we could use the Cholesky decomposition instead of Gaussian elimination. This approach can also solve the problem exactly in $O(dm^2)$ time.

  References: (Trefethen and Bau, 1997, Algorithm 11.1).

Theorem 23.3.2. There is a randomized algorithm to find $\hat{x}$ satisfying

$$\|A\hat{x} - b\| \;\leq\; (1 + O(\epsilon)) \cdot \|Ax^* - b\| \tag{23.3.2}$$

in time

$$O(dm \log d) + \text{poly}(m \log d/\epsilon).$$

The failure probability is at most $2e^{-m}$.

The key point is that the leading term has improved from $O(dm^2)$ to only $O(dm \log d)$. This gives an improvement over conventional results if $\log d \ll m \ll d$ and ϵ is modest.

The main idea is as follows. We will use a Fast JL matrix R to reduce the number of rows of A and b to $t = O((m \log d/\epsilon)^3)$; see Section 22.4.3. After the dimension reduction, we then solve the problem by conventional methods.

Algorithm 23.2 Fast algorithm for least-squares regression. A has size $d \times m$ and b has length d.

1: **function** FastLeastSquares(matrix A, vector b, float ϵ)
2: Let $t \leftarrow O\big((m \log d/\epsilon)^3\big)$
3: Let R be a Fast JL matrix of size $t \times d$
4: Compute the matrix RA and the vector Rb
5: Find $\hat{x} \in \text{argmin}_{x \in \mathbb{R}^m} \|RAx - Rb\|$, by conventional methods.
6: **return** $\hat{x}$
7: **end function**

Runtime analysis. The key computations are on line 4. Computing RA by ordinary matrix multiplication would take time $O(tdm)$, which is too slow. However, since R is a Fast JL matrix, we can do this much more quickly. Recall from Theorem 22.3.2 that matrix-vector multiplication with R takes time $O(d \log d + t)$. So we can construct RA by separately multiplying R by each column of A. Since there are m columns, this takes $O(dm \log d + tm)$ time. The vector Rb can be computed in the same way.

Next, on line 5, we can find $\hat{x}$ by conventional methods in $O(tm^2)$ time. Plugging in the definition of t, the total runtime is

$$O(dm \log d + tm) + O(tm^2) \;=\; O(dm \log d) + \text{poly}(m \log d/\epsilon).$$

Analysis. Define

$$U \;=\; \text{span}\big(\{\, Ax \,:\, x \in \mathbb{R}^m \,\} \cup \{b\}\big). \tag{23.3.3}$$

The vectors in U all have length d, but U is a subspace of dimension at most $m + 1$. By (22.4.7), the matrix R approximately preserves the norm of all vectors in U, with probability at least $1 - 2e^{-m}$. By definition of $\hat{x}$, we have

$$\|RA\hat{x} - Rb\| \;\leq\; \|RAx^* - Rb\| \;=\; \|R(Ax^* - b)\| \;\leq\; (1 + \epsilon) \cdot \|Ax^* - b\|\,,$$

since $Ax^* - b \in U$. On the other hand,

$$\|RA\hat{x} - Rb\| \;=\; \|R(A\hat{x} - b)\| \;\geq\; (1 - \epsilon)\,\|A\hat{x} - b\|\,,$$

since $A\hat{x} - b \in U$. Putting these inequalities together, we obtain

$$\|A\hat{x} - b\| \;\leq\; \frac{1 + \epsilon}{1 - \epsilon} \cdot \|Ax^* - b\|\,.$$

Since $\frac{1}{1-\epsilon} \leq 1 + 2\epsilon$ (see Fact B.4.1), this completes the proof of Theorem 23.3.2.

Discussion. This algorithm is due to Tamas Sarlos. See also Section 10.3 of this survey.

There have been many improvements. Let $\text{nnz}(A)$ denote the number of non-zero entries in the matrix A. It is known that the guarantee (23.3.2) can be achieved in time

$$O(\text{nnz}(A)) + \text{poly}(m/\epsilon).$$

See, for example, this paper or Theorem 21 of this survey.

23.4 Approximate Matrix Multiplication

Let A and B be matrices of size $n \times n$. Computing the product AB exactly takes $O(n^3)$ time by conventional methods, or $O(n^{2.38})$ by a galactic algorithm. The following algorithm is much faster, and is based on a very simple idea:

> *reduce the dimension of the matrices before multiplying them.*

Algorithm 23.3 Algorithm to estimate AB. The matrices A and B have size $n \times n$.

```
1: function FastMatMul(matrix A, B, float ε)
2:     Set t = O(log(n)/ε^2)
3:     Let R be a t × n matrix with independent entries drawn from N(0, 1/t)
4:     return AR^T RB
5: end function
```

Question 23.4.1. What is the runtime of this algorithm?

Answer.
It depends on the order in which the matrices are multiplied. Multiplying $R^\mathsf{T}R$ first will give runtime $O(n^3)$. However, first multiplying AR^T, then RB takes time $O(n^2t)$, and finally combining them takes time $O(n^2t) = O(n^2 \log(n)/\epsilon^2)$.

One appealing aspect of this algorithm is the following claim.

Claim 23.4.2. FastMatMul(A, B, ϵ) is an unbiased estimator of AB.

The proof of this claim is Exercise 23.1.

Notation. The entry of A in the i^{th} row and j^{th} column is denoted $A_{i,j}$. To refer to rows and columns, we will use the following handy notation. Thinking of $*$ as a "wildcard" character, we will let $A_{i,*}$ denote the i^{th} row of A and let $A_{*,k}$ denote the k^{th} column of A. Using this notation, the entries of the product AB can be expressed as a dot product as follows.

$$(AB)_{i,k} \;=\; \sum_{j=1}^{n} A_{i,j}B_{j,k} \;=\; A_{i,*}B_{*,k}$$

Definition 23.4.3. For a matrix A, its ***Frobenius norm*** $\|A\|_F$ is defined by

$$\|A\|_F^2 \;=\; \sum_{i,j\in[n]} (A_{i,j})^2 \;=\; \sum_{i\in[n]} \|A_{i,*}\|_2^2 \;=\; \sum_{j\in[n]} \|A_{*,j}\|_2^2 . \tag{23.4.1}$$

This is just the Euclidean norm of A when viewed as a vector of length n^2.

References: (Trefethen and Bau, 1997, page 22), (Vershynin, 2018, Section 4.1.3), (Murphy, 2022, Section 7.1.3.2), (Blum et al., 2018, Section 12.8.5), Wikipedia.

Our main theorem is that FastMatMul(A, B, ϵ) is a good approximation for AB, with error measured in the Frobenius norm.

Theorem 23.4.4. With probability at least $1 - 1/n$,

$$\|AB - AR^\mathsf{T}RB\|_F \;\le\; \epsilon \|A\|_F \|B\|_F .$$

Proof. We apply dimensionality reduction to the rows of A and the columns of B. Define

$$\mathcal{X} \;=\; \{\, (A_{i,*})^\mathsf{T} \,:\, i \in [n] \,\} \cup \{\, B_{*,k} \,:\, k \in [n] \,\} .$$

Exercise 22.2 shows that, with probability $1 - 1/n$,

$$\begin{aligned} & |x^\mathsf{T}x' - x^\mathsf{T}R^\mathsf{T}Rx'| \;\le\; \epsilon \|x\| \|x'\| && \forall x,\, x' \in \mathcal{X} \\ \implies \quad & |A_{i,*}B_{*,k} - A_{i,*}R^\mathsf{T}RB_{*,k}| \;\le\; \epsilon \|A_{i,*}\| \|B_{*,k}\| && \forall i,\, k \in [n]. \end{aligned}$$

Squaring and summing over i, k, we have

$$\begin{aligned} \sum_{i,k\in[n]} \big(\underbrace{A_{i,*}B_{*,k} - A_{i,*}R^\mathsf{T}RB_{*,k}}_{=(AB - AR^\mathsf{T}RB)_{i,k}} \big)^2 \;&\le\; \epsilon^2 \sum_{i,k\in[n]} \|A_{i,*}\|^2 \|B_{*,k}\|^2 \\ &=\; \epsilon^2 \Big(\sum_{i\in[n]} \|A_{i,*}\|^2 \Big) \Big(\sum_{k\in[n]} \|B_{*,k}\|^2 \Big). \end{aligned}$$

The theorem follows by taking the square root and using the identity (23.4.1). □

Exercises

Exercise 23.1. Prove Claim 23.4.2.

Part III

Back matter

Appendix B

Mathematical Background

B.1 Miscellaneous Facts

Fact B.1.1 (Stirling's Approximation).

$$e\Big(\frac{n}{e}\Big)^n \;<\; n! \;<\; en\Big(\frac{n}{e}\Big)^n$$

References: Wikipedia.

Fact B.1.2. For $0 \le \epsilon \le 1$, we have $(1+\epsilon)^2 \le 1 + 3\epsilon$.

Proof. Expand $(1+\epsilon)^2$ as $1 + 2\epsilon + \epsilon^2$, and use $\epsilon^2 \le \epsilon$ when $\epsilon \le 1$. □

B.2 Geometry

Norms. For a vector $x \in \mathbb{R}^d$, its ℓ_p norm is defined to be

$$\begin{aligned} \|x\|_p &= \Big(\sum_{i=1}^{d} |x_i|^p\Big)^{1/p} \qquad \forall p \in [1,\infty) \\ \|x\|_\infty &= \max_{1\le i\le d} |x_i| \end{aligned}$$

Every norm $\|\cdot\|$ satisfies the

$$\text{Triangle inequality:} \qquad \|a+b\| \le \|a\| + \|b\|\,.$$

The ℓ_2 norm is also called the ***Euclidean norm***. One useful identity it satisfies is

$$\|a-b\|_2^2 \;=\; \|a\|_2^2 - 2a^{\mathsf{T}}b + \|b\|_2^2 \qquad \forall a, b \in \mathbb{R}^n. \tag{B.2.1}$$

This identity might not have a canonical name, but some possibilities are the *generalized Pythagoras identity* or the *law of cosines*.

References: Apostol "Calculus, Volume II", page 17.

Let

$$B(p,r) \;=\; \{\, x \,:\, \|p-x\| \le r \,\} \tag{B.2.2}$$

denote the Euclidean ball in $\mathbb{R}^d$ around p of radius r. If d is even, the ***volume*** of $B(p,r)$ is

$$\operatorname{vol} B(p,r) \;=\; \frac{\pi^{d/2} r^d}{(d/2)!}.$$

References: (Blum et al., 2018, Section 2.4.1), Wikipedia.

By Stirling's formula, we have the bound

$$\operatorname{vol} B(p,r) \;\le\; \frac{\pi^{d/2} r^d}{(d/2e)^{d/2}} \;=\; \frac{(2e\pi)^{d/2} r^d}{d^{d/2}}. \tag{B.2.3}$$

The following fact allows us to compare norms.

Fact B.2.1. For all $x \in \mathbb{R}^d$,

$$\|x\|_p \;\le\; \|x\|_r \;\le\; d^{1/r-1/p} \cdot \|x\|_p \qquad \forall 1 \le r \le p \le \infty.$$

In particular, the most useful cases are

$$\begin{aligned} \|x\|_1 \;&\ge\; \|x\|_2 \;\ge\; \|x\|_\infty \\ \frac{1}{\sqrt{d}} \|x\|_1 \;&\le\; \|x\|_2 \;\le\; \sqrt{d} \cdot \|x\|_\infty . \end{aligned}$$

References: Wikipedia.

B.3 Facts from Convex Analysis

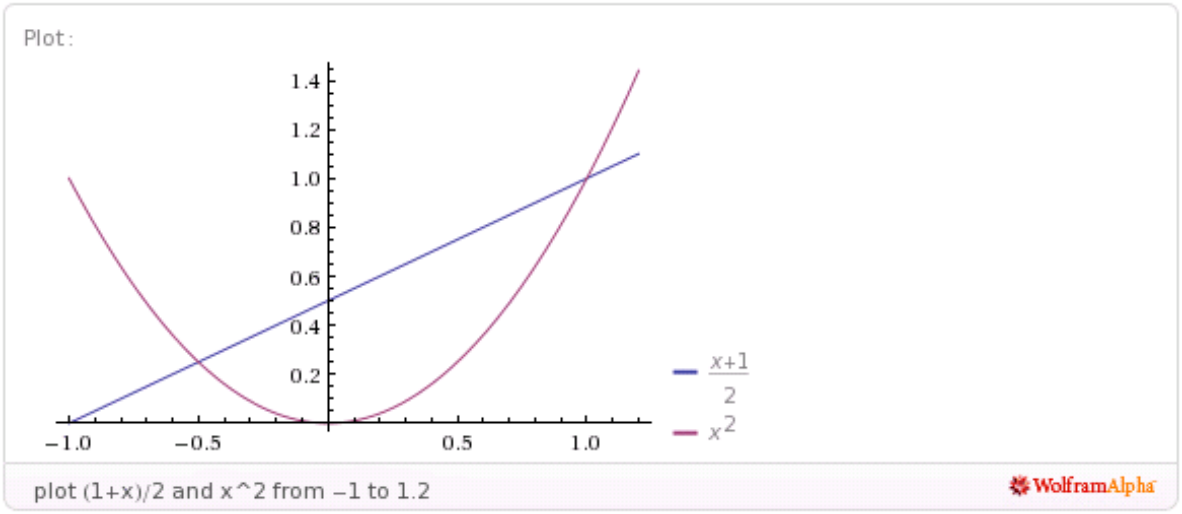
Let $f : S \to \mathbb{R}$ be a function defined on an interval $S \subseteq \mathbb{R}$.

Definition B.3.1. We say that f is convex on S if

$$f(\lambda x + (1-\lambda)y) \;\le\; \lambda f(x) + (1-\lambda) f(y)$$

for all $x, y \in S$ and all $\lambda \in [0,1]$.

Geometrically, this says that the secant line between the points $(x, f(x))$ and $(y, f(y))$ lies above the function f. The following example illustrates that $f(x) = x^2$ is convex.



An equivalent statement of this definition is as follows.

Fact B.3.2. Suppose $f : S \to \mathbb{R}$ is convex. Then, for any points $x, y \in S$ with $x < y$, we have

$$f(z) \;\leq\; \frac{f(y) - f(x)}{y - x} \cdot (z - x) + f(x)$$

for all $z \in [x, y]$.

For sufficiently nice functions, one can easily determine convexity by looking at its second derivative.

Fact B.3.3. Suppose $f : S \to \mathbb{R}$ is twice differentiable. Then f is convex if and only if the second derivative f'' is non-negative on the interior of S.

In our example above we used $f(x) = x^2$. Its second derivative is $f''(x) = 2$, which is non-negative, so f is convex.

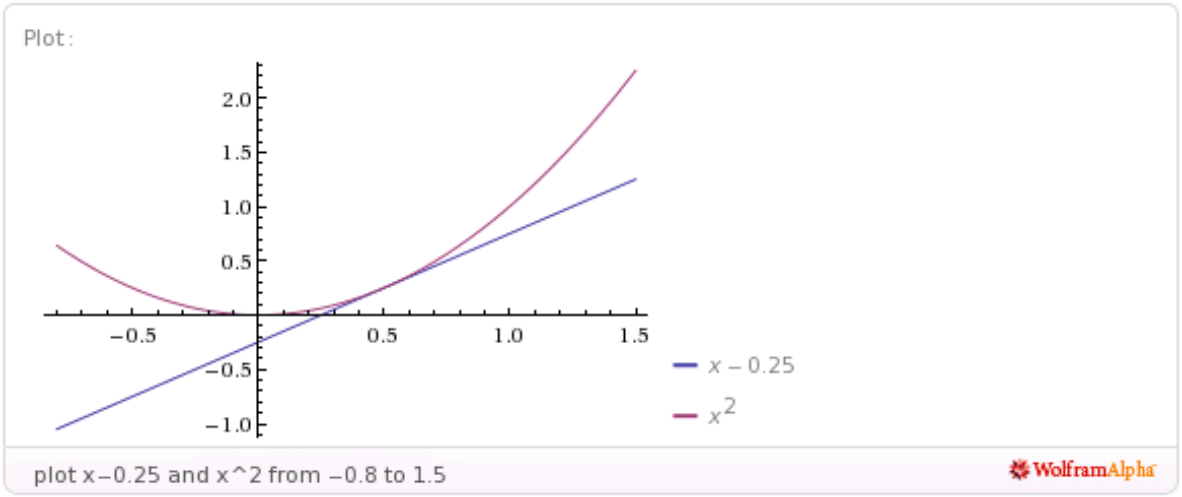
The next fact says that, for convex functions, the linear approximation at any point lies beneath the function.

Fact B.3.4 (Subgradient Inequality). Suppose $f : S \to \mathbb{R}$ is convex, and f is differentiable at a point $x \in S$. Then

$$f(y) \;\geq\; f(x) + f'(x)(y - x),$$

for any point $y \in S$.

The following example illustrates this for $f(x) = x^2$ at the point $x = 0.5$.



B.4 Various Inequalities from Convexity

In the analysis of randomized algorithms, we frequently use various inequalities to simplify expressions or make them easier to work with. These inequalities usually follow by convexity arguments and basic calculus. For example, let us revisit the following familiar fact.

Fact A.2.5 (Approximating e^x near zero). For all real numbers x,

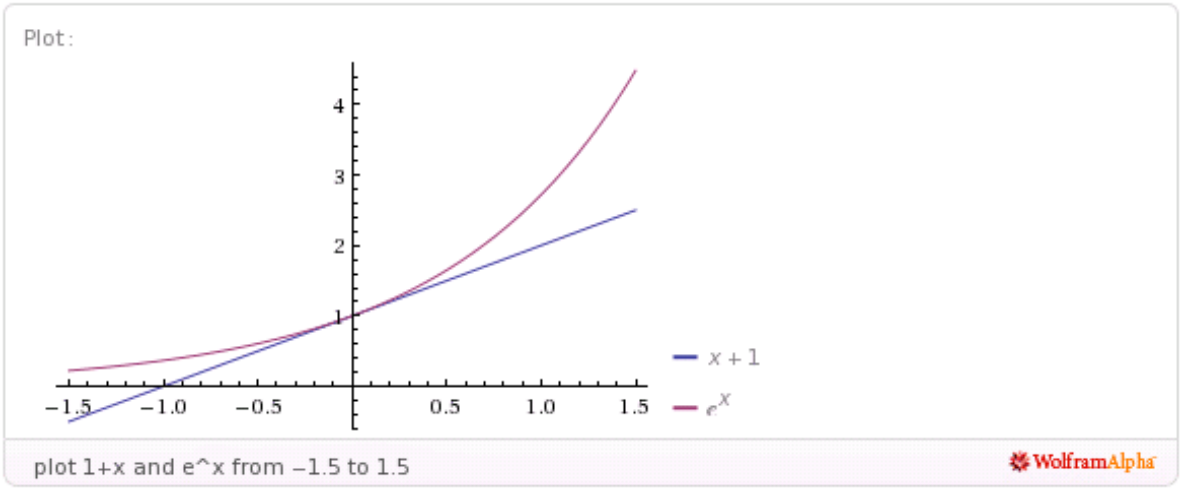
$$1 + x \leq e^x.$$

Moreover, for x close to zero, we have $1 + x \approx e^x$.

Proof. Convexity of e^x follows from Fact B.3.3 since its second derivative is e^x, which is non-negative on $\mathbb{R}$. Applying Fact B.3.4 at the point $x = 0$, we obtain

$$f(y) \;\geq\; f'(x) \cdot (y - x) + f(x) \;=\; y + 1.$$

□



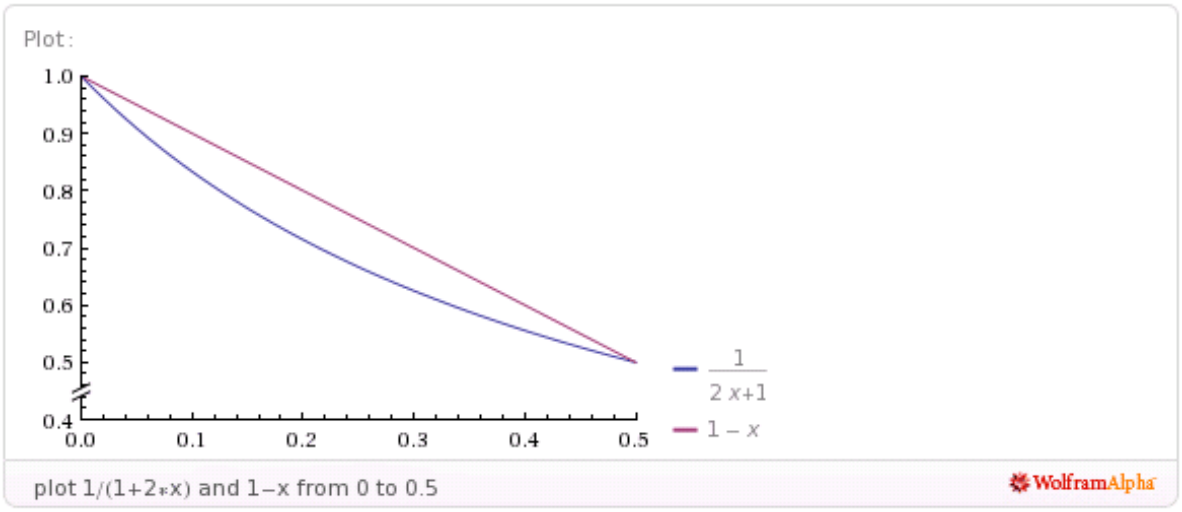
Fact B.4.1.

$$\frac{1}{1+2z} \;\leq\; 1 - z \qquad \forall z \in [0, 1/2].$$

Proof. Convexity of $f(z) = 1/(1+2z)$ on the set $S = (0,\infty)$ follows from Fact B.3.3 since its second derivative is $8/(1+2z)^3$, which is non-negative on S. Applying Fact B.3.2 at $x = 0$ and $y = 1/2$, we obtain

$$\begin{aligned} f(z) &\le \frac{f(y)-f(x)}{y-x}\cdot(z-x)+f(x) \\ &= \frac{1/2-1}{1/2}\cdot(z-0)+1 \;=\; 1-z \qquad \forall z\in[0,1/2]. \end{aligned}$$

□

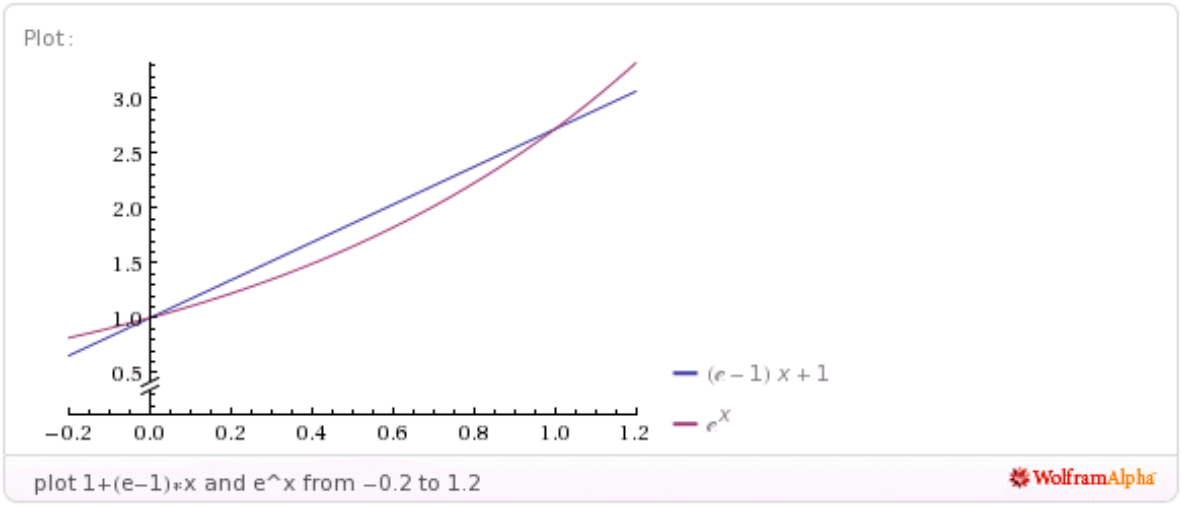


Fact B.4.2. Fix any $c > 0$. Then $c^z \le 1 + (c-1)z$ for all $z \in [0,1]$.

Proof. The second derivative of c^z is $c^z \cdot \ln^2(c)$, which is non-negative, so c^z is convex. Applying Fact B.3.2 at the points $x = 0$ and $y = 1$, we obtain

$$c^z \;\le\; \frac{c^y - c^x}{y-x}\cdot(z-x)+c^x \;=\; (c-1)\cdot z + 1.$$

□



Exercises

Exercise B.1. Prove that

$$\begin{aligned} \ln(1+z) &\leq z && \text{for } z > -1 \\ \ln(1+z) &\geq \ln(2)z \geq z/2 && \text{for } z \in [0,1] \end{aligned}$$

B.5 Probability

B.5.1 Variance

Definition B.5.1. The ***variance*** of a random variable X is defined to be

$$\mathrm{Var}[X] = \mathrm{E}\left[(X - \mathrm{E}[X])^2\right].$$

References: (Lehman et al., 2018, Definition 20.2.2), (Cormen et al., 2001, Equation (C.27)), (Motwani and Raghavan, 1995, page 443), (Mitzenmacher and Upfal, 2005, Definition 3.2), (Grimmett and Stirzaker, 2001, Definition 3.3.5), (Durrett, 2019, page 29).

Fact B.5.2. Variance satisfies the following identity.

$$\mathrm{Var}[X] = \mathrm{E}\left[X^2\right] - \mathrm{E}[X]^2. \tag{B.5.1}$$

Consequently,

$$\mathrm{Var}[X] = \mathrm{E}\left[X^2\right] \quad \text{if} \quad \mathrm{E}[X] = 0. \tag{B.5.2}$$

References: (Lehman et al., 2018, Definition 20.3.1), (Cormen et al., 2001, Equation (C.27)), (Motwani and Raghavan, 1995, Proposition C.8), (Mitzenmacher and Upfal, 2005, Definition 3.2), (Grimmett and Stirzaker, 2001, page 51), (Durrett, 2019, Equation (1.6.2)), (Klenke, 2008, Definition 5.1(iii)).

Proof.

$$\begin{aligned} \mathrm{Var}[X] &= \mathrm{E}\left[(X - \mathrm{E}[X])^2\right] && \text{(by definition)} \\ &= \mathrm{E}\left[X^2 - 2X\,\mathrm{E}[X] + \mathrm{E}[X]^2\right] && \text{(expand the quadratic)} \\ &= \mathrm{E}\left[X^2\right] - 2\,\mathrm{E}[X]\,\mathrm{E}[X] + \mathrm{E}[X]^2 && \text{(linearity of expectation)} \\ &= \mathrm{E}\left[X^2\right] - \mathrm{E}[X]^2. && \text{(simplifying).} \end{aligned}$$

This proves (B.5.1). The equation (B.5.2) is immediate. □

Fact B.5.3. Let $G_1, \ldots, G_d$ be independent random variables with finite variance. Let $\sigma_1, \ldots, \sigma_d \in \mathbb{R}$ be arbitrary. Then $\mathrm{Var}\left[\sum_i \sigma_i G_i\right] = \sum_i \sigma_i^2 \mathrm{Var}[G_i]$.

References: (Lehman et al., 2018, Lemma 20.3.4 and Lemma 20.3.8), (Cormen et al., 2001, page 1200), (Mitzenmacher and Upfal, 2005, Theorem 3.5 and Exercise 3.4), (Grimmett and Stirzaker, 2001, Theorem 3.3.11).

B.5.2 Gaussian random variables

One of the most important continuous distributions is the ***Gaussian distribution***. It is also called the ***Normal distribution***. This distribution has two real parameters, its mean (denoted μ) and its variance (denoted σ^2). The distribution with those parameters is denoted $N(\mu, \sigma^2)$.

References: Wikipedia.

One useful property is that sums of Gaussians are also Gaussian.

Fact B.5.4. Let $g_1, \ldots, g_d$ be independent random variables where g_i has distribution $N(0,1)$. Then, for any scalars $\sigma_1, \ldots, \sigma_d$, the sum $\sum_{i=1}^d \sigma_i g_i$ has distribution $N(0, \sum_{i=1}^d \sigma_i^2) = N(0, \|\sigma\|_2^2)$.

References: (Grimmett and Stirzaker, 2001, Example 4.8.3), (Durrett, 2019, Theorem 2.1.20 and Corollary 3.3.13), Wikipedia.

Remark B.5.5. Fact B.5.4 can be viewed in the more abstract context of ***stable random variables***. The Gaussian distribution is 2-stable. More generally, if X is an α-stable RV, and $X_1, \ldots, X_d$ are independent copies of X, then $\sum_{i=1}^d \sigma_i X_i$ has the distribution $\|\sigma\|_\alpha X$. See Equation (1.8) in "Stable Distributions: Models for Heavy Tailed Data" with $\beta = 0$, $\gamma_i = \sigma_i$ and $\delta = 0$.

References: (Durrett, 2019, Section 3.8), Wikipedia.

A useful fact about the Gaussian distribution is the following bound on its right tail.

Fact B.5.6 (Gaussian tail bound). Let X have the distribution $N(0,1)$. Let $x > 0$. Then

$$\frac{1}{\sqrt{2\pi}}(x^{-1} - x^{-3}) \exp(-x^2/2) \;\le\; \Pr[\, X \ge x \,] \;\le\; \frac{1}{\sqrt{2\pi}} x^{-1} \exp(-x^2/2).$$

References: (Vershynin, 2018, Proposition 2.1.2), (Durrett, 2019, Theorem 1.2.6).

B.5.3 More concentration

Theorem B.5.7 (Hoeffding's General Inequality). Let $X_1, \ldots, X_n$ be independent random variables where $X_i \in [a_i, b_i]$. Let $X = \sum_{i=1}^n X_i$. Then

$$\Pr\left[\left| \sum_{i=1}^n X_i - \mathrm{E}[\, X \,] \right| \ge s \right] \;\le\; 2 \exp\Big(-2 \frac{s^2}{\sum_{i=1}^n (b_i - a_i)^2} \Big).$$

In particular, for any desired $q \in (0,1)$, setting $s = \sqrt{\ln(2/q) \sum_{i=1}^n (b_i - a_i)^2 / 2}$ gives

$$\Pr[\, |\textstyle\sum_i X_i - \mathrm{E}[\, X \,]| \ge s \,] \;\le\; q.$$

References: (McDiarmid, 1998, Theorem 2.5), (Vershynin, 2018, Theorem 2.2.6), (Boucheron et al., 2012, Theorem 2.8), (Roch, 2020, Theorem 2.40), (Dubhashi and Panconesi, 2009, Problem 1.9), (Grimmett and Stirzaker, 2001, Theorem 12.2.3), Wikipedia.

Bibliography

Blum, A., Hopcroft, J., and Kannan, R. (2018). Foundations of data science. https://www.cs.cornell.edu/jeh/book.pdf.

Boucheron, S., Lugosi, G., and Massart, P. (2012). *Concentration Inequalities: A nonasymptotic theory of independence*. Oxford.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. MIT Press, Cambridge, MA, second edition.

Dubhashi, D. P. and Panconesi, A. (2009). *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press.

Durrett, R. (2019). *Probability: Theory and Examples*. Cambridge, fifth edition. https://services.math.duke.edu/~rtd/PTE/PTE5_011119.pdf.

Grimmett, G. and Stirzaker, D. (2001). *Probability and Random Processes*. Oxford University Press, third edition.

Guruswami, V., Rudra, A., and Sudan, M. (2019). Essential coding theory. https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/.

Klenke, A. (2008). *Probability Theory: A Comprehensive Course*. Springer.

Lehman, E., Leighton, F. T., and Meyer, A. R. (2018). Mathematics for computer science. https://courses.csail.mit.edu/6.042/spring18/mcs.pdf.

McDiarmid, C. (1998). Concentration. http://cgm.cs.mcgill.ca/~breed/conc/colin.pdf.

Mitzenmacher, M. and Upfal, E. (2005). *Probability and computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press.

Motwani, R. and Raghavan, P. (1995). *Randomized Algorithms*. Cambridge University Press.

Murphy, K. P. (2022). *Probabilistic Machine Learning: An Introduction*. MIT Press. https://probml.github.io/pml-book/book1.html.

Roch, S. (2020). Modern discrete probability: An essential toolkit. https://people.math.wisc.edu/~roch/mdp/index.html.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

Shmoys, D. P. and Shmoys, D. B. (2010). *The Design of Approximation Algorithms*. Cambridge University Press.
http://designofapproxalgs.com/book.pdf.

Trefethen, L. N. and Bau, III, D. (1997). *Numerical Linear Algebra*. SIAM.

Vershynin, R. (2018). *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press.
https://www.math.uci.edu/~rvershyn/papers/HDP-book/HDP-book.pdf.

Wainwright, M. J. (2019). *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge.