Class notes for Randomized Algorithms

Sariel Har-Peled[①]

May 29, 2013

[①]Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; sariel@uiuc.edu; http://www.uiuc.edu/~sariel/. Work on this paper was partially supported by a NSF CAREER award CCR-0132901.

Contents

Chapter 1

Intro, Quick Sort and BSP

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> Finally: It was stated at the outset, that this system would not be here, and at once, perfected. You cannot but plainly see that I have kept my word. But I now leave my cetological System standing thus unfinished, even as the great Cathedral of Cologne was left, with the crane still standing upon the top of the uncompleted tower. For small erections may be finished by their first architects; grand ones, true ones, ever leave the copestone to posterity. God keep me from ever completing anything. This whole book is but a draft - nay, but the draft of a draft. Oh, Time, Strength, Cash, and Patience!
>
> – Herman Melville, Moby Dick.

1.1 General Introduction

Randomized algorithms are algorithms that makes random decision during their execution. Specifically, they are allowed to use variables that their value is take from some random distribution. It is not immediately clear why adding the ability to consult with randomness would help an algorithm. But it turns out that the benefits are quite substantial:

Best. There are cases were only randomized algorithm is known or possible, especially for games. For example, consider the 3 coins example.

Speed. In some cases randomized algorithms are considerably faster than any deterministic algorithm.

Simplicity. Even if a randomized algorithm is not faster, often it is considerably simpler than its deterministic counterpart.

Derandomization. Some deterministic algorithms arises from derandomizing the randomized algorithms, and this the only algorithm we know for these problems (i.e., discrepancy).

Adversary arguments and lower bounds. The standard worst case analysis relies on the idea that the adversary can select the input on which the algorithm performs worst. Inherently, the adversary is more powerful than the algorithm, since the algorithm is completely predictable. By using a randomized algorithm, we can make the algorithm unpredictable and break the adversary lower bound.

1.1.1 Randomized vs average-case analysis

Randomized algorithms are not the same as ***average-case analysis***. In average case analysis, one assumes

- Probabilistic analysis assuming random input
- randomized algorithms do not assume random inputs
- so analyses are more applicable

1.2 Basic probability

Here we recall some definitions about probability. The reader already familiar with these definition can happily skip this section.

1.2.1 Formal basic definitions: Sample space, σ-algebra, and probability

Here we formally define some basic notions in probability. The reader familiar with these concepts can safely skip this part.

A ***sample space*** Ω is a set of all possible outcomes of an experiment. We also have a set of events $\mathcal{F}$, where every member of $\mathcal{F}$ is a subset of Ω. Formally, we will require that $\mathcal{F}$ is a σ-algebra.

Definition 1.2.1. A set $\mathcal{F}$ of subsets of Ω is a ***σ-algebra*** if:
(i) $\mathcal{F}$ is not empty,
(ii) if $X \in \mathcal{F}$ then $\overline{X} = \Omega \setminus X \in \mathcal{F}$, and
(iii) if $X, Y \in \mathcal{F}$ then $X \cup Y \in \mathcal{F}$.
(More generally, we will require that if $X_i \in \mathcal{F}$, for $i \in \mathbb{Z}$, then $\cup_i X_i \in \mathcal{F}$.)

We will refer to a member of $\mathcal{F}$ as being an ***event***.

As a concrete example, if we are rolling a dice, then $\Omega = \{1, 2, 3, 4, 5, 6\}$ and $\mathcal{F}$ would be the power set of all possible subsets of Ω.

Definition 1.2.2. A ***probability measure*** is a mapping $\mathbf{Pr} : \mathcal{F} \to [0, 1]$ assigning ***probabilities*** to events. The function $\mathbf{Pr}$ needs to have the following properties:
(i) (additive) for $X, Y \in \mathcal{F}$ disjoint sets, we have that $\mathbf{Pr}[X \cup Y] = \mathbf{Pr}[X] + \mathbf{Pr}[Y]$, and
(ii) $\mathbf{Pr}[\Omega] = 1$.

Definition 1.2.3. A ***probability space*** is a triple $(\Omega, \mathcal{F}, \mathbf{Pr})$, where Ω is a sample space, $\mathcal{F}$ is a σ-algebra defined over Ω, and $\mathbf{Pr}$ is a probability measure.

Definition 1.2.4. A ***random variable*** f is a mapping from Ω into some set $\mathcal{G}$. We will require that the probability of the random variable to take on any value in a given subset of values is well defined. Formally, we will require that for any subset $U \subseteq \mathcal{G}$, we have that $f^{-1}(U) \in \mathcal{F}$. That is, $\mathbf{Pr}[f \in U] = \mathbf{Pr}\left[f^{-1}(U)\right]$ is defined.

Going back to the dice example, the number on the top of the dice when we roll it is a random variable. Similarly, let X be one if the number rolled is larger than 3, and zero otherwise. Clearly X is a random variable.

We denote the ***probability*** of a random variable X to get the value x, by $\mathbf{Pr}[X = x]$ (or sometime $\mathbf{Pr}[x]$, if we are really lazy).

1.2.2 Expectation and conditional probability

Definition 1.2.5 (Expectation.). The expectation of a random variable X, is its average. Formally, the ***expectation*** of X is

$$\mathbf{E}\Big[X\Big] = \sum_x x \,\mathbf{Pr}[X = x]\,.$$

Definition 1.2.6 (Conditional Probability.). The ***conditional probability*** of X given Y, is the probability that $X = x$ given that $Y = y$. We denote this quantity by $\mathbf{Pr}\Big[X = x \;\Big|\; Y = y\Big]$.

The conditional probability can be computed using the formula

$$\mathbf{Pr}\Big[X = x \;\Big|\; Y = y\Big] = \frac{\mathbf{Pr}\Big[\;(X = x) \;\cap\; (Y = y)\;\Big]}{\mathbf{Pr}[Y = y]}.$$

For example, let us roll a dice and let X be the number we got. Let Y be the random variable that is true if the number we get is even. Then, we have that

$$\mathbf{Pr}\Big[X = 2 \;\Big|\; Y = \mathit{true}\Big] = 1/3.$$

Definition 1.2.7. Two random variables X and Y are ***independent*** if $\mathbf{Pr}\Big[X = x \;\Big|\; Y = y\Big] = \mathbf{Pr}[X = x]$, for all x.

Lemma 1.2.8 (Linearity of expectation.). **Linearity of expectation** *is the property that for any two random variables X and Y, we have* $\mathbf{E}[X + Y] = \mathbf{E}[X] + \mathbf{E}[Y]$.

1.3 QuickSort

Let the input be a set $t_1, \ldots, t_n$ of n items to be sorted. We remind the reader, that the **QuickSort** algorithm randomly pick a pivot element (uniformly), splits the input into two subarrays of all the elements smaller than the pivot, and all the elements larger than the pivot, and then it recurses on these two subarrays (the pivot is not included in these two subproblems). Here we will show that the expected running time of **QuickSort** is $O(n \log n)$.

Definition 1.3.1. For an event $\mathcal{E}$, let X be a random variable which is 1 if $\mathcal{E}$ occurred and 0 otherwise. The random variable X is an ***indicator variable***.

Observation 1.3.2. *For an indicator variable X of an event $\mathcal{E}$, we have*

$$\mathbf{E}[X] = \mathbf{Pr}[X = 1] = \mathbf{Pr}[\mathcal{E}]\,.$$

Let $S_1, \ldots, S_n$ be the elements in their sorted order (i.e., the output order). Let $X_{ij} = 1$ be the indicator variable which is one iff **QuickSort** compares S_i to S_j, let p_{ij} denote the probability that this happens. Clearly, the number of comparisons performed by the algorithm is $C = \sum_{i<j} X_{ij}$. By linearity of expectations, we have

$$\mathbf{E}[C] = \sum_{i<j} \mathbf{E}\Big[X_{ij}\Big] = \sum_{i<j} p_{ij}.$$

We want to bound p_{ij}, the probability that the S_i is compared to S_j. Consider the last recursive call involving both S_i and S_j. Clearly, the pivot at this step must be one of $S_i, \ldots, S_j$, all equally likely. Indeed, S_i and S_j were separated in the next recursive call.

Observe, that S_i and S_j get compared if and only if pivot is S_i or S_j. Thus, the probability for that is $2/(j-i+1)$. Indeed,

$$p_{ij} = \mathbf{Pr}\Big[S_i \text{ or } S_j \text{ picked} \;\Big|\; \text{picked pivot from } S_i, \ldots, S_j\Big] = \frac{2}{j-i+1}.$$

Thus,

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j>i} p_{ij} &= \sum_{i=1}^{n}\sum_{j>i} 2/(j-i+1) = \sum_{i=1}^{n}\sum_{k=1}^{n-i+1}\frac{2}{k} \le 2\sum_{i=1}^{n}\sum_{k=1}^{n}\frac{1}{k} \\ &\le 2nH_n \le n + 2n\ln n,\end{aligned}$$

where H_n is the ***harmonic number***[①] $H_n = \sum_{i=1}^{n} \frac{1}{i}$. We thus proved the following result.

Lemma 1.3.3. **QuickSort** *performs in expectation at most* $n + 2n\ln n$ *comparisons, when sorting* n *elements.*

Note, that this holds for all inputs. No assumption on the input is made. Similar bounds holds not only in expectation, but also with high probability.

This raises the question, of how does the algorithm pick a random element? We assume we have access to a random source that can get us number between 1 and n uniformly.

Note, that the algorithm always works, but it might take quadratic time in the worst case.

1.4 Binary space partition (BSP)

Let assume that we would like to render an image of a three dimensional scene on the computer screen. The input is in general a collection of polygons in three dimensions. The *painter* algorithm, render the scene by drawing things from back to front; and let front stuff overwrite what was painted before.

The problem is that it is not always possible to order the objects in three dimensions. This ordering might have cycles. So, one possible solution is to build a *binary space partition*. We build a binary tree. In the root, we place a polygon P. Let h be the plane containing P. Next,

[①] Using integration to bound summation, we have $H_n \le 1 + \int_{x=1}^{n} \frac{1}{x}dx \le 1 + \ln n$. Similarly, $H_n \ge \int_{x=1}^{n} \frac{1}{x}dx = \ln n$.

we partition the input polygons into two sets, depending on which side of h they fall into. We recursively construct a BSP for each set, and we hang it from the root node. If a polygon intersects h then we cut it into two polygons as split by h. We continue the construction recursively on the objects on one side of h, and the objects on the other side. What we get, is a binary tree that splits space into cells, and furthermore, one can use the painter algorithm on these objects. The natural question is how big is the resulting partition.

We will study the easiest case, of disjoint segments in the plane.

1.4.1 BSP for disjoint segments

Let $P = \{\mathsf{s}_1, \ldots, \mathsf{s}_n\}$ be n disjoint segments in the plane. We will build the BSP by using the lines defined by these segments. This kind of BSP is called ***autopartition***.

To recap, the BSP is a binary tree, at every internal node we store a segment of P, where the line associated with it splits its region into its two children. Finally, each leaf of the BSP stores a single segment. A ***fragment*** is just going to be a subsegment formed by this splitting. Clearly, every internal node, stores a fragment that defines its split. As such, the size of the BSP is proportional to the number of fragments generated when building the BSP.

One application of such a BSP is ray shooting - given a ray you would like to determine what is the first segment it hits. Start from the root, figure out which child contains the apex of the ray, and first (recursively) compute the first segment stored in this child that the ray intersect. Contain into the second child only if the first subtree does not contain any segment that intersect the ray.

1.4.1.1 The algorithm

We pick a random permutation σ of $1, \ldots, n$, and in the ith step we insert $\mathsf{s}_{\sigma(i)}$ splitting all the cells that s_i intersects.

Observe, that if s_i crosses a cell completely, it just splits it into two and no new fragments are created. As such, the bad case is when a segment s is being inserted, and its line intersect some other segment t.

So, let $\mathcal{E}(\mathsf{s}, \mathsf{t})$ denote the event that when inserted s it had split t. In particular, let $\mathrm{index}(\mathsf{s}, \mathsf{t})$ denote the number of segments on the line of s between s (closer) endpoint and t (including t. If the line of s does not intersect t, then $\mathrm{index}(\mathsf{s}, \mathsf{t}) = \infty$.

We have that

$$\mathbf{Pr}[\mathcal{E}(\mathsf{s}, \mathsf{t})] = \frac{1}{1 + \mathrm{index}(\mathsf{s}, \mathsf{t})}.$$

Let $X_{\mathsf{s},\mathsf{t}}$ be the indicator variable that is 1 if $\mathcal{E}(\mathsf{s}, \mathsf{t})$ happens. We have that

$$S = \text{ number of fragments } == \sum_{i=1}^{n} \sum_{j=1, i \neq j}^{n} X_{\mathsf{s}_i, \mathsf{s}_j}.$$

As such, by linearity of expectations, we have

$$\begin{aligned}\mathbf{E}[S] &= \mathbf{E}\left[\sum_{i=1}^{n}\sum_{j=1,i\neq j}^{n} X_{\mathsf{s}_i,\mathsf{s}_j}\right] = \sum_{i=1}^{n}\sum_{j=1,i\neq j}^{n}\mathbf{E}\left[X_{\mathsf{s}_i,\mathsf{s}_j}\right] = \sum_{i=1}^{n}\sum_{j=1,i\neq j}^{n}\mathbf{Pr}\left[\mathcal{E}(\mathsf{s}_i,\mathsf{s}_j)\right]\\ &= \sum_{i=1}^{n}\sum_{j=1,i\neq j}^{n}\frac{1}{1+\operatorname{index}\left(\mathsf{s}_i,\mathsf{s}_j\right)}\\ &\leq \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{2}{1+j} = 2nH_n.\end{aligned}$$

Since the size of the BSP is proportional to the number of fragments created, we have the following result.

Theorem 1.4.1. *Given n disjoint segments in the plane, one can build a BSP for them of size $O(n \log n)$.*

Csaba Tóth [Tót03] showed that BSP for segments in the plane, in the worst case, has complexity $\Omega\left(n\frac{\log n}{\log\log n}\right)$.

1.5 Extra: QuickSelect running time

We remind the reader that **QuickSelect** receives an array $t[1\ldots n]$ of n real numbers, and a number k, and returns the element of rank k in the sorted order of the elements of t. We can of course, use **QuickSort**, and just return the kth element in the sorted array, but a more efficient algorithm, would be to modify **QuickSelect**, so that it recurses on the subproblem that contains the element we are interested in. Formally, **QuickSelect** chooses a random pivot, splits the array according to the pivot. This implies that we now know the rank of the pivot, and if its equal to $\overline{\mathsf{m}}$, we return it. Otherwise, we recurse on the subproblem containing the required element (modifying $\overline{\mathsf{m}}$ as we go down the recursion. Namely, **QuickSelect** is a modification of **QuickSort** performing only a single recursive call (instead of two).

AS before, to bound the expected running time, we will bound the expected number of comparisons. As before, let $S_1,\ldots,S_n$ be the elements of t in their sorted order. Now, for $i<j$, let X_{ij} be the indicator variable that is one if S_i is being compared to S_j during the execution of **QuickSelect**. There are several possibilities to consider:

(i) If $i<j<\overline{\mathsf{m}}$: Here, S_i is being compared to S_j, if and only if the first pivot in the range $S_i,\ldots,S_k$ is either S_i or S_j. The probability for that is $2/(k-i+1)$. As such, we have that

$$\begin{aligned}\alpha_1 &= \mathbf{E}\left[\sum_{i<j<\overline{\mathsf{m}}} X_{ij}\right] = \mathbf{E}\left[\sum_{i=1}^{med-2}\sum_{j=i+1}^{\overline{\mathsf{m}}-1} X_{ij}\right] = \sum_{i=1}^{med-2}\sum_{j=i+1}^{\overline{\mathsf{m}}-1} 2/(\overline{\mathsf{m}}-i+1)\\ &= \sum_{i=1}^{med-2} 2(\overline{\mathsf{m}}-i-1)/(\overline{\mathsf{m}}-i+1) \leq 2(\overline{\mathsf{m}}-2).\end{aligned}$$

(ii) If $\overline{\mathsf{m}} < i < j$: Using the same analysis as above, we have that $\mathbf{Pr}\Big[X_{ij} = 1\Big] = 2/(j - \overline{\mathsf{m}} + 1)$. As such,

$$\alpha_2 = \mathbf{E}\left[\sum_{j=\overline{\mathsf{m}}+1}^{n}\sum_{i=\overline{\mathsf{m}}+1}^{j-1} X_{ij}\right] = \sum_{j=\overline{\mathsf{m}}+1}^{n}\sum_{i=\overline{\mathsf{m}}+1}^{j-1}\frac{2}{j-\overline{\mathsf{m}}+1} = \sum_{j=\overline{\mathsf{m}}+1}^{n}\frac{2(j-\overline{\mathsf{m}}-1)}{j-\overline{\mathsf{m}}+1} \leq 2(n-\overline{\mathsf{m}}).$$

(iii) $i < \overline{\mathsf{m}} < j$: Here, we compare S_i to S_j if and only if the first indicator in the range $S_i, \ldots, S_j$ is either S_i or S_j. As such, $\mathbf{E}\Big[X_{ij}\Big] = \mathbf{Pr}\Big[X_{ij} = 1\Big] = 2/(j - i + 1)$. As such, we have

$$\alpha_3 = \mathbf{E}\left[\sum_{i=1}^{\overline{\mathsf{m}}-1}\sum_{j=\overline{\mathsf{m}}+1}^{n} X_{ij}\right] = \sum_{i=1}^{\overline{\mathsf{m}}-1}\sum_{j=\overline{\mathsf{m}}+1}^{n}\frac{2}{j-i+1}.$$

Observe, that for a fixed $\Delta = j - i + 1$, we are going to handle the gap Δ in the above summation, at most $\Delta - 2$ times. As such, $\alpha_3 \leq \sum_{\Delta=3}^{n} 2(\Delta-2)/\Delta \leq 2n$.

(iv) $i = \overline{\mathsf{m}}$. We have $\alpha_4 = \displaystyle\sum_{j=\overline{\mathsf{m}}+1}^{n}\mathbf{E}\Big[X_{ij}\Big] = \sum_{j=\overline{\mathsf{m}}+1}^{n}\frac{2}{j-\overline{\mathsf{m}}+1} = \ln n + 1.$

(v) $j = \overline{\mathsf{m}}$. We have $\alpha_5 = \displaystyle\sum_{i=1}^{\overline{\mathsf{m}}-1}\mathbf{E}\Big[X_{ij}\Big] = \sum_{i=1}^{\overline{\mathsf{m}}-1}\frac{2}{\overline{\mathsf{m}}-i+1} \leq \ln\overline{\mathsf{m}} + 1.$

Thus, the expected number of comparisons performed by **QuickSelect** is bounded by

$$\sum_i \alpha_i \leq 2(\overline{\mathsf{m}}-2) + 2(n-\overline{\mathsf{m}}) + 2n + \ln n + 1 + \ln\overline{\mathsf{m}} + = 4n - 2 + \ln n + \ln\overline{\mathsf{m}}.$$

Theorem 1.5.1. *In expectation,* **QuickSelect** *performs at most* $4n - 2 + \ln n + \ln\overline{\mathsf{m}}$ *comparisons, when selecting the* $\overline{\mathsf{m}}$*th element out of* n *elements.*

A different approach can reduce the number of comparisons (in expectation) to $1.5n + o(n)$. More on that later in the course.

Chapter 2

Min Cut

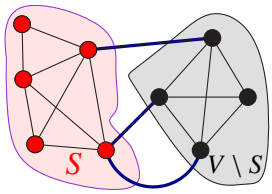
598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> **To acknowledge the corn** - This purely American expression means to admit the losing of an argument, especially in regard to a detail; to retract; to admit defeat. It is over a hundred years old. Andrew Stewart, a member of Congress, is said to have mentioned it in a speech in 1828. He said that haystacks and cornfields were sent by Indiana, Ohio and Kentucky to Philadelphia and New York. Charles A. Wickliffe, a member from Kentucky questioned the statement by commenting that haystacks and cornfields could not walk. Stewart then pointed out that he did not mean literal haystacks and cornfields, but the horses, mules, and hogs for which the hay and corn were raised. Wickliffe then rose to his feet, and said, "Mr. Speaker, I acknowledge the corn".
>
> – Funk, Earle, A Hog on Ice and Other Curious Expressions.

2.1 Min Cut

2.1.1 Problem Definition

Let $\mathsf{G} = (V, E)$ be undirected graph with n vertices and m edges. We are interested in ***cuts*** in G.



Definition 2.1.1. A ***cut*** in G is a partition of the vertices of V into two sets S and $V \setminus S$, where the edges of the cut are

$$(S, V \setminus S) = \left\{ uv \;\middle|\; u \in S, v \in V \setminus S, \text{ and } uv \in E \right\},$$

where $S \neq \emptyset$ and $V \setminus S \neq \emptyset$. We will refer to the number of edges in the cut $(S, V \setminus S)$ as the *size of the cut*. For an example of a cut, see figure on the right.

We are interested in the problem of computing the ***minimum cut*** (i.e., ***mincut***), that is, the cut in the graph with minimum cardinality. Specifically, we would like to find the set $S \subseteq V$ such that $(S, V \setminus S)$ is as small as possible, and S is neither empty nor $V \setminus S$ is empty.

2.1.2 Some Definitions

We remind the reader of the following concepts. The ***conditional probability*** of X given Y is $\mathbf{Pr}\left[X = x \;\middle|\; Y = y\right] = \mathbf{Pr}[(X = x) \cap (Y = y)]/\mathbf{Pr}[Y = y]$. An equivalent, useful restatement of this is that

$$\mathbf{Pr}[(X = x) \cap (Y = y)] = \mathbf{Pr}\left[X = x \;\middle|\; Y = y\right] \cdot \mathbf{Pr}[Y = y]. \tag{2.1}$$

The following is easy to prove by induction using Eq. (2.1).

Lemma 2.1.2. *Let $\mathcal{E}_1, \ldots, \mathcal{E}_n$ be n events which are not necessarily independent. Then,*

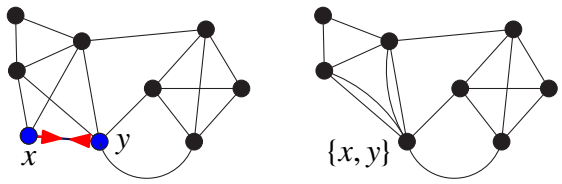
$$\mathbf{Pr}[\cap_{i=1}^{n} \mathcal{E}_i] = \mathbf{Pr}[\mathcal{E}_1] * \mathbf{Pr}\left[\mathcal{E}_2 \;\middle|\; \mathcal{E}_1\right] * \mathbf{Pr}\left[\mathcal{E}_3 \;\middle|\; \mathcal{E}_1 \cap \mathcal{E}_2\right] * \ldots * \mathbf{Pr}\left[\mathcal{E}_n \;\middle|\; \mathcal{E}_1 \cap \ldots \cap \mathcal{E}_{n-1}\right].$$

2.2 The Algorithm

The basic operation used by the algorithm is ***edge contraction***, depicted in Figure 2.1. We take an edge $e = xy$ in G and merge the two vertices into a single vertex. The new resulting graph is denoted by G/xy. Note, that we remove self loops created by the contraction. However, since the resulting graph is no longer a regular graph, it has parallel edges – namely, it is a multi-graph. We represent a multi-graph, as a regular graph with multiplicities on the edges. See Figure 2.2.

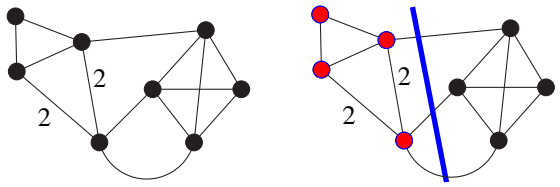
The edge contraction operation can be implemented in $O(n)$ time for a graph with n vertices. This is done by merging the adjacency lists of the two vertices being contracted, and then using hashing to do the fix-ups (i.e., we need to fix the adjacency list of the vertices that are connected to the two vertices).

Note, that the cut is now computed counting multiplicities (i.e., if e is in the cut and it has weight w, then the contribution of e to the cut weight is w).



(a) (b)

Figure 2.1: (a) A contraction of the edge xy. (b) The resulting graph.



(a) (b)

Figure 2.2: (a) A multi-graph. (b) A minimum cut in the resulting multi-graph.

Observation 2.2.1. *A set of vertices in G/xy corresponds to a set of vertices in the graph G. Thus a cut in G/xy always corresponds to a valid cut in G. However, there are cuts in G that do not exist in G/xy. For example, the cut $S = \{x\}$, does not exist in G/xy. As such, the size of the minimum cut in G/xy is at least as large as the minimum cut in G (as long as G/xy has at least one edge). Since any cut in G/xy has a corresponding cut of the same cardinality in G.*

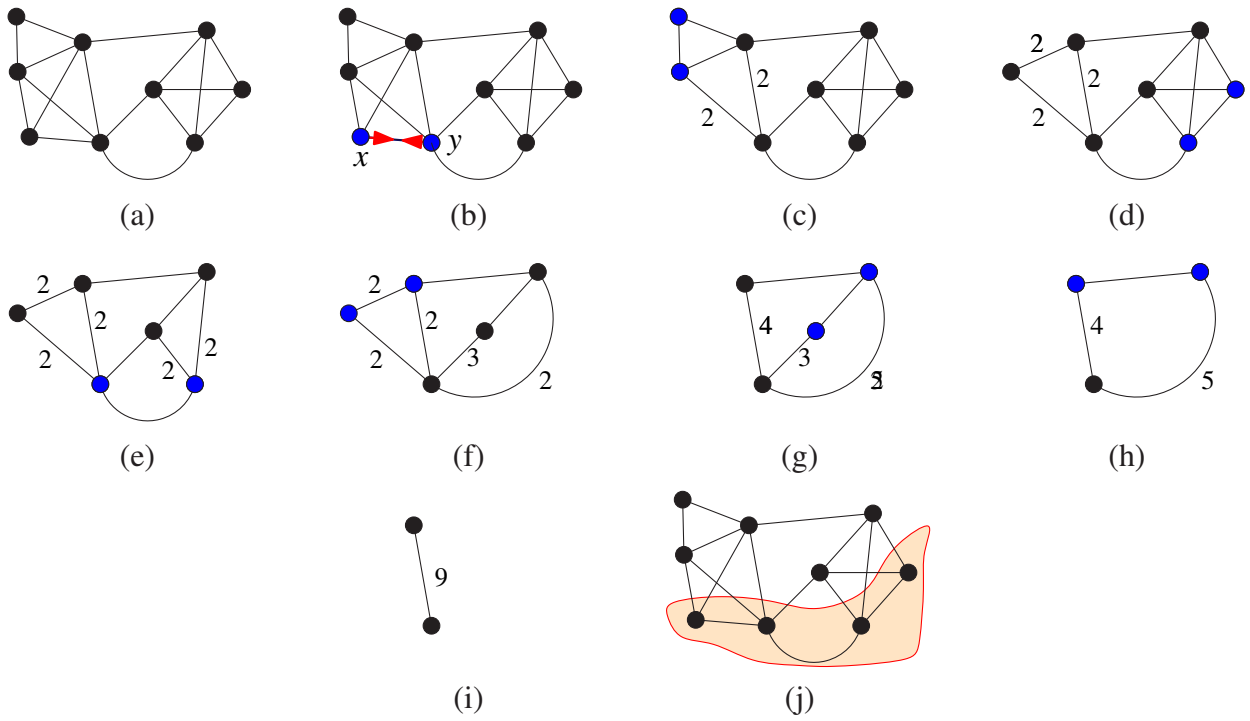


Figure 2.3: (a) Original graph. (b)–(j) a sequence of contractions in the graph, and (h) the cut in the original graph, corresponding to the single edge in (h). Note that the cut of (h) is not a mincut in the original graph.

Our algorithm works by repeatedly performing edge contractions. This is beneficial as this shrinks the underlying graph, and we would compute the cut in the resulting (smaller) graph. An "extreme" example of this, is shown in Figure 2.3, where we contract the graph into a single edge, which (in turn) corresponds to a cut in the original graph. (It might help the reader to think about each vertex in the contracted graph, as corresponding to a connected component in the original graph.)

Figure 2.3 also demonstrates the problem with taking this approach. Indeed, the resulting cut is not the minimum cut in the graph.

So, why did the algorithm fail to find the minimum cut in this case?[①] The failure occurs because of the contraction at Figure 2.3 (e), as we had contracted an edge in the minimum cut. In the new graph, depicted in Figure 2.3 (f), there is no longer a cut of size 3, and all cuts are of size 4 or more. Specifically, the algorithm succeeds only if it does not contract an edge in the minimum cut.

Observation 2.2.2. *Let $e_1, \ldots, e_{n-2}$ be a sequence of edges in G, such that none of them is in the minimum cut, and such that $\mathsf{G}' = G/\{e_1, \ldots, e_{n-2}\}$ is a single multi-edge. Then, this multi-edge corresponds to a minimum cut in G.*

[①] Naturally, if the algorithm had succeeded in finding the minimum cut, this would have been **our** success.

Note, that the claim in the above observation is only in one direction. We might be able to still compute a minimum cut, even if we contract an edge in a minimum cut, the reason being that a minimum cut is not unique. In particular, another minimum cut might survived the sequence of contractions that destroyed other minimum cuts.

Algorithm MinCut(G)
  $\mathsf{G}_0 \leftarrow G$
  $i = 0$
  while G_i has more than two vertices **do**
    Pick randomly an edge e_i from the edges of G_i
    $\mathsf{G}_{i+1} \leftarrow G_i / e_i$
    $i \leftarrow i + 1$
  Let $(S, V \setminus S)$ be the cut in the original graph
    corresponding to the single edge in G_i
  return $(S, V \setminus S)$.

Figure 2.4: The minimum cut algorithm.

Using Observation 2.2.2 in an algorithm is problematic, since the argumentation is circular, how can we find a sequence of edges that are not in the cut without knowing what the cut is? The way to slice the Gordian knot here, is to randomly select an edge at each stage, and contract this random edge.

See Figure 2.4 for the resulting algorithm **MinCut**.

2.2.1 Analysis

2.2.1.1 The probability of success.

Naturally, if we are extremely lucky, the algorithm would never pick an edge in the mincut, and the algorithm would succeed. The ultimate question here is what is the probability of success. If it is relatively "large" then this algorithm is useful since we can run it several times, and return the best result computed. If on the other hand, this probability is tiny, then we are working in vain since this approach would not work.

Lemma 2.2.3. *If a graph* G *has a minimum cut of size* k *and* G *has* n *vertices, then* $|E(G)| \geq \frac{kn}{2}$.

Proof: Each vertex degree is at least k, otherwise the vertex itself would form a minimum cut of size smaller than k. As such, there are at least $\sum_{v \in V} \text{degree(v)}/2 \geq nk/2$ edges in the graph. ■

Lemma 2.2.4. *If we pick in random an edge* e *from a graph* G*, then with probability at most* $2/n$ *it belong to the minimum cut.*

Proof: There are at least $nk/2$ edges in the graph and exactly k edges in the minimum cut. Thus, the probability of picking an edge from the minimum cut is smaller then $k/(nk/2) = 2/n$. ■

The following lemma shows (surprisingly) that **MinCut** succeeds with reasonable probability.

Lemma 2.2.5. **MinCut** *outputs the mincut with probability* $\geq \dfrac{2}{n(n-1)}$.

Proof: Let $\mathcal{E}_i$ be the event that e_i is not in the minimum cut of G_i. By Observation 2.2.2, **MinCut** outputs the minimum cut if the events $\mathcal{E}_0, \ldots, \mathcal{E}_{n-3}$ all happen (namely, all edges picked are outside the minimum cut).

By Lemma 2.2.4, it holds $\mathbf{Pr}\left[\mathcal{E}_i \,\middle|\, \mathcal{E}_0 \cap \mathcal{E}_1 \cap \ldots \cap \mathcal{E}_{i-1}\right] \geq 1 - \frac{2}{|V(G_i)|} = 1 - \frac{2}{n-i}$. Implying that

$$\begin{aligned}\Delta &= \mathbf{Pr}[\mathcal{E}_0 \cap \ldots \cap \mathcal{E}_{n-3}] \\ &= \mathbf{Pr}[\mathcal{E}_0] \cdot \mathbf{Pr}\left[\mathcal{E}_1 \,\middle|\, \mathcal{E}_0\right] \cdot \mathbf{Pr}\left[\mathcal{E}_2 \,\middle|\, \mathcal{E}_0 \cap \mathcal{E}_1\right] \cdot \ldots \cdot \mathbf{Pr}\left[\mathcal{E}_{n-3} \,\middle|\, \mathcal{E}_0 \cap \ldots \cap \mathcal{E}_{n-4}\right]\end{aligned}$$

As such, we have

$$\begin{aligned}\Delta &\geq \prod_{i=0}^{n-3}\left(1 - \frac{2}{n-i}\right) = \prod_{i=0}^{n-3} \frac{n-i-2}{n-i} \\ &= \frac{n-2}{n} * \frac{n-3}{n-1} * \frac{n-4}{n-2} \cdots \frac{2}{4} \cdot \frac{1}{3} \\ &= \frac{2}{n \cdot (n-1)}.\end{aligned}$$

■

2.2.1.2 Running time analysis.

Observation 2.2.6. **MinCut** *runs in* $O(n^2)$ *time.*

Observation 2.2.7. *The algorithm always outputs a cut, and the cut is not smaller than the minimum cut.*

Definition 2.2.8. (informal) Amplification is the process of running an experiment again and again till the things we want to happen, with good probability, do happen.

Let **MinCutRep** be the algorithm that runs **MinCut** $n(n-1)$ times and return the minimum cut computed in all those independent executions of **MinCut**.

Lemma 2.2.9. *The probability that* **MinCutRep** *fails to return the minimum cut is* < 0.14.

Proof: The probability of failure of **MinCut** to output the mincut in each execution is at most $1 - \frac{2}{n(n-1)}$, by Lemma 2.2.5. Now, **MinCutRep** fails, only if all the $n(n-1)$ executions of **MinCut** fail. But these executions are independent, as such, the probability to this happen is at most

$$\left(1 - \frac{2}{n(n-1)}\right)^{n(n-1)} \leq \exp\left(-\frac{2}{n(n-1)} \cdot n(n-1)\right) = \exp(-2) < 0.14,$$

since $1 - x \leq e^{-x}$ for $0 \leq x \leq 1$. ■

Theorem 2.2.10. *One can compute the minimum cut in* $O(n^4)$ *time with constant probability to get a correct result. In* $O\left(n^4 \log n\right)$ *time the minimum cut is returned with high probability.*

```
Contract ( G, t )
begin
   while |(G)| > t do
         Pick a random edge e in G.
         G ← G/e
   return G
end
```

```
FastCut(G = (V, E))
   G – multi-graph
begin
   n ← |V(G)|
   if n ≤ 6 then
         Compute (via brute force) minimum cut
         of G and return cut.
   t ← ⌈1 + n/√2⌉
   H1 ← Contract(G, t)
   H2 ← Contract(G, t)
   /* Contract is randomized!!! */
   X1 ← FastCut(H1),
   X2 ← FastCut(H2)
   return minimum cut out of X1 and X2.
end
```

Figure 2.5: **Contract**(G, t) shrinks G till it has only t vertices. **FastCut** computes the minimum cut using **Contract**.

2.3 A faster algorithm

The algorithm presented in the previous section is extremely simple. Which raises the question of whether we can get a faster algorithm[②]?

So, why **MinCutRep** needs so many executions? Well, the probability of success in the first ν iterations is

$$\begin{aligned}\mathbf{Pr}[\mathcal{E}_0 \cap \ldots \cap \mathcal{E}_{\nu-1}] &\geq \prod_{i=0}^{\nu-1}\left(1 - \frac{2}{n-i}\right) = \prod_{i=0}^{\nu-1}\frac{n-i-2}{n-i} \\ &= \frac{n-2}{n} * \frac{n-3}{n-1} * \frac{n-4}{n-2} \cdots = \frac{(n-\nu)(n-\nu-1)}{n\cdot(n-1)}. \end{aligned} \tag{2.2}$$

Namely, this probability deteriorates very quickly toward the end of the execution, when the graph becomes small enough. (To see this, observe that for $\nu = n/2$, the probability of success is roughly $1/4$, but for $\nu = n - \sqrt{n}$ the probability of success is roughly $1/n$.)

So, the key observation is that as the graph get smaller the probability to make a bad choice increases. So, instead of doing the amplification from the outside of the algorithm, we will run the new algorithm more times when the graph is smaller. Namely, we put the amplification directly into the algorithm.

The basic new operation we use is **Contract**, depicted in Figure 2.5, which also depict the new algorithm **FastCut**.

Lemma 2.3.1. *The running time of* **FastCut**(G) *is* $O\left(n^2 \log n\right)$, *where* $n = |V(G)|$.

[②]This would require a more involved algorithm, thats life.

Proof: Well, we perform two calls to **Contract**(G, t) which takes $O(n^2)$ time. And then we perform two recursive calls on the resulting graphs. We have:

$$T(n) = O\left(n^2\right) + 2T\left(\frac{n}{\sqrt{2}}\right)$$

The solution to this recurrence is $O\left(n^2 \log n\right)$ as one can easily (and should) verify. ■

Exercise 2.3.2. Show that one can modify **FastCut** so that it uses only $O(n^2)$ space.

Lemma 2.3.3. *The probability that* **Contract**$(G, n/\sqrt{2})$ *had not contracted the minimum cut is at least* $1/2$.

Namely, the probability that the minimum cut in the contracted graph is still a minimum cut in the original graph is at least $1/2$.

Proof: Just plug in $v = n - t = n - \left\lceil 1 + n/\sqrt{2} \right\rceil$ into Eq. (2.2). We have

$$\mathbf{Pr}[\mathcal{E}_0 \cap \ldots \cap \mathcal{E}_{n-t}] \geq \frac{t(t-1)}{n \cdot (n-1)} = \frac{\left\lceil 1 + n/\sqrt{2} \right\rceil \left(\left\lceil 1 + n/\sqrt{2} \right\rceil - 1 \right)}{n(n-1)} \geq \frac{1}{2}.$$

■

The following lemma bounds the probability of success. A more elegant argument is given in Section 2.3.1 below.

Lemma 2.3.4. **FastCut** *finds the minimum cut with probability larger than* $\Omega\left(1/\log n\right)$.

Proof: Let $P(n)$ be the probability that the algorithm succeeds on a graph with n vertices.

The probability to succeed in the first call on H_1 is the probability that **Contract** did not hit the minimum cut (this probability is larger than $1/2$ by Lemma 2.3.3), times the probability that the algorithm succeeded on H_1 in the recursive call (those two events are independent). Thus, the probability to succeed on the call on H_1 is at least $(1/2) * P(n/\sqrt{2})$, Thus, the probability to fail on H_1 is $\leq 1 - \frac{1}{2} P\left(\frac{n}{\sqrt{2}}\right)$.

The probability to fail on both H_1 and H_2 is smaller than

$$\left(1 - \frac{1}{2} P\left(\frac{n}{\sqrt{2}}\right)\right)^2,$$

since H_1 and H_2 are being computed independently. Note that if the algorithm, say, fails on H_1 but succeeds on H_2 then it succeeds to return the mincut. Thus the above expression bounds the probability of failure. And thus, the probability for the algorithm to succeed is

$$P(n) \geq 1 - \left(1 - \frac{1}{2} P\left(\frac{n}{\sqrt{2}}\right)\right)^2 = P\left(\frac{n}{\sqrt{2}}\right) - \frac{1}{4}\left(P\left(\frac{n}{\sqrt{2}}\right)\right)^2.$$

We need to solve this recurrence. (This is very tedious, but since the details are non-trivial we provide the details of how to do so.) Divide both sides of the equation by $P\left(n/\sqrt{2}\right)$ we have:

$$\frac{P(n)}{P(n/\sqrt{2})} \geq 1 - \frac{1}{4} P(n/\sqrt{2}).$$

It is now easy to verify that this inequality holds for $P(n) \geq c/\log n$ (since the worst case is $P(n) = c/\log n$ we verify this inequality for this value). Indeed,

$$\frac{c/\log n}{c/\log(n/\sqrt{2})} \geq 1 - \frac{c}{4\log(n/\sqrt{2})}.$$

As such, letting $\Delta = \log n$, we have

$$\frac{\log n - \log\sqrt{2}}{\log n} = \frac{\Delta - \log\sqrt{2}}{\Delta} \geq \frac{4(\log n - \log\sqrt{2}) - c}{4(\log n - \log\sqrt{2})} = \frac{4(\Delta - \log\sqrt{2}) - c}{4(\Delta - \log\sqrt{2})}.$$

Equivalently, $4(\Delta - \log\sqrt{2})^2 \geq 4\Delta(\Delta - \log\sqrt{2}) - c\Delta$. Which implies $-8\Delta\log\sqrt{2} + 4\log^2\sqrt{2} \geq -4\Delta\log\sqrt{2} - c\Delta$. Namely,

$$c\Delta - 4\Delta\log\sqrt{2} + 4\log^2\sqrt{2} \geq 0,$$

which clearly holds for $c \geq 4\log\sqrt{2}$.

We conclude, that the algorithm succeeds in finding the minimum cut in probability

$$\geq 2\log 2/\log n.$$

(Note that the base of the induction holds because we use brute force, and then $P(i) = 1$ for small i.) ∎

Exercise 2.3.5. Prove, that running **FastCut** repeatedly $c \cdot \log^2 n$ times, guarantee that the algorithm outputs the minimum cut with probability $\geq 1 - 1/n^2$, say, for c a constant large enough.

Theorem 2.3.6. *One can compute the minimum cut in a graph* G *with* n *vertices in* $O(n^2\log^3 n)$ *time. The algorithm succeeds with probability* $\geq 1 - 1/n^2$.

Proof: We do amplification on **FastCut** by running it $O(\log^2 n)$ times. The running time bound follows from Lemma 2.3.1. The bound on the probability follows from Lemma 2.3.4, and using the amplification analysis as done in Lemma 2.2.9 for **MinCutRep**. ∎

2.3.1 On coloring trees and min-cut

Let T_h be a complete binary tree of height h. We randomly color its edges by black and white. Namely, for each edge we independently choose its color to be either black or white, with equal probability. We are interested in the event that there exists a path from the root of T_h to one of its leafs, that is all black. Let $\mathcal{E}_h$ denote this event, and let $\rho_h = \mathbf{Pr}[\mathcal{E}_h]$. Observe that $\rho_0 = 1$ and $\rho_1 = 3/4$ (see below).

To bound this probability, consider the root u of T_h and its two children u_l and u_r. The probability that there is a black path from u_l to one of its children is ρ_{h-1}, and as such, the probability that there is a black path from u through u_l to a leaf of the subtree of u_l is $\mathbf{Pr}\Big[\text{the edge } uu_l \text{ is colored black}\Big]\cdot \rho_{h-1} = \rho_{h-1}/2$. As such, the probability that there is no black path through u_l is $1 - \rho_{h-1}/2$. As such, the probability of not having a black path from u to a leaf (through either children) is $(1 - \rho_{h-1}/2)^2$. In particular, there desired probability, is the complement; that is

$$\rho_h = 1 - \left(1 - \frac{\rho_{h-1}}{2}\right)^2 = \frac{\rho_{h-1}}{2}\left(2 - \frac{\rho_{h-1}}{2}\right) = \rho_{h-1} - \frac{\rho_{h-1}^2}{4}.$$

Lemma 2.3.7. *We have that* $\rho_h \geq 1/(h+1)$.

Proof: The proof is by induction. For $h = 1$, we have $\rho_1 = 3/4 \geq 1/(1+1)$.

Observe that $\rho_h = f(\rho_{h-1})$ for $f(x) = x - x^2/4$, and $f'(x) = 1 - x/2$. As such, $f'(x) > 0$ for $x \in [0,1]$ and $f(x)$ is increasing in the range $[0,1]$. As such, by induction, we have that $\rho_h = f(\rho_{h-1}) \geq f\left(\frac{1}{(h-1)+1}\right) = \frac{1}{h} - \frac{1}{4h^2}$. We need to prove that $\rho_h \geq 1/(h+1)$, which is implied by the above if

$$\frac{1}{h} - \frac{1}{4h^2} \geq \frac{1}{h+1} \quad \Leftrightarrow \quad 4h(h+1) - (h+1) \geq 4h^2 \quad \Leftrightarrow \quad 4h^2 + 4h - h - 1 \geq 4h^2 \quad \Leftrightarrow \quad 3h \geq 1,$$

which trivially holds. ∎

The recursion tree for **FastCut** corresponds to such a coloring. Indeed, it is a binary tree as every call performs two recursive calls. Inside such a call, we independently perform two (independent) contractions reducing the given graph with n vertices to have $n/\sqrt{2}$ vertices. If this contraction succeeded (i.e., it did not hit the min-cut), then consider this edge to be colored by black (and white otherwise). Clearly, the algorithm succeeds, if and only if, there is black colored path from the root of the tree to the leaf. Since the tree has depth $H \leq 2 + \log_{\sqrt{2}} n$, and by Lemma 2.3.7, we have that the probability of **FastCut** to succeed is at least $1/(h+1) \geq 1/(3 + \log_{\sqrt{2}} n)$.

2.4 Bibliographical Notes

The **MinCut** algorithm was developed by David Karger during his PhD thesis in Stanford. The fast algorithm is a joint work with Clifford Stein. The basic algorithm of the mincut is described in [MR95, pages 7–9], the faster algorithm is described in [MR95, pages 289–295].

Galton-Watson process. The idea of using coloring of the edges of a tree to analyze **FastCut** might be new (i.e., Section 2.3.1). It is inspired by ***Galton-Watson processes*** (which is a special case of a branching process). The problem that initiated the study of these processes goes back to the 19th century [WG75]. Victorians were worried that aristocratic surnames were disappearing, as family names passed on only through the male children. As such, a family with no male children had its family name disappear. So, imagine the number of male children of a person is an independent random variable $X \in \{0, 1, 2, \ldots\}$. Starting with a single person, its family (as far as male children are concerned) is a random tree with the degree of a node being distributed according to X. We continue recursively in constructing this tree, again, sampling the number of children for each current leaf according to the distribution of X. It is not hard to see that a family disappears if $\mathbf{E}[X] \leq 1$, and it has a constant probability of surviving if $\mathbf{E}[X] > 1$. In our case, X was the number of the two children of a node that their edges were colored black.

Of course, since infant mortality is dramatically down (as is the number of aristocrat males dying to maintain the British empire), the probability of family names to disappear is now much lower than it was in the 19th century. Interestingly, countries with family names that were introduced long time ago have very few surnames (i.e., Korean have 250 surnames, and three surnames form 45% of the population). On the other hand, countries that introduced surnames more recently have dramatically more surnames (for example, the Dutch have surnames only for the last 200 years, and there are 68,000 different family names).

Chapter 3

Complexity, the Changing Minimum and Closest Pair

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> The events of 8 September prompted Foch to draft the later legendary signal: "My centre is giving way, my right is in retreat, situation excellent. I attack." It was probably never sent.
> – John Keegan, The first world war.

3.1 Las Vegas and Monte Carlo algorithms

Definition 3.1.1. A ***Las Vegas algorithm*** is a randomized algorithms that *always* return the correct result. The only variant is that it's running time might change between executions.

An example for a Las Vegas algorithm is the **QuickSort** algorithm.

Definition 3.1.2. A ***Monte Carlo algorithm*** is a randomized algorithm that might output an incorrect result. However, the probability of error can be diminished by repeated executions of the algorithm.

The **MinCut** algorithm was an example of a Monte Carlo algorithm.

3.1.1 Complexity Classes

I assume people know what are Turing machines, **NP**, **NPC**, RAM machines, uniform model, logarithmic model. **PSPACE**, and **EXP**. If you do now know what are those things, you should read about them. Some of that is covered in the randomized algorithms book, and some other stuff is covered in any basic text on complexity theory.

Definition 3.1.3. The class **P** consists of all languages L that have a polynomial time algorithm **Alg**, such that for any input Σ^*, we have

- $x \in L \Rightarrow$ **Alg**(x) accepts,

- $x \notin L \Rightarrow$ **Alg**(x) rejects.

Definition 3.1.4. The class **NP** consists of all languages L that have a polynomial time algorithm **Alg**, such that for any input Σ^*, we have:

(i) If $x \in L \Rightarrow$ then $\exists y \in \Sigma^*$, **Alg**(x, y) accepts, where $|y|$ (i.e. the length of y) is bounded by a polynomial in $|x|$.

(ii) If $x \notin L \Rightarrow$ then $\forall y \in \Sigma^*$**Alg**(x, y) rejects.

Definition 3.1.5. For a complexity class $\mathcal{C}$, we define the complementary class co-$\mathcal{C}$ as the set of languages whose complement is in the class $\mathcal{C}$. That is

$$co - \mathcal{C} = \left\{ L \;\middle|\; \overline{L} \in \mathcal{C} \right\},$$

where $\overline{L} = \Sigma^* \setminus L$.

It is obvious that **P** = co−**P** and **P** $\subseteq$ **NP** $\cap$ co−**NP**. (It is currently unknown if **P** = **NP** $\cap$ co−**NP** or whether **NP** = co−**NP**, although both statements are believed to be false.)

Definition 3.1.6. The class **RP** (for Randomized Polynomial time) consists of all languages L that have a randomized algorithm **Alg** with worst case polynomial running time such that for any input $x \in \Sigma^*$, we have

(i) If $x \in L$ then **Pr**[**Alg**(x) accepts] $\geq 1/2$.

(ii) $x \notin L$ then **Pr**[**Alg**(x) accepts] $= 0$.

An **RP** algorithm is a Monte Carlo algorithm, but this algorithm can make a mistake only if $x \in L$. As such, co−**RP** is all the languages that have a Monte Carlo algorithm that make a mistake only if $x \notin L$. A problem which is in **RP** $\cap$ co−**RP** has an algorithm that does not make a mistake, namely a Las Vegas algorithm.

Definition 3.1.7. The class **ZPP** (for Zero-error Probabilistic Polynomial time) is the class of languages that have Las Vegas algorithms in expected polynomial time.

Definition 3.1.8. The class **PP** (for Probabilistic Polynomial time) is the class of languages that have a randomized algorithm **Alg** with worst case polynomial running time such that for any input $x \in \Sigma^*$, we have

(i) If $x \in L$ then **Pr**$\Big[$**Alg**(x) accepts$\Big] > 1/2$.

(ii) If $x \notin L$ then **Pr**$\Big[$**Alg**(x) accepts$\Big] < 1/2$.

The class **PP** is not very useful. Why?

Well, lets think about it. A randomized algorithm that just return yes/no with probability half is almost in **PP**, as it return the correct answer with probability half. An algorithm is in **PP** needs to be slightly better, and be correct with probability better than half, but how much better can be made to be arbitrarily close to $1/2$. In particular, there is no way to do effective amplification with such an algorithm.

Definition 3.1.9. The class **BPP** (for Bounded-error Probabilistic Polynomial time) is the class of languages that have a randomized algorithm **Alg** with worst case polynomial running time such that for any input $x \in \Sigma^*$, we have

(i) If $x \in L$ then $\mathbf{Pr}[\mathbf{Alg}(x) \text{ accepts}] \geq 3/4$.

(ii) If $x \notin L$ then $\mathbf{Pr}[\mathbf{Alg}(x) \text{ accepts}] \leq 1/4$.

3.2 How many times can a minimum change, before it is THE minimum?

Let $a_1, \ldots, a_n$ be a set of n numbers, and let us randomly permute them into the sequence $b_1, \ldots, b_n$. Next, let $c_i = \min_{k=1}^{i} b_i$, and let X be the random variable which is the number of distinct values that appears in the sequence $c_1, \ldots, c_n$. What is the expectation of X?

Lemma 3.2.1. *In expectation, the number of times the minimum of a prefix of n randomly permuted numbers change, is $O(\log n)$. That is $\mathbf{E}[X] = O(\log n)$.*

Proof: Consider the indicator variable X_i, such that $X_i = 1$ if $c_i \neq c_{i-1}$. The probability for that is $\leq q1/i$, since this is the probability that the smallest number if $b_1, \ldots, b_i$ is b_i. As such, we have $X = \sum_i X_i$, and $\mathbf{E}[X] = \sum_i \mathbf{E}[X_i] = \sum_{i=1}^{n} \frac{1}{i} = O(\log n)$. ∎

3.3 Closest Pair

Assumption 3.3.1. Throughout the discourse, we are going to assume that every hashing operation takes (worst case) constant time. This is quite a reasonable assumption when true randomness is available (using for example perfect hashing [CLRS01]). We probably will revisit this issue later in the course.

For a real positive number r and a point $\mathsf{p} = (x, y)$ in $\mathbb{R}^2$, define

$$\mathrm{G}_r(\mathsf{p}) := \left(\left\lfloor \frac{x}{r} \right\rfloor r \ , \ \left\lfloor \frac{y}{r} \right\rfloor r \right) \in \mathbb{R}^2.$$

We call r the ***width*** of the ***grid*** G_r. Observe that G_r partitions the plane into square regions, which we call grid ***cells***. Formally, for any $i, j \in \mathbb{Z}$, the intersection of the half-planes $x \geq ri$, $x < r(i+1)$, $y \geq rj$ and $y < r(j+1)$ is said to be a ***grid cell***. Further we define a ***grid cluster*** as a block of 3×3 contiguous grid cells.

For a point set P, and a parameter r, the partition of P into subsets by the grid G_r, is denoted by $\mathrm{G}_r(\mathsf{P})$. More formally, two points $\mathsf{p}, \mathsf{q} \in \mathsf{P}$ belong to the same set in the partition $\mathrm{G}_r(\mathsf{P})$, if both points are being mapped to the same grid point or equivalently belong to the same grid cell.

Note, that every grid cell C of G_r, has a unique ID; indeed, let $\mathsf{p} = (x, y)$ be any point in C, and consider the pair of integer numbers $\mathrm{id}_C = \mathrm{id}(\mathsf{p}) = (\lfloor x/r \rfloor, \lfloor y/r \rfloor)$. Clearly, only points inside

C are going to be mapped to id_C. This is very useful, since we can store a set P of points inside a grid efficiently. Indeed, given a point p, compute its $\mathrm{id}(\mathsf{p})$. We associate with each unique id a data-structure that stores all the points falling into this grid cell (of course, we do not maintain such data-structures for grid cells which are empty). So, once we computed $\mathrm{id}(\mathsf{p})$, we fetch the data structure for this cell, by using hashing. Namely, we store pointers to all those data-structures in a hash table, where each such data-structure is indexed by its unique id. Since the ids are integer numbers, we can do the hashing in constant time.

We are interested in solving the following problem.

Problem 3.3.2. Given a set P of n points in the plane, find the pair of points closest to each other. Formally, return the pair of points realizing $\mathcal{CP}(\mathsf{P}) = \min_{\mathsf{p},\mathsf{q}\in P} \|\mathsf{pq}\|$.

Lemma 3.3.3. *Given a set P of n points in the plane, and a distance r, one can verify in linear time, whether or not $\mathcal{CP}(\mathsf{P}) < r$ or $\mathcal{CP}(\mathsf{P}) \geq r$.*

Proof: Indeed, store the points of P in the grid G_r. For every non-empty grid cell, we maintain a linked list of the points inside it. Thus, adding a new point p takes constant time. Indeed, compute $\mathrm{id}(p)$, check if $\mathrm{id}(p)$ already appears in the hash table, if not, create a new linked list for the cell with this ID number, and store p in it. If a data-structure already exist for $\mathrm{id}(p)$, just add p to it.

This takes $O(n)$ time. Now, if any grid cell in $\mathrm{G}_r(\mathsf{P})$ contains more than, say, 9 points of P, then it must be that the $\mathcal{CP}(\mathsf{P}) < r$. Indeed, consider a cell C containing more than four points of P, and partition C into 3×3 equal squares. Clearly, one of those squares must contain two points of P, and let C' be this square. Clearly, the diameter of $C' = \mathrm{diam}(C)/3 = \sqrt{r^2 + r^2}/3 < r$. Thus, the (at least) two points of P in C' are in distance smaller than r from each other.

Thus, when we insert a point p, we can fetch all the points of P that were already inserted, for the cell of P, and the 8 adjacent cells. All those cells, must contain at most 9 points of P (otherwise, we would already have stopped since the $\mathcal{CP}(\cdot)$ of inserted points, is smaller than r). Let S be the set of all those points, and observe that $|S| \leq 9 \cdot 9 = O(1)$. Thus, we can compute by brute force the closest point to p in S. This takes $O(1)$ time. If $\mathsf{d}(p, S) < r$, we stop, otherwise, we continue to the next point, where $\mathsf{d}(p, S) = \min_{s\in S} \|ps\|$.

Overall, this takes $O(n)$ time. As for correctness, first observe that if $\mathcal{CP}(\mathsf{P}) > r$ then the algorithm would never make a mistake, since it returns '$\mathcal{CP}(\mathsf{P}) < r$' only after finding a pair of points of P with distance smaller than r. Thus, assume that p, q are the pair of points of P realizing the closest pair, and $\|pq\| = \mathcal{CP}(\mathsf{P}) < r$. Clearly, when the later of them, say p, is being inserted, the set S would contain q, and as such the algorithm would stop and return "$\mathcal{CP}(\mathsf{P}) < r$". ∎

Lemma 3.3.3 hints on a natural way to compute $\mathcal{CP}(\mathsf{P})$. Indeed, permute the points of P in arbitrary fashion, and let $P = \langle p_1, \ldots, p_n \rangle$. Next, let $r_i = \mathcal{CP}\Big(\{p_1, \ldots, p_i\}\Big)$. We can check if $r_{i+1} < r_i$, by just calling the algorithm for Lemma 3.3.3 on P_{i+1} and r_i. In fact, if $r_{i+1} < r_i$, the algorithm of Lemma 3.3.3, would give us back the distance r_{i+1} (with the other point realizing this distance).

In fact, consider the "good" case, where $r_{i+1} = r_i = r_{i-1}$. Namely, the length of the shortest pair does not change. In this case, we do not need to rebuild the data structure of Lemma 3.3.3, for each point. We can just reuse it from the previous iteration. Thus, inserting a single point takes constant time, as long as the closest pair does not change.

Things become bad, when $r_i < r_{i-1}$. Because then, we need to rebuild the grid, and reinsert all the points of $\mathsf{P}_i = \langle p_1, \ldots, p_i \rangle$ into the new grid $\mathrm{G}_{r_i}(\mathsf{P}_i)$. This takes $O(i)$ time.

So, if the closest pair radius, in the sequence $r_1, \ldots, r_n$ changes only k times, then the running time of our algorithm would be $O(nk)$. In fact, we can do even better.

Theorem 3.3.4. *Let* P *be a set of n points in the plane, one can compute the closest pair of points of* P *in expected linear time.*

Proof: Pick a random permutation of the points of P, let $\langle p_1, \ldots, p_n \rangle$ be this permutation. Let $r_2 = \|p_1 p_2\|$, and start inserting the points into the data structure of Lemma 3.3.3. In the ith iteration, if $r_i = r_{i-1}$, then this insertion takes constant time. If $r_i < r_{i-1}$, then we rebuild the grid and reinsert the points. Namely, we recompute $\mathrm{G}_{r_i}(\mathsf{P}_i)$.

To analyze the running time of this algorithm, let X_i be the indicator variable which is 1 if $r_i \neq r_{i-1}$, and 0 otherwise. Clearly, the running time is proportional to

$$R = 1 + \sum_{i=2}^{n} (1 + X_i \cdot i) .$$

Thus, the expected running time is

$$\mathbf{E}[R] \quad = \quad 1 + \mathbf{E}\left[1 + \sum_{i=2}^{n} (1 + X_i \cdot i) \right] = n + \sum_{i=2}^{n} (\mathbf{E}[X_i] \cdot i) = n + \sum_{i=2}^{n} i \cdot \mathbf{Pr}[X_1 = 1] ,$$

by linearity of expectation and since for an indicator variable X_i, we have that $\mathbf{E}[X_i] = \mathbf{Pr}[X_i = 1]$.

Thus, we need to bound $\mathbf{Pr}[X_i = 1] = \mathbf{Pr}[r_i < r_{i-1}]$. To bound this quantity, fix the points of P_i, and randomly permute them. A point $\mathsf{q} \in \mathsf{P}_i$ is called ***critical***, if $\mathcal{CP}(\mathsf{P}_i \setminus \{\mathsf{q}\}) > \mathcal{CP}(\mathsf{P}_i)$. If there are no critical points, then $r_{i-1} = r_i$ and then $\mathbf{Pr}[X_i = 1] = 0$. If there is one critical point, than $\mathbf{Pr}[X_i = 1] = 1/i$, as this is the probability that this critical point, would be the last point in the random permutation of P_i.

If there are two critical points, and let p, q be this unique pair of points of P_i realizing $\mathcal{CP}(\mathsf{P}_i)$. The quantity r_i is smaller than r_{i-1}, if either p or q are p_i. But the probability for that is $2/i$ (i.e., the probability in a random permutation of i objects, that one of two marked objects would be the last element in the permutation).

Observe, that there can not be more than two critical points. Indeed, if p and q are two points that realize the closest distance, than if there is a third critical point r, then $\mathcal{CP}(\mathsf{P}_i \setminus \{\mathsf{r}\}) = \|\mathsf{pq}\|$, and r is not critical.

We conclude that

$$\mathbf{E}[R] = n + \sum_{i=2}^{n} i \cdot \mathbf{Pr}[X_1 = 1] \leq n + \sum_{i=2}^{n} i \cdot \frac{2}{i} \leq 3n.$$

As such, the expected running time of this algorithm is $O\Big(\mathbf{E}\Big[R\Big]\Big) = O(n)$. ∎

Theorem 3.3.4 is a surprising result, since it implies that ***uniqueness*** (i.e., deciding if n real numbers are all distinct) can be solved in linear time. However, there is a lower bound of $\Omega(n \log n)$ on uniqueness, using the comparison tree model. This reality dysfunction, can be easily explained, once one realizes that the model of computation of Theorem 3.3.4 is considerably stronger, using hashing, randomization, and the floor function.

3.4 Bibliographical notes

Section 3.1 follows [MR95, Section 1.5]. The closest-pair algorithm follows Golin *et al.* [GRSS95]. This is in turn a simplification of a result of Rabin [Rab76]. Smid provides a survey of such algorithms [Smi00].

Chapter 4

The Occupancy and Coupon Collector problems

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

4.1 Preliminaries

Definition 4.1.1 (Variance and Standard Deviation). For a random variable X, let $\mathbf{V}[X] = \mathbf{E}\left[(X - \mu_X)^2\right] = \mathbf{E}\left[X^2\right] - \mu_X^2$ denote the ***variance*** of X, where $\mu_X = \mathbf{E}[X]$. Intuitively, this tells us how concentrated is the distribution of X.

The ***standard deviation*** of X, denoted by σ_X is the quantity $\sqrt{\mathbf{V}[X]}$.

Observation 4.1.2. *(i)* $\mathbf{V}[cX] = c^2\,\mathbf{V}[X]$.

(ii) For X and Y independent variables, we have $\mathbf{V}[X + Y] = \mathbf{V}[X] + \mathbf{V}[Y]$.

Definition 4.1.3 (Bernoulli distribution). Assume, that one flips a coin and get 1 (heads) with probability p, and 0 (i.e., tail) with probability $q = 1 - p$. Let X be this random variable. The variable X is has *Bernoulli distribution with parameter p*. Then $\mathbf{E}[X] = p$, and $\mathbf{V}[X] = pq$.

Definition 4.1.4 (Binomial distribution). Assume that we repeat a Bernoulli experiments n times (independently!). Let $X_1, \ldots, X_n$ be the resulting random variables, and let $X = X_1 + \cdots + X_n$. The variable X has the *binomial distribution* with parameters n and p. We denote this fact by $X \sim B(n, p)$. We have

$$b(k; n, p) = \mathbf{Pr}[X = k] = \binom{n}{k} p^k q^{n-k}.$$

Also, $\mathbf{E}[X] = np$, and $\mathbf{V}[X] = npq$.

Observation 4.1.5. *Let $C_1, \ldots, C_n$ be random events (not necessarily independent). Than*

$$\mathbf{Pr}\left[\bigcup_{i=1}^{n} C_i\right] \leq \sum_{i=1}^{n} \mathbf{Pr}[C_i]\,.$$

(This is usually referred to as the **union bound**.*) If $C_1, \ldots, C_n$ are* disjoint *events then*

$$\mathbf{Pr}\left[\bigcup_{i=1}^{n} C_i\right] = \sum_{i=1}^{n} \mathbf{Pr}[C_i]\,.$$

Lemma 4.1.6. *For any positive integer n, we have:*

(i) $(1 + 1/n)^n \leq e$.

(ii) $(1 - 1/n)^{n-1} \geq e^{-1}$.

(iii) $n! \geq (n/e)^n$.

(iv) For any $k \leq n$, *we have:* $\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{ne}{k}\right)^k$.

Proof: (i) Indeed, $1 + 1/n \leq \exp(1/n)$, since $1 + x \leq e^x$, for $x \geq 0$. As such $(1 + 1/n)^n \leq \exp(n(1/n)) = e$.

(ii) Rewriting the inequality, we have that we need to prove $\left(\frac{n-1}{n}\right)^{n-1} \geq \frac{1}{e}$. This is equivalence to proving $e \geq \left(\frac{n}{n-1}\right)^{n-1} = \left(1 + \frac{1}{n-1}\right)^{n-1}$, which is our friend from (i).

(iii) Indeed,

$$\frac{n^n}{n!} \leq \sum_{i=0}^{\infty} \frac{n^i}{i!} = e^n,$$

by the Taylor expansion of $e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$. This implies that $(n/e)^n \leq n!$, as required.

(iv) Indeed, for any $k \leq n$, we have $\frac{n}{k} \leq \frac{n-1}{k-1}$ since $kn - n = n(k-1) \leq k(n-1) = kn - k$. As such, $\frac{n}{k} \leq \frac{n-i}{k-i}$, for $1 \leq i \leq k-1$. As such,

$$\left(\frac{n}{k}\right)^k \leq \frac{n}{k} \cdot \frac{n-1}{k-1} \cdots \frac{n-i}{k-i} \cdots \frac{n-k+1}{1} = \frac{n!}{(n-k)!k!} = \binom{n}{k}.$$

As for the other direction, we have

$$\binom{n}{k} \leq \frac{n^k}{k!} \leq \frac{n^k}{\left(\frac{k}{e}\right)^k} = \left(\frac{ne}{k}\right)^k,$$

by (iii). ∎

4.2 Occupancy Problems

Problem 4.2.1. We are throwing m balls into n bins randomly (i.e., for every ball we randomly and uniformly pick a bin from the n available bins, and place the ball in the bin picked). What is the maximum number of balls in any bin? What is the number of bins which are empty? How many balls do we have to throw, such that all the bins are non-empty, with reasonable probability?

Let X_i be the number of balls in the ith bins, when we throw n balls into n bins (i.e., $m = n$). Clearly,

$$\mathbf{E}[X_i] = \sum_{j=1}^{n} \mathbf{Pr}\Big[\text{The } j\text{th ball fall in } i\text{th bin}\Big] = n \cdot \frac{1}{n} = 1,$$

by linearity of expectation. The probability that the first bin has exactly i balls is

$$\binom{n}{i}\left(\frac{1}{n}\right)^i\left(1 - \frac{1}{n}\right)^{n-i} \leq \binom{n}{i}\left(\frac{1}{n}\right)^i \leq \left(\frac{ne}{i}\right)^i\left(\frac{1}{n}\right)^i = \left(\frac{e}{i}\right)^i$$

This follows by Lemma 4.1.6 (iv).

Let $C_j(k)$ be the event that the jth bin has k or more balls in it. Then,

$$\mathbf{Pr}[C_1(k)] \leq \sum_{i=k}^{n} \left(\frac{e}{i}\right)^i \leq \left(\frac{e}{k}\right)^k \left(1 + \frac{e}{k} + \frac{e^2}{k^2} + \ldots\right) = \left(\frac{e}{k}\right)^k \frac{1}{1 - e/k}.$$

Let $k^* = \lceil (3 \ln n)/\ln\ln n \rceil$. Then,

$$\begin{aligned}
\mathbf{Pr}[C_1(k^*)] &\leq \left(\frac{e}{k^*}\right)^{k^*} \frac{1}{1 - e/k^*} \leq 2\left(\frac{e}{(3\ln n)/\ln\ln n}\right)^{k^*} = 2(\exp(1 - \ln 3 - \ln\ln n + \ln\ln\ln n))^{k^*} \\
&\leq 2\Big(\exp(-\ln\ln n + \ln\ln\ln n)\Big)^{k^*} \\
&\leq 2\exp\left(-3\ln n + 6\ln n \frac{\ln\ln\ln n}{\ln\ln n}\right) \leq 2\exp(-2.5\ln n) \leq \frac{1}{n^2},
\end{aligned}$$

for n large enough. We conclude, that since there are n bins and they have identical distributions that

$$\mathbf{Pr}\Big[\text{any bin contains more than } k^* \text{ balls}\Big] \leq \sum_{i=1}^{n} C_i(k^*) \leq \frac{1}{n}.$$

Theorem 4.2.2. *With probability at least $1 - 1/n$, no bin has more than $k^* = \left\lceil \frac{3\ln n}{\ln\ln n} \right\rceil$ balls in it.*

Exercise 4.2.3. Show that for $m = n \ln n$, with probability $1 - o(1)$, every bin has $O(\log n)$ balls.

It is interesting to note, that if at each iteration we randomly pick d bins, and throw the ball into the bin with the smallest number of balls, then one can do much better. We currently do not have the machinery to prove the following theorem, but hopefully we would prove it later in the course.

Theorem 4.2.4. *Suppose that n balls are sequentially places into n bins in the following manner. For each ball, $d \geq 2$ bins are chosen independently and uniformly at random (with replacement). Each ball is placed in the least full of the d bins at the time of placement, with ties broken randomly. After all the balls are places, the maximum load of any bin is at most $\ln\ln n/(\ln d) + O(1)$, with probability at least $1 - o(1/n)$.*

Note, even by setting $d = 2$, we get considerable improvement. A proof of this theorem can be found in the work by Azar *et al.* [ABKU00].

4.2.1 The Probability of all bins to have exactly one ball

Next, we are interested in the probability that all m balls fall in distinct bins. Let X_i be the event that the ith ball fell in a distinct bin from the first $i - 1$ balls. We have:

$$\begin{aligned}
\mathbf{Pr}\left[\bigcap_{i=2}^{m} X_i\right] &= \mathbf{Pr}[X_2] \prod_{i=3}^{m} \mathbf{Pr}\left[X_i \;\middle|\; \bigcap_{j=2}^{i-1} X_j\right] \leq \prod_{i=2}^{m} \left(\frac{n - i + 1}{n}\right) \leq \prod_{i=2}^{m} \left(1 - \frac{i-1}{n}\right) \\
&\leq \prod_{i=2}^{m} e^{-(i-1)/n} \leq \exp\left(-\frac{m(m-1)}{2n}\right),
\end{aligned}$$

thus for $m = \left\lceil \sqrt{2n} + 1 \right\rceil$, the probability that all the m balls fall in different bins is smaller than $1/e$.

This is sometime referred to as the ***birthday paradox***. You have $m = 30$ people in the room, and you ask them for the date (day and month) of their birthday (i.e., $n = 365$). The above shows that the probability of all birthdays to be distinct is $\exp(-30 \cdot 29/730) \leq 1/e$. Namely, there is more than 50% chance for a birthday collision, a simple but counterintuitive phenomena.

4.3 The Markov and Chebyshev inequalities

We remind the reader that for a random variable X assuming real values, its *expectation* is $\mathbf{E}[Y] = \sum_y y \cdot \mathbf{Pr}[Y = y]$. Similarly, for a function $f(\cdot)$, we have $\mathbf{E}[f(Y)] = \sum_y f(y) \cdot \mathbf{Pr}[Y = y]$.

Theorem 4.3.1 (Markov Inequality). *Let Y be a random variable assuming only non-negative values. Then for all $t > 0$, we have*

$$\mathbf{Pr}[Y \geq t] \leq \frac{\mathbf{E}[Y]}{t}$$

Proof: Indeed,

$$\begin{aligned} \mathbf{E}[Y] &= \sum_{y \geq t} y \,\mathbf{Pr}[Y = y] + \sum_{y < t} y \,\mathbf{Pr}[Y = y] \geq \sum_{y \geq t} y \,\mathbf{Pr}[Y = y] \\ &\geq \sum_{y \geq t} t \,\mathbf{Pr}[Y = y] = t \,\mathbf{Pr}[Y \geq t]\,. \end{aligned}$$

■

Markov inequality is tight, as the following exercise testifies.

Exercise 4.3.2. For any (integer) $k > 1$, define a random positive variable X_k such that $\mathbf{Pr}\Big[X_k \geq k\,\mathbf{E}[X_k]\Big] = \frac{1}{k}$.

Theorem 4.3.3 (Chebyshev inequality). $\mathbf{Pr}[|X - \mu_X| \geq t\sigma_X] \leq \frac{1}{t^2}$, *where $\mu_X = \mathbf{E}[X]$ and $\sigma_X = \sqrt{\mathbf{V}[X]}$.*

Proof: Note that

$$\mathbf{Pr}\Big[\,|X - \mu_X| \geq t\sigma_X\Big] = \mathbf{Pr}\Big[(X - \mu_X)^2 \geq t^2\sigma_X^2\Big]\,.$$

Set $Y = (X - \mu_X)^2$. Clearly, $\mathbf{E}[Y] = \sigma_X^2$. Now, apply Markov inequality to Y. ■

4.4 The Coupon Collector's Problem

There are n types of coupons, and at each trial one coupon is picked in random. How many trials one has to perform before picking all coupons? Let m be the number of trials performed. We would like to bound the probability that m exceeds a certain number, and we still did not pick all coupons.

Let $C_i \in \{1, \ldots, n\}$ be the coupon picked in the ith trial. The jth trial is a success, if C_j was not picked before in the first $j-1$ trials. Let X_i denote the number of trials from the ith success, till after the $(i+1)$th success. Clearly, the number of trials performed is

$$X = \sum_{i=0}^{n-1} X_i.$$

Now, the probability of X_i to succeed in a trial is $p_i = (n-i)/n$, and X_i has the geometric distribution with probability p_i. As such $\mathbf{E}[X_i] = 1/p_i$, and $\mathbf{V}[X_i] = q/p^2 = (1-p_i)/p_i^2$.

Thus,

$$\mathbf{E}[X] = \sum_{i=0}^{n-1} \mathbf{E}[X_i] = \sum_{i=0}^{n-1} \frac{n}{n-i} = nH_n = n(\ln n + \Theta(1)) = n \ln n + O(n),$$

where $H_n = \sum_{i=1}^{n} 1/i$ is the nth Harmonic number.

As for variance, using the independence of $X_0, \ldots, X_{n-1}$, we have

$$\begin{aligned}
\mathbf{V}[X] &= \sum_{i=0}^{n-1} \mathbf{V}[X_i] = \sum_{i=0}^{n-1} \frac{1-p_i}{p_i^2} = \sum_{i=0}^{n-1} \frac{1-(n-i)/n}{\left(\frac{n-i}{n}\right)^2} = \sum_{i=0}^{n-1} \frac{i/n}{\left(\frac{n-i}{n}\right)^2} = \sum_{i=0}^{n-1} \frac{i}{n}\left(\frac{n}{n-i}\right)^2 \\
&= n \sum_{i=0}^{n-1} \frac{i}{(n-i)^2} = n \sum_{i=1}^{n} \frac{n-i}{i^2} = n\left(\sum_{i=1}^{n} \frac{n}{i^2} - \sum_{i=1}^{n} \frac{1}{i}\right) = n^2 \sum_{i=1}^{n} \frac{1}{i^2} - nH_n.
\end{aligned}$$

Since, $\lim_{n\to\infty} \sum_{i=1}^{n} \frac{1}{i^2} = \pi^2/6$, we have $\lim_{n\to\infty} \dfrac{\mathbf{V}[X]}{n^2} = \dfrac{\pi^2}{6}$.

This implies a weak bound on the concentration of X, using Chebyshev inequality, but this is going to be quite weaker than what we implied we can do. Indeed, we have

$$\mathbf{Pr}\left[X \geq n \log n + n + t \cdot n \frac{\pi}{\sqrt{6}}\right] \leq \mathbf{Pr}\Big[\, |X - \mathbf{E}[X]| \geq t\, \mathbf{V}[X]\Big] \leq \frac{1}{t^2},$$

for any t.

Stronger bounds will be shown in the next lecture.

4.5 Notes

The material in this note covers parts of [MR95, sections 3.1,3.2,3.6]

Chapter 5

The Occupancy and Coupon Collector Problems II

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> There is not much talking now. A silence falls upon them all. This is no time to talk of hedges and fields, or the beauties of any country. Sadness and fear and hate, how they well up in the heart and mind, whenever one opens the pages of these messengers of doom. Cry for the broken tribe, for the law and custom that is gone. Aye, and cry aloud for the man who is dead, for the woman and children bereaved. Cry, the beloved country, these things are not yet at an end. The sun pours down on the earth, on the lovely land that man cannot enjoy. He knows only the fear of his heart.
>
> – Alan Paton, Cry, the beloved country.

5.1 The Coupon Collector's Problem Revisited

There are n types of coupons, and at each trial one coupon is picked in random. How many trials one has to perform before picking all coupons? Let m be the number of trials performed. We would like to bound the probability that m exceeds a certain number, and we still did not pick all coupons.

In the previous lecture, we showed that

$$\mathbf{Pr}\left[\#\text{ of trials} \geq n\log n + n + t\cdot n\frac{\pi}{\sqrt{6}}\right] \leq \frac{1}{t^2},$$

for any t.

A stronger bound, follows from the following observation. Let Z_i^r denote the event that the ith coupon was not picked in the first r trials. Clearly,

$$\mathbf{Pr}[Z_i^r] = \left(1-\frac{1}{n}\right)^r \leq \exp\left(-\frac{r}{n}\right).$$

Thus, for $r = \beta n\log n$, we have $\mathbf{Pr}\left[Z_i^r\right] \leq \exp(-(\beta n\log n)/n) = n^{-\beta}$. Thus,

$$\mathbf{Pr}[X > \beta n\log n] \leq \mathbf{Pr}\left[\bigcup_i Z_i^{\beta n\log n}\right] \leq n\cdot\mathbf{Pr}[Z_1] \leq n^{-\beta+1}.$$

Lemma 5.1.1. *Let the random variable X denote the number of trials for collecting each of the n types of coupons. Then, we have $\mathbf{Pr}[X > n \ln n + cn] \leq e^{-c}$.*

Proof: The probability we fail to pick the first type of coupon is $\alpha = (1 - 1/n)^m \leq \exp\left(-\frac{n \ln n + cn}{n}\right) = \exp(-c)/n$. As such, using the union bound, the probability we fail to pick all n types of coupons is bounded by $n\alpha = \exp(-c)$, as claimed. ∎

In the following, we show a slightly stronger bound on the probability, which is $1 - \exp(-e^{-c})$. To see that it is indeed stronger, observe that $e^{-c} \geq 1 - \exp(-e^{-c})$.

5.1.0.1 A slightly stronger bound

Lemma 5.1.2. *Let $c > 0$ be a constant, $m = n \ln n + cn$ for a positive integer n. Then for any constant k, we have*

$$\lim_{n\to\infty} \binom{n}{k}\left(1 - \frac{k}{n}\right)^m = \frac{\exp(-ck)}{k!}.$$

Proof: By Lemma 5.3.3, we have

$$\left(1 - \frac{k^2 m}{n^2}\right)\exp\left(-\frac{km}{n}\right) \leq \left(1 - \frac{k}{n}\right)^m \leq \exp\left(-\frac{km}{n}\right).$$

Observe also that $\lim_{n\to\infty}\left(1 - \frac{k^2 m}{n}\right) = 1$, and $\exp(-km/n) = n^{-k}\exp(-ck)$. Also,

$$\lim_{n\to\infty} \binom{n}{k}\frac{k!}{n^k} = \lim_{n\to\infty} \frac{n(n-1)\cdots(n-k+1)}{n^k} = 1.$$

Thus,

$$\lim_{n\to\infty} \binom{n}{k}\left(1 - \frac{k}{n}\right)^m = \lim_{n\to\infty} \frac{n^k}{k!}\exp\left(-\frac{km}{n}\right) = \lim_{n\to\infty} \frac{n^k}{k!} n^{-k}\exp(-ck) = \frac{\exp(-ck)}{k!}.$$

∎

Theorem 5.1.3. *Let the random variable X denote the number of trials for collecting each of the n types of coupons. Then, for any constant $c \in \mathbb{R}$, and $m = n \ln n + cn$, we have*

$$\lim_{n\to\infty} \mathbf{Pr}[X > m] = 1 - \exp(-e^{-c}).$$

Before dwelling into the proof, observe that $1 - \exp(-e^{-c}) \approx 1 - (1 - e^{-c}) = e^{-c}$, as such the bound in the above theorem is indeed a considerable improvement over the previous bounds.

Proof: We have $\mathbf{Pr}[X > m] = \mathbf{Pr}\left[\cup_i Z_i^m\right]$. By inclusion-exclusion, we have

$$\mathbf{Pr}\left[\bigcup_i Z_i^m\right] = \sum_{i=1}^{n} (-1)^{i+1} P_i^n,$$

where

$$P_j^n = \sum_{1 \leq i_1 < i_2 < \ldots < i_j \leq n} \mathbf{Pr}\left[\bigcap_{v=1}^{j} Z_{i_v}^m\right].$$

Let $S^n_k = \sum_{i=1}^{k} (-1)^{i+1} P^n_i$. We know that $S^n_{2k} \leq \mathbf{Pr}\left[\bigcup_i Z^m_i\right] \leq S^n_{2k+1}$.

By symmetry,

$$P^n_k = \binom{n}{k} \mathbf{Pr}\left[\bigcap_{v=1}^{k} Z^m_v\right] = \binom{n}{k}\left(1 - \frac{k}{n}\right)^m,$$

Thus, $P_k = \lim_{n\to\infty} P^n_k = \exp(-ck)/k!$, by Lemma 5.1.2.

Let

$$S_k = \sum_{j=1}^{k} (-1)^{j+1} P_j = \sum_{j=1}^{k} (-1)^{j+1} \cdot \frac{\exp(-cj)}{j!}$$

Clearly, $\lim_{k\to\infty} S_k = 1 - \exp(-e^{-c})$ by the Taylor expansion of $\exp(x)$ for $x = -e^{-c}$. Indeed,

$$\exp(x) = \sum_{j=0}^{\infty} \frac{x^j}{j!} = \sum_{j=0}^{\infty} \frac{(-e^{-c})^j}{j!} = 1 + \sum_{j=0}^{\infty} \frac{(-1)^j e^{-cj}}{j!}$$

Clearly, $\lim_{n\to\infty} S^n_k = S_k$ and $\lim_{k\to\infty} S_k = 1 - \exp(-e^{-c})$. Thus, (using fluffy math), we have

$$\lim_{n\to\infty} \mathbf{Pr}[X > m] = \lim_{n\to\infty} \mathbf{Pr}[\cup_{i=1}^{n} Z^m_i] = \lim_{n\to\infty} \lim_{k\to\infty} S^n_k = \lim_{k\to\infty} S_k = 1 - \exp(-e^{-c}).$$

■

5.2 Randomized Selection

We are given a set S of n distinct elements, with an associated ordering. For $t \in S$, let $r_S(t)$ denote the rank of t (the smallest element in S has rank 1). Let $S_{(i)}$ denote the ith element in the sorted list of S.

Given k, we would like to compute S_k (i.e., select the kth element). The code of **LazySelect** is depicted in Figure 5.1.

Exercise 5.2.1. Show how to compute the ranks of $r_S(a)$ and $r_S(b)$, such that the expected number of comparisons performed is $1.5n$.

Consider the element $S_{(k)}$ and where it is mapped to in the random sample R. Consider the interval of values

$$\mathrm{I}(j) = \left[R_{(\alpha(j))}, R_{(\beta(j))}\right] = \left\{R_{(k)} \;\middle|\; \alpha(j) \leq k \leq \beta(j)\right\},$$

where $\alpha(j) = j \cdot n^{-1/4} - \sqrt{n}$ and $\beta(j) = j \cdot n^{-1/4} + \sqrt{n}$.

Lemma 5.2.2. *For a fixed j, we have that* $\mathbf{Pr}\left[S_{(j)} \in \mathrm{I}(j)\right] \geq 1 - 1/(4n^{1/4})$.

Proof: There are two possible bad events: (i) $S_{(j)} < R_{\alpha(j)}$ and (ii) $R_{\beta(j)} < S_{(j)}$. Let X_i be an indicator variable which is 1 if the ith sample is smaller equal to $S_{(j)}$, otherwise 0. We have $p = \mathbf{Pr}[X_i]] = j/n$ and $q = 1 - j/n$. The random variable $X = \sum_{i=1}^{n^{3/4}} X_i$ is the rank of $S_{(j)}$ in the random sample. Clearly, $X \sim B\left(3/4, j/n\right)$ (i.e., X has a binomial distribution with $p = j/n$, and $n^{3/4}$ trials). As such, we have $\mathbf{E}[X] = pn^{3/4}$ and $\mathbf{V}[X] = n^{3/4}pq$.

Now, by Chebyshev inequality

$$\mathbf{Pr}\left[\left|X - pn^{3/4}\right| \geq t\sqrt{n^{3/4}pq}\right] \leq \frac{1}{t^2}.$$

Func **LazySelect**(S , k)
`Input:` S - set of n elements, k - index of element to be output.
begin
repeat
$R \leftarrow \left\{\text{Sample with replacement of } n^{3/4} \text{ elements from } S\right\}$
$\cup \{-\infty, +\infty\}$.
Sort R.
$l \leftarrow \max\left(1, \left\lfloor kn^{-1/4} - \sqrt{n} \right\rfloor\right)$, $h \leftarrow \min\left(n^{3/4}, \left\lfloor kn^{-1/4} + \sqrt{n} \right\rfloor\right)$
$a \leftarrow R_{(l)}$, $b \leftarrow R_{(h)}$.
Compute the ranks $r_S(a)$ and $r_S(b)$ of b in S
`/* using` $2n$ `comparisons */`
$P \leftarrow \left\{ y \in S \;\middle|\; a \le y \le b \right\}$
`/* done when computing the rank of` a `and` b `*/`
Until $(r_S(a) \le k \le r_S(b))$ **and** $\left(|P| \le 8n^{3/4} + 2\right)$
Sort P in $O(n^{3/4} \log n)$ time.
return $P_{k-r_S(a)+1}$
end **LazySelect**

Figure 5.1: The **LazySelect** algorithm.

Since $pn^{3/4} = jn^{-1/4}$ and $\sqrt{n^{3/4}(j/n)(1-j/n)} \le n^{3/8}/2$, we have that the probability of $a > S_{(j)}$ or $b > S_{(j)}$ is

$$\begin{aligned}
\mathbf{Pr}\left[S_{(j)} < R_{\alpha(j)} \text{ or } R_{\beta(j)} < S_{(j)}\right] &= \mathbf{Pr}\left[X < (jn^{-1/4} - \sqrt{n}) \;\text{ or }\; X > (jn^{-1/4} + \sqrt{n})\right] \\
&= \mathbf{Pr}\left[|X - jn^{-1/4}| \ge 2n^{1/8} \cdot \frac{n^{3/8}}{2}\right] \\
&\le \frac{1}{(2n^{1/8})^2} = \frac{1}{4n^{1/4}}.
\end{aligned}$$

■

Lemma 5.2.3. **LazySelect** *succeeds with probability* $\ge 1 - O(n^{-1/4})$ *in the first iteration. And it performs only* $2n + o(n)$ *comparisons.*

Proof: By Lemma 5.2.2, we know that $S_{(k)} \in \mathrm{I}(k)$ with probability $\ge 1 - 1/(4n^{1/4})$. This in turn implies that $S_{(k)} \in P$. Thus, the only possible bad event is that the set P is too large. To this end, set $k^- = k - 3n^{3/4}$ and $k^+ = k + 3n^{3/4}$, and observe that, by definition, it holds $\mathrm{I}(k^-) \cap \mathrm{I}(k) = \emptyset$ and $\mathrm{I}(k) \cap \mathrm{I}(k^+) = \emptyset$. As such, we know by Lemma 5.2.2, that $S_{(k^-)} \in \mathrm{I}(k^-)$ and $S_{(k^+)} \in \mathrm{I}(k^+)$, and this holds with probability $\ge 1 - \frac{2}{4n^{1/4}}$. As such, the set P, which is by definition contained in the range $\mathrm{I}(k)$, has only elements that are larger than $S_{(k^-)}$ and smaller than $S_{(k^+)}$. As such, the size of P is bounded by $k^+ - k^- = 6n^{3/4}$. Thus, the algorithm succeeds in the first iteration, with probability $\ge 1 - \frac{3}{4n^{1/4}}$.

As for the number of comparisons, an iteration requires

$$O(n^{3/4} \log n) + 2n + O(n^{3/4} \log n) = 2n + o(n)$$

comparisons ■

Any deterministic selection algorithm requires $2n$ comparisons, and **LazySelect** can be changed to require only $1.5n + o(n)$ comparisons (expected).

5.3 A technical lemma

Lemma 5.3.1. *For $x \geq 0$, we have $1 - x \leq \exp(-x)$ and $1 + x \leq e^x$.*
Namely, for all x, we have $1 + x \leq e^x$.

Proof: For $x = 0$ we have equality. Next, computing the derivative on both sides, we have that we need to prove that $-\exp(-x) \geq -1$. Namely, that $1 \geq \exp(-x)$, which is trivially true, since $e^x \geq 1$, for $x \geq 0$. The second argument works for the second inequality. ■

Lemma 5.3.2. *For any $y \geq 1$, and $|x| \leq 1$, we have $(1 - x^2)^y \geq 1 - yx^2$.*

Proof: Observe that the inequality holds with equality for $x = 0$. So compute the derivative of x of both sides of the inequality. We need to prove that

$$y(-2x)\left(1 - x^2\right)^{y-1} \geq -2yx \quad \Leftrightarrow \quad \left(1 - x^2\right)^{y-1} \leq 1,$$

which holds since $1 - x^2 \leq 1$, and $y - 1 \geq 0$. ■

Lemma 5.3.3. *For any $y \geq 1$, and $|x| \leq 1$, we have $\left(1 - x^2 y\right) e^{xy} \leq (1 + x)^y \leq e^{xy}$.*

Proof: The right side of the inequality is standard by now. As for the left side. Observe that

$$(1 - x^2)e^x \leq 1 + x,$$

since dividing both sides by $(1 + x)e^x$, we get $1 - x \leq e^{-x}$, which we know holds for any x. By Lemma 5.3.2, we have

$$\left(1 - x^2 y\right) e^{xy} \leq \left(1 - x^2\right)^y e^{xy} = \left(\left(1 - x^2\right) e^x\right)^y \leq (1 + x)^y \leq e^{xy}.$$

■

Chapter 6

Sampling and other Stuff

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

6.1 Two-Point Sampling

Definition 6.1.1. A collection of random variables $X_1, \ldots, X_n$ is ***pairwise-independent***, if for any pair of variables X_i and X_j, and any pair of values α and β we have that $\mathbf{Pr}\Big[X_i = \alpha \cap X_j = \beta\Big] = \mathbf{Pr}[X_i = \alpha]\,\mathbf{Pr}\Big[X_j = \beta\Big]$.

Similarly, this collection is ***k-wise independent***, if for any $t \leq k$ variables $X_{i_1}, \ldots, X_{i_t}$ in this collection, and any set of t values, $\alpha_1, \ldots, \alpha_t'$ we have that

$$\mathbf{Pr}\Big[(X_{i_1} = \alpha_1) \cap \ldots \cap (X_{i_t} = \alpha_t)\Big] = \prod_{j=1}^{t} \mathbf{Pr}\Big[X_{i_j} = \alpha_j\Big].$$

Namely, pairwise independent variables behaves like independent random variables as long as you look only in pairs.

Example 6.1.2. Consider the following probability space, where the triple of variables X, Y, Z can be assigned any of the rows with equal probability (i.e., 1/4).

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

Clearly, for any $\alpha, \beta \in \{0, 1\}$ we have $\mathbf{Pr}[(X = \alpha) \cap (Y = \beta)] = \mathbf{Pr}[(X = \alpha)]\,\mathbf{Pr}[(Y = \beta)] = 1/4$ (this also holds for X, Z and Y, Z). Namely, X, Y, Z are all pairwise independent. However, they are not 3-wise independent (or just independent). Indeed, $\mathbf{Pr}[(X = 1) \cap (Y = 1) \cap (Z = 1)] = 0$, while it should have been $1/8$ if they were truely independent, or even just 3-wise independent.

6.1.1 About Modulo Rings and Pairwise Independence

Let p be a prime number, and let $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$ denote the ring of integers modules p. Two integers x and y are ***equivalent modulo*** p, if $x \equiv y \mod p$; namely, the reminder of dividing x and y by p is the same.

Lemma 6.1.3. *Given $y, i \in \mathbb{Z}_p$, and choosing a, b randomly and uniformly from $\mathbb{Z}_p$, the probability of $y \equiv ai + b \pmod p$ is $1/p$.*

Proof: Imagine that we first choose a, then the required probability, is that we choose b such that $y - ai \equiv b \pmod p$. And the probability for that is $1/p$, as we choose b uniformly. ∎

Lemma 6.1.4. *Let p be a prime, and fix $a \in \{1, \ldots, p-1\}$. Then,*

$$\left\{ ai \pmod p \;\middle|\; i = 0, \ldots, p-1 \right\} = \mathbb{Z}_p.$$

Putting it differently, for any non-zero $a \in \mathbb{Z}_p$, there is a unique inverse $b \in \mathbb{Z}_p$ such that $ab \pmod p = 1$.

Proof: Assume, for the sake of contradiction, that the claim is false. Then, by the pigeon hole principle, there must exists $1 \le j < i \le p-1$ such that $ai \pmod p = aj \pmod p$. Namely, there are k', k, u such that

$$ai = u + kp \quad \text{and} \quad aj = u + k'p.$$

(Here, we know that $0 \le k < p$, $0 \le k' < p$ and $0 \le u < p$.) Since $i > j$ it must be that $k > k'$. Subtracting the two equalities, we get that $a(i-j) = (k-k')p > 0$. Now, $i - j$ must be larger than one, since if $i - j = 1$ then $a = p$, which is impossible. Similarly, $i - j < p$. Also, $i - j$ can not divide p, since p is a prime. Thus, it must be that $i - j$ must divide $k - k'$. So, let us set $\beta = (k-k')/(i-j) \ge 1$. This implies that $a = \beta p \ge p$, which is impossible. Thus, our assumption is false. ∎

Lemma 6.1.5. *Given $y, z, x, w \in \mathbb{Z}_p$, such that $x \neq w$, and choosing a and b randomly and uniformly from $\mathbb{Z}_p$, the probability that $y \equiv ax + b \pmod p$ and $z = aw + b \pmod p$ is $1/p^2$.*

Proof: This equivalent to claiming that the system of equalities $y \equiv ax + b \pmod p$ and $z = aw + b$ have a unique solution in a and b.

To see why this is true, subtract one equation from the other. We get $y - z \equiv a(x - w) \pmod p$. Since $x - w \not\equiv 0 \pmod p$, it must be that there is a unique value of a such that the equation holds. This in turns, imply a specific value for b. The probability that a and b get those two specific values is $1/p^2$. ∎

Lemma 6.1.6. *Let i and j be two distinct elements of $\mathbb{Z}_p$. And choose a and b randomly and independently from $\mathbb{Z}_p$. Then, the two random variables $Y_i = ai+b \pmod p$ and $Y_j = aj+b \pmod p$ are uniformly distributed on $\mathbb{Z}_p$, and are pairwise independent.*

Proof: The claim about the uniform distribution follows from Lemma 6.1.3, as $\mathbf{Pr}[Y_i = \alpha] = 1/p$, for any $\alpha \in \mathbb{Z}_p$. As for being pairwise independent, observe that

$$\mathbf{Pr}\Big[Y_i = \alpha \;\Big|\; Y_j = \beta\Big] = \frac{\mathbf{Pr}\Big[Y_i = \alpha \cap Y_j = \beta\Big]}{\mathbf{Pr}\Big[Y_j = \beta\Big]} = \frac{1/n^2}{1/n} = \frac{1}{n} = \mathbf{Pr}[Y_i = \alpha],$$

by Lemma 6.1.3 and Lemma 6.1.5. Thus, Y_i and Y_j are pairwise independent. ∎

Remark 6.1.7. It is important to understand what independence between random variables mean: It means that having information about the value of X, gives you no information about Y. But this is only pairwise independence. Indeed, consider the variables Y_1, Y_2, Y_3, Y_4 defined above. Every pair of them are pairwise independent. But, if you give the value of Y_1 and Y_2, I know the value of Y_3 and Y_4 immediately. Indeed, giving me the value of Y_1 and Y_2 is enough to figure out the value of a and b. Once we know a and b, we immediately can compute all the Y_is.

Thus, the notion of independence can be extended to k-pairwise independence of n random variables, where only if you know the value of k variables, you can compute the value of all the other variables. More on that later in the course.

Lemma 6.1.8. *If X and Y are pairwise independent then $\mathbf{E}[XY] = \mathbf{E}[X]\,\mathbf{E}[Y]$.*

Proof: By definition $\mathbf{E}[XY] = \sum_{x,y} xy\,\mathbf{Pr}[(X = x) \cap (Y = y)] = \sum_{x,y} xy\,\mathbf{Pr}[X = x]\,\mathbf{Pr}[Y = y] = \sum_x x\,\mathbf{Pr}[X = x] \sum_y y\,\mathbf{Pr}[Y = y] = \Big(\sum_x x\,\mathbf{Pr}[X = x]\Big)\Big(\sum_y y\,\mathbf{Pr}[Y = y]\Big) = \mathbf{E}[X]\,\mathbf{E}[Y]$. ■

Lemma 6.1.9. *Let $X_1, X_2, \ldots, X_n$ be pairwise independent random variables, and $X = \sum_{i=1}^n X_i$. Then $\mathbf{V}[X] = \sum_{i=1}^n \mathbf{V}[X_i]$.*

Proof: Observe, that

$$\mathbf{V}[X] = \mathbf{E}\Big[(X - \mathbf{E}[X])^2\Big] = \mathbf{E}\Big[X^2\Big] - (\mathbf{E}[X])^2 .$$

Let X and Y be pairwise independent variables. Observe that $\mathbf{E}[XY] = \mathbf{E}[X]\,\mathbf{E}[Y]$, as can be easily verified. Thus,

$$\begin{aligned}
\mathbf{V}[X + Y] &= \mathbf{E}\Big[(X + Y - \mathbf{E}[X] - \mathbf{E}[Y])^2\Big] \\
&= \mathbf{E}\Big[(X + Y)^2 - 2(X + Y)(\mathbf{E}[X] + \mathbf{E}[Y]) + (\mathbf{E}[X] + \mathbf{E}[Y])^2\Big] \\
&= \mathbf{E}\Big[(X + Y)^2\Big] - (\mathbf{E}[X] + \mathbf{E}[Y])^2 \\
&= \mathbf{E}\Big[X^2 + 2XY + Y^2\Big] - (\mathbf{E}[X])^2 - 2\,\mathbf{E}[X]\,\mathbf{E}[Y] - (\mathbf{E}[Y])^2 \\
&= \Big(\mathbf{E}\Big[X^2\Big] - (\mathbf{E}[X])^2\Big) + \Big(\mathbf{E}\Big[Y^2\Big] - (\mathbf{E}[Y])^2\Big) + 2\,\mathbf{E}[XY] - 2\,\mathbf{E}[X]\,\mathbf{E}[Y] \\
&= \mathbf{V}[X] + \mathbf{V}[Y] + +2\,\mathbf{E}[X]\,\mathbf{E}[Y] - 2\,\mathbf{E}[X]\,\mathbf{E}[Y] \\
&= \mathbf{V}[X] + \mathbf{V}[Y] ,
\end{aligned}$$

by Lemma 6.1.8. Using the above argumentation for several variables, instead of just two, implies the lemma. ■

6.1.2 Using less randomization for a randomized algorithm

We can consider a randomized algorithm, to be a deterministic algorithm $A(x, r)$ that receives together with the input x, a random string r of bits, that it uses to read random bits from. Let us redefine **RP**:

Definition 6.1.10. The class **RP** (for Randomized Polynomial time) consists of all languages L that have a deterministic algorithm $A(x, r)$ with worst case polynomial running time such that for any input $x \in \Sigma^*$,

- $x \in L \Rightarrow A(x, r) = 1$ for half the possible values of r.
- $x \notin L \Rightarrow A(x, r) = 0$ for all values of r.

Let assume that we now want to minimize the number of random bits we use in the execution of the algorithm (Why?). If we run the algorithm t times, we have confidence 2^{-t} in our result, while using $t \log n$ random bits (assuming our random algorithm needs only $\log n$ bits in each execution). Similarly, let us choose two random numbers from $\mathbb{Z}_n$, and run $A(x, a)$ and $A(x, b)$, gaining us only confidence $1/4$ in the correctness of our results, while requiring $2 \log n$ bits.

Can we do better? Let us define $r_i = ai + b \mod n$, where a, b are random values as above (note, that we assume that n is prime), for $i = 1, \ldots, t$. Thus $Y = \sum_{i=1}^{t} A(x, r_i)$ is a sum of random variables which are pairwise independent, as the r_i are pairwise independent. Assume, that $x \in L$, then we have $\mathbf{E}[Y] = t/2$, and $\sigma_Y^2 = \mathbf{V}[Y] = \sum_{i=1}^{t} \mathbf{V}[A(x, r_i)] \leq t/4$, and $\sigma_Y \leq \sqrt{t}/2$. The probability that all those executions failed, corresponds to the event that $Y = 0$, and

$$\mathbf{Pr}[Y = 0] \leq \mathbf{Pr}\left[|Y - \mathbf{E}[Y]| \geq \frac{t}{2}\right] = \mathbf{Pr}\left[|Y - \mathbf{E}[Y]| \geq \frac{\sqrt{t}}{2} \cdot \sqrt{t}\right] \leq \frac{1}{t},$$

by the Chebyshev inequality. Thus we were able to "extract" from our random bits, much more than one would naturally suspect is possible.

6.2 Chernoff Inequality - A Special Case

Theorem 6.2.1. *Let $X_1, \ldots, X_n$ be n independent random variables, such that $\mathbf{Pr}[X_i = 1] = \mathbf{Pr}[X_i = -1] = \frac{1}{2}$, for $i = 1, \ldots, n$. Let $Y = \sum_{i=1}^{n} X_i$. Then, for any $\Delta > 0$, we have*

$$\mathbf{Pr}[Y \geq \Delta] \leq e^{-\Delta^2/2n}.$$

Proof: Clearly, for an arbitrary t, to specified shortly, we have

$$\mathbf{Pr}[Y \geq \Delta] = \mathbf{Pr}[\exp(tY) \geq \exp(t\Delta)] \leq \frac{\mathbf{E}[\exp(tY)]}{\exp(t\Delta)},$$

the first part follows by the fact that $\exp(\cdot)$ preserve ordering, and the second part follows by the Markov inequality.

Observe that

$$\begin{aligned}
\mathbf{E}[\exp(tX_i)] &= \frac{1}{2}e^{t} + \frac{1}{2}e^{-t} = \frac{e^{t} + e^{-t}}{2} \\
&= \frac{1}{2}\left(1 + \frac{t}{1!} + \frac{t^2}{2!} + \frac{t^3}{3!} + \cdots\right) \\
&+ \frac{1}{2}\left(1 - \frac{t}{1!} + \frac{t^2}{2!} - \frac{t^3}{3!} + \cdots\right) \\
&= \left(1 + \quad + \frac{t^2}{2!} + \quad + \cdots + \frac{t^{2k}}{(2k)!} + \cdots\right),
\end{aligned}$$

by the Taylor expansion of $\exp(\cdot)$. Note, that $(2k)! \geq (k!)2^k$, and thus

$$\mathbf{E}[\exp(tX_i)] = \sum_{i=0}^{\infty} \frac{t^{2i}}{(2i)!} \leq \sum_{i=0}^{\infty} \frac{t^{2i}}{2^i(i!)} = \sum_{i=0}^{\infty} \frac{1}{i!}\left(\frac{t^2}{2}\right)^i = \exp\left(t^2/2\right),$$

again, by the Taylor expansion of $\exp(\cdot)$. Next, by the independence of the X_is, we have

$$\begin{aligned}\mathbf{E}[\exp(tY)] &= \mathbf{E}\left[\exp\left(\sum_i tX_i\right)\right] = \mathbf{E}\left[\prod_i \exp(tX_i)\right] = \prod_{i=1}^{n} \mathbf{E}[\exp(tX_i)] \\ &\leq \prod_{i=1}^{n} e^{t^2/2} = e^{nt^2/2}.\end{aligned}$$

We have

$$\mathbf{Pr}[Y \geq \Delta] \leq \frac{\exp\left(nt^2/2\right)}{\exp(t\Delta)} = \exp\left(nt^2/2 - t\Delta\right).$$

Next, by minimizing the above quantity for t, we set $t = \Delta/n$. We conclude,

$$\mathbf{Pr}[Y \geq \Delta] \leq \exp\left(\frac{n}{2}\left(\frac{\Delta}{n}\right)^2 - \frac{\Delta}{n}\Delta\right) = \exp\left(-\frac{\Delta^2}{2n}\right).$$

■

By the symmetry of Y, we get the following:

Corollary 6.2.2. *Let $X_1, \ldots, X_n$ be n independent random variables, such that $\mathbf{Pr}[X_i = 1] = \mathbf{Pr}[X_i = -1] = \frac{1}{2}$, for $i = 1, \ldots, n$. Let $Y = \sum_{i=1}^{n} X_i$. Then, for any $\Delta > 0$, we have*

$$\mathbf{Pr}[|Y| \geq \Delta] \leq 2e^{-\Delta^2/2n}.$$

Corollary 6.2.3. *Let $X_1, \ldots, X_n$ be n independent* coin flips*, such that $\mathbf{Pr}[X_i = 0] = \mathbf{Pr}[X_i = 1] = \frac{1}{2}$, for $i = 1, \ldots, n$. Let $Y = \sum_{i=1}^{n} X_i$. Then, for any $\Delta > 0$, we have*

$$\mathbf{Pr}\left[\left|Y - \frac{n}{2}\right| \geq \Delta\right] \leq 2e^{-2\Delta^2/n}.$$

Remark 6.2.4. Before going any further, it is might be instrumental to understand what this inequalities imply. Consider then case where X_i is either zero or one with probability half. In this case $\mu = \mathbf{E}[Y] = n/2$. Set $\delta = t\sqrt{n}$ ($\sqrt{\mu}$ is approximately the standard deviation of X if $p_i = 1/2$). We have by

$$\mathbf{Pr}\left[\left|Y - \frac{n}{2}\right| \geq \Delta\right] \leq 2\exp\left(-2\Delta^2/n\right) = 2\exp\left(-2(t\sqrt{n})^2/n\right) = 2\exp\left(-2t^2\right).$$

Thus, Chernoff inequality implies exponential decay (i.e., $\leq 2^{-t}$) with t standard deviations, instead of just polynomial (i.e., $\leq 1/t^2$) by the Chebychev's inequality.

6.2.1 Application – QuickSort is Quick

We revisit **QuickSort**. We remind the reader that the running time of **QuickSort** is proportional to the number of comparisons performed by the algorithm. Next, consider an arbitrary element u being sorted. Consider the ith level recursive subproblem that contains u, and let S_i be the set of elements in this subproblems. We consider u to be *successful* in the ith level, if $|S_{i+1}| \leq |S_i|/2$. Namely, if u is successful, then the next level in the recursion involving u would include a considerably smaller subproblem. Let X_i be the indicator variable which is 1 if u is successful.

We first observe that if **QuickSort** is applied to an array with n elements, then u can be successful at most $T = \lceil \lg n \rceil$ times, before the subproblem it participates in is of size one, and the recursion stops. Thus, consider the indicator variable X_i which is 1 if u is successful in the ith level, and zero otherwise. Note that the X_is are independent, and $\mathbf{Pr}[X_i = 1] = 1/2$.

If u participates in v levels, then we have the random variables $X_1, X_2, \ldots, X_v$. To make things simpler, we will extend this series by adding independent random variables, such that $\mathbf{Pr}[`] X_i = 1 = 1/2$, for $i \geq v$. Thus, we have an infinite sequence of independent random variables, that are 0/1 and get 1 with probability $1/2$. The question is how many elements in the sequence we need to read, till we get T ones.

Lemma 6.2.5. *Let $X_1, X_2, \ldots$ be an infinite sequence of independent random 0/1 variables. Let M be an arbitrary parameter. Then the probability that we need to read more than $2M + 4t\sqrt{M}$ variables of this sequence till we collect M ones is at most $2\exp\left(-t^2\right)$, for $t \leq \sqrt{M}$. If $t \geq \sqrt{M}$ then this probability is at most $2\exp\left(-t\sqrt{M}\right)$.*

Proof: Consider the random variable $Y = \sum_{i=1}^{L} X_i$, where $L = 2M + 4t\sqrt{M}$. Its expectation is $L/2$, and using the Chernoff inequality, we get

$$\begin{aligned}
\alpha &= \mathbf{Pr}[Y \leq M] \leq \mathbf{Pr}\left[\left|Y - \frac{L}{2}\right| \geq \frac{L}{2} - M\right] \leq 2\exp\left(-\frac{2}{L}\left(\frac{L}{2} - M\right)^2\right) \\
&\leq 2\exp\left(-\frac{2}{L}\left(M + 2t\sqrt{M} - M\right)^2\right) \leq 2\exp\left(-\frac{2}{L}\left(2t\sqrt{M}\right)^2\right) = 2\exp\left(-\frac{8t^2M}{L}\right),
\end{aligned}$$

by Corollary 6.2.3. For $t \leq \sqrt{M}$ we have that $L = 2M + 4t\sqrt{M} \leq 8M$, as such in this case $\mathbf{Pr}[Y \leq M] \leq 2\exp\left(-t^2\right)$.

If $t \geq \sqrt{M}$, then $\alpha = 2\exp\left(-\frac{8t^2M}{2M + 4t\sqrt{M}}\right) \leq 2\exp\left(-\frac{8t^2M}{6t\sqrt{M}}\right) \leq 2\exp\left(-t\sqrt{M}\right)$. ∎

Going back to the **QuickSort** problem, we have that if we sort n elements, the probability that u will participate in more than $L = (4 + c)\lceil \lg n \rceil = 2\lceil \lg n \rceil + 4c\sqrt{\lg n}\sqrt{\lg n}$, is smaller than $2\exp\left(-c\sqrt{\lg n}\sqrt{\lg n}\right) \leq 1/n^c$, by Lemma 6.2.5. There are n elements being sorted, and as such the probability that any element would participate in more than $(4 + c + 1)\lceil \lg n \rceil$ recursive calls is smaller than $1/n^c$.

Lemma 6.2.6. *For any $c > 0$, the probability that **QuickSort** performs more than $(6 + c)n \lg n$, is smaller than $1/n^c$.*

Chapter 7

Chernoff Inequality - Part II

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

7.1 Tail Inequalities

7.1.1 The Chernoff Bound — General Case

Here we present the Chernoff bound in a more general settings.

Question 7.1.1. *Let $X_1, \ldots, X_n$ be n independent Bernoulli trials, where*

$$\mathbf{Pr}[X_i = 1] = p_i, \text{ and } \mathbf{Pr}[X_i = 0] = q_i = 1 - p_i.$$

(Each X_i is known as a Poisson trials.) And let $X = \sum_{i=1}^{b} X_i$. $\mu = \mathbf{E}[X] = \sum_i p_i$. We are interested in the question of what is the probability that $X > (1+\delta)\mu$?

Theorem 7.1.2. *For any $\delta > 0$, we have $\mathbf{Pr}[X > (1+\delta)\mu] < \left(\dfrac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}$.*

Or in a more simplified form, for any $\delta \leq 2e - 1$,

$$\mathbf{Pr}[X > (1+\delta)\mu] < \exp\left(-\mu\delta^2/4\right), \tag{7.1}$$

and

$$\mathbf{Pr}[X > (1+\delta)\mu] < 2^{-\mu(1+\delta)}, \tag{7.2}$$

for $\delta \geq 2e - 1$.

Proof: We have $\mathbf{Pr}[X > (1+\delta)\mu] = \mathbf{Pr}\left[e^{tX} > e^{t(1+\delta)\mu}\right]$. By the Markov inequality, we have:

$$\mathbf{Pr}[X > (1+\delta)\mu] < \frac{\mathbf{E}\left[e^{tX}\right]}{e^{t(1+\delta)\mu}}$$

On the other hand,

$$\mathbf{E}\left[e^{tX}\right] = \mathbf{E}\left[e^{t(X_1+X_2\ldots+X_n)}\right] = \mathbf{E}\left[e^{tX_1}\right]\cdots\mathbf{E}\left[e^{tX_n}\right].$$

Namely,

$$\mathbf{Pr}[X > (1+\delta)\mu] < \frac{\prod_{i=1}^{n} \mathbf{E}\left[e^{tX_i}\right]}{e^{t(1+\delta)\mu}} = \frac{\prod_{i=1}^{n}\left((1-p_i)e^0 + p_i e^t\right)}{e^{t(1+\delta)\mu}} = \frac{\prod_{i=1}^{n}(1+p_i(e^t-1))}{e^{t(1+\delta)\mu}}.$$

Let $y = p_i(e^t - 1)$. We know that $1 + y < e^y$ (since $y > 0$). Thus,

$$\begin{aligned}
\mathbf{Pr}[X > (1+\delta)\mu] &< \frac{\prod_{i=1}^{n} \exp(p_i(e^t-1))}{e^{t(1+\delta)\mu}}. = \frac{\exp(\sum_{i=1}^{n} p_i(e^t-1))}{e^{t(1+\delta)\mu}} \\
&= \frac{\exp((e^t-1)\sum_{i=1}^{n} p_i)}{e^{t(1+\delta)\mu}} = \frac{\exp((e^t-1)\mu)}{e^{t(1+\delta)\mu}} = \left(\frac{\exp(e^t-1)}{e^{t(1+\delta)}}\right)^{\mu} \\
&= \left(\frac{\exp(\delta)}{(1+\delta)^{(1+\delta)}}\right)^{\mu},
\end{aligned}$$

if we set $t = \log(1+\delta)$.

For the proof of the simplified form, see Section 7.1.2. ∎

Definition 7.1.3. $F^+(\mu,\delta) = \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{\mu}$.

Example 7.1.4. Arkansas Aardvarks win a game with probability $1/3$. What is their probability to have a winning season with n games. By Chernoff inequality, this probability is smaller than

$$F^+(n/3, 1/2) = \left[\frac{e^{1/2}}{1.5^{1.5}}\right]^{n/3} = (0.89745)^{n/3} = 0.964577^n.$$

For $n = 40$, this probability is smaller than 0.236307. For $n = 100$ this is less than 0.027145. For $n = 1000$, this is smaller than $2.17221 \cdot 10^{-16}$ (which is pretty slim and shady). Namely, as the number of experiments is increases, the distribution converges to its expectation, and this converge is exponential.

Theorem 7.1.5. *Under the same assumptions as Theorem 7.1.2, we have:*

$$\mathbf{Pr}[X < (1-\delta)\mu] < e^{-\mu\delta^2/2}.$$

Definition 7.1.6. $F^-(\mu,\delta) = e^{-\mu\delta^2/2}$.

Let $\Delta^-(\mu,\varepsilon)$ denote the quantity, which is what should be the value of δ, so that the probability is smaller than ε. We have that

$$\Delta^-(\mu,\varepsilon) = \sqrt{\frac{2\log 1/\varepsilon}{\mu}}.$$

And for large δ

$$\Delta^+(\mu,\varepsilon) < \frac{\log_2(1/\varepsilon)}{\mu} - 1.$$

Values	Probabilities	Inequality	Ref
$-1, +1$	$\mathbf{Pr}[X_i = -1] =$ $\mathbf{Pr}[X_i = 1] = \frac{1}{2}$	$\mathbf{Pr}[Y \geq \Delta] \leq e^{-\Delta^2/2n}$	Theorem 6.2.1
		$\mathbf{Pr}[Y \leq -\Delta] \leq e^{-\Delta^2/2n}$	Theorem 6.2.1
		$\mathbf{Pr}[\lvert Y\rvert \geq \Delta] \leq 2e^{-\Delta^2/2n}$	Corollary 6.2.2
$0, 1$	$\mathbf{Pr}[X_i = 0] =$ $\mathbf{Pr}[X_i = 1] = \frac{1}{2}$	$\mathbf{Pr}\left[\left\lvert Y - \frac{n}{2}\right\rvert \geq \Delta\right] \leq 2e^{-2\Delta^2/n}$	Corollary 6.2.3
0,1	$\mathbf{Pr}[X_i = 0] = 1 - p_i$ $\mathbf{Pr}[X_i = 1] = p_i$	$\mathbf{Pr}[Y > (1+\delta)\mu] < \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu$	Theorem 7.1.2
	For $\delta \leq 2e - 1$	$\mathbf{Pr}[Y > (1+\delta)\mu] < \exp\left(-\mu\delta^2/4\right)$	Theorem 7.1.2
	$\delta \geq 2e - 1$	$\mathbf{Pr}[Y > (1+\delta)\mu] < 2^{-\mu(1+\delta)}$	
	For $\delta \geq 0$	$\mathbf{Pr}[Y < (1-\delta)\mu] < \exp\left(-\mu\delta^2/2\right)$	Theorem 7.1.5

Table 7.1: Summary of Chernoff type inequalities covered. Here we have n variables $X_1, \ldots, X_n$, $Y = \sum_i X_i$ and $\mu = \mathbf{E}[Y]$.

7.1.2 A More Convenient Form

Proof: **(of simplified form of Theorem 7.1.2)** Eq. (7.2) is easy. Indeed, we have

$$\left[\frac{e}{1+\delta}\right]^{(1+\delta)\mu} \leq \left[\frac{e}{1+2e-1}\right]^{(1+\delta)\mu} \leq 2^{-(1+\delta)\mu},$$

since $\delta > 2e - 1$.

As for Eq. (7.1), we prove this only for $\delta \leq 1/2$. For details about the case $1/2 \leq \delta \leq 2e - 1$, see [MR95]. By Theorem 7.1.2, we have

$$\mathbf{Pr}[X > (1+\delta)\mu] < \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu = \exp(\mu\delta - \mu(1+\delta)\ln(1+\delta)).$$

The Taylor expansion of $\ln(1+\delta)$ is

$$\delta - \frac{\delta^2}{2} + \frac{\delta^3}{3} - \frac{\delta^4}{4} + \cdot \geq \delta - \frac{\delta^2}{2},$$

for $\delta \leq 1$. Thus,

$$\begin{aligned}\mathbf{Pr}[X > (1+\delta)\mu] &< \exp\left(\mu\left(\delta - (1+\delta)\left(\delta - \delta^2/2\right)\right)\right) = \exp\left(\mu\left(\delta - \delta + \delta^2/2 - \delta^2 + \delta^3/2\right)\right) \\ &\leq \exp\left(\mu\left(-\delta^2/2 + \delta^3/2\right)\right) \leq \exp\left(-\mu\delta^2/4\right),\end{aligned}$$

for $\delta \leq 1/2$. ∎

7.2 Application: Routing in a Parallel Computer

Let G be a graph of a network, where every node is a processor. The processor communicate by sending packets on the edges. Let $[0, \ldots, N]$ denote be vertices (i.e., processors) of G, where

(i) Pick a *random* intermediate destination $\sigma(i)$ from $[1, \ldots, N]$. Packet v_i travels to $\sigma(i)$.

(ii) Wait till all the packets arrive to their intermediate destination.

(iii) Packet v_i travels from $\sigma(i)$ to its destination $d(i)$.

Figure 7.1: The routing algorithm

$N = 2^n$, and G is the hypercube. As such, each processes is identified with a binary string $b_1 b_2 \ldots b_n$. Two nodes are connected if their binary string differs only in a single bit. Namely, G is the hypercube over n vertices.

We want to investigate the best routing strategy for this topology of network. We assume that every processor need to send a message to a single other processor. This is representation by a permutation π, and we would like to figure out how to send the permutation and create minimum delay?

In our model, every edge has a FIFO queue of the packets it has to transmit. At every clock tick, one message get sent. All the processors start sending the packets in their permutation in the same time.

Theorem 7.2.1. *For any deterministic oblivious permutation routing algorithm on a network of N nodes each of out-degree n, there is a permutation for which the routing of the permutation takes $\Omega\left(\sqrt{N/n}\right)$ time.*

Oblivious here refers to the fact that the routing of packet is determined only by inspecting only the packet, and without referring to other things in the network.

How do we sent a packet? We use ***bit fixing***. Namely, the packet from the i node, always go to the current adjacent node that have the first different bit as we scan the destination string $d(i)$. For example, packet from (0000) going to (1101), would pass through (1000), (1100), (1101).

We assume each edge have a FIFO queue. The routing algorithm is depicted in Figure 7.1.

We analyze only (i) as (iii) follows from the same analysis. In the following, let ρ_i denote the route taken by v_i in (i).

Exercise 7.2.2. Once a packet v_j that travel along a path ρ_j can not leave a path ρ_i, and then join it again later. Namely, $\rho_i \cap \rho_j$ is (maybe an empty) path.

Lemma 7.2.3. *Let the route of a message* **c** *follow the sequence of edges $\pi = (e_1, e_2, \ldots, e_k)$. Let S be the set of packets whose routes pass through at least one of $(e_1, \ldots, e_k)$. Then, the delay incurred by* **c** *is at most $|S|$.*

Proof: A packet in S is said to leave π at that time step at which it traverses an edge of π for the last time. If a packet is ready to follow edge e_j at time t, we define its *lag* at time t to be $t - j$. The lag of **c** is initially zero, and the delay incurred by **c** is its lag when it traverse e_k. We will show that each step at which the lag of **c** increases by one can be charged to a distinct member of S.

We argue that if the lag of **c** reaches $\ell + 1$, some packet in S leaves π with lag ℓ. When the lag of **c** increases from ℓ to $\ell + 1$, there must be at least one packet (from S) that wishes to traverse the

same edge as **c** at that time step, since otherwise **c** would be permitted to traverse this edge and its lag would not increase. Thus, S contains at least one packet whose lag reach the value ℓ.

Let τ be the last time step at which any packet in S has lag ℓ. Thus there is a packet **d** ready to follow edge e_μ at τ, such that $\tau - \mu = \ell$. We argue that some packet of S leaves π at τ; this establishes the lemma since once a packet leaves π, it would never join it again and as such will never again delay **c**.

Since **d** is ready to follow e_μ at τ, some packet ω (which may be **d** itself) in S follows e_μ at time τ. Now ω leaves π at time τ; if not, some packet will follow $e_{\mu+1}$ at step $\mu + 1$ with lag still at ℓ, violating the maximality of τ. We charge to ω the increase in the lag of **c** from ℓ to $\ell + 1$; since ω leaves π, it will never be charged again. Thus, each member of S whose route intersects π is charge for at most one delay, establishing the lemma. ∎

Let H_{ij} be an indicator variable that is 1 if ρ_i and ρ_j share an edge, and 0 otherwise. The total delay for v_i is at most $\leq \sum_j H_{ij}$. Note, that for a fixed i, the variables $H_{i1}, \ldots, H_{iN}$ are independent (note however, that $H_{11}, \ldots, H_{NN}$ are not independent!). For $\rho_i = (e_1, \ldots, e_k)$, let $T(e)$ be the number of packets (i.e., paths) that pass through e.

$$\sum_{j=1}^{N} H_{ij} \leq \sum_{j=1}^{k} T(e_j) \text{ and thus } \mathbf{E}\left[\sum_{j=1}^{N} H_{ij}\right] \leq \mathbf{E}\left[\sum_{j=1}^{k} T(e_j)\right].$$

Because of symmetry, the variables $T(e)$ have the same distribution for all the edges of G. On the other hand, the expected length of a path is $n/2$, there are N packets, and there are $Nn/2$ edges. We conclude $E[T(e)] = 1$. Thus

$$\mu = \mathbf{E}\left[\sum_{j=1}^{N} H_{ij}\right] \leq \mathbf{E}\left[\sum_{j=1}^{k} T(e_j)\right] = \mathbf{E}[|\rho_i|] \leq \frac{n}{2}.$$

By the Chernoff inequality, we have

$$\mathbf{Pr}\left[\sum_{j} H_{ij} > 7n\right] \leq \mathbf{Pr}\left[\sum_{j} H_{ij} > (1 + 13)\mu\right] < 2^{-13\mu} \leq 2^{-6n}.$$

Since there are $N = 2^n$ packets, we know that with probability $\leq 2^{-5n}$ all packets arrive to their temporary destination in a delay of most $7n$.

Theorem 7.2.4. *Each packet arrives to its destination in $\leq 14n$ stages, in probability at least $1 - 1/N$ (note that this is very conservative).*

7.3 Application of the Chernoff Inequality – Faraway Strings

Consider the Hamming distance between binary strings. It is natural to ask how many strings of length n can one have, such that any pair of them, is of Hamming distance at least t from each other. Consider two random strings, generated by picking at each bit randomly and independently. Thus,

$\mathbf{E}[d_H(x,y)] = n/2$, where $d_H(x,y)$ denote the hamming distance between x and y. In particular, using the Chernoff inequality, we have that

$$\mathbf{Pr}[d_H(x,y) \leq n/2 - \Delta] \leq \exp\left(-2\Delta^2/n\right).$$

Next, consider generating M such string, where the value of M would be determined shortly. Clearly, the probability that any pair of strings are at distance at most $n/2 - \Delta$, is

$$\alpha \leq \binom{M}{2} \exp\left(-2\Delta^2/n\right) < M^2 \exp\left(-2\Delta^2/n\right).$$

If this probability is smaller than one, then there is some probability that all the M strings are of distance at least $n/2 - \Delta$ from each other. Namely, there exists a set of M strings such that every pair of them is far. We used here the fact that if an event has probability larger than zero, then it exists. Thus, set $\Delta = n/4$, and observe that

$$\alpha < M^2 \exp\left(-2n^2/16n\right) = M^2 \exp(-n/8).$$

Thus, for $M = \exp(n/16)$, we have that $\alpha < 1$. We conclude:

Lemma 7.3.1. *There exists a set of* $\exp(n/16)$ *binary strings of length n, such that any pair of them is at Hamming distance at least* $n/4$ *from each other.*

This is our first introduction to the beautiful technique known as the probabilistic method — we will hear more about it later in the course.

This result has also interesting interpretation in the Euclidean setting. Indeed, consider the sphere $\mathbb{S}$ of radius $\sqrt{n}/2$ centered at $(1/2, 1/2, \ldots, 1/2) \in \mathbb{R}^n$. Clearly, all the vertices of the binary hypercube $\{0,1\}^n$ lie on this sphere. As such, let P be the set of points on $\mathbb{S}$ that exists according to Lemma 7.3.1. A pair p, q of points of P have *Euclidean* distance at least $\sqrt{d_H(p,q)} = \sqrt{n4} = \sqrt{n}/2$ from each other. We conclude:

Lemma 7.3.2. *Consider the unit hypersphere* $\mathbb{S}$ *in* $\mathbb{R}^n$*. The sphere* $\mathbb{S}$ *contains a set* Q *of points, such that each pair of points is at (Euclidean) distance at least one from each other, and* $|Q| \geq \exp(n/16)$.

7.4 Bibliographical notes

The exposition here follows more or less the exposition in [MR95]. Exercise 7.5.1 (without the hint) is from [Mat99].

Section 7.2 is based on Section 4.2 in [MR95]. A similar result to Theorem 7.2.4 is known for the case of the wrapped butterfly topology (which is similar to the hypercube topology but every node has a constant degree, and there is no clear symmetry). The interested reader is referred to [MU05].

7.5 Exercises

Exercise 7.5.1. [**10 points**] Let $S = \sum_{i=1}^{n} S_i$ be a sum of n independent random variables each attaining values $+1$ and -1 with equal probability. Let $P(n, \Delta) = \mathbf{Pr}[S > \Delta]$. Prove that for $\Delta \leq n/C$,

$$P(n, \Delta) \geq \frac{1}{C} \exp\left(-\frac{\Delta^2}{Cn}\right),$$

where C is a suitable constant. That is, the well-known Chernoff bound $P(n, \Delta) \leq \exp(-\Delta^2/2n))$ is close to the truth.

[**Hint:** Use Stirling's formula. There is also an elementary solution, using estimates for the middle binomial coefficients [MN98, pages 83–84], but this solution is considerably more involved and yields unfriendly constants.**]**

Exercise 7.5.2. To some extent, Lemma 7.3.1 is somewhat silly, as one can prove a better bound by direct argumentation. Indeed, for a fixed binary string x of length n, show a bound on the number of strings in the Hamming ball around x of radius $n/4$ (i.e., binary strings of distance at most $n/4$ from x). (Hint: interpret the special case of the Chernoff inequality as an inequality over binomial coefficients.)

Next, argue that the greedy algorithm which repeatedly pick a string which is in distance $\geq n/4$ from all strings picked so far, stops after picking at least $\exp(n/8)$ strings.

Chapter 8

Martingales

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

'After that he always chose out a "dog command" and sent them ahead. It had the task of informing the inhabitants in the village where we were going to stay overnight that no dog must be allowed to bark in the night otherwise it would be liquidated. I was also on one of those commands and when we came to a village in the region of Milevsko I got mixed up and told the mayor that every dog-owner whose dog barked in the night would be liquidated for strategic reasons. The mayor got frightened, immediately harnessed his horses and rode to headquarters to beg mercy for the whole village. They didn't let him in, the sentries nearly shot him and so he returned home, but before we got to the village everybody on his advice had tied rags round the dogs muzzles with the result that three of them went mad.'

– The good soldier Svejk, Jaroslav Hasek

8.1 Martingales

8.1.1 Preliminaries

Let X and Y be two random variables. Let $\rho(x,y) = \mathbf{Pr}[(X = x) \cap (Y = y)]$. Then,

$$\mathbf{Pr}\Big[X = x \;\Big|\; Y = y\Big] = \frac{\rho(x,y)}{\mathbf{Pr}[Y = y]} = \frac{\rho(x,y)}{\sum_z \rho(z,y)}$$

and

$$\mathbf{E}\Big[X \;\Big|\; Y = y\Big] = \sum_x x \,\mathbf{Pr}\Big[X = x \;\Big|\; Y = y\Big] = \frac{\sum_x x\rho(x,y)}{\sum_z \rho(z,y)} = \frac{\sum_x x\rho(x,y)}{\mathbf{Pr}[Y = y]}.$$

Definition 8.1.1. The random variable $\mathbf{E}\Big[X \;\Big|\; Y\Big]$ is the random variable $f(y) = \mathbf{E}\Big[X \;\Big|\; Y = y\Big]$.

Lemma 8.1.2. $\mathbf{E}\Big[\mathbf{E}\Big[X \;\Big|\; Y\Big]\Big] = \mathbf{E}[X]$.

Proof:

$$\begin{aligned}
\mathbf{E}\Big[\mathbf{E}\Big[X \;\Big|\; Y\Big]\Big] &= \mathop{\mathbf{E}}_{Y}\Big[\mathbf{E}\Big[X \;\Big|\; Y = y\Big]\Big] = \sum_y \mathbf{Pr}[Y = y]\,\mathbf{E}\Big[X \;\Big|\; Y = y\Big] \\
&= \sum_y \mathbf{Pr}[Y = y]\,\frac{\sum_x x\,\mathbf{Pr}[X = x \cap Y = y]}{\mathbf{Pr}[Y = y]} \\
&= \sum_y \sum_x x\,\mathbf{Pr}[X = x \cap Y = y] = \sum_x x \sum_y \mathbf{Pr}[X = x \cap Y = y] \\
&= \sum_x x\,\mathbf{Pr}[X = x] = \mathbf{E}[X]\,.
\end{aligned}$$

■

Lemma 8.1.3. $\mathbf{E}\Big[Y \cdot \mathbf{E}\Big[X \;\Big|\; Y\Big]\Big] = \mathbf{E}[XY]$.

Proof:

$$\begin{aligned}
\mathbf{E}\Big[Y \cdot \mathbf{E}\Big[X \;\Big|\; Y\Big]\Big] &= \sum_y \mathbf{Pr}[Y = y] \cdot y \cdot \mathbf{E}\Big[X \;\Big|\; Y = y\Big] \\
&= \sum_y \mathbf{Pr}[Y = y] \cdot y \cdot \frac{\sum_x x\,\mathbf{Pr}[X = x \cap Y = y]}{\mathbf{Pr}[Y = y]} \\
&= \sum_x \sum_y xy \cdot \mathbf{Pr}[X = x \cap Y = y] = \mathbf{E}[XY]\,.
\end{aligned}$$

■

8.1.2 Martingales

Definition 8.1.4. A sequence of random variables $X_0, X_1, \ldots$, is said to be a ***martingale sequence*** if for all $i > 0$, we have $\mathbf{E}\Big[X_i \;\Big|\; X_0, \ldots, X_{i-1}\Big] = X_{i-1}$.

Lemma 8.1.5. *Let $X_0, X_1, \ldots$, be a martingale sequence. Then, for all $i \geq 0$, we have* $\mathbf{E}[X_i] = \mathbf{E}[X_0]$.

Example 8.1.6. An example of martingales is the sum of money after participating in a sequence of fair bets. That is, let X_i be the amount of money a gambler has after playing i rounds. In each round it either gains one dollar, or loses one dollar. Clearly, we have $\mathbf{E}\Big[X_i \;\Big|\; X_0, \ldots, X_{i-1}\Big] = \mathbf{E}\Big[X_i \;\Big|\; X_{i-1}\Big] = X_i$.

Example 8.1.7. Let $Y_i = X_i^2 - i$, where X_i is as defined in the above example. We claim that $Y_0, Y_1, \ldots$ is a martingale. Let us verify that this is true. Given Y_{i-1}, we have $Y_{i-1} = X_{i-1}^2 - (i-1)$. We have that

$$\begin{aligned}
\mathbf{E}\Big[Y_i \;\Big|\; Y_{i-1}\Big] = \mathbf{E}\Big[X_i^2 - i \;\Big|\; X_{i-1}^2 - (i-1)\Big] &= \frac{1}{2}\Big((X_{i-1}+1)^2 - i)\Big) + \frac{1}{2}\Big((X_{i-1}-1)^2 - i\Big) \\
&= X_{i-1}^2 + 1 - i = Y_{i-1},
\end{aligned}$$

which implies that indeed it is a martingale.

Example 8.1.8. Let U be a urn with b black balls, and w white balls. We repeatedly select a ball and replace it by c balls having the same color. Let X_i be the fraction of black balls after the first i trials. This sequence is a martingale.

Indeed, let $n_i = b + w + i(c-1)$ be the number of balls in the urn after the ith trial. Clearly,

$$\begin{aligned}\mathbf{E}\left[X_i \,\middle|\, X_{i-1}, \ldots, X_0\right] &= X_{i-1} \cdot \frac{(c-1) + X_{i-1} n_{i-1}}{n_i} + (1 - X_{i-1}) \cdot \frac{X_{i-1} n_{i-1}}{n_i} \\ &= \frac{X_{i-1}(c-1) + X_{i-1} n_{i-1}}{n_i} = X_{i-1} \frac{c - 1 + n_{i-1}}{n_i} = X_{i-1} \frac{n_i}{n_i} = X_{i-1}.\end{aligned}$$

Example 8.1.9. Let G be a random graph on the vertex set $V = \{1, \ldots, n\}$ obtained by independently choosing to include each possible edge with probability p. The underlying probability space is called $\mathbf{G}_{n,p}$. Arbitrarily label the $m = n(n-1)/2$ possible edges with the sequence $1, \ldots, m$. For $1 \le j \le m$, define the indicator random variable I_j, which takes values 1 if the edge j is present in G, and has value 0 otherwise. These indicator variables are independent and each takes value 1 with probability p.

Consider any real valued function f defined over the space of all graphs, e.g., the clique number, which is defined as being the size of the largest complete subgraph. The ***edge exposure martingale*** is defined to be the sequence of random variables $X_0, \ldots, X_m$ such that

$$X_i = \mathbf{E}\left[f(G) \,\middle|\, I_1, \ldots, I_i\right],$$

while $X_0 = \mathbf{E}[f(G)]$ and $X_m = f(G)$. The fact that this sequence of random variable is a martingale follows immediately from a theorem that would be described in the next lecture.

One can define similarly a ***vertex exposure martingale***, where the graph G_i is the graph induced on the first i vertices of the random graph G.

Theorem 8.1.10 (Azuma's Inequality.). *Let $X_0, \ldots, X_m$ be a martingale with $X_0 = 0$, and $|X_{i+1} - X_i| \le 1$ for all $0 \le i < m$. Let $\lambda > 0$ be arbitrary. Then*

$$\mathbf{Pr}\left[X_m > \lambda \sqrt{m}\right] < e^{-\lambda^2/2}.$$

Proof: Let $\alpha = \lambda / \sqrt{m}$. Let $Y_i = X_i - X_{i-1}$, so that $|Y_i| \le 1$ and $\mathbf{E}\left[Y_i \,\middle|\, X_0, \ldots, X_{i-1}\right] = 0$.

We are interested in bounding $\mathbf{E}\left[e^{\alpha Y_i} \,\middle|\, X_0, \ldots, X_{i-1}\right]$. Note that, for $-1 \le x \le 1$, we have

$$e^{\alpha x} \le h(x) = \frac{e^{\alpha} + e^{-\alpha}}{2} + \frac{e^{\alpha} - e^{-\alpha}}{2} x,$$

as $e^{\alpha x}$ is a convex function, $h(-1) = e^{-\alpha}$, $h(1) = e^{\alpha}$, and $h(x)$ is a linear function. Thus,

$$\begin{aligned}
\mathbf{E}\Big[e^{\alpha Y_i} \;\Big|\; X_0, \ldots, X_{i-1}\Big] &\leq \mathbf{E}\Big[h(Y_i) \;\Big|\; X_0, \ldots, X_{i-1}\Big] = h\Big(\mathbf{E}\Big[Y_i \;\Big|\; X_0, \ldots, X_{i-1}\Big]\Big) \\
&= h(0) = \frac{e^{\alpha} + e^{-\alpha}}{2} \\
&= \frac{(1 + \alpha + \frac{\alpha^2}{2!} + \frac{\alpha^3}{3!} + \cdots) + (1 - \alpha + \frac{\alpha^2}{2!} - \frac{\alpha^3}{3!} + \cdots)}{2} \\
&= 1 + \frac{\alpha^2}{2} + \frac{\alpha^4}{4!} + \frac{\alpha^6}{6!} + \cdots \\
&\leq 1 + \frac{1}{1!}\left(\frac{\alpha^2}{2}\right) + \frac{1}{2!}\left(\frac{\alpha^2}{2}\right)^2 + \frac{1}{3!}\left(\frac{\alpha^2}{2}\right)^3 + \cdots = e^{\alpha^2/2}
\end{aligned}$$

Hence,

$$\begin{aligned}
\mathbf{E}\Big[e^{\alpha X_m}\Big] &= \mathbf{E}\left[\prod_{i=1}^{m} e^{\alpha Y_i}\right] = \mathbf{E}\left[\left(\prod_{i=1}^{m-1} e^{\alpha Y_i}\right) e^{\alpha Y_m}\right] \\
&= \mathbf{E}\left[\left(\prod_{i=1}^{m-1} e^{\alpha Y_i}\right) \mathbf{E}\Big[e^{\alpha Y_m} \;\Big|\; X_0, \ldots, X_{m-1}\Big]\right] \leq e^{\alpha^2/2}\, \mathbf{E}\left[\prod_{i=1}^{m-1} e^{\alpha Y_i}\right] \\
&\leq e^{m\alpha^2/2}
\end{aligned}$$

Therefore, by Markov's inequality, we have

$$\begin{aligned}
\mathbf{Pr}\Big[X_m > \lambda\sqrt{m}\Big] &= \mathbf{Pr}\Big[e^{\alpha X_m} > e^{\alpha\lambda\sqrt{m}}\Big] = \frac{\mathbf{E}\Big[e^{\alpha X_m}\Big]}{e^{\alpha\lambda\sqrt{m}}} = e^{m\alpha^2/2 - \alpha\lambda\sqrt{m}} \\
&= \exp\Big(m(\lambda/\sqrt{m})^2/2 - (\lambda/\sqrt{m})\lambda\sqrt{m}\Big) = e^{-\lambda^2/2},
\end{aligned}$$

implying the result. ∎

Here is an alternative form.

Theorem 8.1.11 (Azuma's Inequality). *Let $X_0, \ldots, X_m$ be a martingale sequence such that and $|X_{i+1} - X_i| \leq 1$ for all $0 \leq i < m$. Let $\lambda > 0$ be arbitrary. Then*

$$\mathbf{Pr}\Big[|X_m - X_0| > \lambda\sqrt{m}\Big] < 2e^{-\lambda^2/2}.$$

Example 8.1.12. Let $\chi(H)$ be the chromatic number of a graph H. What is chromatic number of a random graph? How does this random variable behaves?

Consider the vertex exposure martingale, and let $X_i = \mathbf{E}\Big[\chi(G) \;\Big|\; G_i\Big]$. Again, without proving it, we claim that $X_0, \ldots, X_n = X$ is a martingale, and as such, we have: $\mathbf{Pr}\Big[|X_n - X_0| > \lambda\sqrt{n}\Big] \leq e^{-\lambda^2/2}$. However, $X_0 = \mathbf{E}[\chi(G)]$, and $X_n = \mathbf{E}\Big[\chi(G) \;\Big|\; G_n\Big] = \chi(G)$. Thus,

$$\mathbf{Pr}\Big[\Big|\chi(G) - \mathbf{E}[\chi(G)]\Big| > \lambda\sqrt{n}\Big] \leq e^{-\lambda^2/2}.$$

Namely, the chromatic number of a random graph is highly concentrated! And we do not even know what is the expectation of this variable!

8.2 Even more probability

Definition 8.2.1. A *σ-field* $(\Omega, \mathcal{F})$ consists of a sample space Ω (i.e., the atomic events) and a collection of subsets $\mathcal{F}$ satisfying the following conditions:

1. $\emptyset \in \mathcal{F}$.
2. $C \in \mathcal{F} \Rightarrow \overline{C} \in \mathcal{F}$.
3. $C_1, C_2, \ldots \in \mathcal{F} \Rightarrow C_1 \cup C_2 \ldots \in \mathcal{F}$.

Definition 8.2.2. Given a σ-field $(\Omega, \mathcal{F})$, a ***probability measure*** $\mathbf{Pr} : \mathcal{F} \to \mathbb{R}^+$ is a function that satisfies the following conditions.

1. $\forall A \in \mathcal{F}, 0 \leq \mathbf{Pr}[A] \leq 1$.
2. $\mathbf{Pr}[\Omega] = 1$.
3. For mutually disjoint events $C_1, C_2, \ldots$, we have $\mathbf{Pr}[\cup_i C_i] = \sum_i \mathbf{Pr}[C_i]$.

Definition 8.2.3. A ***probability space*** $(\Omega, \mathcal{F}, \mathbf{Pr})$ consists of a σ-field $(\Omega, \mathcal{F})$ with a probability measure $\mathbf{Pr}$ defined on it.

Chapter 9

Martingales II

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> "The Electric Monk was a labor-saving device, like a dishwasher or a video recorder. Dishwashers washed tedious dishes for you, thus saving you the bother of washing them yourself, video recorders watched tedious television for you, thus saving you the bother of looking at it yourself; Electric Monks believed things for you, thus saving you what was becoming an increasingly onerous task, that of believing all the things the world expected you to believe."
>
> – — Dirk Gently's Holistic Detective Agency, Douglas Adams..

9.1 Filters and Martingales

Definition 9.1.1. Given a σ-field $(\Omega, \mathcal{F})$ with $\mathcal{F} = 2^{\Omega}$, a ***filter*** (also ***filtration***) is a nested sequence $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \subseteq \mathcal{F}_n$ of subsets of 2^{Ω} such that

1. $\mathcal{F}_0 = \{\emptyset, \Omega\}$.
2. $\mathcal{F}_n = 2^{\Omega}$.
3. For $0 \leq i \leq n$, $(\Omega, \mathcal{F}_i)$ is a σ-field.

Definition 9.1.2. An ***elementary event*** or ***atomic event*** is a subset of a sample space that contains only one element.

Intuitively, when we consider a probability space, we usually consider a random variable X. The value of X is a function of the elementary event that happened in the probability space.

Thus, each $\mathcal{F}_i$ define a partition of Ω into *atomic events*. This partition is getting more and more refined as we progress down the filter.

Example 9.1.3. Consider an algorithm **Alg** that uses n random bits. As such, the underlying sample space is $\Omega = \left\{ b_1 b_2 \ldots b_n \;\middle|\; b_1, \ldots, b_n \in \{0,1\} \right\}$; that is, the set of all binary strings of length n. Next, let $\mathcal{F}_i$ be the σ-field generated by the partition of Ω into the atomic events B_w, where $w \in \{0,1\}^i$; here w is the string encoding the first i random bits used by the algorithm. Specifically,

$$B_w = \left\{ wx \;\middle|\; x \in \{0,1\}^{n-i} \right\}$$

and the set of atomic events in $\mathcal{F}_i$ is $\left\{ B_w \;\middle|\; w \in \{0,1\}^i \right\}$. The set $\mathcal{F}_i$ is the closure of this set of atomic events under complement and union.

We conclude that $\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_n$ form a filter.

Definition 9.1.4. A random variable X is said to be ***$\mathcal{F}_i$-measurable*** if for each $x \in \mathbb{R}$, the event $\{X \leq x\}$ is contained in $\mathcal{F}_i$.

Example 9.1.5. Let $\mathcal{F}_0, \ldots, \mathcal{F}_n$ be the filter defined in Example 9.1.3. Let X be the parity of the n bits. Clearly, X is a valid event only in $\mathcal{F}_n$ (why?). Namely, it is only measurable in $\mathcal{F}_n$, but not in $\mathcal{F}_i$, for $i < n$.

As such, a random variable X is $\mathcal{F}_i$-measurable, only if it is a constant on the elementary events of $\mathcal{F}_i$.

This gives us a new interpretation of what a filter is – its a sequence of refinements of the underlying probability space, that is achieved by splitting the atomic events of $\mathcal{F}_i$ into smaller atomic events in $\mathcal{F}_{i+1}$. Putting it explicitly, an atomic event $\mathcal{E}$ of $\mathcal{F}_i$, is a subset of 2^{Σ}. As we move to $\mathcal{F}_{i+1}$ the event $\mathcal{E}$ might now be split into several atomic (and disjoint events) $\mathcal{E}_1, \ldots, \mathcal{E}_k$. Now, naturally, the atomic event that really happens is an atomic event of $\mathcal{F}_n$. As we progress down the filter, we "zoom" into this event.

Definition 9.1.6. Let $(\Omega, \mathcal{F})$ be any σ-field, and Y any random variable that takes on distinct values on the elementary events in $\mathcal{F}$. Then $\mathbf{E}\left[X \;\middle|\; \mathcal{F} \right] = \mathbf{E}\left[X \;\middle|\; Y \right]$.

9.2 Martingales

Definition 9.2.1. A sequence of random variables $Y_1, Y_2, \ldots$, is said to be a ***martingale difference*** sequence if for all $i \geq 0$,

$$\mathbf{E}\left[Y_i \;\middle|\; Y_1, \ldots, Y_{i-1} \right] = 0.$$

Clearly, $X_1, \ldots,$ is a martingale sequence **iff** $Y_1, Y_2, \ldots,$ is a martingale difference sequence where $Y_i = X_i - X_{i-1}$.

Definition 9.2.2. A sequence of random variables $Y_1, Y_2, \ldots$, is said to be a ***super martingale*** sequence if for all $i \geq$,

$$\mathbf{E}\left[Y_i \;\middle|\; Y_1, \ldots, Y_{i-1} \right] \leq Y_{i-1},$$

and a ***sub martingale*** sequence if

$$\mathbf{E}\left[Y_i \;\middle|\; Y_1, \ldots, Y_{i-1} \right] \geq Y_{i-1}.$$

9.2.1 Martingales, an alternative definition

Definition 9.2.3. Let $(\Omega, \mathcal{F}, \mathbf{Pr})$ be a probability space with a filter $\mathcal{F}_0, \mathcal{F}_1, \ldots$. Suppose that $X_0, X_1, \ldots$, are random variables such that for all $i \geq 0$, X_i is $\mathcal{F}_i$-measurable. The sequence $X_0, \ldots, X_n$ is a ***martingale*** provided that, for all $i \geq 0$, we have

$$\mathbf{E}\Big[X_{i+1} \;\Big|\; \mathcal{F}_i\Big] = X_i.$$

Lemma 9.2.4. *Let $(\Omega, \mathcal{F})$ and $(\Omega, \mathcal{G})$ be two σ-fields such that $\mathcal{F} \subseteq \mathcal{G}$. Then, for any random variable X, $\mathbf{E}\Big[\mathbf{E}\Big[X \;\Big|\; \mathcal{G}\Big] \;\Big|\; \mathcal{F}\Big] = \mathbf{E}\Big[X \;\Big|\; \mathcal{F}\Big]$.*

Proof: $\mathbf{E}\Big[\mathbf{E}\Big[X \;\Big|\; \mathcal{G}\Big] \;\Big|\; \mathcal{F}\Big] = \mathbf{E}\Big[\mathbf{E}\Big[X \;\Big|\; G = g\Big] \;\Big|\; F = f\Big]$

$$\begin{aligned}
&= \mathbf{E}\left[\frac{\sum_x x \mathbf{Pr}[X = x \cap G = g]}{\mathbf{Pr}[G = g]} \;\middle|\; F = f\right] = \sum_{g \in G} \frac{\frac{\sum_x x \mathbf{Pr}[X = x \cap G = g]}{\mathbf{Pr}[G = g]} \cdot \mathbf{Pr}[G = g \cap F = f]}{\mathbf{Pr}[F = f]} \\
&= \sum_{g \in \mathcal{G}, g \subseteq f} \frac{\frac{\sum_x x \mathbf{Pr}[X = x \cap G = g]}{\mathbf{Pr}[G = g]} \cdot \mathbf{Pr}[G = g \cap F = f]}{\mathbf{Pr}[F = f]} = \sum_{g \in \mathcal{G}, g \subseteq f} \frac{\frac{\sum_x x \mathbf{Pr}[X = x \cap G = g]}{\mathbf{Pr}[G = g]} \cdot \mathbf{Pr}[G = g]}{\mathbf{Pr}[F = f]} \\
&= \sum_{g \in \mathcal{G}, g \subseteq f} \frac{\sum_x x \mathbf{Pr}[X = x \cap G = g]}{\mathbf{Pr}[F = f]} = \frac{\sum_x x \Big(\sum_{g \in \mathcal{G}, g \subseteq f} \mathbf{Pr}[X = x \cap G = g]\Big)}{\mathbf{Pr}[F = f]} \\
&= \frac{\sum_x x \mathbf{Pr}[X = x \cap F = f]}{\mathbf{Pr}[F = f]} = \mathbf{E}\Big[X \;\Big|\; \mathcal{F}\Big].
\end{aligned}$$

■

Theorem 9.2.5. *Let $(\Omega, \mathcal{F}, \mathbf{Pr})$ be a probability space, and let $\mathcal{F}_0, \ldots, \mathcal{F}_n$ be a filter with respect to it. Let X be any random variable over this probability space and define $X_i = \mathbf{E}\Big[X \;\Big|\; \mathcal{F}_i\Big]$ then, the sequence $X_0, \ldots, X_n$ is a martingale.*

Proof: We need to show that $\mathbf{E}\Big[X_{i+1} \;\Big|\; \mathcal{F}_i\Big] = X_i$. Namely,

$$\mathbf{E}\Big[X_{i+1} \;\Big|\; \mathcal{F}_i\Big] = \mathbf{E}\Big[\mathbf{E}\Big[X \;\Big|\; \mathcal{F}_{i+1}\Big] \;\Big|\; \mathcal{F}_i\Big] = \mathbf{E}\Big[X \;\Big|\; \mathcal{F}_i\Big] = X_i,$$

by Lemma 9.2.4 and by definition of X_i. ■

Definition 9.2.6. Let $f : \mathcal{D}_1 \times \cdots \times \mathcal{D}_n \to \mathbb{R}$ be a real-valued function with a arguments from possibly distinct domains. The function f is said to satisfy the ***Lipschitz condition*** if for any $x_1 \in \mathcal{D}_1, \ldots, x_n \in \mathcal{D}_n$, and $i \in \{1, \ldots, n\}$ and any $y_i \in \mathcal{D}_i$,

$$|f(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) - f(x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)| \leq 1.$$

Definition 9.2.7. Let $X_1, \ldots, X_n$ be a sequence of ***independent*** random variables, and a function $f(X_1, \ldots, X_n)$ defined over them that such that f satisfies the Lipschitz condition. The ***Doob martingale*** sequence $Y_0, \ldots, Y_m$ is defined by $Y_0 = \mathbf{E}[f(X_1, \ldots, X_n)]$ and

$$Y_i = \mathbf{E}\Big[f(X_1, \ldots, X_n) \;\Big|\; X_1, \ldots, X_i \Big],$$

for $i = 1, \ldots, n$. Clearly, $Y_0, \ldots, Y_n$ is a martingale, by Theorem 9.2.5.

Furthermore, if $|X_i - X_{i-1}| \leq 1$, for $i = 1, \ldots, n$ then $|X_i - X_{i-1}| \leq 1$. Thus, in this case, we can use Azuma's inequality on such a sequence.

9.3 Occupancy Revisited

We have m balls thrown independently and uniformly into n bins. Let Z denote the number of bins that remains empty. Let X_i be the bin chosen in the ith trial, and let $Z = F(X_1, \ldots, X_m)$. Clearly, we have by Azuma's inequality that $\mathbf{Pr}\Big[|Z - \mathbf{E}[Z]| > \lambda \sqrt{m}\Big] \leq 2e^{-\lambda^2/2}$.

The following is an extension of Azuma's inequality shown in class. We do not provide a proof but it is similar to what we saw.

Theorem 9.3.1 (Azuma's Inequality - Stronger Form). *Let $X_0, X_1, \ldots,$ be a martingale sequence such that for each k,*

$$|X_k - X_{k-1}| \leq c_k,$$

where c_k may depend on k. Then, for all $t \geq 0$, and any $\lambda > 0$,

$$\mathbf{Pr}[|X_t - X_0| \geq \lambda] \leq 2 \exp\left(-\frac{\lambda^2}{2 \sum_{k=1}^{t} c_k^2} \right).$$

Theorem 9.3.2. *Let $r = m/n$, and Z_{end} be the number of empty bins when m balls are thrown randomly into n bins. Then*

$$\mu = \mathbf{E}[Z_{end}] = n\left(1 - \frac{1}{n}\right)^m \approx ne^{-r}$$

and for $\lambda > 0$,

$$\mathbf{Pr}[|Z_{end} - \mu| \geq \lambda] \leq 2 \exp\left(-\frac{\lambda^2 (n - 1/2)}{n^2 - \mu^2} \right).$$

Proof: Let $z(Y, t)$ be the expected number of empty bins, if there are Y empty bins in time t. Clearly,

$$z(Y, t) = Y\left(1 - \frac{1}{n}\right)^{m-t}.$$

In particular, $\mu = z(n, 0) = n\left(1 - \frac{1}{n}\right)^m$.

Let $\mathcal{F}_t$ be the σ-field generated by the bins chosen in the first t steps. Let Z_{end} be the number of empty balls at time m, and let $Z_t = \mathbf{E}\Big[Z_{\text{end}} \;\Big|\; \mathcal{F}_t \Big]$. Namely, Z_t is the expected number of empty bins after we know where the first t balls had been placed. The random variables $Z_0, Z_1, \ldots, Z_m$ form a martingale. Let Y_t be the number of empty bins after t balls where thrown. We have $Z_{t-1} = z(Y_{t-1}, t - 1)$. Consider the ball thrown in the t-step. Clearly:

1. With probability $1 - Y_{t-1}/n$ the ball falls into a non-empty bin. Then $Y_t = Y_{t-1}$, and $Z_t = z(Y_{t-1}, t)$. Thus,

$$\begin{aligned}\Delta_t &= Z_t - Z_{t-1} = z(Y_{t-1}, t) - z(Y_{t-1}, t-1) = Y_{t-1}\left(\left(1 - \frac{1}{n}\right)^{m-t} - \left(1 - \frac{1}{n}\right)^{m-t+1}\right) \\ &= \frac{Y_{t-1}}{n}\left(1 - \frac{1}{n}\right)^{m-t} \leq \left(1 - \frac{1}{n}\right)^{m-t}.\end{aligned}$$

2. Otherwise, with probability Y_{t-1}/n the ball falls into an empty bin, and $Y_t = Y_{t-1} - 1$. Namely, $Z_t = z(Y_t - 1, t)$.

$$\begin{aligned}\Delta_t &= Z_t - Z_{t-1} = z(Y_{t-1} - 1, t) - z(Y_{t-1}, t-1) \\ &= (Y_{t-1} - 1)\left(1 - \frac{1}{n}\right)^{m-t} - Y_{t-1}\left(1 - \frac{1}{n}\right)^{m-t+1} \\ &= \left(1 - \frac{1}{n}\right)^{m-t}\left(Y_{t-1} - 1 - Y_{t-1}\left(1 - \frac{1}{n}\right)\right) \\ &= \left(1 - \frac{1}{n}\right)^{m-t}\left(-1 + \frac{Y_{t-1}}{n}\right) = -\left(1 - \frac{1}{n}\right)^{m-t}\left(1 - \frac{Y_{t-1}}{n}\right) \\ &\geq -\left(1 - \frac{1}{n}\right)^{m-t}.\end{aligned}$$

Thus, $Z_0, \ldots, Z_m$ is a martingale sequence, where $|Z_t - Z_{t-1}| \leq |\Delta_t| \leq c_t$, where $c_t = \left(1 - \frac{1}{n}\right)^{m-t}$. We have

$$\sum_{t=1}^{n} c_t^2 = \frac{1 - (1 - 1/n)^{2m}}{1 - (1 - 1/n)^2} = \frac{n^2\left(1 - (1 - 1/n)^{2m}\right)}{2n - 1} = \frac{n^2 - \mu^2}{2n - 1}.$$

Now, deploying Azuma's inequality, yield the result. ■

Chapter 10

The Probabilistic Method

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> "Shortly after the celebration of the four thousandth anniversary of the opening of space, Angary J. Gustible discovered Gustible's planet. The discovery turned out to be a tragic mistake.
>
> Gustible's planet was inhabited by highly intelligent life forms. They had moderate telepathic powers. They immediately mind-read Angary J. Gustible's entire mind and life history, and embarrassed him very deeply by making up an opera concerning his recent divorce."
>
> – – From Gustible's Planet, Cordwainer Smith.

10.1 Introduction

The probabilistic method is a combinatorial technique to use probabilistic algorithms to create objects having desirable properties, and furthermore, prove that such objects exist. The basic technique is based on two basic observations:

1. If $\mathbf{E}[X] = \mu$, then there exists a value x of X, such that $x \geq \mathbf{E}[X]$.
2. If the probability of event $\mathcal{E}$ is larger than zero, then $\mathcal{E}$ exists and it is not empty.

The surprising thing is that despite the elementary nature of those two observations, they lead to a powerful technique that leads to numerous nice and strong results. Including some elementary proofs of theorems that previously had very complicated and involved proofs.

The main proponent of the probabilistic method, was Paul Erdős. An excellent text on the topic is the book by Noga Alon and Joel Spencer [AS00].

This topic is worthy of its own course. The interested student is refereed to the course "Math 475 — The Probabilistic Method".

10.1.1 Examples

Theorem 10.1.1. *For any undirected graph $G(V, E)$ with n vertices and m edges, there is a partition of the vertex set V into two sets A and B such that*

$$\left|\left\{uv \in E \;\middle|\; u \in A \text{ and } v \in B\right\}\right| \geq \frac{m}{2}.$$

Proof: Consider the following experiment: randomly assign each vertex to A or B, independently and equal probability.

For an edge $e = uv$, the probability that one endpoint is in A, and the other in B is $1/2$, and let X_e be the indicator variable with value 1 if this happens. Clearly,

$$\mathbf{E}\left[\left|\left\{uv \in E \;\middle|\; (u,v) \in (A \times B) \cup (B \times A)\right\}\right|\right] = \sum_{e \in E(G)} \mathbf{E}[X_e] = \sum_{e \in E(G)} \frac{1}{2} = \frac{m}{2}.$$

Thus, there must be a partition of V that satisfies the theorem. ■

Definition 10.1.2. For a vector $v = (v_1, \ldots, v_n) \in \mathbb{R}^n$, $\|v\|_\infty = \max_i |v_i|$.

Theorem 10.1.3. *Let* M *be an* $n \times n$ *binary matrix (i.e., each entry is either* 0 *or* 1*), then there always exists a vector* $b \in \{-1, +1\}^n$ *such that* $\|\mathsf{M}b\|_\infty \leq 4\sqrt{n \log n}$.

Proof: Let $v = (v_1, \ldots, v_n)$ be a row of M. Chose a random $b = (b_1, \ldots, b_n) \in \{-1, +1\}^n$. Let $i_1, \ldots, i_m$ be the indices such that $v_{i_j} = 1$, and let

$$Y = \langle v, b \rangle = \sum_{i=1}^{n} v_i b_i = \sum_{j=1}^{m} v_{i_j} b_{i_j} = \sum_{j=1}^{m} b_{i_j}.$$

As such Y is the sum of m independent random variables that accept values in $\{-1, +1\}$. Clearly,

$$\mathbf{E}[Y] = \mathbf{E}\Big[\langle v, b \rangle\Big] = \mathbf{E}\left[\sum_i v_i b_i\right] = \sum_i \mathbf{E}[v_i b_i] = \sum_i v_i \,\mathbf{E}[b_i] = 0.$$

By Chernoff inequality (Theorem 6.2.1) and the symmetry of Y, we have that, for $\Delta = 4\sqrt{n \ln n}$, it holds

$$\mathbf{Pr}\Big[|Y| \geq \Delta\Big] = 2\,\mathbf{Pr}\Big[v \cdot b \geq \Delta\Big] = 2\,\mathbf{Pr}\left[\sum_{j=1}^{m} b_{i_j} \geq \Delta\right] \leq 2\exp\left(-\frac{\Delta^2}{2m}\right) = 2\exp\left(-8\frac{n \ln n}{m}\right) \leq \frac{2}{n^8}.$$

Thus, the probability that any entry in $\mathsf{M}b$ exceeds $4\sqrt{n \ln n}$ is smaller than $2/n^7$. Thus, with probability at least $1 - 2/n^7$, all the entries of $\mathsf{M}b$ have value smaller than $4\sqrt{n \ln n}$.

In particular, there exists a vector $b \in \{-1, +1\}^n$ such that $\left\| \mathsf{M}b \right\|_\infty \leq 4\sqrt{n \ln n}$. ■

10.2 Maximum Satisfiability

In the MAX-SAT problem, we are given a binary formula F in ***[CNF]*** (Conjunctive normal form), and we would like to find an assignment that satisfies as many clauses as possible of F, for example $F = (x \vee y) \wedge (\overline{x} \vee z)$. Of course, an assignment satisfying all the clauses of the formula, and thus F itself, would be even better – but this problem is of course NPC. As such, we are looking for how well can be we do when we relax the problem to maximizing the number of clauses to be satisfied..

Theorem 10.2.1. *For any set of* m *clauses, there is a truth assignment of variables that satisfies at least* $m/2$ *clauses.*

Proof: Assign every variable a random value. Clearly, a clause with k variables, has probability $1 - 2^{-k}$ to be satisfied. Using linearity of expectation, and the fact that every clause has at least one variable, it follows, that $\mathbf{E}[X] = m/2$, where X is the random variable counting the number of clauses being satisfied. In particular, there exists an assignment for which $X \geq m/2$. ■

For an instant I, let $m_{\text{opt}}(I)$, denote the maximum number of clauses that can be satisfied by the "best" assignment. For an algorithm **Alg**, let $m_{\textbf{Alg}}(I)$ denote the number of clauses satisfied computed by the algorithm **Alg**. The ***approximation factor*** of **Alg**, is $m_{\textbf{Alg}}(I)/m_{\text{opt}}(I)$. Clearly, the algorithm of Theorem 10.2.1 provides us with $1/2$-approximation algorithm.

For every clause, C_j in the given instance, let $z_j \in \{0, 1\}$ be a variable indicating whether C_j is satisfied or not. Similarly, let $x_i = 1$ if the ith variable is being assigned the value TRUE. Let C_j^+ be indices of the variables that appear in C_j in the positive, and C_j^- the indices of the variables that appear in the negative. Clearly, to solve MAX-SAT, we need to solve:

$$\begin{aligned} \textit{maximize} \quad & \sum_{j=1}^{m} z_j \\ \textit{subject to} \quad & x_i, z_j \in \{0,1\} \text{ for all } i, j \\ & \sum_{i \in C_j^+} x_i + \sum_{i \in C_j^-} (1 - x_i) \geq z_j \text{ for all } j. \end{aligned}$$

We relax this into the following linear program:

$$\begin{aligned} \textit{maximize} \quad & \sum_{j=1}^{m} z_j \\ \textit{subject to} \quad & 0 \leq y_i, z_j \leq 1 \text{ for all } i, j \\ & \sum_{i \in C_j^+} y_i + \sum_{i \in C_j^-} (1 - y_i) \geq z_j \text{ for all } j. \end{aligned}$$

Which can be solved in polynomial time. Let $\widehat{t}$ denote the values assigned to the variable t by the linear-programming solution. Clearly, $\sum_{j=1}^{m} \widehat{z_j}$ is an upper bound on the number of clauses of I that can be satisfied.

We set the variable y_i to 1 with probability $\widehat{y_i}$. This is ***randomized rounding***.

Lemma 10.2.2. *Let C_j be a clause with k literals. The probability that it is satisfied by randomized rounding is at least $\beta_k \widehat{z_j} \geq (1 - 1/e)\widehat{z_j}$, where*

$$\beta_k = 1 - \left(1 - \frac{1}{k}\right)^k.$$

Proof: Assume $C_j = y_1 \vee v_2 \ldots \vee v_k$. By the LP, we have $\widehat{y_1} + \cdots + \widehat{y_k} \geq \widehat{z_j}$. Furthermore, the probability that C_j is not satisfied is $\prod_{i=1}^{k}(1-\widehat{y_i})$. Note that $1 - \prod_{i=1}^{k}(1-\widehat{y_i})$ is minimized when all the $\widehat{y_i}$'s are equal (by symmetry). Namely, when $\widehat{y_i} = \widehat{z_j}/k$. Consider the function $f(x) = 1-(1-x/k)^k$. This is a concave function, which is larger than $g(x) = \beta_k x$ for all $0 \leq x \leq 1$, as can be easily verified, by checking the inequality at $x = 0$ and $x = 1$.

Thus,

$$\mathbf{Pr}\Big[C_j \text{ is satisfied}\Big] = 1 - \prod_{i=1}^{k}(1-\widehat{y_i}) \geq f\left(\widehat{z_j}\right) \geq \beta_k \widehat{z_j}.$$

The second part of the inequality, follows from the fact that $\beta_k \geq 1 - 1/e$, for all $k \geq 0$. Indeed, for $k = 1, 2$ the claim trivially holds. Furthermore,

$$1 - \left(1 - \frac{1}{k}\right)^k \geq 1 - \frac{1}{e} \quad \Leftrightarrow \quad \left(1 - \frac{1}{k}\right)^k \leq \frac{1}{e},$$

but this holds since $1 - x \leq e^{-x}$ implies that $1 - \frac{1}{k} \leq e^{-1/k}$, and as such $\left(1 - \frac{1}{k}\right)^k \leq e^{-k/k} = 1/e$. ■

Theorem 10.2.3. *Given an instance I of MAX-SAT, the expected number of clauses satisfied by linear programming and randomized rounding is at least $(1 - 1/e) \approx 0.632 m_{\text{opt}}(I)$, where $m_{\text{opt}}(I)$ is the maximum number of clauses that can be satisfied on that instance.*

Theorem 10.2.4. *Given an instance I of MAX-SAT, let n_1 be the expected number of clauses satisfied by randomized assignment, and let n_2 be the expected number of clauses satisfied by linear programming followed by randomized rounding. Then, $\max(n_1, n_2) \geq (3/4)\sum_j \widehat{z_j} \geq (3/4) m_{\text{opt}}(I)$.*

Proof: It is enough to show that $(n_1 + n_2)/2 \geq \frac{3}{4}\sum_j \widehat{z_j}$. Let S_k denote the set of clauses that contain k literals. We know that

$$n_1 = \sum_k \sum_{C_j \in S_k} \left(1 - 2^{-k}\right) \geq \sum_k \sum_{C_j \in S_k} \left(1 - 2^{-k}\right) \widehat{z_j}.$$

By Lemma 10.2.2 we have $n_2 \geq \sum_k \sum_{C_j \in S_k} \beta_k \widehat{z_j}$. Thus,

$$\frac{n_1 + n_2}{2} \geq \sum_k \sum_{C_j \in S_k} \frac{1 - 2^{-k} + \beta_k}{2} \widehat{z_j}.$$

One can verify that $\left(1 - 2^{-k}\right) + \beta_k \geq 3/2$, for all k. [①] Thus, we have

$$\frac{n_1 + n_2}{2} \geq \frac{3}{4} \sum_k \sum_{C_j \in S_k} \widehat{z_j} = \frac{3}{4} \sum_j \widehat{z_j}.$$

■

[①] Indeed, by the proof of Lemma 10.2.2, we have that $\beta_k \geq 1 - 1/e$. Thus, $\left(1 - 2^{-k}\right) + \beta_k \geq 2 - 1/e - 2^{-k} \geq 3/2$ for $k \geq 3$. Thus, we only need to check the inequality for $k = 1$ and $k = 2$, which can be done directly.

Chapter 11

The Probabilistic Method II

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> "Today I know that everything watches, that nothing goes unseen, and that even wallpaper has a better memory than ours. It isn't God in His heaven that sees all. A kitchen chair, a coat-hanger a half-filled ash tray, or the wood replica of a woman name Niobe, can perfectly well serve as an unforgetting witness to every one of our acts."
>
> – Gunter Grass, The tin drum.

11.1 Expanding Graphs

In this lecture, we are going to discuss *expanding graphs*.

Definition 11.1.1. An (n, d, α, c) OR-concentrator is a bipartite multigraph $G(L, R, E)$, with the independent sets of vertices L and R each of cardinality n, such that

(i) Every vertex in L has degree at most d.

(ii) Any subset S of vertices of L, with $|S| \leq \alpha n$ has at least $c|S|$ neighbors in R.

A good (n, d, α, c) OR-concentrator would have d should as small as possible, and c as large as possible.

Theorem 11.1.2. *There is an integer n_0 such that for all $n \geq n_0$, there is an $(n, 18, 1/3, 2)$ OR-concentrator.*

Proof: Let every vertex of L choose neighbors by sampling (with replacement) d vertices independently and uniformly from R. We discard multiple edges in the resulting graph.

Let $\mathcal{E}_s$ be the event that a subset of s vertices of L has fewer than cs neighbors in R. Clearly,

$$\mathbf{Pr}[\mathcal{E}_s] \leq \binom{n}{s}\binom{n}{cs}\left(\frac{cs}{n}\right)^{ds} \leq \left(\frac{ne}{s}\right)^{s}\left(\frac{ne}{cs}\right)^{cs}\left(\frac{cs}{n}\right)^{ds} = \left(\left(\frac{s}{n}\right)^{d-c-1} e^{1+c} c^{d-c}\right)^{s},$$

since $\binom{n}{k} \leq \left(\frac{ne}{k}\right)^k$. Setting $\alpha = 1/3$ using $s \leq \alpha n$, and $c = 2$, we have

$$\begin{aligned}\mathbf{Pr}[\mathcal{E}_s] \leq \left(\left(\frac{1}{3}\right)^{d-c-1} e^{1+c} c^{d-c}\right)^s &\leq \left(\left(\frac{1}{3}\right)^d 3^{1+c} e^{1+c} c^{d-c}\right)^s \leq \left(\left(\frac{1}{3}\right)^d 3^{1+c} e^{1+c} c^d\right)^s \\ &\leq \left(\left(\frac{c}{3}\right)^d (3e)^{1+c}\right)^s \leq \left(\left(\frac{2}{3}\right)^{18} (3e)^{1+2}\right)^s \leq (0.4)^s,\end{aligned}$$

as $c = 2$ and $d = 18$. Thus,

$$\sum_{s \geq 1} \mathbf{Pr}[\mathcal{E}_s] \leq \sum_{s \geq 1} (0.4)^s < 1.$$

It thus follows that the random graph we generated has the required properties with positive probability. ■

11.2 Probability Amplification

Let **Alg** be an algorithm in **RP**, such that given x, **Alg** picks a random number r from the range $\mathbb{Z}_n = \{0, \ldots, n-1\}$, for a suitable choice of a prime n, and computes a binary value **Alg**(x, r) with the following properties:

1. If $x \in L$, then **Alg**$(x, r) = 1$ for at least half the possible values of r.
2. If $x \notin L$, then **Alg**$(x, r) = 0$ for all possible choices of r.

Next, we show that using $\lg^2 n$ bits, one can achieve $1/n^{\lg n}$ confidence, compared with the naive $1/n$, and the $1/t$ confidence achieved by t (dependent) executions of the algorithm using two-point sampling.

Theorem 11.2.1. *For n large enough, there exists a bipartite graph $G(V, R, E)$ with $|V| = n$, $|R| = 2^{\lg^2 n}$ such that:*

(i) Every subset of $n/2$ vertices of V has at least $\left(2^{\lg^2 n} - n\right)$ neighbors in R.

(ii) No vertex of R has more than $12 \lg^2 n$ neighbors.

Proof: Each vertex of R chooses $d = 2^{\lg^2 n}(4 \lg^2 n)/n$ neighbors in R. We show that the resulting graph violate the required properties with probability less than half.

The probability for a set of $n/2$ vertices on the left to fail to have enough neighbors, is

$$\begin{aligned}\tau &= \binom{n}{n/2}\binom{2^{\lg^2 n}}{n}\left(1 - \frac{n}{2^{\lg^2 n}}\right)^{dn/2} \leq \left(\frac{ne}{n/2}\right)^n \left(\frac{2^{\lg^2 n} e}{n}\right)^n \exp\left(-\frac{dn}{2}\frac{n}{2^{\lg^2 n}}\right) \\ &\leq (2e)^n \left(\frac{2^{\lg^2 n} e}{n}\right)^n \exp\left(-\frac{2^{\lg^2 n}(4 \lg^2 n)/n}{2}\frac{n^2}{2^{\lg^2 n}}\right) \leq \exp\left(2n + n \ln\left(\frac{2^{\lg^2 n} e}{n}\right) - 2n \lg^2 n\right) \\ &\leq \exp\left(3n + (\ln 2 - 2)\, n \lg^2 n - n \ln n\right) \leq \exp\left(3n - 1.3 n \lg^2 n - n \ln n\right) \ll \frac{1}{2}.\end{aligned}$$

As for the second property, note that the expected number of neighbors of a vertex of R is $4\lg^2 n$; the Chernoff bound now shows that the probability of exceeding $12\lg^2 n$ neighbors is less than $(e/3)^{12\lg^2 n} = (1/3)^{\lg^2 n}$. Since R contains $2^{\lg^2 n}$ vertices this implies, that the probability for a bad vertex is bounded by $(2/3)^{\lg^2 n} \ll 1/2$.

Thus, with constant positive probability, the random graph has the required property. ■

We assume that given a vertex (of the above graph) we can compute its neighbors, without computing the whole graph.

So, we are given an input x. Use $\lg^2 n$ bits to pick a vertex $v \in R$. We next identify the neighbors of v in V: $r_1, \ldots, r_k$. We then compute **Alg**(x, r_i) for $1 \le i \le k$. Note that $k = O\left(\lg^2 n\right)$. If all k calls return 0, then we return that **Alg** is not in the language. Otherwise, we return that x belong to V.

If x is in the language, then Consider the subset $U \subseteq V$, such that running **Alg** on any of the strings of U returns TRUE. We know that $|U| \ge n/2$. The set U is connected to all the vertices of R except for at most $|R| - \left(2^{\lg^2 n} - n\right) = n$ of them. As such, the probability of a failure in this case, is

$$\mathbf{Pr}[x \in L \text{ but } r_1, r_2, \ldots, r_k \notin U] = \mathbf{Pr}[v \text{ not connected to } U] \le \frac{n}{|R|} \le \frac{n}{2^{\lg^2 n}}.$$

Unfortunately, there is no explicit construction of the expanders used here. However, there are alternative techniques that achieve a similar result.

11.3 Oblivious routing revisited

Theorem 11.3.1. *Consider any randomized oblivious algorithm for permutation routing on the hypercube with $N = 2^n$ nodes. If this algorithm uses k random bits, then its expected running time is $\Omega\left(2^{-k}\sqrt{N/n}\right)$.*

Corollary 11.3.2. *Any randomized oblivious algorithm for permutation routing on the hypercube with $N = 2^n$ nodes must use $\Omega(n)$ random bits in order to achieve expected running time $O(n)$.*

Theorem 11.3.3. *For every n, there exists a randomized oblivious scheme for permutation routing on a hypercube with $n = 2^n$ nodes that uses $3n$ random bits and runs in expected time at most $15n$.*

Chapter 12

The Probabilistic Method III

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> At other times you seemed to me either pitiable or contemptible, eunuchs, artificially confined to an eternal childhood, childlike and childish in your cool, tightly fenced, neatly tidied playground and kindergarten, where every nose is carefully wiped and every troublesome emotion is soothed, every dangerous thought repressed, where everyone plays nice, safe, bloodless games for a lifetime and every jagged stirring of life, every strong feeling, every genuine passion, every rapture is promptly checked, deflected and neutralized by meditation therapy.
>
> – – The Glass Bead Game, Hermann Hesse .

12.1 The Lovász Local Lemma

Lemma 12.1.1. *(i)* $\mathbf{Pr}\left[A \;\middle|\; B \cap C\right] = \dfrac{\mathbf{Pr}\left[A \cap B \;\middle|\; C\right]}{\mathbf{Pr}\left[B \;\middle|\; C\right]}$

(ii) Let $\eta_1, \ldots, \eta_n$ be n events which are not necessarily independent. Then,

$$\mathbf{Pr}[\cap_{i=1}^{n} \eta_i] = \mathbf{Pr}[\eta_1] * \mathbf{Pr}\left[\eta_2 \;\middle|\; \eta_1\right] \mathbf{Pr}\left[\eta_3 \;\middle|\; \eta_1 \cap \eta_2\right] * \ldots * \mathbf{Pr}\left[\eta_n \;\middle|\; \eta_1 \cap \ldots \cap \eta_{n-1}\right].$$

Proof: (i) We have that

$$\frac{\mathbf{Pr}\left[A \cap B \;\middle|\; C\right]}{\mathbf{Pr}\left[B \;\middle|\; C\right]} = \frac{\mathbf{Pr}[A \cap B \cap C]}{\mathbf{Pr}[C]} \Big/ \frac{\mathbf{Pr}[B \cap C]}{\mathbf{Pr}[C]} = \frac{\mathbf{Pr}[A \cap B \cap C]}{\mathbf{Pr}[B \cap C]} = \mathbf{Pr}\left[A \;\middle|\; B \cap C\right].$$

As for (ii), we already saw it and used it in the minimum cut algorithm lecture. ■

Definition 12.1.2. An event $\mathcal{E}$ is mutually independent of a set of events $\mathcal{C}$, if for any subset $\mathcal{U} \subseteq \mathcal{C}$, we have that $\mathbf{Pr}[\mathcal{E} \cap (\bigcap_{\mathcal{E}' \in U} \mathcal{E}')] = \mathbf{Pr}[\mathcal{E}]\,\mathbf{Pr}[\bigcap_{\mathcal{E}' \in U} \mathcal{E}']$.

Let $\mathcal{E}_1, \ldots, \mathcal{E}_n$ be events. A ***dependency graph*** for these events is a directed graph $G = (V, E)$, where $\{1, \ldots, n\}$, such that $\mathcal{E}_i$ is mutually independent of all the events in $\left\{\mathcal{E}_j \;\middle|\; (i, j) \notin E\right\}$.

Intuitively, an edge (i, j) in a dependency graph indicates that $\mathcal{E}_i$ and $\mathcal{E}_j$ have (maybe) some dependency between them. We are interested in settings where this dependency is limited enough, that we can claim something about the probability of all these events happening simultaneously.

Lemma 12.1.3 (Lovász Local Lemma). *Let $G(V, E)$ be a dependency graph for events $\mathcal{E}_1, \ldots, \mathcal{E}_n$. Suppose that there exist $x_i \in [0, 1]$, for $1 \leq i \leq n$ such that $\mathbf{Pr}[\mathcal{E}_i] \leq x_i \prod_{(i,j)\in E} \Big(1 - x_j\Big)$. Then $\mathbf{Pr}\Big[\cap_{i=1}^{n} \overline{\mathcal{E}_i}\Big] \geq \prod_{i=1}^{n} (1 - x_i)$.*

We need the following technical lemma.

Lemma 12.1.4. *Let $G(V, E)$ be a dependency graph for events $\mathcal{E}_1, \ldots, \mathcal{E}_n$. Suppose that there exist $x_i \in [0, 1]$, for $1 \leq i \leq n$ such that $\mathbf{Pr}[\mathcal{E}_i] \leq x_i \prod_{(i,j)\in E} \Big(1 - x_j\Big)$. Now, let S be a subset of the vertices from $\{1, \ldots, n\}$, and let i be an index not in S. We have that*

$$\mathbf{Pr}\Big[\mathcal{E}_i \;\Big|\; \cap_{j\in S} \overline{\mathcal{E}_j}\Big] \leq x_i. \tag{12.1}$$

Proof: The proof is by induction on $k = |S|$.

For $k = 0$, we have by assumption that $\mathbf{Pr}\Big[\mathcal{E}_i \;\Big|\; \cap_{j\in S} \overline{\mathcal{E}_j}\Big] = \mathbf{Pr}[\mathcal{E}_i] \leq x_i \prod_{(i,j)\in E} \Big(1 - x_j\Big) \leq x_i$.

Thus, let $N = \Big\{ j \in S \;\Big|\; (i, j) \in E \Big\}$, and let $R = S \setminus N$. If $N = \emptyset$, then we have that $\mathcal{E}_i$ is mutually independent of the events of $\mathcal{C}(R) = \Big\{ \mathcal{E}_j \;\Big|\; j \in R \Big\}$. Thus, $\mathbf{Pr}\Big[\mathcal{E}_i \;\Big|\; \cap_{j\in S} \overline{\mathcal{E}_j}\Big] = \mathbf{Pr}\Big[\mathcal{E}_i \;\Big|\; \cap_{j\in R} \overline{\mathcal{E}_j}\Big] = \mathbf{Pr}[\mathcal{E}_i] \leq x_i$, by arguing as above.

By Lemma 12.1.1 (i), we have that

$$\mathbf{Pr}\left[\mathcal{E}_i \;\middle|\; \bigcap_{j\in S} \overline{\mathcal{E}_j}\right] = \frac{\mathbf{Pr}\Big[\mathcal{E}_i \cap \Big(\cap_{j\in N} \overline{\mathcal{E}_j}\Big) \;\Big|\; \cap_{m\in R} \overline{\mathcal{E}_m}\Big]}{\mathbf{Pr}\Big[\cap_{j\in N} \overline{\mathcal{E}_j} \;\Big|\; \cap_{m\in R} \overline{\mathcal{E}_m}\Big]}.$$

We bound the numerator by

$$\mathbf{Pr}\Big[\mathcal{E}_i \cap \Big(\cap_{j\in N} \overline{\mathcal{E}_j}\Big) \;\Big|\; \cap_{m\in R} \overline{\mathcal{E}_m}\Big] \leq \mathbf{Pr}\Big[\mathcal{E}_i \;\Big|\; \cap_{m\in R} \overline{\mathcal{E}_m}\Big] = \mathbf{Pr}[\mathcal{E}_i] \leq x_i \prod_{(i,j)\in E} \Big(1 - x_j\Big),$$

since $\mathcal{E}_i$ is mutually independent of $\mathcal{C}(R)$. As for the denominator, let $N = \{j_1, \ldots, j_r\}$. We have, by Lemma 12.1.1 (ii), that

$$\begin{aligned}
\mathbf{Pr}\Big[\overline{\mathcal{E}_{j_1}} \cap \ldots \cap \overline{\mathcal{E}_{j_r}} \;\Big|\; \cap_{m\in R} \overline{\mathcal{E}_m}\Big] &= \mathbf{Pr}\Big[\overline{\mathcal{E}_{j_1}} \;\Big|\; \cap_{m\in R} \overline{\mathcal{E}_m}\Big] \mathbf{Pr}\Big[\overline{\mathcal{E}_{j_2}} \;\Big|\; \overline{\mathcal{E}_{j_1}} \cap \Big(\cap_{m\in R} \overline{\mathcal{E}_m}\Big)\Big] \\
&\quad \cdots \mathbf{Pr}\Big[\overline{\mathcal{E}_{j_r}} \;\Big|\; \overline{\mathcal{E}_{j_1}} \cap \ldots \cap \overline{\mathcal{E}_{j_{r-1}}} \cap \Big(\cap_{m\in R} \overline{\mathcal{E}_m}\Big)\Big] \\
&= \Big(1 - \mathbf{Pr}\Big[\mathcal{E}_{j_1} \;\Big|\; \cap_{m\in R} \overline{\mathcal{E}_m}\Big]\Big) \Big(1 - \mathbf{Pr}\Big[\mathcal{E}_{j_2} \;\Big|\; \overline{\mathcal{E}_{j_1}} \cap \Big(\cap_{m\in R} \overline{\mathcal{E}_m}\Big)\Big]\Big) \\
&\quad \cdots \Big(1 - \mathbf{Pr}\Big[\mathcal{E}_{j_r} \;\Big|\; \overline{\mathcal{E}_{j_1}} \cap \ldots \cap \overline{\mathcal{E}_{j_{r-1}}} \cap \Big(\cap_{m\in R} \overline{\mathcal{E}_m}\Big)\Big]\Big) \\
&\geq \Big(1 - x_{j_1}\Big) \cdots \Big(1 - x_{j_r}\Big) \geq \prod_{(i,j)\in E} \Big(1 - x_j\Big),
\end{aligned}$$

by Eq. (12.1) and induction, as every probability term in the above expression has less than $|S|$ items involved. It thus follows, that $\mathbf{Pr}\left[\mathcal{E}_i \;\middle|\; \bigcap_{j\in S} \overline{\mathcal{E}_j}\right] \leq x_i$. ∎

Proof of Lovász local lemma (Lemma 12.1.3): Using Lemma 12.1.4, we have that

$$\mathbf{Pr}\left[\cap_{i=1}^{n} \overline{\mathcal{E}_i}\right] = (1 - \mathbf{Pr}[\mathcal{E}_1])\left(1 - \mathbf{Pr}\left[\mathcal{E}_2 \;\middle|\; \overline{\mathcal{E}_1}\right]\right) \cdots \left(1 - \mathbf{Pr}\left[\mathcal{E}_n \;\middle|\; \cap_{i=1}^{n-1} \overline{\mathcal{E}_i}\right]\right) \geq \prod_{i=1}^{n} (1 - x_i).$$ ∎

Corollary 12.1.5. *Let $\mathcal{E}_1, \ldots, \mathcal{E}_n$ be events, with $\mathbf{Pr}[\mathcal{E}_i] \leq p$ for all i. If each event is mutually independent of all other events except for at most d, and if $ep(d+1) \leq 1$, then $\mathbf{Pr}\left[\cap_{i=1}^{n} \overline{\mathcal{E}_i}\right] > 0$.*

Proof: If $d = 0$ the result is trivial, as the events are independent. Otherwise, there is a dependency graph, with every vertex having degree at most d. Apply Lemma 12.1.3 with $x_i = \frac{1}{d+1}$. Observe that

$$x_i(1 - x_i)^d = \frac{1}{d+1}\left(1 - \frac{1}{d+1}\right)^d > \frac{1}{d+1} \cdot \frac{1}{e} \geq p,$$

by assumption and the since $\left(1 - \frac{1}{d+1}\right)^d > 1/e$, see Lemma 12.1.6 below. ∎

The following is standard by now, and we include it only for the sake of completeness.

Lemma 12.1.6. *For any $n \geq 1$, we have* $\left(1 - \frac{1}{n+1}\right)^n > \frac{1}{e}$.

Proof: This is equivalent to $\left(\frac{n}{n+1}\right)^n > \frac{1}{e}$. Namely, we need to prove $e > \left(\frac{n+1}{n}\right)^n$. But this obvious, since $\left(\frac{n+1}{n}\right)^n = \left(1 + \frac{1}{n}\right)^n < \exp(n(1/n)) = e$. ∎

12.2 Application to k-SAT

We are given a instance I of k-SAT, where every clause contains k literals, there are m clauses, and every one of the n variables, appears in at most $2^{k/50}$ clauses.

Consider a random assignment, and let $\mathcal{E}_i$ be the event that the ith clause was not satisfied. We know that $p = \mathbf{Pr}[\mathcal{E}_i] = 2^{-k}$, and furthermore, $\mathcal{E}_i$ depends on at most $d = k2^{k/50}$ other events. Since $ep(d+1) = e\left(k^{k/50} + 1\right)2^{-k} < 1$, for $k \geq 4$, we conclude that by Corollary 12.1.5, that

$$\mathbf{Pr}[I \text{ have a satisfying assignment}] = \mathbf{Pr}[\cup_i \mathcal{E}_i] > 0.$$

12.2.1 An efficient algorithm

The above just proves that a satisfying assignment exists. We next show a polynomial algorithm (in m) for the computation of such an assignment (the algorithm will not be polynomial in k).

Let G be the dependency graph for I, where the vertices are the clauses of I, and two clauses are connected if they share a variable. In the first stage of the algorithm, we assign values to the variables one by one, in an arbitrary order. In the beginning of this process all variables are unspecified, at each step, we randomly assign a variable either 0 or 1 with equal probability.

Definition 12.2.1. A clause $\mathcal{E}_i$ is ***dangerous*** if both the following conditions hold:

(i) $k/2$ literals of $\mathcal{E}_i$ have been fixed.

(ii) $\mathcal{E}_i$ is still unsatisfied.

After assigning each value, we discover all the dangerous clauses, and we defer ("freeze") all the unassigned variables participating in such a clause. We continue in this fashion till all the unspecified variables are frozen. This completes the first stage of the algorithm.

At the second stage of the algorithm, we will compute a satisfying assignment to the variables using brute force. This would be done by taking the surviving formula I' and breaking it into fragments, so that each fragment does not share any variable with any other fragment (naively, it might be that all of I' is one fragment). We can find a satisfying assignment to each fragment separately, and if each such fragment is "small" the resulting algorithm would be "fast".

We need to show that I' has a satisfying assignment and that the fragments are indeed small.

12.2.1.1 Analysis

A clause had ***survived*** if it is not satisfied by the variables fixed in the first stage. Note, that a clause that survived must have a dangerous clause as a neighbor in the dependency graph G. Not that I', the instance remaining from I after the first stage, has at least $k/2$ unspecified variables in each clause. Furthermore, every clause of I' has at most $d = k2^{k/50}$ neighbors in G', where G' is the dependency graph for I'. It follows, that again, we can apply Lovász local lemma to conclude that I' has a satisfying assignment.

Definition 12.2.2. Two connected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, where $V_1, V_2 \subseteq \{1, \ldots, n\}$ are ***unique*** if $V_1 \neq V_2$.

Lemma 12.2.3. *Let G be a graph with degree at most d and with n vertices. Then, the number of unique subgraphs of G having r vertices is at most nd^{2r}.*

Proof: Consider a unique subgraph $\widehat{G}$ of G, which by definition is connected. Let H be a connected subtree of G spanning $\widehat{G}$. Duplicate every edge of H, and let H' denote the resulting graph. Clearly, H' is Eulerian, and as such posses a Eulerian path π of length at most $2(r-1)$, which can be specified, by picking a starting vertex v, and writing down for the i-th vertex of π which of the d possible neighbors, is the next vertex in π. Thus, there are st most $nd^{2(r-1)}$ ways of specifying π, and thus, there are at most $nd^{2(r-1)}$ unique subgraphs in G of size r. ■

Lemma 12.2.4. *With probability* $1 - o(1)$*, all connected components of* G' *have size at most* $O(\log m)$*, where* G' *denote the dependency graph for* I'*.*

Proof: Let G_4 be a graph formed from G by connecting any pair of vertices of G of distance *exactly* 4 from each other. The degree of a vertex of G_4 is at most $O(d^4)$.

Let U be a set of r vertices of G, such that every pair is in distance at least 4 from each other in G. We are interested in bounding the probability that all the clauses of U survive the first stage.

The probability of a clause to be dangerous is at most $2^{-k/2}$, as we assign (random) values to half of the variables of this clause. Now, a clause survive only if it is dangerous or one of its neighbors is dangerous. Thus, the probability that a clause survive is bounded by $2^{-k/2}(d+1)$.

Furthermore, the survival of two clauses $\mathcal{E}_i$ and $\mathcal{E}_j$ in U is an independent event, as no *neighbor* of $\mathcal{E}_i$ shares a variable with a neighbor of $\mathcal{E}_j$ (because of the distance 4 requirement). We conclude, that the probability that all the vertices of U to appear in G' is bounded by

$$\left(2^{-k/2}(d+1)\right)^r.$$

In fact, we are interested in sets U that induce a connected subgraphs of G_4. The number of unique such sets of size r is bounded by the number of unique subgraphs of G_4 of size r, which is bounded by md^{8r}, by Lemma 12.2.3. Thus, the probability of any connected subgraph of G_4 of size $r = \log_2 m$ to survive in G' is smaller than

$$md^{8r}\left(2^{-k/2}(d+1)\right)^r = m\left(k2^{k/50}\right)^{8r}\left(2^{-k/2}(k2^{k/50}+1)\right)^r \leq m2^{kr/5}\cdot 2^{-kr/4} = m2^{-kr/20} = o(1),$$

since $k \geq 50$. (Here, a subgraph survive of G_4 survive, if all its vertices appear in G'.) Note, however, that if a connected component of G' has more than L vertices, than there must be a connected component having L/d^3 vertices in G_4 that had survived in G'. We conclude, that with probability $o(1)$, no connected component of G' has more than $O(d^3 \log m) = O(\log m)$ vertices (note, that we consider k to be a constant, and thus, also d). ∎

Thus, after the first stage, we are left with fragments of $(k/2)$-SAT, where every fragment has size at most $O(\log m)$, and thus having at most $O(\log m)$ variables. Thus, we can by brute force find the satisfying assignment to each such fragment in time polynomial in m. We conclude:

Theorem 12.2.5. *The above algorithm finds a satisfying truth assignment for any instance of k-SAT containing m clauses, which each variable is contained in at most* $2^{k/50}$ *clauses, in expected time polynomial in m.*

Chapter 13

The Probabilistic Method IV

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> Once I sat on the steps by a gate of David's Tower, I placed my two heavy baskets at my side. A group of tourists was standing around their guide and I became their target marker. "You see that man with the baskets? Just right of his head there's an arch from the Roman period. Just right of his head." "But he's moving, he's moving!" I said to myself: redemption will come only if their guide tells them, "You see that arch from the Roman period? It's not important: but next to it, left and down a bit, there sits a man who's bought fruit and vegetables for his family."
>
> – — Yehuda Amichai, Tourists .

13.1 The Method of Conditional Probabilities

In previous lectures, we encountered the following problem.

Problem 13.1.1 (Set Balancing). Given a binary matrix A of size $n \times n$, find a vector $\vec{v} \in \{-1, +1\}^n$, such that $\left\|A\vec{v}\right\|_\infty$ is minimized.

Using random assignment and the Chernoff inequality, we showed that there exists $\vec{v}$, such that $\left\|A\vec{v}\right\|_\infty \leq 4\sqrt{n \ln n}$. Can we derandomize this algorithm? Namely, can we come up with an efficient *deterministic* algorithm that has low discrepancy?

To derandomize our algorithm, construct a computation tree of depth n, where in the ith level we expose the ith coordinate of $\vec{v}$. This tree T has depth n. The root represents all possible random choices, while a node at depth i, represents all computations when the first i bits are fixed. For a node $v \in T$, let $P(v)$ be the probability that a random computation starting from v succeeds. Let v_l and v_r be the two children of v. Clearly, $P(v) = (P(v_l) + P(v_r))/2$. In particular, $\max(P(v_l), P(v_r)) \geq P(v)$. Thus, if we could compute $P(\cdot)$ quickly (and deterministically), then we could derandomize the algorithm.

Let C_m^+ be the bad event that $r_m \cdot \vec{v} > 4\sqrt{n \log n}$, where r_m is the mth row of A. Similarly, C_m^- is the bad event that $r_m \cdot \vec{v} < -4\sqrt{n \log n}$, and let $C_m = C_m^+ \cup C_m^-$. Consider the probability, $\mathbf{Pr}\left[C_m^+ \,\middle|\, \vec{v}_1, \ldots, \vec{v}_k\right]$ (namely, the first k coordinates of $\vec{v}$ are specified). Let $r_m = (\alpha_1, \ldots, \alpha_n)$. We

have that

$$\begin{aligned}\mathbf{Pr}\left[C_m^+ \;\middle|\; \vec{v}_1,\ldots,\vec{v}_k\right] &= \mathbf{Pr}\left[\sum_{i=k+1}^{n} \vec{v}_i\alpha_i > 4\sqrt{n\log n} - \sum_{i=1}^{k}\vec{v}_i\alpha_i\right] \\ &= \mathbf{Pr}\left[\sum_{i\geq k+1,\alpha_i\neq 0} \vec{v}_i\alpha_i > L\right] = \mathbf{Pr}\left[\sum_{i\geq k+1,\alpha_i=1} \vec{v}_i > L\right],\end{aligned}$$

where $L = 4\sqrt{n\log n} - \sum_{i=1}^{k}\vec{v}_i\alpha_i$ is a known quantity (since $\vec{v}_1,\ldots,\vec{v}_k$ are known). Let $V = \sum_{i\geq k+1,\alpha_i=1} 1$. We have,

$$\mathbf{Pr}\left[C_i^+ \;\middle|\; \vec{v}_1,\ldots,\vec{v}_k\right] = \mathbf{Pr}\left[\sum_{\substack{i\geq k+1\\ \alpha_i=1}} (\vec{v}_i+1) > L+V\right] = \mathbf{Pr}\left[\sum_{\substack{i\geq k+1\\ \alpha_i=1}} \frac{\vec{v}_i+1}{2} > \frac{L+V}{2}\right],$$

The last probability, is the probability that in V flips of a fair coin we will get more than $(L+V)/2$ heads. Thus,

$$P_m^+ = \mathbf{Pr}\left[C_m^+ \;\middle|\; \vec{v}_1,\ldots,\vec{v}_k\right] = \sum_{i=\lceil (L+V)/2\rceil}^{V}\binom{V}{i}\frac{1}{2^n} = \frac{1}{2^n}\left(\sum_{i=\lceil (L+V)/2\rceil}^{V}\binom{V}{i}\right).$$

This implies, that we can compute P_m^+ in polynomial time! Indeed, we are adding $V \leq n$ numbers, each one of them is a binomial coefficient that has polynomial size representation in n, and can be computed in polynomial time (why?). One can define in similar fashion P_m^-, and let $P_m = P_m^+ + P_m^-$. Clearly, P_m can be computed in polynomial time, by applying a similar argument to the computation of $P_m^- = \mathbf{Pr}\left[C_m^- \;\middle|\; \vec{v}_1,\ldots,\vec{v}_k\right]$.

For a node $v \in T$, let $\vec{v}_v$ denote the portion of $\vec{v}$ that was fixed when traversing from the root of T to v. Let $P(v) = \sum_{m=1}^{n}\mathbf{Pr}\left[C_m \;\middle|\; \vec{v}_v\right]$. By the above discussion $P(v)$ can be computed in polynomial time. Furthermore, we know, by the previous result on set balancing that $P(r) < 1$ (thats was the bound used to show that there exist a good assignment).

As before, for any $v \in T$, we have $P(v) \geq \min(P(v_l), P(v_r))$. Thus, we have a polynomial *deterministic* algorithm for computing a set balancing with discrepancy smaller than $4\sqrt{n\log n}$. Indeed, set $v = root(T)$. And start traversing down the tree. At each stage, compute $P(v_l)$ and $P(v_r)$ (in polynomial time), and set v to the child with lower value of $P(\cdot)$. Clearly, after n steps, we reach a leaf, that corresponds to a vector $\vec{v}'$ such that $\left\|A\vec{v}'\right\|_\infty \leq 4\sqrt{n\log n}$.

Theorem 13.1.2. *Using the method of conditional probabilities, one can compute in polynomial time in n, a vector $\vec{v} \in \{-1,1\}^n$, such that $\left\|A\vec{v}\right\|_\infty \leq 4\sqrt{n\log n}$.*

Note, that this method might fail to find the best assignment.

13.2 A Very Short Excursion into Combinatorics using the Probabilistic Method

In this section, we provide some additional examples of the Probabilistic Method to prove some results in combinatorics and discrete geometry. While the results are not directly related to our main course, their beauty, hopefully, will speak for itself.

13.2.1 High Girth and High Chromatic Number

Definition 13.2.1. For a graph G, let $\alpha(G)$ be the cardinality of the largest independent set in G, $\chi(G)$ denote the chromatic number of G, and let $\text{girth}(G)$ denote the length of the shortest circle in G.

Theorem 13.2.2. *For all K, L there exists a graph G with* $\text{girth}(G) > L$ *and* $\chi(G) > K$.

Proof: Fix $\mu < 1/L$, and let $G \approx G(n,p)$ with $p = n^{\mu-1}$; namely, G is a random graph on n vertices chosen by picking each pair of vertices to be an edge in G, randomly and independently with probability p. Let X be the number of cycles of size at most L. Then

$$\mathbf{E}[X] = \sum_{i=3}^{L} \frac{n!}{(n-i)!} \cdot \frac{1}{2i} \cdot p^i \leq \sum_{i=3}^{L} \frac{n^i}{2i} \cdot \left(n^{\mu-1}\right)^i \leq \sum_{i=3}^{L} \frac{n^{\mu i}}{2i} = o(n),$$

as $\mu L < 1$, and since the number of different sequence of i vertices is $\frac{n!}{(n-i)!}$, and every cycle is being counted in this sequence $2i$ times.

In particular, $\mathbf{Pr}[X \geq n/2] = o(1)$.

Let $x = \left\lceil \frac{3}{p} \ln n \right\rceil + 1$. We remind the reader that $\alpha(G)$ denotes the size of the largest independent set in G. We have that

$$\begin{aligned} \mathbf{Pr}[\alpha(G) \geq x] &\leq \binom{n}{x}(1-p)^{\binom{x}{2}} < \left(n \exp\left(-\frac{p(x-1)}{2}\right)\right)^x < \left(n \exp\left(-\frac{3}{2}\ln n\right)\right)^x \\ &< (o(1))^x = o(1). \end{aligned}$$

Let n be sufficiently large so that both these events have probabilist less than $1/2$. Then there is a specific G with less than $n/2$ cycles of length at most L and with $\alpha(G) < 3n^{1-\mu}\ln n + 1$.

Remove from G a vertex from each cycle of length at most L. This gives a graph G^* with at least $n/2$ vertices. G^* has girth greater than L and $\alpha(G^*) \leq \alpha(G)$ (any independent set in G^* is also an independent set in G). Thus

$$\chi(G^*) \geq \frac{|V(G^*)|}{\alpha(G^*)} \geq \frac{n/2}{3n^{1-\mu}\ln n} \geq \frac{n^\mu}{12 \ln n}.$$

To complete the proof, let n be sufficiently large so that this is greater than K. ∎

13.2.2 Crossing Numbers and Incidences

The following problem has a long and very painful history. It is truly amazing that it can be solved by such a short and elegant proof.

And ***embedding*** of a graph $G = (V, E)$ in the plane is a planar representation of it, where each vertex is represented by a point in the plane, and each edge uv is represented by a curve connecting the points corresponding to the vertices u and v. The ***crossing number*** of such an embedding is the number of pairs of intersecting curves that correspond to pairs of edges with no common endpoints. The ***crossing number*** $\text{cr}(G)$ of G is the minimum possible crossing number in an embedding of it in the plane.

Theorem 13.2.3. *The crossing number of any simple graph $G = (V, E)$ with $|E| \geq 4|V|$ is $\geq \frac{|E|^3}{64|V|^2}$.*

Proof: By Euler's formula any simple planar graph with n vertices has at most $3n - 6$ edges. (Indeed, $f - e + v = 2$ in the case with maximum number of edges, we have that every face, has 3 edges around it. Namely, $3f = 2e$. Thus, $(2/3)e - e + v = 2$ in this case. Namely, $e = 3v - 6$.) This implies that the crossing number of any simple graph with n vertices and m edges is at least $m - 3n + 6 > m - 3n$. Let $G = (V, E)$ be a graph with $|E| \geq 4|V|$ embedded in the plane with $t = \mathrm{cr}(G)$ crossings. Let H be the random induced subgraph of G obtained by picking each vertex of G randomly and independently, to be a vertex of H with probabilistic p (where P will be specified shortly). The expected number of vertices of H is $p|V|$, the expected number of its edges is $p^2|E|$, and the expected number of crossings in the given embedding is $p^4 t$, implying that the expected value of its crossing number is at most $p^4 t$. Therefore, we have $p^4 t \geq p^2|E| - 3p|V|$, implying that

$$\mathrm{cr}(G) \geq \frac{|E|}{p^2} - \frac{3|V|}{p^3},$$

let $p = 4|V|/|E| < 1$, and we have $\mathrm{cr}(G) \geq (1/16 - 3/64)|E|^3/|V|^2 = |E|^3/(64|V|^2)$. ∎

Theorem 13.2.4. *Let P be a set of n distinct points in the plane, and let L be a set of m distinct lines. Then, the number of incidences between the points of P and the lines of L (that is, the number of pairs (p, ℓ) with $p \in P$, $\ell \in L$, and $p \in \ell$) is at most $c(m^{2/3}n^{2/3} + m + n)$, for some absolute constant c.*

Proof: Let I denote the number of such incidences. Let $G = (V, E)$ be the graph whose vertices are all the points of P, where two are adjacent if and only if they are consecutive points of P on some line in L. Clearly $|V| = n$, and $|E| = I - m$. Note that G is already given embedded in the plane, where the edges are presented by segments of the corresponding lines of L.

Either, we can not apply Theorem 13.2.3, implying that $I - m = |E| < 4|V| = 4n$. Namely, $I \leq m + 4n$. Or alliteratively,

$$\frac{(I-m)^3}{(64n^2)} \leq \mathrm{cr}(G) \leq \binom{m}{2} \leq \frac{m^2}{2}.$$

Implying that $I \leq (32)^{1/3}m^{2/3}n^{2/3} + m$. In both cases, $I \leq 4(m^{2/3}n^{2/3} + m + n)$. ∎

This technique has interesting and surprising results, as the following theorem shows.

Theorem 13.2.5. *For any three sets A, B and C of s real numbers each,*

$$|A \cdot B + C| = \left|\left\{ab + c \;\middle|\; a \in A, b \in B, mc \in C\right\}\right| \geq \Omega\left(s^{3/2}\right).$$

Proof: Let $R = A \cdot B + C$, $|R| = r$ and define $P = \left\{(a, t) \;\middle|\; a \in A, t \in R\right\}$, and $L = \left\{y = bx + c \;\middle|\; b \in B, c \in C\right\}$.

Clearly $n = |P| = sr$, and $m = |L| = s^2$. Furthermore, a line $y = bx + c$ of L is incident with s points of R, namely with $\left\{(a,t) \;\middle|\; a \in A, t = ab + c\right\}$. Thus, the overall number of incidences is at at least s^3. By Theorem 13.2.4, we have

$$s^3 \leq 4(m^{2/3}n^{2/3} + m + n) = 4\left(\left(s^2\right)^{2/3}(sr)^{2/3} + s^2 + sr\right) = 4\left(s^2r^{2/3} + s^2 + sr\right).$$

For $r < s^3$, we have that $sr \leq s^2r^{2/3}$. Thus, for $r < s^3$, we have $s^3 \leq 12s^2r^{2/3}$, implying that $s^{3/2} \leq 12r$. Namely, $|R| = \Omega(s^{3/2})$, as claimed. ■

Among other things, the crossing number technique implies a better bounds for k-sets in the plane than what was previously known. The k-set problem had attracted a lot of research, and remains till this day one of the major open problems in discrete geometry.

Chapter 14

Random Walks I

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> "A drunk man will find his way home; a drunk bird may wander forever."
> – – Anonymous.

14.1 Definitions

Let $G = G(V, E)$ be an undirected connected graph. For $v \in V$, let $\Gamma(v)$ denote the set of neighbors of v in G; that is, $\Gamma(v) = \left\{ u \;\middle|\; vu \in E(G) \right\}$. A ***random walk*** on G is the following process: Starting from a vertex v_0, we randomly choose one of the neighbors of v_0, and set it to be v_1. We continue in this fashion, in the ith step choosing v_i, such that $v_i \in \Gamma(v_{i-1})$. It would be interesting to investigate the random walk process. Questions of interest include:

(A) How long does it take to arrive from a vertex v to a vertex u in G?
(B) How long does it take to visit all the vertices in the graph.
(C) If we start from an arbitrary vertex v_0, how long the random walk has to be such that the location of the random walk in the ith step is uniformly (or near uniformly) distributed on $V(G)$?

Example 14.1.1. In the complete graph K_n, visiting all the vertices takes in expectation $O(n \log n)$ time, as this is the coupon collector problem with $n - 1$ coupons. Indeed, the probability we did not visit a specific vertex v by the ith step of the random walk is $\leq (1 - 1/n)^{i-1} \leq e^{-(i-1)/n} \leq 1/n^{10}$, for $i = \Omega(n \log n)$. As such, with high probability, the random walk visited all the vertex of K_n. Similarly, arriving from u to v, takes in expectation $n - 1$ steps of a random walk, as the probability of visiting v at every step of the walk is $p = 1/(n - 1)$, and the length of the walk till we visit v is a geometric random variable with expectation $1/p$.

14.1.1 Walking on grids and lines

Lemma 14.1.2 (Stirling's formula.). *For any integer $n \geq 1$, it holds $n! \approx \sqrt{2\pi n}(n/e)^n$.*

Lemma 14.1.3. *Consider the infinite random walk on the integer line, starting from* 0*. Here, the vertices are the integer numbers, and from a vertex k, one walks with probability* $1/2$ *either to* $k-1$ *or* $k+1$*. The expected number of times that such a walk visits* 0 *is unbounded.*

Proof: The probability that in the $2i$th step we visit 0 is $\frac{1}{2^{2i}}\binom{2i}{i}$, As such, the expected number of times we visit the origin is

$$\sum_{i=1}^{\infty} \frac{1}{2^{2i}}\binom{2i}{i} \geq \sum_{i=1}^{\infty} \frac{1}{2\sqrt{i}} = \infty,$$

since $\dfrac{2^{2i}}{2\sqrt{i}} \leq \binom{2i}{i} \leq \dfrac{2^{2i}}{\sqrt{2i}}$ [MN98, p. 84]. This can also be verified using the Stirling formula, and the resulting sequence diverges. ∎

A random walk on the integer grid $\mathbb{Z}^d$, starts from a point of this integer grid, and at each step if it is at point $(i_1, i_2, \ldots, i_d)$, it chooses a coordinate and either increases it by one, or decreases it by one, with equal probability.

Lemma 14.1.4. *Consider the infinite random walk on the two dimensional integer grid* $\mathbb{Z}^2$*, starting from* $(0,0)$*. The expected number of times that such a walk visits the origin is unbounded.*

Proof: Rotate the grid by 45 degrees, and consider the two new axises X' and Y'. Let x_i be the projection of the location of the ith step of the random walk on the X'-axis, and define y_i in a similar fashion. Clearly, x_i are of the form $j/\sqrt{2}$, where j is an integer. By scaling by a factor of $\sqrt{2}$, consider the resulting random walks $x'_i = \sqrt{2}x_i$ and $y'_i = \sqrt{2}y_i$. Clearly, x_i and y_i are random walks on the integer grid, and furthermore, they are *independent*. As such, the probability that we visit the origin at the $2i$th step is $\mathbf{Pr}\left[x'_{2i} = 0 \cap y'_{2i} = 0\right] = \mathbf{Pr}\left[x'_{2i} = 0\right]^2 = \left(\frac{1}{2^{2i}}\binom{2i}{i}\right)^2 \geq 1/4i$. We conclude, that the infinite random walk on the grid $\mathbb{Z}^2$ visits the origin in expectation

$$\sum_{i=0}^{\infty} \mathbf{Pr}[x'_i = 0 \cap y'_i = 0] \geq \sum_{i=0}^{\infty} \frac{1}{4i} = \infty,$$

as this sequence diverges. ∎

In the following, let $\binom{i}{a\ b\ c} = \dfrac{i!}{a!\,b!\,c!}$.

Lemma 14.1.5. *Consider the infinite random walk on the three dimensional integer grid* $\mathbb{Z}^3$*, starting from* $(0,0,0)$*. The expected number of times that such a walk visits the origin is bounded.*

Proof: The probability of a neighbor of a point (x, y, z) to be the next point in the walk is $1/6$. Assume that we performed a walk for $2i$ steps, and decided to perform $2a$ steps parallel to the x-axis, $2b$ steps parallel to the y-axis, and $2c$ steps parallel to the z-axis, where $a + b + c = i$. Furthermore, the walk on each dimension is balanced, that is we perform a steps to the left on the x-axis, and a steps to the right on the x-axis. Clearly, this corresponds to the only walks in $2i$ steps that arrives to the origin.

Next, the number of different ways we can perform such a walk is $\frac{(2i)!}{a!a!b!b!c!c!}$, and the probability to perform such a walk, summing over all possible values of a, b and c, is

$$\begin{aligned}
\alpha_i &= \sum_{\substack{a+b+c=i \\ a,b,c\geq 0}} \frac{(2i)!}{a!a!b!b!c!c!}\frac{1}{6^{2i}} \\
&= \binom{2i}{i}\frac{1}{2^{2i}} \sum_{\substack{a+b+c=i \\ a,b,c\geq 0}} \left(\frac{i!}{a!\,b!\,c!}\right)^2 \left(\frac{1}{3}\right)^{2i} \\
&= \binom{2i}{i}\frac{1}{2^{2i}} \sum_{\substack{a+b+c=i \\ a,b,c\geq 0}} \left(\binom{i}{a\ b\ c}\left(\frac{1}{3}\right)^i\right)^2
\end{aligned}$$

Consider the case where $i = 3m$. We have that $\binom{i}{a\ b\ c} \leq \binom{i}{m\ m\ m}$. As such,

$$\begin{aligned}
\alpha_i &\leq \binom{2i}{i}\frac{1}{2^{2i}}\left(\frac{1}{3}\right)^i \binom{i}{m\ m\ m} \sum_{\substack{a+b+c=i \\ a,b,c\geq 0}} \binom{i}{a\ b\ c}\left(\frac{1}{3}\right)^i \\
&= \binom{2i}{i}\frac{1}{2^{2i}}\left(\frac{1}{3}\right)^i \binom{i}{m\ m\ m}.
\end{aligned}$$

By the Stirling formula, we have

$$\binom{i}{m\ m\ m} \approx \frac{\sqrt{2\pi i}(i/e)^i}{\left(\sqrt{2\pi i/3}\left(\frac{i}{3e}\right)^{i/3}\right)^3} = c\frac{3^i}{i},$$

for some constant c. As such,

$$\alpha_i = O\left(\frac{1}{\sqrt{i}}\left(\frac{1}{3}\right)^i \frac{3^i}{i}\right) = O\left(\frac{1}{i^{3/2}}\right).$$

Thus,

$$\sum_{m=1}^{\infty} \alpha_{6m} = \sum_i O\left(\frac{1}{i^{3/2}}\right) = O(1).$$

Finally, observe that $\alpha_{6m} \geq (1/6)^2\alpha_{6m-2}$ and $\alpha_{6m} \geq (1/6)^4\alpha_{6m-4}$. Thus,

$$\sum_{m=1}^{\infty} \alpha_m = O(1).$$

■

Notes

The presentation here follows [Nor98].

Chapter 15

Random Walks II

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> "Then you must begin a reading program immediately so that you man understand the crises of our age," Ignatius said solemnly. "Begin with the late Romans, including Boethius, of course. Then you should dip rather extensively into early Medieval. You may skip the Renaissance and the Enlightenment. That is mostly dangerous propaganda. Now, that I think about of it, you had better skip the Romantics and the Victorians, too. For the contemporary period, you should study some selected comic books."
>
> "You're fantastic."
>
> "I recommend Batman especially, for he tends to transcend the abysmal society in which he's found himself. His morality is rather rigid, also. I rather respect Batman."
>
> – John Kennedy Toole, A confederacy of Dunces.

15.1 The 2SAT example

Let $G = G(V, E)$ be a undirected connected graph. For $v \in V$, let $\Gamma(v)$ denote the neighbors of v in G. A random walk on G is the following process: Starting from a vertex v_0, we randomly choose one of the neighbors of v_0, and set it to be v_1. We continue in this fashion, such that $v_i \in \Gamma(v_{i-1})$. It would be interesting to investigate the process of the random walk. For example, questions like: (i) how long does it take to arrive from a vertex v to a vertex u in G? and (ii) how long does it take to visit all the vertices in the graph.

15.1.1 Solving 2SAT

Consider a 2SAT formula F with m clauses defined over n variables. Start from an arbitrary assignment to the variables, and consider a non-satisfied clause in F. Randomly pick one of the clause variables, and change its value. Repeat this till you arrive to a satisfying assignment.

Consider the random variable X_i, which is the number of variables assigned the correct value (according to the satisfying assignment) in the current assignment. Clearly, with probability (at least) half $X_i = X_{i-1} + 1$.

Thus, we can think about this algorithm as performing a random walk on the numbers $0, 1, \ldots, n$, where at each step, we go to the right probability at least half. The question is, how long does it take to arrive to n in such a settings.

Theorem 15.1.1. *The expected number of steps to arrive to a satisfying assignment is $O(n^2)$.*

Proof: Consider the random walk on the integer line, starting from zero, where we go to the left with probability $1/2$, and to the right probability $1/2$. Let Y_i be the location of the walk at the i step. Clearly, $\mathbf{E}[Y_i] \geq \mathbf{E}[X_i]$. In fact, by defining the random walk on the integer line more carefully, one can ensure that $Y_i \leq X_i$. Thus, the expected number of steps till Y_i is equal to n is an upper bound on the required quantity.

To this end, observe that the probability that in the ith step we have $Y_i \geq n$ is

$$\sum_{m=n/2}^{i} \frac{1}{2^i} \binom{i}{i/2+m} > 1/3,$$

for $i > \mu = c'n^2$, where c' is a large enough constant. To see that, observe that if we get $i/2+m$ times $+1$, and $i-(i/2+m) = i/2-m$ times -1, then we have that $Y_i = (i/2+m) - ((i/2)-m) = 2m \geq n$.

Next, if X_i fails to arrive to n at the first μ steps, we will reset $Y_\mu = X_\mu$ and continue the random walk, repeating this process as many phases as necessary. The probability that the number of phases exceeds i is $\leq (2/3)^i$. As such, the expected number of steps in the walk is at most

$$\sum_i c'n^2 i \left(\frac{2}{3}\right)^i = O(n^2),$$

as claimed. ∎

15.2 Markov Chains

Let S denote a state space, which is either finite or countable. A ***Markov chain*** is at one state at any given time. There is a ***transition probability*** $\mathtt{P}_{ij}$, which is the probability to move to the state j, if the Markov chain is currently at state i. As such, $\sum_j \mathtt{P}_{ij} = 1$ and $\forall i, j, 0 \leq \mathtt{P}_{ij} \leq 1$. The matrix $\mathbf{P} = \left\{\mathtt{P}_{ij}\right\}_{ij}$ is the ***transition probabilities matrix***.

The Markov chain start at an initial state X_0, and at each point in time moves according to the transition probabilities. This form a sequence of states $\{X_t\}$. We have a distribution over those sequences. Such a sequence would be referred to as a ***history***.

Similar to Martingales, the behavior of a Markov chain in the future, depends only on its location X_t at time t, and does not depends on the earlier stages that the Markov chain went through. This is the ***memorylessness property*** of the Markov chain, and it follows as $\mathtt{P}_{ij}$ is independent of time. Formally, the memorylessness property is

$$\mathbf{Pr}\Big[X_{t+1} = j \;\Big|\; X_0 = i_0, X_1 = i_1, \ldots, X_{t-1} = i_{t-1}, X_t = i\Big] = \mathbf{Pr}\Big[X_{t+1} = j \;\Big|\; X_t = i\Big] = \mathtt{P}_{ij}.$$

The initial state of the Markov chain might also be chosen randomly.

For states $i, j \in \mathsf{S}$, the *t-step transition probability* is $\mathtt{P}_{ij}^{(t)} = \mathbf{Pr}\Big[X_t = j \;\Big|\; X_0 = i\Big]$. The probability that we visit j for the first time, starting from i after t steps, is denoted by

$$\mathtt{r}_{ij}^{(t)} = \mathbf{Pr}\Big[X_t = j \text{ and } X_1 \neq j, X_2 \neq j, \ldots, X_{t-1} \neq j \;\Big|\; X_0 = i\Big].$$

Let $\mathtt{f}_{ij} = \sum_{t>0} \mathtt{r}_{ij}^{(t)}$ denote the probability that the Markov chain visits state j, at any point in time, starting from state i. The expected number of steps to arrive to state j starting from i is

$$\mathtt{h}_{ij} = \sum_{t>0} t \cdot \mathtt{r}_{ij}^{(t)}.$$

Of course, if $\mathtt{f}_{ij} < 1$, then there is a positive probability that the Markov chain never arrives to j, and as such $\mathtt{h}_{ij} = \infty$ in this case.

Definition 15.2.1. A state $i \in \mathsf{S}$ for which $\mathtt{f}_{ii} < 1$ (i.e., the chain has positive probability of never visiting i again), is a ***transient*** state. If $\mathtt{f}_{ii} = 1$ then the state is ***persistent***.

If a state is persistent, but $\mathtt{h}_{ii} = \infty$ are called ***null persistent***. If the state i is persistent and $\mathtt{h}_{ii} \neq \infty$ then it is ***non null persistent***.

Example 15.2.2. Consider the state 0 in the random walk on the integers. We already know that in expectation the random walk visits the origin infinite number of times. Let figure out the probability $\mathtt{r}_{00}^{(2n)}$. To this end, consider a walk $X_0, X_1, \ldots, X_{2n}$ that starts at 0 and return to 0 only in the $2n$ step. Let $S_i = X_i - X_{i-1}$, for all i. Clearly, we have $S_i \in -1, +1$ (i.e., move left or move right). Assume the walk starts by $S_1 = +1$ (the case -1 is handled similarly). Clearly, the walk $S_2, \ldots, S_{2n-1}$ must be prefix balanced; that is, the number of 1s is always bigger (or equal) for any prefix of this sequence.

Strings with this property are known as ***Dyck words***, and the number of such words of length $2m$ is the ***Catalan number*** $C_m = \frac{1}{m+1}\binom{2m}{m}$. As such, the probability of the random walk to visit 0 for the first time (starting from 0 after $2n$ steps, is

$$\mathtt{r}_{00}^{(2n)} = 2\frac{1}{n}\binom{2n-2}{n-1}\frac{1}{2^{2n}} = \Theta\left(\frac{1}{n} \cdot \frac{1}{\sqrt{n}}\right) = \Theta\left(\frac{1}{n^{3/2}}\right).$$

(the 2 here is because the other option is that the sequence starts with -1), using the fact that $\binom{2n}{n}\Theta\left(2^{2n}/\sqrt{n}\right)$.

It is not hard to show that $\mathtt{f}_{00} = 1$ (this requires a clever trick). On the other hand, we have that

$$\mathtt{h}_{00} = \sum_{t>0} t \cdot \mathtt{r}_{00}^{(t)} \geq \sum_{n=1}^{\infty} 2n\mathtt{r}_{00}^{(2n)} = \sum_{n=1}^{\infty} \Theta\left(1/\sqrt{n}\right) = \infty.$$

Namely, 0 (and in fact all integers) are null persistent.

In finite Markov chains, there are no null persistent states (this requires a proof, which is left as exercise). There is a natural directed graph associated with a Markov chain. The states are the vertices, and the transition probability $\mathtt{P}_{ij}$ is the weight assigned to the edge $(i \to j)$. Note that we include only edges with $\mathtt{P}_{ij} > 0$.

Definition 15.2.3. A ***strong component*** of a directed graph G is a maximal subgraph C of G such that for any pair of vertices i and j in the vertex set of C, there is a directed path from i to j, as well as a directed path from j to i.

Definition 15.2.4. A strong component C is said to be a ***final strong component*** if there is no edge going from a vertex in C to a vertex not in C.

In a finite Markov chain, there is positive probability to arrive from any vertex on C to any other vertex of C in a finite number of steps. If C is a final strong component, then probability is 1, since the Markov chain can never leave C once it enters it. It follows that a state is persistent if and only if it lies in a final strong component.

Definition 15.2.5. A Markov chain is ***irreducible*** when its underlying graph consists of single strong component.

Clearly, if a Markov chain is irreducible, then all states are persistent.

Definition 15.2.6. Let $\mathbf{q}^{(t)} = \left(q_1^{(t)}, q_2^{(t)}, \ldots, q_n^{(t)}\right)$ be the *state probability vector* (also called the distribution of the chain at time t), to be the row vector whose ith component is the probability that the chain is in state i at time t.

The key observation is that

$$\mathbf{q}^{(t)} = \mathbf{q}^{(t-1)}\mathbf{P} = \mathbf{q}^{(0)}\mathbf{P}^t.$$

Namely, a Markov chain is fully defined by $\mathbf{q}^{(0)}$ and $\mathbf{P}$.

Definition 15.2.7. A ***stationary distribution*** for a Markov chain with the transition matrix $\mathbf{P}$ is a probability distribution π such that $\pi = \pi\mathbf{P}$.

In general, stationary distribution does not necessarily exist. We will mostly be interested in Markov chains that have stationary distribution. Intuitively it is clear, that if a stationary distribution exists, then the Markov chain, given enough time, will converge to the stationary distribution.

Definition 15.2.8. The ***periodicity*** of a state i is the maximum integer T for which there exists an initial distribution $\mathbf{q}^{(0)}$ and positive integer a such that, for all t if at time t we have $q_i^{(t)} > 0$ then t belongs to the arithmetic progression $\left\{a + ti \;\middle|\; i \geq 0\right\}$. A state is said to be ***periodic*** if it has periodicity greater than 1, and is ***aperiodic*** otherwise. A Markov chain in which every state is aperiodic is ***aperiodic***.

A neat trick that forces a Markov chain to be aperiodic, is to shrink all the probabilities by a factor of 2, and make every state to have a transition probability to itself equal to $1/2$. Clearly, the resulting Markov chain is aperiodic.

Definition 15.2.9. An ***ergodic*** state is aperiodic and (non-null) persistent.
An ***ergodic*** Markov chain is one in which all states are ergodic.

The following theorem is the fundamental fact about Markov chains that we will need. The interested reader, should check the proof in [Nor98].

Theorem 15.2.10 (Fundamental theorem of Markov chains). *Any irreducible, finite, and aperiodic Markov chain has the following properties.*

(i) All states are ergodic.

(ii) There is a unique stationary distribution π such that, for $1 \leq i \leq n$, $\pi_i > 0$.

(iii) *For* $1 \leq i \leq n$, $\mathtt{f}_{ii} = 1$ *and* $\mathtt{h}_{ii} = 1/\pi_i$.

(iv) *Let* $N(i,t)$ *be the number of times the Markov chain visits state* i *in* t *steps. Then*

$$\lim_{t\to\infty} \frac{N(i,t)}{t} = \pi_i.$$

Namely, independent of the starting distribution, the process converges to the stationary distribution.

Chapter 16

Random Walks III

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> "I gave the girl my protection, offering in my equivocal way to be her father. But I came too late, after she had ceased to believe in fathers. I wanted to do what was right, I wanted to make reparation: I will not deny this decent impulse, however mixed with more questionable motives: there must always be a place for penance and reparation. Nevertheless, I should never have allowed the gates of the town to be opened to people who assert that there are higher considerations that those of decency. They exposed her father to her naked and made him gibber with pain, they hurt her and he could not stop them (on a day I spent occupied with the ledgers in my office). Thereafter she was no longer fully human, sister to all of us. Certain sympathies died, certain movements of the heart became no longer possible to her. I too, if I live longer enough in this cell with its ghost not only of the father and the daughter but of the man who even by lamplight did not remove the black discs from his eyes and the subordinate whose work it was to keep the brazier fed, will be touched with the contagion and turned into a create that believes in nothing."
>
> – J. M. Coetzee, Waiting for the Barbarians.

16.1 Random Walks on Graphs

Let $G = (V, E)$ be a connected, non-bipartite, undirected graph, with n vertices. We define the natural Markov chain on G, where the transition probability is

$$\mathtt{P}_{uv} = \begin{cases} \frac{1}{d(u)} & \text{if } uv \in E \\ 0 & \text{otherwise,} \end{cases}$$

where $d(w)$ is the degree of vertex w. Clearly, the resulting Markov chain M_G is irreducible. Note, that the graph must have an odd cycle, and it has a cycle of length 2. Thus, the gcd of the lengths of its cycles is 1. Namely, M_G is aperiodic. Now, by the Fundamental theorem of Markov chains, M_G has a unique stationary distribution π.

Lemma 16.1.1. *For all $v \in V$, $\pi_v = d(v)/2m$.*

Proof: Since π is stationary, and the definition of $\mathtt{P}_{uv}$, we get

$$\pi_v = [\pi \mathbf{P}]_v = \sum_{uv} \pi_u \mathtt{P}_{uv},$$

and this holds for all v. We only need to verify the claimed solution, since there is a unique stationary distribution. Indeed,

$$\frac{d(v)}{2m} = \pi_v = [\pi \mathbf{P}]_v = \sum_{uv} \frac{d(u)}{2m} \frac{1}{d(u)} = \frac{d(v)}{2m},$$

as claimed. ∎

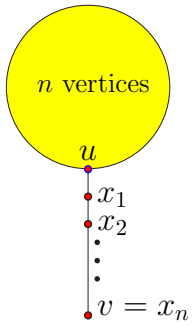
Lemma 16.1.2. *For all $v \in V$, $\mathtt{h}_{vv} = 1/\pi_v = 2m/d(v)$.*

Definition 16.1.3. The ***hitting time*** $\mathtt{h}_{uv}$ is the expected number of steps in a random walk that starts at u and ends upon first reaching v.

The ***commute time*** between u and v is denoted by $\mathbf{CT}_{uv} = \mathtt{h}_{uv} + \mathtt{h}_{vu}$.

Let $\mathcal{C}_u(G)$ denote the expected length of a walk that starts at u and ends upon visiting every vertex in G at least once. The ***cover time*** of G denotes by $\mathcal{C}(G)$ is defined by $\mathcal{C}(G) = \max_u \mathcal{C}_u(G)$.

Example 16.1.4 (Lollipop.). Let L_{2n} be the $2n$-vertex ***lollipop graph***, this graph consists of a clique on n vertices, and a path on the remaining n vertices. There is a vertex u in the clique which is where the path is attached to it. Let v denote the end of the path, see figure on the right.



Taking a random walk from u to v requires in expectation $O(n^2)$ steps, as we already saw in class. This ignores the fact that with probability $(n-1)/n$ we enter the clique K_n. As such, it turns out that $\mathtt{h}_{uv} = \Theta(n^3)$, and $\mathtt{h}_{vu} = \Theta(n^2)$.

Note, that the cover time is not monotone decreasing with the number of edges. Indeed, the path of length n, has cover time $O(n^2)$, but the larger graph L_n has cover time $\Omega(n^3)$.

Example 16.1.5 (More on walking on the Lollipop.). So see why $\mathtt{h}_{uv} = \Theta(n^3)$, number the vertices on the stem $x_1, \ldots, x_n$. Let T_i be the expected time to arrive to the vertex x_i when starting a walk from u. Observe, that surprisingly, $T_1 = \Theta(n^2)$. Indeed, the walk has to visit the vertex u about n times in expectation, till the walk would decide to go to x_1 instead of falling back into the clique. The time between visits to u is in expectation $O(n)$ (assuming the walk is inside the clique).

Now, observe that $T_{2i} = T_i + \Theta(i^2) + \frac{1}{2}T_{2i}$. Indeed, starting with x_i, it takes in expectation $\Theta(i^2)$ steps of the walk to either arrive (with equal probability) at x_{2i} (good), or to get back to u (oopsi). In the later case, the game begins from scratch. As such, we have that

$$T_{2i} = 2T_i + \Theta(i^2) = \cdots = 2iT_1 + \Theta(i^2),$$

assuming i is a power of two (why not?). As such, $T_n = nT_1 + \Theta(n^2)$. Since $T_1 = \Theta(n^2)$, we have that $T_n = \Theta(n^3)$.

Definition 16.1.6. A $n \times n$ matrix M is ***stochastic*** if all its entries are non-negative and for each row i, it holds $\sum_k \mathsf{M}_{ik} = 1$. It is ***doubly stochastic*** if in addition, for any i, it holds $\sum_k \mathsf{M}_{ki} = 1$.

Lemma 16.1.7. *Let MC be a Markov chain, such that transition probability matrix $\mathbf{P}$ is* doubly stochastic. *Then, the distribution $u = (1/n, 1/n, \ldots, 1/n)$ is stationary for MC.*

Proof: $[u\mathbf{P}]_i = \sum_{k=1}^{n} \frac{\mathtt{P}_{ki}}{n} = \frac{1}{n}.$ ■

Lemma 16.1.8. *For any edge* $(u \to v) \in E$, $\mathtt{h}_{uv} + \mathtt{h}_{vu} \leq 2m$.

(Note, that the fact that $(u \to v)$ is an edge in the graph is crucial. Indeed, without it a worst bound holds, see Theorem 16.2.1.)

Proof: Consider a new Markov chain defined by the edges of the graph (where every edge is taken twice as two directed edges), where the current state is the last (directed) edge visited. There are $2m$ edges in the new Markov chain, and the new transition matrix, has $Q_{(u\to v),(v\to w)} = \mathtt{P}_{vw} = \frac{1}{d(v)}$. This matrix is *doubly stochastic*, meaning that not only do the rows sum to one, but the columns sum to one as well. Indeed, for the $(v \to w)$ we have

$$\sum_{x\in V, y\in\Gamma(x)} Q_{(x\to y),(v\to w)} = \sum_{u\in\Gamma(v)} Q_{(u\to v),(v\to w)} = \sum_{u\in\Gamma(v)} \mathtt{P}_{vw} = d(v) \times \frac{1}{d(v)} = 1.$$

Thus, the stationary distribution for this Markov chain is uniform, by Lemma 16.1.7. Namely, the stationary distribution of $e = (u \to v)$ is $\mathtt{h}_{ee} = \pi_e = 1/(2m)$. Thus, the expected time between successive traversals of e is $1/\pi_e = 2m$, by Theorem 15.2.10 (iii).

Consider $\mathtt{h}_{uv} + \mathtt{h}_{vu}$ and interpret this as the time to go from u to v and then return to u. Conditioned on the event that the initial entry into u was via the $(v \to u)$, we conclude that the expected time to go from there to v and then finally use $(v \to u)$ is $2m$. The memorylessness property of a Markov chains now allows us to remove the conditioning: since how we arrived to u is not relevant. Thus, the expected time to travel from u to v and back is at most $2m$. ■

16.2 Electrical networks and random walks

A ***resistive electrical network*** is an undirected graph; each edge has ***branch resistance*** associated with it. The electrical flow is determined by two laws: ***Kirchhoff's law*** (preservation of flow - all the flow coming into a node, leaves it) and ***Ohm's law*** (the voltage across a resistor equals the product of the resistance times the current through it). Explicitly, Ohm's law states

$$\text{voltage} = \text{resistance} * \text{current}.$$

The ***effective resistance*** between nodes u and v is the voltage difference between u and v when one ampere is injected into u and removed from v (or injected into v and removed from u). The effective resistance is always bounded by the branch resistance, but it can be much lower.

Given an undirected graph G, let $\mathcal{N}(G)$ be the electrical network defined over G, associating one ohm resistance on the edges of $\mathcal{N}(G)$.

You might now see the connection between a random walk on a graph and electrical network. Intuitively (used in the most unscientific way possible), the electricity, is made out of electrons each one of them is doing a random walk on the electric network. The resistance of an edge, corresponds to the probability of taking the edge. The higher the resistance, the lower the probability that we will travel on this edge. Thus, if the effective resistance $\mathbf{R}_{uv}$ between u and v is low, then there is a good probability that travel from u to v in a random walk, and $\mathtt{h}_{uv}$ would be small.

Theorem 16.2.1. *For any two vertices u and v in G, the commute time $\mathbf{CT}_{uv} = 2m\mathbf{R}_{uv}$.*

Proof: Let ϕ_{uv} denote the voltage at u in $\mathcal{N}(G)$ with respected to v, where $d(x)$ amperes of current are injected into each node $x \in V$, and $2m$ amperes are removed from v. We claim that

$$\mathtt{h}_{uv} = \phi_{uv}.$$

Note, that the voltage on an edge xy is $\phi_{xy} = \phi_{xv} - \phi_{yv}$. Thus, using Kirchhoff's Law and Ohm's Law, we obtain that

$$x \in V \setminus \{v\} \qquad d(x) = \sum_{w \in \Gamma(x)} \text{current}(xw) = \sum_{w \in \Gamma(x)} \frac{\phi_{xw}}{\text{resistance}(xw)} = \sum_{w \in \Gamma(x)} (\phi_{xv} - \phi_{wv}), \tag{16.1}$$

since the resistance of every edge is 1 ohm. (We also have the "trivial" equality that $\phi_{vv} = 0$.) Furthermore, we have only n variables in this system; that is, for every $x \in V$, we have the variable ϕ_{xv}.

Now, for the random walk interpretation – by the definition of expectation, we have

$$\begin{aligned} x \in V \setminus \{v\} \qquad \mathtt{h}_{xv} = \frac{1}{d(x)} \sum_{w \in \Gamma(x)} (1 + \mathtt{h}_{wv}) \qquad &\iff \qquad d(x)\,\mathtt{h}_{xv} = \sum_{w \in \Gamma(x)} 1 + \sum_{w \in \Gamma(x)} \mathtt{h}_{wv} \\ &\iff \qquad \sum_{w \in \Gamma(x)} 1 = d(x)\,\mathtt{h}_{xv} - \sum_{w \in \Gamma(x)} \mathtt{h}_{wv} = \sum_{w \in \Gamma(x)} (\mathtt{h}_{xv} - \mathtt{h}_{wv}). \end{aligned}$$

Since $d(x) = \sum_{w \in \Gamma(x)} 1$, this is equivalent to

$$x \in V \setminus \{v\} \qquad d(x) = \sum_{w \in \Gamma(x)} (\mathtt{h}_{xv} - \mathtt{h}_{wv}). \tag{16.2}$$

Again, we also have the trivial equality $\mathtt{h}_{vv} = 0$.[①] Note, that this system also has n equalities and n variables.

Eq. (16.1) and Eq. (16.2) show two systems of linear equalities. Furthermore, if we identify $\mathtt{h}_{uv}$ with ϕ_{xv} then they are exactly the same system of equalities. Furthermore, since Eq. (16.1) represents a physical system, we know that it has a unique solution. This implies, that $\phi_{xv} = \mathtt{h}_{xv}$, for all $x \in V$.

Imagine the network where u is injected with $2m$ amperes, and for all nodes w remove $d(w)$ units from w. In this new network, $\mathtt{h}_{vu} = -\phi'_{vu} = \phi'_{uv}$. Now, since flows behaves linearly, we can superimpose them (i.e., add them up). We have that in this new network $2m$ unites are being injected at u, and $2m$ units are being extracted at v, all other nodes the charge cancel itself out. The voltage difference between u and v in the new network is $\widehat{\phi} = \phi_{uv} + \phi'_{uv} = \mathtt{h}_{uv} + \mathtt{h}_{vu} = \mathbf{CT}_{uv}$. Now, in the new network there are $2m$ amperes going from u to v, and by Ohm's law, we have

$$\widehat{\phi} = \text{voltage} = \text{resistance} * \text{current} = 2m\mathbf{R}_{uv},$$

as claimed. ∎

[①] In previous lectures, we interpreted $\mathtt{h}_{vv}$ as the expected length of a walk starting at v and coming back to v.

Example 16.2.2. Recall the lollipop from Exercise 16.1.4 L_n. Let u be the connecting vertex between the clique and the stem (i.e., the path). We inject $d(x)$ units of flow for each vertex x of L_n, and collect $2m$ units at u. Next, let $u = x_0, x_1, \ldots, x_n = v$ be the vertices of the stem. Clearly, there are $2(n-i)-1$ units of electricity flowing on the edge $(x_{i+1} \to x_i)$. Thus, the voltage on this edge is $2(n-i)$, by Ohm's law (every edge has resistance one). The effective resistance from v to u is as such $\Theta(n^2)$, which implies that $\mathtt{h}_{vu} = \Theta(n^2)$.

Similarly, it is easy to show $\mathtt{h}_{uv} = \Theta(n^3)$.

A similar analysis works for the random walk on the integer line in the range 1 to n.

Lemma 16.2.3. *For any n vertex connected graph G, and for all $u, v \in V(G)$, we have $\mathbf{CT}_{uv} < n^3$.*

Proof: The effective resistance between any two nodes in the network is bounded by the length of the shortest path between the two nodes, which is at most $n-1$. As such, plugging this into Theorem 16.2.1, yields the bound, since $m < n^2$. ■

16.3 Bibliographical Notes

A nice survey of the material covered here, is available at `http://arxiv.org/abs/math.PR/0001057`.

Chapter 17

Random Walks IV

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> "Do not imagine, comrades, that leadership is a pleasure! On the contrary, it is a deep and heavy responsibility. No one believes more firmly than Comrade Napoleon that all animals are equal. He would be only too happy to let you make your decisions for yourselves. But sometimes you might make the wrong decisions, comrades, and then where should we be? Suppose you had decided to follow Snowball, with his moonshine of windmills-Snowball, who, as we now know, was no better than a criminal?"
>
> – Animal Farm, George Orwell.

17.1 Cover times

We remind the reader that the cover time of a graph is the expected time to visit all the vertices in the graph, starting from an arbitrary vertex (i.e., worst vertex). The cover time is denoted by $\mathcal{C}(\mathsf{G})$.

Theorem 17.1.1. *Let* G *be an undirected connected graph, then* $\mathcal{C}(\mathsf{G}) \leq 2m(n-1)$*, where* $n = |V(\mathsf{G})|$ *and* $m = |E(\mathsf{G})|$.

Proof: (Sketch.) Construct a spanning tree T of G, and consider the time to walk around T. The expected time to travel on this edge on both directions is $\mathbf{CT}_{uv} = \mathtt{h}_{uv} + \mathtt{h}_{vu}$, which is smaller than $2m$, by Lemma 16.1.8. Now, just connect up those bounds, to get the expected time to travel around the spanning tree. Note, that the bound is independent of the starting vertex. ■

Definition 17.1.2. The ***resistance*** of G is $\mathbf{R}(\mathsf{G}) = \max_{u,v \in V(\mathsf{G})} \mathbf{R}_{uv}$; namely, it is the maximum effective resistance in G.

Theorem 17.1.3. $m\mathbf{R}(\mathsf{G}) \leq \mathcal{C}(\mathsf{G}) \leq 2e^3 m\mathbf{R}(\mathsf{G}) \ln n + 2n$.

Proof: Consider the vertices u and v realizing $\mathbf{R}(\mathsf{G})$, and observe that $\max(\mathtt{h}_{uv}, \mathtt{h}_{vu}) \geq \mathbf{CT}_{uv}/2$, and $\mathbf{CT}_{uv} = 2m\mathbf{R}_{uv}$ by Theorem 16.2.1. Thus, $\mathcal{C}(\mathsf{G}) \geq \mathbf{CT}_{uv}/2 \geq m\mathbf{R}(\mathsf{G})$.

As for the upper bound. Consider a random walk, and divide it into ***epochs***, where a epoch is a random walk of length $2e^3 m\mathbf{R}(\mathsf{G})$. For any vertex v, the expected time to hit u is $\mathtt{h}_{vu} \leq 2m\mathbf{R}(\mathsf{G})$, by Theorem 16.2.1. Thus, the probability that u is not visited in a epoch is $1/e^3$ by the Markov

inequality. Consider a random walk with $\ln n$ epochs. We have that the probability of not visiting u is $\leq (1/e^3)^{\ln n} \leq 1/n^3$. Thus, all vertices are visited after $\ln n$ epochs, with probability $\geq 1 - 1/n^3$. Otherwise, after this walk, we perform a random walk till we visit all vertices. The length of this (fix-up) random walk is $\leq 2n^3$, by Theorem 17.1.1. Thus, expected length of the walk is $\leq 2e^3 m\mathbf{R}(\mathsf{G}) \ln n + 2n^3(1/n^2)$. ■

17.1.1 Rayleigh's Short-cut Principle.

Observe that effective resistance is never raised by lowering the resistance on an edge, and it is never lowered by raising the resistance on an edge. Similarly, resistance is never lowered by removing a vertex.

Another interesting fact, is that effective resistance comply with the triangle inequality.

Observation 17.1.4. *For a graph with minimum degree d, we have $\mathbf{R}(\mathsf{G}) \geq 1/d$ (collapse all vertices except the minimum-degree vertex into a single vertex).*

Lemma 17.1.5. *Suppose that G contains p edge-disjoint paths of length at most ℓ from s to t. Then $\mathbf{R}_{st} \leq \ell/p$.*

17.2 Graph Connectivity

Definition 17.2.1. A ***probabilistic log-space Turing machine*** for a language L is a Turing machine using space $O(\log n)$ and running in time $O(\text{poly}(n))$, where n is the input size. A problem A is in RLP, if there exists a probabilistic log-space Turing machine M such that M accepts $x \in L(A)$ with probability larger than $1/2$, and if $x \notin L(A)$ then $M(x)$ always reject.

Theorem 17.2.2. *Let* USTCON *denote the problem of deciding if a vertex s is connected to a vertex t in an undirected graph. Then $USTCON \in$ RLP.*

Proof: Perform a random walk of length $2n^3$ in the input graph G, starting from s. Stop as soon as the random walk hit t. If u and v are in the same connected component, then $\mathtt{h}_{st} \leq n^3$. Thus, by the Markov inequality, the algorithm works. It is easy to verify that it can be implemented in $O(\log n)$ space. ■

Definition 17.2.3. A graph is ***d-regular***, if all its vertices are of degree d.

A d-regular graph is ***labeled*** if at each vertex of the graph, each of the d edges incident on that vertex has a unique label in $\{1, \ldots, d\}$.

Any sequence of symbols $\sigma = (\sigma_1, \sigma_2, \ldots)$ from $\{1, \ldots, d\}$ together with a starting vertex s in a labeled graph describes a ***walk*** in the graph. For our purposes, such a walk would almost always be finite.

A sequence σ is said to ***traverse*** a labeled graph if the walk visits every vertex of G regardless of the starting vertex. A sequence σ is said to be a ***universal traversal sequence*** of a labeled graph if it traverses all the graphs in this class.

Given such a universal traversal sequence, we can construct (a non-uniform) Turing machine that can solve USTCON for such d-regular graphs, by encoding the sequence in the machine.

Let $\mathcal{F}$ denote a family of graphs, and let $U(\mathcal{F})$ denote the length of the shortest universal traversal sequence for all the labeled graphs in $\mathcal{F}$. Let $\mathbf{R}(\mathcal{F})$ denote the maximum resistance of graphs in this family.

Theorem 17.2.4. $U(\mathcal{F}) \leq 5m\mathbf{R}(\mathcal{F}) \lg(n|\mathcal{F}|)$.

Let $U(d,n)$ denote the length of the shortest universal traversal sequence of connected, labeled n-vertex, d-regular graphs.

Lemma 17.2.5. *The number of labeled n-vertex graphs that are d-regular is $(nd)^{O(nd)}$.*

Proof: There are at most n^{nd} choices for edges in the graph. Every vertex has $d!$ possible labeling of the edges adjacent to it. ■

Lemma 17.2.6. $U(d,n) = O\left(n^3 d \log n\right)$.

Proof: The diameter of every connected n-vertex, d-regular graph is $O(n/d)$. And so, this also bounds the resistance of such a graph. The number of edges is $m = nd/2$. Now, combine Lemma 17.2.5 and Theorem 17.2.4. ■

This is, as mentioned before, not uniform solution. There is by now a known log-space deterministic algorithm for this problem, which is uniform.

17.2.1 Directed graphs

Theorem 17.2.7. *One can solve the $\overrightarrow{\text{STCON}}$ problem with a log-space randomized algorithm, that always output NO if there is no path from s to t, and output YES with probability at least $1/2$ if there is a path from s to t.*

17.3 Graphs and Eigenvalues

Consider an undirected graph $\mathsf{G} = \mathsf{G}(V,E)$ with n vertices. The adjacency matrix $\mathsf{M}(\mathsf{G})$ of G is the $n \times n$ symmetric matrix where $\mathsf{M}_{ij} = \mathsf{M}_{ji}$ is the number of edges between the vertices v_i and v_j. If G is bipartite, we assume that V is made out of two independent sets X and Y. In this case the matrix $\mathsf{M}(\mathsf{G})$ can be written in block form.

Since $\mathsf{M}(\mathsf{G})$ is symmetric, all its eigenvalues exists $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_n$, and their corresponding orthonormal basis vectors are $e_1, \ldots, e_n$. We will need the following theorem.

Theorem 17.3.1 (Fundamental theorem of algebraic graph theory.). *Let $\mathsf{G} = \mathsf{G}(V,E)$ be an n-vertex, undirected (multi)graph with maximum degree d. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be the eigenvalues of $\mathsf{M}(\mathsf{G})$ and the corresponding orthonormal eigenvectors are $e_1, \ldots, e_n$. The following holds.*

(i) If G is connected then $\lambda_2 < \lambda_1$.

(ii) *For $i = 1, \ldots, n$, we have $|\lambda_i| \leq d$.*

(iii) *d is an eigenvalue if and only if G is regular.*

(iv) *If G is d-regular then the eigenvalue $\lambda_1 = d$ has the eigenvector $e_1 = \frac{1}{\sqrt{n}}(1, 1, 1, \ldots, 1)$.*

(v) *The graph G is bipartite if and only if for every eigenvalue λ there is an eigenvalue $-\lambda$ of the same multiplicity.*

(vi) *Suppose that G is connected. Then G is bipartite if and only if $-\lambda_1$ is an eigenvalue.*

(vii) *If G is d-regular and bipartite, then $\lambda_n = d$ and $e_n = \frac{1}{\sqrt{n}}(1, 1, \ldots, 1, -1, \ldots, -1)$, where there are equal numbers of 1s and -1s in e_n.*

17.4 Bibliographical Notes

A nice survey of algebraic graph theory appears in [Wes01] and in [Bol98].

Chapter 18

Expanders I

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> "Mr. Matzerath has just seen fit to inform me that this partisan, unlike so many of them, was an authentic partisan. For - to quote the rest of my patient's lecture - there is no such thing as a part-time partisan. Real partisans are partisans always and as long as they live. They put fallen governments back in power and over throw governments that have just been put in power with the help of partisans. Mr. Matzerath contended - and this thesis struck me as perfectly plausible - that among all those who go in for politics your incorrigible partisan, who undermines what he has just set up, is closest to the artiest because he consistently rejects what he has just created."
>
> – Gunter Grass, The tin drum.

18.1 Preliminaries on expanders

18.1.1 Definitions

Let $\mathsf{G} = (V, E)$ Be an undirected graph, where $V = \{1, \ldots, n\}$. A ***d*-regular graph** is a graph where all vertices have degree d. A d-regular graph $\mathsf{G} = (V, E)$ is a δ-edge expander (or just, ***δ*-expander**) if for every set $S \subseteq V$ of size at most $|V|/2$, there are at least $\delta d|S|$ edges connecting S and $\overline{S} = V \setminus S$; that is

$$e(S, \overline{S}) \geq \delta d |S|, \tag{18.1}$$

where

$$e(X, Y) = \left| \left\{ uv \;\middle|\; u \in X, v \in Y \right\} \right|.$$

A graph is **$[n, d, \delta]$-expander** if it is a n vertex, d-regular, δ-expander.

A **(n, d)-graph** G is a connected d-regular undirected (multi) graph. We will consider the set of vertices of such a graph to be the set $[\![n]\!] = \{1, \ldots, n\}$.

For a (multi) graph G with n nodes, its ***adjacency matrix*** is a $n \times n$ matrix M, where M_{ij} is the number of edges between i and j. It would be convenient to work the ***transition matrix*** Q associated with the random walk on G. If G is d-regular then $\mathsf{Q} = \mathsf{M}(\mathsf{G})/d$ and it is doubly stochastic.

A vector x is ***eigenvector*** of a matrix M with ***eigenvalue*** μ, if $x\mathsf{M} = \mu x$. In particular, by taking the dot product of both size by x, we get $\langle x\mathsf{M}, x\rangle = \langle \mu x, x\rangle$, which implies $\mu = \langle x\mathsf{M}, x\rangle / \langle x, x\rangle$. Since

the adjacency matrix M of G is symmetric, all its eigenvalues are real numbers (this is a special case of the spectral theorem from linear algebra). Two eigenvectors with different eigenvectors are orthogonal to each other.

We denote the eigenvalues of M by $\widehat{\lambda_1} \geq \widehat{\lambda_2} \geq \cdots \widehat{\lambda_n}$, and the eigenvalues of Q by $\widehat{\lambda_1} \geq \widehat{\lambda_2} \geq \cdots \widehat{\lambda_n}$. Note, that for a d-regular graph, the eigenvalues of Q are the eigenvalues of M scaled down by a factor of $1/d$; that is $\widehat{\lambda_i} = \widehat{\lambda_i}/d$.

Lemma 18.1.1. *Let G be an undirected graph, and let Δ denote the maximum degree in G. Then, $\left|\widehat{\lambda_1}(\mathsf{G})\right| = \left|\widehat{\lambda_1}(\mathsf{M})\right| = \Delta$ if and only one connected component of G is Δ-regular. The multiplicity of Δ as an eigenvector is the number of Δ-regular connected components. Furthermore, we have $\left|\widehat{\lambda_i}(\mathsf{G})\right| \leq \Delta$, for all i.*

Proof: The ith entry of $\mathsf{M}\mathbf{1}_n$ is the degree of the ith vertex v_i of G (i.e., $\mathsf{M}\mathbf{1}_n = d(v_i)$, where $\mathbf{1}_n = (1, 1, \ldots, 1) \in \mathbb{R}^n$. So, let x be an eigenvector of M with eigenvalue λ, and let $x_j \neq 0$ be the coordinate with the largest (absolute value) among all coordinates of x corresponding to a connect component H of G. We have that

$$|\lambda| \left|x_j\right| = \left|(\mathsf{M}x)_j\right| = \left|\sum_{v_i \in N(v_j)} x_i\right| \leq \Delta \left|x_j\right|,$$

where $N(v_j)$ are the neighbors of v_i in G. Thus, all the eigenvalues of G have $\left|\widehat{\lambda_i}\right| \leq \Delta$, for $i = 1, \ldots, n$. If $\lambda = \Delta$, then this implies that $x_i = x_j$ if $v_i \in N(v_j)$, and $d(v_j) = \Delta$. Applying this argument to the vertices of $N(v_j)$, implies that H must be Δ-regular, and furthermore, $x_j = x_i$, if $x_i \in V(H)$. Clearly, the dimension of the subspace with eigenvalue (in absolute value) Δ is exactly the number of such connected components. ∎

The following is also known. We do not provide a proof since we do not need it in our argumentation.

Lemma 18.1.2. *If G is bipartite, then if λ is eigenvalue of $\mathsf{M}(\mathsf{G})$ with multiplicity k, then $-\lambda$ is also its eigenvalue also with multiplicity k.*

18.2 Tension and expansion

Let $\mathsf{G} = (V, E)$, where $V = \{1, \ldots, n\}$ and G is a d regular graph.

Definition 18.2.1. For a graph G, let $\gamma(\mathsf{G})$ denote the ***tension*** of G; that is, the smallest constant, such that for any function $f : V(\mathsf{G}) \to \mathbb{R}$, we have that

$$\mathop{\mathbf{E}}_{x,y \in V}\left[|f(x) - f(y)|^2\right] \leq \gamma(\mathsf{G}) \mathop{\mathbf{E}}_{xy \in E}\left[|f(x) - f(y)|^2\right]. \tag{18.2}$$

Intuitively, the tension captures how close is estimating the variance of a function defined over the vertices of G, by just considering the edges of G. Note, that a disconnected graph would have infinite tension, and the clique has tension 1.

Surprisingly, tension is directly related to expansion as the following lemma testifies.

Lemma 18.2.2. *Let $\mathsf{G} = (V, E)$ be a given connected d-regular graph with n vertices. Then, G is a δ-expander, where $\delta \geq \dfrac{1}{2\gamma(\mathsf{G})}$ and $\gamma(\mathsf{G})$ is the tension of G.*

Proof: Consider a set $S \subseteq V$, where $|S| \leq n/2$. Let $f_S(v)$ be the function assigning 1 if $v \in S$, and zero otherwise. Observe that if $(u, v) \in \left(S \times \overline{S}\right) \cup \left(\overline{S} \times S\right)$ then $|f_S(u) - f_S(v)| = 1$, and $|f_S(u) - f_S(v)| = 0$ otherwise. As such, we have

$$\frac{2\,|S|\,(n - |S|)}{n^2} = \mathop{\mathbf{E}}_{x,y \in V}\left[|f_S(x) - f_S(y)|^2\right] \leq \gamma(\mathsf{G}) \mathop{\mathbf{E}}_{xy \in E}\left[|f_S(x) - f_S(y)|^2\right] = \gamma(\mathsf{G})\,\frac{e\left(S, \overline{S}\right)}{|E|},$$

by Lemma 18.2.4. Now, since G is d-regular, we have that $|E| = nd/2$. Furthermore, $n - |S| \geq n/2$, which implies that

$$e\left(S, \overline{S}\right) \geq \frac{2\,|E| \cdot |S|\,(n - |S|)}{\gamma(\mathsf{G})\, n^2} = \frac{2(nd/2)(n/2)\,|S|}{\gamma(\mathsf{G})\, n^2} = \frac{1}{2\gamma(\mathsf{G})}\, d\,|S|\,.$$

which implies the claim (see Eq. (18.1)). ∎

Now, a clique has tension 1, and it has the best expansion possible. As such, the smaller the tension of a graph, the better expander it is.

Definition 18.2.3. Given a random walk matrix Q associated with a d-regular graph, let $\mathcal{B}(\mathsf{Q}) = \langle v_1, \ldots, v_n \rangle$ denote the ***orthonormal eigenvector basis*** defined by Q. That is, $v_1, \ldots, v_n$ is an orthonormal basis for $\mathbb{R}^n$, where all these vectors are eigenvectors of Q and $v_1 = 1^n/\sqrt{n}$. Furthermore, let $\widehat{\lambda_i}$ denote the ith eigenvalue of Q, associated with the eigenvector v_i, such that $\widehat{\lambda_1} \geq \widehat{\lambda_2} \geq \cdots \geq \widehat{\lambda_n}$.

Lemma 18.2.4. *Let $\mathsf{G} = (V, E)$ be a given connected d-regular graph with n vertices. Then $\gamma(\mathsf{G}) = \frac{1}{1 - \widehat{\lambda_2}}$, where $\widehat{\lambda_2} = \lambda_2/d$ is the second largest eigenvalue of Q.*

Proof: Let $f : V \to \mathbb{R}$. Since in Eq. (18.2), we only look on the difference between two values of f, we can add a constant to f, and would not change the quantities involved in Eq. (18.2). As such, we assume that $\mathbf{E}[f(x)] = 0$. As such, we have that

$$\begin{aligned}
\mathop{\mathbf{E}}_{x,y \in V}\left[|f(x) - f(y)|^2\right] &= \mathop{\mathbf{E}}_{x,y \in V}\left[(f(x) - f(y))^2\right] = \mathop{\mathbf{E}}_{x,y \in V}\left[(f(x))^2 - 2f(x)f(y) + (f(y))^2\right] \qquad (18.3)\\
&= \mathop{\mathbf{E}}_{x,y \in V}\left[(f(x))^2\right] - 2 \mathop{\mathbf{E}}_{x,y \in V}[f(x)f(y)] + \mathop{\mathbf{E}}_{x,y \in V}\left[(f(y))^2\right]\\
&= \mathop{\mathbf{E}}_{x \in V}\left[(f(x))^2\right] - 2 \mathop{\mathbf{E}}_{x \in V}[f(x)] \mathop{\mathbf{E}}_{y \in V}[f(y)] + \mathop{\mathbf{E}}_{y \in V}\left[(f(y))^2\right] = 2 \mathop{\mathbf{E}}_{x \in V}\left[(f(x))^2\right].
\end{aligned}$$

Now, let $\mathcal{I}$ be the $n \times n$ identity matrix (i.e., one on its diagonal, and zero everywhere else). We have that

$$\begin{aligned}
\rho &= \frac{1}{d}\sum_{xy \in E}(f(x) - f(y))^2 = \frac{1}{d}\left(\sum_{x \in V} d(f(x))^2 - 2\sum_{xy \in E} f(x)f(y)\right) = \sum_{x \in V}(f(x))^2 - \frac{2}{d}\sum_{xy \in E} f(x)f(y)\\
&= \sum_{x,y \in V}(\mathcal{I} - \mathsf{Q})_{xy}\, f(x)f(y).
\end{aligned}$$

Note, that 1^n is an eigenvector of Q with eigenvalue 1, and this is the largest eigenvalue of Q. Let $\mathcal{B}(\mathsf{Q}) = \langle v_1, \ldots, v_n \rangle$ be the orthonormal eigenvector basis defined by Q, with eigenvalues $\widehat{\lambda_1} \geq \widehat{\lambda_2} \geq \cdots \geq \widehat{\lambda_n}$, respectively. Write $f = \sum_{i=1}^{n} \alpha_i v_i$, and observe that

$$0 = \mathbf{E}[f(x)] = \sum_{i=1}^{n} \frac{f(i)}{n} = \left\langle f, \frac{v_1}{\sqrt{n}} \right\rangle = \left\langle \sum_i \alpha_i v_i, \frac{v_1}{\sqrt{n}} \right\rangle = \frac{1}{\sqrt{n}} \langle \alpha_1 v_1, v_1 \rangle = \frac{\alpha_1}{\sqrt{n}},$$

since $v_i \perp v_1$ for $i \geq 2$. Hence $\alpha_1 = 0$, and we have

$$\begin{aligned}\rho &= \sum_{x,y \in V} (\mathcal{I} - \mathsf{Q})_{xy} f(x) f(y) = \sum_{x,y \in V} (\mathcal{I} - \mathsf{Q})_{xy} \sum_{i=2}^{n} \alpha_{i=1}^{n} \alpha_i v_i(x) \sum_{j=1}^{n} \alpha_j v_j(y) \\ &= \sum_{i,j} \alpha_i \alpha_j \sum_{x \in V} v_i(x) \sum_{y \in V} (\mathcal{I} - \mathsf{Q})_{xy} v_j(y).\end{aligned}$$

Now, we have that

$$\sum_{y \in V} (\mathcal{I} - \mathsf{Q})_{xy} v_j(y) = \left\langle \begin{bmatrix} x\text{th row of} \\ (\mathcal{I} - \mathsf{Q}) \end{bmatrix}, v_j \right\rangle = \left((\mathcal{I} - \mathsf{Q}) v_j \right)(x) = \left(\left(1 - \widehat{\lambda_j}\right) v_j \right)(x) = \left(1 - \widehat{\lambda_j}\right) v_j(x),$$

since v_j is eigenvector of Q with eigenvalue $\widehat{\lambda_j}$. Since $v_1, \ldots, v_n$ is an orthonormal basis, and $f = \sum_{i=1}^{n} \alpha_i v_i$, we have that $\|f\|^2 = \sum_j \alpha_j^2$. Going back to ρ, we have that

$$\begin{aligned}\rho &= \sum_{i,j} \alpha_i \alpha_j \sum_{x \in V} v_i(x) \left(1 - \widehat{\lambda_j}\right) v_j(x) = \sum_{i,j} \alpha_i \alpha_j \left(1 - \widehat{\lambda_j}\right) \sum_{x \in V} v_i(x) v_j(x) \\ &= \sum_{i,j} \alpha_i \alpha_j \left(1 - \widehat{\lambda_j}\right) \left\langle v_i, v_j \right\rangle = \sum_{j=1}^{n} \alpha_j^2 \left(1 - \widehat{\lambda_j}\right) \left\langle v_j, v_j \right\rangle \\ &\geq \left(1 - \widehat{\lambda_2}\right) \sum_{j=2}^{n} \alpha_j^2 \sum_{x \in V} \left(v_j(x)\right)^2 = \left(1 - \widehat{\lambda_2}\right) \sum_{j=2}^{n} \alpha_j^2 = \left(1 - \widehat{\lambda_2}\right) \|f\|^2 = \left(1 - \widehat{\lambda_2}\right) \sum_{j=1}^{n} (f(x))^2 \qquad (18.4) \\ &= n\left(1 - \widehat{\lambda_2}\right) \mathop{\mathbf{E}}_{x \in V} \left[(f(x))^2 \right],\end{aligned}$$

since $\alpha_1 = 0$ and $\widehat{\lambda_1} \geq \widehat{\lambda_2} \geq \cdots \geq \widehat{\lambda_n}$.

We are now ready for the kill. Indeed, by Eq. (18.3), and the above, we have that

$$\begin{aligned}\mathop{\mathbf{E}}_{x,y \in V} \left[|f(x) - f(y)|^2 \right] &= 2 \mathop{\mathbf{E}}_{x \in V} \left[(f(x))^2 \right] \leq \frac{2}{n\left(1 - \widehat{\lambda_2}\right)} \rho = \frac{2}{dn\left(1 - \widehat{\lambda_2}\right)} \sum_{xy \in E} (f(x) - f(y))^2 \\ &= \frac{1}{1 - \widehat{\lambda_2}} \cdot \frac{1}{|E|} \sum_{xy \in E} (f(x) - f(y))^2 = \frac{1}{1 - \widehat{\lambda_2}} \mathop{\mathbf{E}}_{xy \in E} \left[|f(x) - f(y)|^2 \right].\end{aligned}$$

This implies that $\gamma(\mathsf{G}) \leq \dfrac{1}{1 - \widehat{\lambda_2}}$. Observe, that the inequality in our analysis, had risen from Eq. (18.4), but if we take $f = v_2$, then the inequality there holds with equality, which implies that $\gamma(\mathsf{G}) \geq \frac{1}{1 - \widehat{\lambda_2}}$, which implies the claim. ∎

Lemma 18.2.2 together with the above lemma, implies that the expansion δ of a d-regular graph G is at least $\delta = 1/2\gamma(\mathsf{G}) = (1 - \lambda_2/d)/2$, where λ_2 is the second eigenvalue of the adjacency matrix of G. Since the tension of a graph is direct function of its second eigenvalue, we could either argue about the tension of a graph or its second eigenvalue when bounding the graph expansion.

Chapter 19

Expanders II

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> Be that as it may, it is to night school that I owe what education I possess; I am the first to own that it doesn't amount to much, though there is something rather grandiose about the gaps in it.
> – Gunter Grass, The tin drum.

19.1 Bi-tension

Our construction of good expanders, would use the idea of composing graphs together. To this end, in our analysis, we will need the notion of bi-tension. Let $\widetilde{\mathsf{E}}(\mathsf{G})$ be the set of *directed* edges of G; that is, every edge $xy \in E(\mathsf{G})$ appears twice as $(x \to y)$ and $(y \to x)$ in $\widetilde{\mathsf{E}}$.

Definition 19.1.1. For a graph G, let $\gamma_2(\mathsf{G})$ denote the ***bi-tension*** of G; that is, the smallest constant, such that for any two function $f, g : V(\mathsf{G}) \to \mathbb{R}$, we have that

$$\mathop{\mathbf{E}}_{x,y\in V}\Big[|f(x) - g(y)|^2\Big] \leq \gamma_2(\mathsf{G}) \mathop{\mathbf{E}}_{(x\to y)\in\widetilde{\mathsf{E}}}\Big[|f(x) - g(y)|^2\Big]. \tag{19.1}$$

The proof of the following lemma is similar to the proof of Lemma 18.2.4. The proof is provided for the sake of completeness, but there is little new in it.

Lemma 19.1.2. *Let $\mathsf{G} = (V, E)$ be a connected d-regular graph with n vertices. Then $\gamma_2(\mathsf{G}) = \dfrac{1}{1-\widehat{\lambda}}$, where $\widehat{\lambda} = \widehat{\lambda}(\mathsf{G})$, where $\widehat{\lambda}(\mathsf{G}) = \max\Big(\widehat{\lambda_2}, -\widehat{\lambda_n}\Big)$, where $\widehat{\lambda_i}$ is the ith largest eigenvalue of the random walk matrix associated with G.*

Proof: We can assume that $\mathbf{E}[f(x)] = 0$. As such, we have that

$$\mathop{\mathbf{E}}_{x,y\in V}\Big[|f(x) - g(y)|^2\Big] = \mathop{\mathbf{E}}_{x,y\in V}\Big[(f(x))^2\Big] - 2\mathop{\mathbf{E}}_{x,y\in V}[f(x)g(y)] + \mathop{\mathbf{E}}_{y\in V}\Big[(g(y))^2\Big] = \mathop{\mathbf{E}}_{x,y\in V}\Big[(f(x))^2\Big] + \mathop{\mathbf{E}}_{y\in V}\Big[(g(y))^2\Big]. \tag{19.2}$$

Let Q be the matrix associated with the random walk on G (each entry is either zero or $1/d$), we have

$$\rho = \mathop{\mathbf{E}}_{(x\to y)\in\widetilde{\mathsf{E}}}\left[|f(x)-g(y)|^2\right] = \frac{1}{nd}\sum_{(x\to y)\in\widetilde{\mathsf{E}}}(f(x)-g(y))^2 = \frac{1}{n}\sum_{x,y\in V}\mathsf{Q}_{xy}(f(x)-g(y))^2$$
$$= \frac{1}{n}\sum_{x\in V}\Big((f(x))^2+(g(x))^2\Big) - \frac{2}{n}\sum_{x,y\in V}\mathsf{Q}_{xy}f(x)g(y).$$

Let $\mathcal{B}(\mathsf{Q}) = \langle v_1,\ldots,v_n\rangle$ be the orthonormal eigenvector basis defined by Q (see Definition 18.2.3), with eigenvalues $\widehat{\lambda_1} \geq \widehat{\lambda_2} \geq \cdots \geq \widehat{\lambda_n}$, respectively. Write $f = \sum_{i=1}^n \alpha_i v_i$ and $g = \sum_{i=1}^n \beta_i v_i$. Since $\mathbf{E}[f(x)] = 0$, we have that $\alpha_1 = 0$. Now, $\mathsf{Q}_{xy} = \mathsf{Q}_{yx}$, and we have

$$\sum_{x,y\in V}\mathsf{Q}_{xy}f(x)g(y) = \sum_{x,y\in V}\mathsf{Q}_{yx}\left(\sum_i \alpha_i v_i(x)\right)\left(\sum_j \beta_j v_j(y)\right) = \sum_{i,j}\alpha_i\beta_j\sum_{y\in V}v_j(y)\sum_{x\in V}\mathsf{Q}_{yx}v_i(x)$$
$$= \sum_{i,j}\alpha_i\beta_j\sum_{y\in V}v_j(y)\Big(\widehat{\lambda_i}v_i(y)\Big) = \sum_{i,j}\alpha_i\beta_j\widehat{\lambda_i}\left\langle v_j, v_i\right\rangle = \sum_{i=2}^n \alpha_i\beta_i\widehat{\lambda_i}\sum_{y\in V}(v_i(y))^2$$
$$\leq \widehat{\lambda}\sum_{i=2}^n\frac{\alpha_i^2+\beta_i^2}{2}\sum_{y\in V}(v_i(y))^2 \leq \frac{\widehat{\lambda}}{2}\sum_{i=1}^n\sum_{y\in V}\Big((\alpha_i v_i(y))^2+(\beta_i v_i(y))^2\Big)$$
$$= \frac{\widehat{\lambda}}{2}\sum_{y\in V}\Big((f(y))^2+(g(y))^2\Big)$$

As such,

$$\mathop{\mathbf{E}}_{(x\to y)\in\widetilde{\mathsf{E}}}\left[|f(x)-g(y)|^2\right] = \frac{1}{nd}\sum_{(x\to y)\in\widetilde{\mathsf{E}}}|f(x)-g(y)|^2 = \frac{1}{n}\sum_{y\in V}\Big((f(y))^2+(g(y))^2\Big) - \frac{1}{n}\sum_{x,y\in V}\frac{2f(x)g(y)}{d}$$
$$= \frac{1}{n}\sum_{y\in V}\Big((f(y))^2+(g(y))^2\Big) - \frac{2}{n}\sum_{x,y\in V}\mathsf{Q}_{xy}f(x)g(y)$$
$$\geq \left(\frac{1}{n}-\frac{2}{n}\cdot\frac{\widehat{\lambda}}{2}\right)\sum_{y\in V}\Big((f(y))^2+(g(y))^2\Big) = \Big(1-\widehat{\lambda}\Big)\left(\mathop{\mathbf{E}}_{y\in V}\left[(f(y))^2\right]+\mathop{\mathbf{E}}_{y\in V}\left[(g(y))^2\right]\right)$$
$$= \Big(1-\widehat{\lambda}\Big)\mathop{\mathbf{E}}_{x,y\in V}\left[|f(x)-g(y)|^2\right],$$

by Eq. (19.2). This implies that $\gamma_2(\mathsf{G}) \leq 1/\Big(1-\widehat{\lambda}\Big)$. Again, by trying either $f = g = v_2$ or $f = v_n$ and $g = -v_n$, we get that the inequality above holds with equality, which implies $\gamma_2(\mathsf{G}) \geq 1/\Big(1-\widehat{\lambda}\Big)$. Together, the claim now follows. ∎

19.2 Explicit construction

For a set $U \subseteq V$ of vertices, its ***characteristic vector***, denoted by $x = \chi_U$, is the n dimensional vector, where $x_i = 1$ if and only if $i \in U$.

The following is an easy consequence of Lemma 18.1.1.

Lemma 19.2.1. *For a d-regular graph G the vector $\mathbf{1}^n = (1, 1, \ldots, 1)$ is the only eigenvector with eigenvalue d (of the adjacency matrix $\mathsf{M}(\mathsf{G})$, if and only if G is connected. Furthermore, we have $|\lambda_i| \leq d$, for all i.*

Our main interest would be in the second largest eigenvalue of M. Formally, let

$$\lambda_2(\mathsf{G}) = \max_{x \perp \mathbf{1}^n, x \neq 0} \left| \frac{\langle x\mathsf{M}, x \rangle}{\langle x, x \rangle} \right|.$$

We state the following result but do not prove it since we do not need it for our nafarious purposes (however, we did prove the left side of the inequality).

Theorem 19.2.2. *Let G be a δ-expander with adjacency matrix M and let $\lambda_2 = \lambda_2(\mathsf{G})$ be the second-largest eigenvalue of M. Then*

$$\frac{1}{2}\left(1 - \frac{\lambda_2}{d}\right) \leq \delta \leq \sqrt{2\left(1 - \frac{\lambda_2}{d}\right)}.$$

What the above theorem says, is that the expansion of a $[n, d, \delta]$-expander is a function of how far is its second eigenvalue (i.e., λ_2) from its first eigenvalue (i.e., d). This is usually referred to as the ***spectral gap***.

We will start by explicitly constructing an expander that has "many" edges, and then we will show to reduce its degree till it become a constant degree expander.

19.2.1 Explicit construction of a small expander

19.2.1.1 A quicky reminder of fields

A ***field*** is a set $\mathbb{F}$ together with two operations, called addition and multiplication, and denoted by $+$ and $\cdot$, respectively, such that the following axioms hold:

(i) Closure: $\forall x, y \in \mathbb{F}$, we have $x + y \in \mathbb{F}$ and $x \cdot y \in \mathbb{F}$.

(ii) Associativity: $\forall x, y, z \in \mathbb{F}$, we have $x + (y + z) = (x + y) + z$ and $(x \cdot y) \cdot z = x \cdot (y \cdot z)$.

(iii) Commutativity: $\forall x, y \in \mathbb{F}$, we have $x + y = y + x$ and $x \cdot y = y \cdot x$.

(iv) Identity: There exists two distinct special elements $0, 1 \in \mathbb{F}$. We have that $\forall x \in \mathbb{F}$ it holds $x + 0 = a$ and $x \cdot 1 = x$.

(v) Inverse: There exists two distinct special elements $0, 1 \in \mathbb{F}$, and we have that $\forall x \in \mathbb{F}$ there exists an element $-x \in \mathbb{F}$, such that $x + (-x) = 0$.

Similarly, $\forall x \in \mathbb{F}$, $x \neq 0$, there exists an element $y = x^{-1} = 1/x \in \mathbb{F}$ such that $x \cdot y = 1$.

(vi) Distributivity: $\forall x, y, z \in \mathbb{F}$ we have $x \cdot (y + z) = x \cdot y + x \cdot z$.

Let $q = 2^t$, and $r > 0$ be an integer. Consider the finite field $\mathbb{F}_q$. It is the field of polynomials of degree at most $t-1$, where the coefficients are over $\mathbb{Z}_2$ (i.e., all calculations are done modulus 2). Formally, consider the polynomial

$$p(x) = x^t + x + 1.$$

It it irreducible over $\mathbb{F}_2 = \{0, 1\}$ (i.e., $p(0) = p(1) \neq 0$). We can now do polynomial arithmetic over polynomials (with coefficients from $\mathbb{F}_2$), where we do the calculations modulus $p(x)$. Note, that any irreducible polynomial of degree n yields the same field up to isomorphism. Intuitively, we are introducing the n distinct roots of $p(x)$ into $\mathbb{F}$ by creating an extension field of $\mathbb{F}$ with those roots.

An element of $\mathbb{F}_q = \mathbb{F}_{2^t}$ can be interpreted as a binary string $b = b_0 b_1 \ldots, b_{t-1}$ of length t, where the corresponding polynomial is

$$\text{poly}(b) = \sum_{i=0}^{t-1} b_i x^i.$$

The nice property of $\mathbb{F}_q$ is that addition can be interpreted as a `xor` operation. That is, for any $x, y \in \mathbb{F}_q$, we have that $x + y + y = x$ and $x - y - y = x$. The key properties of $\mathbb{F}_q$ we need is that multiplications and addition can be computed in it in polynomial time in t, and it is a field (i.e., each non-zero element has a unique inverse).

Computing multiplication in $\mathbb{F}_q$. Consider two elements $\alpha, \beta \in \mathbb{F}_q$. Multiply the two polynomials $\text{poly}(\alpha)$ by $\text{poly}(\beta)$, let $\text{poly}(\gamma)$ be the resulting polynomial (of degree at most $2t-2$), and compute the remainder $\text{poly}(\beta)$ when dividing it by the irreducible polynomial $p(x)$. For this remainder polynomial, normalize the coefficients by computing their modules base 2. The resulting polynomial is the product of α and β.

For more details on this field, see any standard text on abstract algebra.

19.2.1.2 The construction

Let $q = 2^t$, and $r > 0$ be an integer. Consider the linear space $\mathbb{G} = \mathbb{F}_q^r$. Here, a member $\alpha = (\alpha_0, \ldots, \alpha_r) \in \mathbb{G}$ can be thought of as being a string (of length $r+1$) over $\mathbb{F}_q$, or alternatively, as a binary string of length $n = t(r+1)$.

For $\alpha = (\alpha_0, \ldots, \alpha_r) \in \mathbb{G}$, and $x, y \in \mathbb{F}_q$, define the operator

$$\rho(\alpha, x, y) = \alpha + y \cdot \left(1, x, x^2, \ldots, x^r\right) = \left(\alpha_0 + y, \alpha_1 + yx, \alpha_2 + yx^2, \ldots, \alpha_r + yx^r\right) \in \mathbb{G}.$$

Since addition over $\mathbb{F}_q$ is equivalent to a xor operation we have that

$$\begin{aligned}\rho\Big(\rho(\alpha, x, y), x, y\Big) &= \left(\alpha_0 + y + y, \alpha_1 + yx + yx, \alpha_2 + yx^2 + yx^2, \ldots, \alpha_r + yx^r + yx^r\right) \\ &= (\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_r) = \alpha.\end{aligned}$$

Furthermore, if $(x, y) \neq (x', y')$ then $\rho(\alpha, x, y) \neq \rho(\alpha, x', y')$.

We now define a graph $\mathsf{LD}(q, r) = (\mathbb{G}, E)$, where

$$E = \left\{ \alpha\beta \;\middle|\; \begin{array}{l} \alpha \in \mathbb{G}, x, y \in \mathbb{F}_q \\ \beta = \rho(\alpha, x, y) \end{array} \right\}$$

Note, that this graph is well defined, as $\rho(\beta, x, y) = \alpha$. The degree of a vertex of $\mathsf{LD}(q, r)$ is $\left|\mathbb{F}_q\right|^2 = q^2$, and $\mathsf{LD}(q, r)$ has $N = |\mathbb{G}| = q^{r+1} = 2^{t(r+1)} = 2^n$ vertices.

Theorem 19.2.3. *For any $t > 0, r > 0$ and $q = 2^t$, where $r < q$, we have that $\mathsf{LD}(q,r)$ is a graph with q^{r+1} vertices. Furthermore, $\lambda_1(\mathsf{LD}(q,r)) = q^2$, and $\lambda_i(\mathsf{LD}(q,r)) \leq rq$, for $i = 2,\ldots,n$.*

In particular, if $r \leq q/2$, then $\mathsf{LD}(q,r)$ is a $\left[q^{r+1}, q^2, \frac{1}{4}\right]$-expander.

Proof: Let M be the $N \times N$ adjacency matrix of $\mathsf{LD}(q,r)$. Let $L : \mathbb{F}_q \to \{0,1\}$ be a linear map which is onto. It is easy to verify that $\left|L^{-1}(0)\right| = \left|L^{-1}(1)\right|$[①]

We are interested in the eigenvalues of the matrix M. To this end, we consider vectors in $\mathbb{R}^N$. The ith row an ith column of M is associated with a unique element $b_i \in \mathbb{G}$. As such, for a vector $v \in \mathbb{R}^N$, we denote by $v[b_i]$ the ith coordinate of v. In particular, for $\alpha = (\alpha_0,\ldots,\alpha_r) \in \mathbb{G}$, let $v_\alpha \in \mathbb{R}^N$ denote the vector, where its $\beta = (\beta_0,\ldots,\beta_r) \in \mathbb{G}$ coordinate is

$$v_\alpha[\beta] = (-1)^{L\left(\sum_{i=0}^r \alpha_i\beta_i\right)}.$$

Let $V = \left\{ v_\alpha \;\middle|\; \alpha \in \mathbb{G} \right\}$. For $\alpha \neq \alpha' \in V$, observe that

$$\langle v_\alpha, v_{\alpha'}\rangle = \sum_{\beta\in\mathbb{G}} (-1)^{L\left(\sum_{i=0}^r \alpha_i\beta_i\right)} \cdot (-1)^{L\left(\sum_{i=0}^r \alpha'_i\beta_i\right)} = \sum_{\beta\in\mathbb{G}} (-1)^{L\left(\sum_{i=0}^r (\alpha_i+\alpha'_i)\beta_i\right)} = \sum_{\beta\in\mathbb{G}} v_{\alpha+\alpha'}[\beta].$$

So, consider $\psi = \alpha + \alpha' \neq 0$. Assume, for the simplicity of exposition that all the coordinates of ψ are non-zero. We have, by the linearity of L that

$$\langle v_\alpha, v_{\alpha'}\rangle = \sum_{\beta\in\mathbb{G}} (-1)^{L\left(\sum_{i=0}^r \alpha_i\beta_i\right)} = \sum_{\beta_0\in\mathbb{F}_q,\ldots,\beta_{r-1}\in\mathbb{F}_q} (-1)^{L(\psi_0\beta_0+\cdots+\psi_{r-1}\beta_{r-1})} \sum_{\beta_r\in\mathbb{F}_q} (-1)^{L(\psi_r\beta_r)}.$$

However, since $\psi_r \neq 0$, the quantity $\left\{ \psi_r\beta_r \;\middle|\; \beta_r \in \mathbb{F}_q \right\} = \mathbb{F}_q$. Thus, the summation $\sum_{\beta_r\in\mathbb{F}_q} (-1)^{L(\psi_r\beta_r)}$ gets $\left|L^{-1}(0)\right|$ terms that are 1, and $\left|L^{-1}(0)\right|$ terms that are -1. As such, this summation is zero, implying that $\langle v_\alpha, v_{\alpha'}\rangle = 0$. Namely, the vectors of V are orthogonal.

Observe, that for $\alpha,\beta,\psi \in \mathbb{G}$, we have $v_\alpha[\beta+\psi] = v_\alpha[\beta]\, v_\alpha[\psi]$. For $\alpha \in \mathbb{G}$, consider the vector $\mathsf{M}v_\alpha$. We have, for $\beta \in \mathbb{G}$, that

$$\begin{aligned}(\mathsf{M}v_\alpha)[\beta] = \sum_{\psi\in\mathbb{G}} \mathsf{M}_{\beta\psi}\cdot v_\alpha[\psi] &= \sum_{\substack{x,y\,\in\,\mathbb{F}_q \\ \psi=\rho(\beta,x,y)}} v_\alpha[\psi] = \sum_{x,y\,\in\,\mathbb{F}_q} v_\alpha[\beta + y(1,x,\ldots,x^r)] \\ &= \left(\sum_{x,y\,\in\,\mathbb{F}_q} v_\alpha[y(1,x,\ldots,x^r)]\right)\cdot v_\alpha[\beta].\end{aligned}$$

Thus, setting $\lambda(\alpha) = \sum_{x,y\,\in\,\mathbb{F}_q} v_\alpha[y(1,x,\ldots,x^r)] \in \mathbb{R}$, we have that $\mathsf{M}v_\alpha = \lambda(\alpha)\cdot v_\alpha$. Namely, v_α is an eigenvector, with eigenvalue $\lambda(\alpha)$.

Let $p_\alpha(x) = \sum_{i=0}^r \alpha_i x^i$, and let

$$\begin{aligned}\lambda(\alpha) &= \sum_{x,y\,\in\,\mathbb{F}_q} v_\alpha[y(1,x,\ldots,x^r)] \in \mathbb{R} = \sum_{x,y\in\mathbb{F}_q} (-1)^{L(yp_\alpha(x))} \\ &= \sum_{\substack{x,y\in\mathbb{F}_q \\ p_\alpha(x)=0}} (-1)^{L(y\,p_\alpha(x))} + \sum_{\substack{x,y\in\mathbb{F}_q \\ p_\alpha(x)\neq 0}} (-1)^{L(y\,p_\alpha(x))}.\end{aligned}$$

[①]Indeed, if $Z = L^{-1}(0)$, and $L(x) = 1$, then $L(y) = 1$, for all $y \in U = \left\{ x + z \;\middle|\; z \in Z \right\}$. Now, its clear that $|Z| = |U|$.

If $p_\alpha(x) = 0$ then $(-1)^{L(y\ p_\alpha(x))} = 1$, for all y. As such, each such x contributes q to $\lambda(\alpha)$.

If $p_\alpha(x) \neq 0$ then $y\ p_\alpha(x)$ takes all the values of $\mathbb{F}_q$, and as such, $L(y\ p_\alpha(x))$ is 0 for half of these values, and 1 for the other half. Implying that these kind terms contribute 0 to $\lambda(\alpha)$. But $p_\alpha(x)$ is a polynomial of degree r, and as such there could be at most r values of x for which the first term is taken. As such, if $\alpha \neq 0$ then $\lambda(\alpha) \leq rq$. If $\alpha = 0$ then $\lambda(\alpha) = q^2$, which implies the theorem. ■

This construction provides an expander with constant degree only if the number of vertices is a constant. Indeed, if we want an expander with constant degree, we have to take q to be as small as possible. We get the relation $n = q^{r+1} \leq q^q$, since $r \leq r$, which implies that $q = \Omega(\log n / \log\log n)$. Now, the expander of Theorem 19.2.3 is q^2-regular, which means that it is not going to provide us with a constant degree expander.

However, we are going to use it as our building block in a construction that would start with this expander and would inflate it up to the desired size.

Chapter 20

Expanders III - The Zig Zag Product

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> Gradually, but not as gradually as it seemed to some parts of his brain, he began to infuse his tones with a sarcastic wounding bitterness. Nobody outside a madhouse, he tried to imply, could take seriously a single phrase of this conjectural, nugatory, deluded, tedious rubbish. Within quite a short time he was contriving to sound like an unusually fanatical Nazi trooper in charge of a book-burning reading out to the crowd excerpts from a pamphlet written by a pacifist, Jewish, literate Communist. A growing mutter, half-amused, half-indignant, arose about him, but he closed his ears to it and read on. Almost unconsciously he began to adopt an unnameable foreign accent and to read faster and faster, his head spinning. As if in a dream he heard Welch stirring, then whispering, then talking at his side. he began punctuating his discourse with smothered snorts of derision. He read on, spitting out the syllables like curses, leaving mispronunciations, omissions, spoonerisms uncorrected, turning over the pages of his script like a score-reader following a presto movement, raising his voice higher and higher. At last he found his final paragraph confronting him, stopped, and look at his audience.
>
> – Kingsley Amis, Lucky Jim.

20.1 Building a large expander with constant degree

20.1.1 Notations

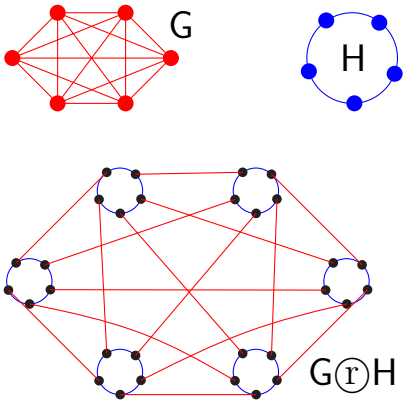
For a vertex $v \in V(\mathsf{G})$, we will denote by $v_{\mathsf{G}}[i] = v[i]$ the ith neighbor of v in the graph G (we order the neighbors of a vertex in an arbitrary order).

The regular graphs we next discuss have ***consistent labeling***. That is, for a regular graph G (we assume here that G is regular). This means that if u is the ith neighbor v then v is the ith neighbor of u. Formally, this means that $v[i][i] = v$, for all v and i. This is a non-trivial property, but its easy to verify that the low quality expander of Theorem 19.2.3 has this property. It is also easy to verify that the complete graph can be easily be made into having consistent labeling (exercise). These two graphs would be sufficient for our construction.

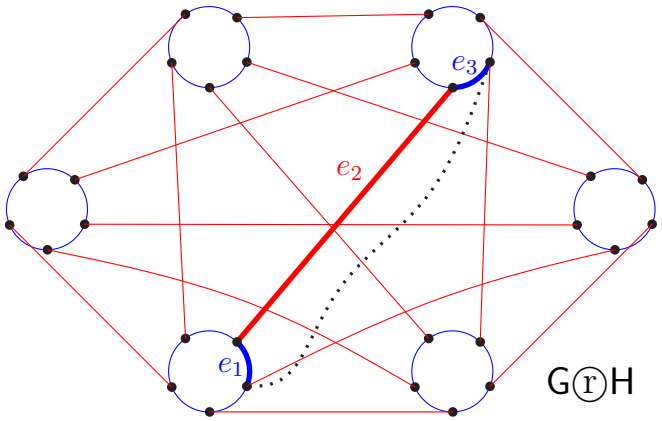
20.1.2 The Zig-Zag product

At this point, we know how to construct a good "small" expander. The question is how to build a large expander (i.e., large number of vertices) and with constant degree.

The intuition of the construction is the following: It is easy to improve the expansion qualities of a graph by squaring it. The problem is that the resulting graph G has a degree which is too large. To overcome this, we will replace every vertex in G by a copy of a small graph that is connected and has low degree. For example, we could replace every vertex of degree d in G by a path having d vertices. Every such vertex is now in charge of original edge of the graph. Naturally, such a replacement operation reduces the quality of the expansion of the resulting graph. In this case, replacing a vertex with a path is a potential "disaster", since every such subpath increases the lengths of the paths of the original graph by a factor of d (and intuitively, a good expander have "short" paths between any pair of vertices).



Consider a "large" (n, D)-graph G and a "small" (D, d)-graph H. As a first stage, we replace every vertex of G by a copy of H. The new graph K has $[\![n]\!] \times [\![D]\!]$ as a vertex set. Here, the edge $vu \in V(\mathsf{G})$, where $u = v[i]$ and $v = u[j]$, is replaced by the edge connecting $(v, i) \in V(\mathsf{K})$ with $(u, j) \in V(\mathsf{K})$. We will refer to this resulting edge $(v, i)(u, j)$ as a ***long*** edge. Also, we copy all the edges of the small graph to each one of its copies. That is, for each $i \in [\![n]\!]$, and $uv \in E(\mathsf{H})$, we add the edge $(i, u)(i, v)$ to K, which is a ***short*** edge. We will refer to K, which is a $(nD, d + 1)$-graph, as a ***replacement product*** of G and H, denoted by G ⓡ H. See figure on the right for an example.



Again, intuitively, we are losing because the expansion of the resulting graph had deteriorated too much. To overcome this problem, we will perform local shortcuts to shorten the paths in the resulting graph (and thus improve its expansion properties). A ***zig-zag-zig path*** in the replacement product graph K, is a three edge path $e_1e_2e_3$, where e_1 and e_3 are short edges, and the middle edge e_2 is a long edge. That is, if $e_1 = (i, u)(i, v)$, $e_2 = (i, v)(j, v')$, and $e_3 = (j, v')(j, u')$, then $e_1, e_2, e_3 \in E(\mathsf{K})$, $ij \in E(\mathsf{G})$, $uv \in E(\mathsf{H})$ and $v'u' \in E(\mathsf{H})$. Intuitively, you can think about e_1 as a small "zig" step in H, e_2 is a long "zag" step in G, and finally e_3 is a "zig" step in H.

Another way of representing a zig-zag-zig path $v_1v_2v_3v_4$ starting at the vertex $v_1 = (i, v) \in V(\mathsf{F})$, is to parameterize it by two integers $\ell, \ell' \in [\![d]\!]$, where

$$v_1 = (i, v), \quad v_2 = (i, v_{\mathsf{H}}[\ell]) \quad v_3 = \left(i_{\mathsf{G}}\left[v_{\mathsf{H}}[\ell]\right], v_{\mathsf{H}}[\ell]\right) \quad v_4 = \left(i_{\mathsf{G}}\left[v_{\mathsf{H}}[\ell]\right], \left(v_{\mathsf{H}}[\ell]\right)_{\mathsf{H}}[\ell']\right).$$

Let Z be the set of all (unordered) pairs of vertices of K connected by such a zig-zag-zig path. Note, that every vertex (i, v) of K has d^2 paths having (i, v) as an end point. Consider the graph $\mathsf{F} = (V(\mathsf{K}), Z)$. The graph F has nD vertices, and it is d^2 regular. Furthermore, since we shortcut all these zig-zag-zig paths in K, the graph F is a much better expander (intuitively) than K. We will refer to the graph F as the ***zig-zag*** product of G and H.

Definition 20.1.1. The ***zig-zag product*** of (n, D)-graph G and a (D, d)-graph H, is the (nD, d^2) graph $\mathsf{F} = \mathsf{G} \textcircled{z} \mathsf{H}$, where the set of vertices is $[\![n]\!] \times [\![D]\!]$ and for any $v \in [\![n]\!]$, $i \in [\![D]\!]$, and $\ell, \ell' \in [\![d]\!]$ we have in F the edge connecting the vertex (i, v) with the vertex $\left(i_{\mathsf{G}}\left[v_{\mathsf{H}}[\ell]\right], \left(v_{\mathsf{H}}[\ell]\right)_{\mathsf{H}}[\ell']\right)$.

Remark 20.1.2. We need the resulting zig-zag graph to have consistent labeling. For the sake of simplicity of exposition, we are just going to assume this property.

We next bound the tension of the zig-zag product graph.

Theorem 20.1.3. *We have* $\gamma(\mathsf{G} \textcircled{z} \mathsf{H}) \leq \gamma_2(\mathsf{G}) (\gamma_2(\mathsf{H}))^2$. *and* $\gamma_2(\mathsf{G} \textcircled{z} \mathsf{H}) \leq \gamma_2(\mathsf{G}) (\gamma_2(\mathsf{H}))^2$.

Proof: Let $\mathsf{G} = ([\![n]\!], E)$ be a (n, D)-graph and $\mathsf{H} = ([\![D]\!], E')$ be a (D, d)-graph. Fix any function $f : [\![n]\!] \times [\![D]\!] \to \mathbb{R}$, and observe that

$$\begin{aligned}
\psi &= \mathop{\mathbf{E}}_{\substack{u,v \in [\![n]\!] \\ k,\ell \in [\![D]\!]}} \left[|f(u,k) - f(v,\ell)|^2 \right] = \mathop{\mathbf{E}}_{k,\ell \in [\![D]\!]} \left[\mathop{\mathbf{E}}_{u,v \in [\![n]\!]} \left[|f(u,k) - f(v,\ell)|^2 \right] \right] \\
&\leq \mathop{\mathbf{E}}_{k,\ell \in [\![D]\!]} \left[\gamma_2(\mathsf{G}) \mathop{\mathbf{E}}_{uv \in E(\mathsf{G})} \left[|f(u,k) - f(v,\ell)|^2 \right] \right] = \gamma_2(\mathsf{G}) \underbrace{\mathop{\mathbf{E}}_{k,\ell \in [\![D]\!]} \left[\mathop{\mathbf{E}}_{\substack{u \in [\![n]\!] \\ p \in [\![D]\!]}} \left[\left| f(u,k) - f(u[p],\ell) \right|^2 \right] \right]}_{=\Delta_1}.
\end{aligned}$$

Now,

$$\begin{aligned}
\Delta_1 &= \mathop{\mathbf{E}}_{\substack{u \in [\![n]\!] \\ \ell \in [\![D]\!]}} \left[\mathop{\mathbf{E}}_{k,p \in [\![D]\!]} \left[\left| f(u,k) - f(u[p],\ell) \right|^2 \right] \right] \leq \mathop{\mathbf{E}}_{\substack{u \in [\![n]\!] \\ \ell \in [\![D]\!]}} \left[\gamma_2(\mathsf{H}) \mathop{\mathbf{E}}_{kp \in E(\mathsf{H})} \left[\left| f(u,k) - f(u[p],\ell) \right|^2 \right] \right] \\
&= \gamma_2(\mathsf{H}) \underbrace{\mathop{\mathbf{E}}_{\substack{u \in [\![n]\!] \\ \ell \in [\![D]\!]}} \left[\mathop{\mathbf{E}}_{\substack{p \in [\![D]\!] \\ j \in [\![d]\!]}} \left[\left| f(u,p[j]) - f(u[p],\ell) \right|^2 \right] \right]}_{=\Delta_2}.
\end{aligned}$$

Now,

$$\begin{aligned}
\Delta_2 &= \mathop{\mathbf{E}}_{\substack{j \in [\![d]\!] \\ \ell \in [\![D]\!]}} \left[\mathop{\mathbf{E}}_{\substack{u \in [\![n]\!] \\ p \in [\![D]\!]}} \left[\left| f(u,p[j]) - f(u[p],\ell) \right|^2 \right] \right] = \mathop{\mathbf{E}}_{\substack{j \in [\![d]\!] \\ \ell \in [\![D]\!]}} \left[\mathop{\mathbf{E}}_{\substack{v \in [\![n]\!] \\ p \in [\![D]\!]}} \left[\left| f(v[p],p[j]) - f(v,\ell) \right|^2 \right] \right] \\
&= \mathop{\mathbf{E}}_{\substack{j \in [\![d]\!] \\ v \in [\![n]\!]}} \left[\mathop{\mathbf{E}}_{\substack{p \in [\![D]\!] \\ \ell \in [\![D]\!]}} \left[\left| f(v[p],p[j]) - f(v,\ell) \right|^2 \right] \right] \\
&= \gamma_2(\mathsf{H}) \underbrace{\mathop{\mathbf{E}}_{\substack{j \in [\![d]\!] \\ v \in [\![n]\!]}} \left[\mathop{\mathbf{E}}_{p\ell \in E(\mathsf{H})} \left[\left| f(v[p],p[j]) - f(v,\ell) \right|^2 \right] \right]}_{=\Delta_3}.
\end{aligned}$$

Now, we have

$$\Delta_3 = \mathop{\mathbf{E}}_{\substack{j \in [\![d]\!] \\ v \in [\![n]\!]}} \left[\mathop{\mathbf{E}}_{\substack{p \in [\![D]\!] \\ i \in [\![d]\!]}} \left[\left| f(v[p],p[j]) - f(v,p[i]) \right|^2 \right] \right] = \mathop{\mathbf{E}}_{(u,k)(\ell,v) \in E(\mathsf{G} \textcircled{z} \mathsf{H})} \left[\left| f(u,k) - f(\ell,v) \right| \right],$$

as $(v[p], p[j])$ is adjacent to $(v[p], p)$ (a short edge), which is in turn adjacent to (v, p) (a long edge), which is adjacent to $(v, p[i])$ (a short edge). Namely, $(v[p], p[j])$ and $(v, p[i])$ form the endpoints of a zig-zag path in the replacement product of G and H. That is, these two endpoints are connected by an edge in the zig-zag product graph. Furthermore, it is easy to verify that each zig-zag edge get accounted for in this representation exactly once, implying the above inequality. Thus, we have $\psi \leq \gamma_2(\mathsf{G})(\gamma_2(\mathsf{H}))^2 \Delta_3$, which implies the claim.

The second claim follows by similar argumentation. ■

20.1.3 Squaring

The last component in our construction, is ***squaring***squaring!graph a graph. Given a (n, d)-graph G, consider the multigraph G^2 formed by connecting any vertices connected in G by a path of length 2. Clearly, if M is the adjacency matrix of G, then the adjacency matrix of G^2 is the matrix M^2. Note, that $\left(\mathsf{M}^2\right)_{ij}$ is the number of distinct paths of length 2 in G from i to j. Note, that the new graph might have self loops, which does not effect our analysis, so we keep them in.

Lemma 20.1.4. *Let G be a (n, d)-graph. The graph G^2 is a (n, d^2)-graph. Furthermore $\gamma_2\left(\mathsf{G}^2\right) = \frac{(\gamma_2(\mathsf{G}))^2}{2\gamma_2(\mathsf{G})-1}$.*

Proof: The graph G^2 has eigenvalues $\left(\widehat{\lambda}_1(\mathsf{G})\right)^2, \ldots, \left(\widehat{\lambda}_1(\mathsf{G})\right)^2$ for its matrix Q^2. As such, we have that

$$\widehat{\lambda}\left(\mathsf{G}^2\right) = \max\left(\widehat{\lambda}_2\left(\mathsf{G}^2\right), -\widehat{\lambda}_n\left(\mathsf{G}^2\right)\right).$$

Now, $\widehat{\lambda}_1\left(\mathsf{G}^2\right) = 1$. Now, if $\widehat{\lambda}_2(\mathsf{G}) \geq \left|\widehat{\lambda}_n(\mathsf{G})\right| < 1$ then $\widehat{\lambda}\left(\mathsf{G}^2\right) = \widehat{\lambda}_2\left(\mathsf{G}^2\right) = \left(\widehat{\lambda}_2(\mathsf{G})\right)^2 = \left(\widehat{\lambda}(\mathsf{G})\right)^2$.

If $\widehat{\lambda}_2(\mathsf{G}) < \left|\widehat{\lambda}_n(\mathsf{G})\right|$ then $\widehat{\lambda}\left(\mathsf{G}^2\right) = \widehat{\lambda}_2\left(\mathsf{G}^2\right) = \left(\widehat{\lambda}_n(\mathsf{G})\right)^2 = \left(\widehat{\lambda}(\mathsf{G})\right)^2$..

Thus, in either case $\widehat{\lambda}\left(\mathsf{G}^2\right) = \left(\widehat{\lambda}(\mathsf{G})\right)^2$. Now, By Lemma 19.1.2 $\gamma_2(\mathsf{G}) = \frac{1}{1-\widehat{\lambda}(\mathsf{G})}$, which implies that $\widehat{\lambda}(\mathsf{G}) = 1 - 1/\gamma_2(\mathsf{G})$, and thus

$$\gamma_2\left(\mathsf{G}^2\right) = \frac{1}{1-\widehat{\lambda}(\mathsf{G}^2)} = \frac{1}{1-\left(\widehat{\lambda}(\mathsf{G})\right)^2} = \frac{1}{1-\left(1-\frac{1}{\gamma_2(\mathsf{G})}\right)^2} = \frac{\gamma_2(\mathsf{G})}{2-\frac{1}{\gamma_2(\mathsf{G})}} = \frac{(\gamma_2(\mathsf{G}))^2}{2\gamma_2(\mathsf{G})-1}.$$

■

20.1.4 The construction

So, let build an expander using Theorem 19.2.3, with parameters $r = 7$ $q = 2^4 = 32$. Let $d = q^2 = 256$. The resulting graph H has $N = q^{r+1} = d^4$ vertices, and it is $d = q^2$ regular. Furthermore, $\widehat{\lambda}_i \leq r/q = 7/32$, for all $i \geq 2$. As such, we have

$$\gamma(\mathsf{H}) = \gamma_2(\mathsf{H}) = \frac{1}{1-7/32} = \frac{32}{25}.$$

Let G_0 be any graph that its square is the complete graph over $n_0 = N + 1$ vertices. Observe that G_0^2 is d^4-regular. Set $\mathsf{G}_i = \left(\mathsf{G}_{i-1}^2 \circledcirc \mathsf{H}\right)$, Clearly, the graph G_i has

$$n_i = n_{i-1}N$$

vertices. The graph $\mathsf{G}_{i-1}^2 \textcircled{z} \mathsf{H}$ is d^2 regular. As far as the bi-tension, let $\alpha_i = \gamma_2(\mathsf{G}_i)$. We have that

$$\alpha_i = \frac{\alpha_{i-1}^2}{2\alpha_{i-1} - 1}(\gamma_2(\mathsf{H}))^2 = \frac{\alpha_{i-1}^2}{2\alpha_{i-1} - 1}\left(\frac{32}{25}\right)^2 \leq 1.64\frac{\alpha_{i-1}^2}{2\alpha_{i-1} - 1}.$$

It is now easy to verify, that α_i can not be bigger than 5.

Theorem 20.1.5. *For any $i \geq 0$, one can compute deterministically a graph G_i with $n_i = (d^4+1)d^{4i}$ vertices, which is d^2 regular, where $d = 256$. The graph G_i is a $(1/10)$-expander.*

Proof: The construction is described above. As for the expansion, since the bi-tension bounds the tension of a graph, we have that $\gamma(\mathsf{G}_i) \leq \gamma_2(\mathsf{G}_i) \leq 5$. Now, by Lemma 18.2.2, we have that G_i is a δ-expander, where $\delta \geq 1/(2\gamma(\mathsf{G}_i)) \geq 1/10$. ■

20.2 Bibliographical notes

A good survey on expanders is the monograph by Hoory *et al.* [HLW06]. The small expander construction is from the paper by Alon *et al.* [ASS08] (but its originally from the work by Along and Roichman [AR94]). The work by Alon *et al.* [ASS08] contains a construction of an expander that is constant degree, which is of similar complexity to the one we presented here. Instead, we used the zig-zag expander construction from the influential work of Reingold *et al.* [RVW02]. Our analysis however, is from an upcoming paper by Mendel and Naor [MN08]. This analysis is arguably reasonably simple (as simplicity is in the eye of the beholder, we will avoid claim that its the simplest), and (even better) provide a good intuition and a systematic approach to analyzing the expansion.

We took a creative freedom in naming notations, and the name tension and bi-tension are the author's own invention.

20.3 Exercises

Exercise 20.3.1 (Expanders made easy.). By considering a random bipartite three-regular graph on $2n$ vertices obtained by picking three random permutations between the two sides of the bipartite graph, prove that there is a $c > 0$ such that for every n there exits a $(2n, 3, c)$-expander. (What is the value of c in your construction?)

Exercise 20.3.2 (Is your consistency in vain?). In the construction, we assumed that the graphs we are dealing with when building expanders have consistent labeling. This can be enforced by working with bipartite graphs, which implies modifying the construction slightly.

(A) Prove that a d-regular bipartite graph always has a consistent labeling (hint: consider matchings in this graph).

(B) Prove that if G is bipartite so is the graph G^3 (the cubed graph).

(C) Let G be a (n, D)-graph and let H be a (D, d)-graph. Prove that if G is bipartite then $\mathsf{G}\mathsf{G}\textcircled{z}\mathsf{H}$ is bipartite.

(D) Describe in detail a construction of an expander that is: (i) bipartite, and (ii) has consistent labeling at every stage of the construction (prove this property if necessary). For the ith graph in your series, what is its vertex degree, how many vertices it has, and what is the quality of expansion it provides?

Exercise 20.3.3 (TENSION AND BI-TENSION.). [30 points]

Disprove (i.e., give a counter example) that there exists a universal constant c, such that for any connected graph G, we have that $\gamma(\mathsf{G}) \leq \gamma_2(\mathsf{G}) \leq c\gamma(\mathsf{G})$.

Acknowledgements

Much of the presentation was followed suggestions by Manor Mendel. He also contributed some of the figures.

Chapter 21

Random Walks V

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> "Is there anything in the Geneva Convention about the rules of war in peacetime?" Stanko wanted to know, crawling back toward the truck. "Absolutely nothing," Caulec assured him. "The rules of war apply only in wartime. In peacetime, anything goes."
>
> – Romain Gary, Gasp.

21.1 Rapid mixing for expanders

We remind the reader of the following definition of expander.

Definition 21.1.1. Let $\mathsf{G} = (V, E)$ be an undirected d-regular graph. The graph G is a (n, d, c)-***expander*** (or just c-***expander***), for every set $S \subseteq V$ of size at most $|V|/2$, there are at least $cd|S|$ edges connecting S and $\overline{S} = V \setminus S$; that is $e\left(S, \overline{S}\right) \geq cd|S|$,

Guaranteeing aperiodicity Let G be a (n, d, c)-expander. We would like to perform a random walk on G. The graph G is connected, but it might be periodic (i.e., bipartite). To overcome this, consider the random walk on G that either stay in the current state with probability $1/2$ or traverse one of the edges. Clearly, the resulting Markov Chain (MC) is aperiodic. The resulting transition matrix is

$$\mathsf{Q} = \mathsf{M}/2d + \mathcal{I}/2,$$

where M is the adjacency matrix of G and $\mathcal{I}$ is the identity $n \times n$ matrix. Clearly Q is doubly stochastic. Furthermore, if $\widehat{\lambda_i}$ is an eigenvalue of M, with eigenvector v_i, then

$$\mathsf{Q}v_i = \frac{1}{2}\left(\frac{\mathsf{M}}{d} + \mathcal{I}\right)v_i = \frac{1}{2}\left(\frac{\widehat{\lambda_i}}{d} + 1\right)v_i.$$

As such, $\left(\widehat{\lambda_i}/d + 1\right)/2$ is an eigenvalue of Q. Namely, if there is a spectral gap in the graph G, there would also be a similar spectral gap in the resulting MC. This MC can be generated by adding to each vertex d self loops, ending up with a $2d$-regular graph. Clearly, this graph is still an expander if the original graph is an expander, and the random walk on it is aperiodic.

From this point on, we would just assume our expander is aperiodic.

21.1.1 Bounding the mixing time

For a MC with n states, we denote by $\pi = (\pi_1, \ldots, \pi_n)$ its stationary distribution. We consider only nicely behave MC that fall under Theorem 15.2.10. As such, no state in the MC has zero stationary probability.

Definition 21.1.2. Let $\mathsf{q}^{(t)}$ denote the state probability vector of a Markov chain defined by a transition matrix Q at time $t \geq 0$, given an initial distribution $\mathsf{q}^{(0)}$. The ***relative pairwise distance*** of the Markov chain at time t is

$$\Delta(t) = \max_i \frac{\left|\mathsf{q}_i^{(t)} - \pi_i\right|}{\pi_i}.$$

Namely, if $\Delta(t)$ approaches zero then $\mathsf{q}^{(t)}$ approaches π.

We remind the reader that we saw a construction of a constant degree expander with constant expansion. In its transition matrix Q, we have that $\widehat{\lambda_1} = 1$, and $-1 \leq \widehat{\lambda_2} < 1$, and furthermore the ***spectral gap*** $\widehat{\lambda_1} - \widehat{\lambda_2}$ was a constant (the two properties are equivalent, but we proved only one direction of this).

In fact, we will need a slightly stronger property (that does hold for our expander construction). We have that $\widehat{\lambda_2} \geq \max_{i=2}^{n} \left|\widehat{\lambda_i}\right|$.

Theorem 21.1.3. *Let Q be the transition matrix of an aperiodic (n, d, c)-expander. Then, for any initial distribution $\mathsf{q}^{(0)}$, we have that*

$$\Delta(t) \leq n^{3/2} \widehat{\lambda_2}^t.$$

Namely, since $\widehat{\lambda_2}$ is a constant smaller than 1, the distance Δt drops exponentially with t.

Proof: We have that $\mathsf{q}^{(t)} = \mathsf{q}^{(0)}\mathsf{Q}^t$. Let $\mathcal{B}(\mathsf{Q}) = \langle v_1, \ldots, v_n \rangle$ denote the orthonormal eigenvector basis of Q (see Definition 18.2.3), and write $\mathsf{q}^{(0)} = \sum_{i=1}^{n} \alpha_i v_i$. Since $\widehat{\lambda_1} = 1$, we have that

$$\mathsf{q}^{(t)} = \mathsf{q}^{(0)}\mathsf{Q}^t = \sum_{i=1}^{n} \alpha_i (v_i \mathsf{Q}^t) = \sum_{i=1}^{n} \alpha_i \widehat{\lambda_i}^t v_i = \alpha_1 v_1 + \sum_{i=2}^{n} \alpha_i \widehat{\lambda_i}^t v_i.$$

Since $v_1 = 1^n/\sqrt{n}$, and $\left|\widehat{\lambda_i}\right| \leq \widehat{\lambda_2} < 1$, for $i > 1$, we have that $\lim_{t\to\infty} \widehat{\lambda_i}^t = 0$, and thus

$$\pi = \lim_{t\to\infty} \mathsf{q}^{(t)} = \alpha_1 v_1 + \sum_{i=2}^{n} \alpha_i \left(\lim_{t\to\infty} \widehat{\lambda_i}^t\right) v_i = \alpha_1 v_1.$$

Now, since $v_1, \ldots, v_n$ is an orthonormal basis, and $\mathsf{q}^{(0)} = \sum_{i=1}^{n} \alpha_i v_i$, we have that $\left\|\mathsf{q}^{(0)}\right\|_2 = \sqrt{\sum_{i=1}^{n} \alpha_i^2}$.

Thuis implies that

$$\begin{aligned}
\left\|\mathsf{q}^{(t)} - \pi\right\|_1 &= \left\|\mathsf{q}^{(t)} - \alpha_1 v_1\right\|_1 = \left\|\sum_{i=2}^{n} \alpha_i \left(\widehat{\lambda_i}\right)^t v_i\right\|_1 \leq \sqrt{n} \left\|\sum_{i=2}^{n} \alpha_i \left(\widehat{\lambda_i}\right)^t v_i\right\|_2 = \sqrt{n} \sqrt{\sum_{i=2}^{n} \left(\alpha_i \left(\widehat{\lambda_i}\right)^t\right)^2} \\
&\leq \sqrt{n}\left(\widehat{\lambda_2}\right)^t \sqrt{\sum_{i=2}^{n} (\alpha_i)^2} \leq \sqrt{n}\left(\widehat{\lambda_2}\right)^t \left\|\mathsf{q}^{(0)}\right\|_2 \leq \sqrt{n}\left(\widehat{\lambda_2}\right)^t \left\|\mathsf{q}^{(0)}\right\|_1 = \sqrt{n}\left(\widehat{\lambda_2}\right)^t,
\end{aligned}$$

since $\mathsf{q}^{(0)}$ is a distribution. Now, since $\pi_i = 1/n$, we have

$$\Delta(t) = \max_i \frac{\left|\mathsf{q}_i^{(t)} - \pi_i\right|}{\pi_i} = \max_i n\left|\mathsf{q}_i^{(t)} - \pi_i\right| \leq n \max_i \left\|\mathsf{q}^{(t)} - \pi\right\|_1 \leq n\sqrt{n}\left(\widehat{\lambda}_2\right)^t.$$

■

21.2 Probability amplification by random walks on expanders

We are interested in performing probability amplification for an algorithm that is a **BPP** algorithm (see Definition 3.1.9). It would be convenient to work with an algorithm which is already somewhat amplified. That is, we assume that we are given a **BPP** algorithm **Alg** for a language L, such that :

(i) If $x \in L$ then $\mathbf{Pr}[\mathbf{Alg}(x) \text{ accepts}] \geq 199/200$.

(ii) If $x \notin L$ then $\mathbf{Pr}[\mathbf{Alg}(x) \text{ accepts}] \leq 1/200$.

We assume that **Alg** requires a random bit string of length n. So, we have a constant degree expander G (say of degree d) that has at least $200 \cdot 2^n$ vertices. In particular, let $U = |V(\mathsf{G})|$ and observe that since our expander construction grow exponentially in size (but the base of the exponent is a constant), we have that $U = O(2^n)$. (Translation: We can not quite get an expander with a specific number of vertices. Rather, we can guarantee an expander that has more vertices than we need, but not many more.)

We label the vertices of G with all the binary strings of length n, in a round robin fashion (thus, each binary string of length n appears either $\lceil |V(\mathsf{G})| / 2^n \rceil$ or $\lfloor |V(\mathsf{G})| / 2^n \rfloor$ times). For a vertex $v \in V(\mathsf{G})$, let $\mathsf{s}(v)$ denote the binary string associated with v.

Consider a string x that we would like to decide if it is in L or not. We know that at least $99/100U$ vertices of G are labeled with "random" strings that would yield the right result if we feed them into **Alg** (the constant here deteriorated from 199/200 to 99/100 because the number of times a string appears is not identically the same for all strings).

The algorithm. We perform a random walk of length $\mu = \alpha\beta k$ on G, where α and β are constants to be determined shortly. To this end, we randomly choose a starting vertex X_0 (this would require $n + O(1)$ bits). Every step in the random walk, would require $O(1)$ random bits, as the expander is a constant degree expander, and as such overall, this would require $n + O(k)$ random bits.

Now, lets $X_0, X_1, \ldots, X_\mu$ be the resulting random walk. We compute the result of

$$Y_i = \mathbf{Alg}(x, r_i), \text{ for } i = 0, \ldots, K = \alpha k,$$

where $r_i = \mathsf{s}\left(X_{i \cdot \beta}\right)$. Specifically, we use the strings associated with nodes that are in distance β from each other along the path of the random walk. We return the majority of the bits $Y_0, \ldots, Y_{\alpha k}$ as the decision of whether $x \in L$ or not.

We assume here that we have a ***fully explicit*** construction of an expander. That is, given a vertex of an expander, we can compute all its neighbors in polynomial time (in the length of the index of the vertex). While the construction of expander shown is only explicit it can be made fully explicit with more effort.

21.2.1 The analysis

Intuition. Skipping every β nodes in the random walk corresponds to performing a random walk on the graph G^k; that is, we raise the graph to power k. This new graph is a much better expander (but the degree had deteriorated). Now, consider a specific input x, and mark the bad vertices for it in the graph G. Clearly, we mark at most $1/100$ fraction of the vertices. Conceptually, think about these vertices as being uniformly spread in the graph and far apart. From the execution of the algorithm to fail, the random walk needs to visit $\alpha k/2$ bad vertices in the random walk in G^k. However, the probability for that is extremely small - why would the random walk keep stumbling into bad vertices, when they are so infrequent?

The real thing. Let Q be the transition matrix of G. We assume, as usual, that the random walk on G is aperiodic (if not, we can easily fix it using standard tricks), and thus ergodic. Let $\mathsf{B} = \mathsf{Q}^\beta$ be the transition matrix of the random walk of the states we use in the algorithm. Note, that the eigenvalues (except the first one) of B "shrink". In particular, by picking β to be a sufficiently large constant, we have that $\widehat{\lambda_1}(\mathsf{B}) = 1$ and $\left|\widehat{\lambda_i}(\mathsf{B})\right| \leq 1/10$, for $i = 2, \ldots, n$.

For the input string x, let W be the matrix that has 1 in W_{ii} if and only $\mathbf{Alg}(x, \mathsf{s}(i))$ returns the right answer, and zero everywhere else. Similarly, let $\overline{\mathsf{W}} = \mathcal{I} - \mathsf{W}$ be the "complement" matrix having 1 at $\overline{\mathsf{W}}_{ii}$ iff $\mathbf{Alg}(x, \mathsf{s}(i))$ is incorrect. We know that W is a $U \times U$ matrix, that has at least $(99/100)U$ ones on its diagonal.

Lemma 21.2.1. *Let Q be a symmetric transition matrix, then all its eigenvalues of Q are in the range $[-1, 1]$.*

Proof: Let $p \in \mathbb{R}^n$ be an eigenvector with eigenvalue λ. Let p_i be the coordinate with the maximum absolute value in p. We have that

$$|\lambda p_i| = \left|(p\mathsf{Q})_i\right| = \left|\sum_{j=1}^{n} p_j \mathsf{Q}_{ji}\right| \leq \sum_{j=1}^{n} \left|p_j\right| \left|\mathsf{Q}_{ji}\right| \leq |p_i| \sum_{j=1}^{n} \left|\mathsf{Q}_{ji}\right| = |p_i| .$$

This implies that $|\lambda| \leq 1$.

(We used the symmetry of the matrix, in implying that Q eigenvalues are all real numbers.) ∎

Lemma 21.2.2. *Let Q be a symmetric transition matrix, then for any $p \in \mathbb{R}^n$, we have that $\|p\mathsf{Q}\|_2 \leq \|p\|_2$.*

Proof: Let $\mathcal{B}(\mathsf{Q}) = \langle v_1, \ldots, v_n \rangle$ denote the orthonormal eigenvector basis of Q, with eigenvalues $1 = \lambda_1, \ldots, \lambda_n$. Write $p = \sum_i \alpha_i v_i$, and observe that

$$\|p\mathsf{Q}\|_2 = \left\|\sum_i \alpha_i v_i \mathsf{Q}\right\|_2 = \left\|\sum_i \alpha_i \lambda_i v_i\right\|_2 = \sqrt{\sum_i \alpha_i^2 \lambda_i^2} \leq \sqrt{\sum_i \alpha_i^2} = \|p\|_2 ,$$

since $|\lambda_i| \leq 1$, for $i = 1, \ldots, n$, by Lemma 21.2.1. ∎

Lemma 21.2.3. *Let $\mathsf{B} = \mathsf{Q}^\beta$ be the transition matrix of the graph G^β. For all vectors $p \in \mathbb{R}^n$, we have: (i) $\|p\mathsf{B}\mathsf{W}\|_2 \leq \|p\|_2$, and (ii) $\left\|p\mathsf{B}\overline{\mathsf{W}}\right\| \leq \|p\|/5$.*

Proof: (i) Since multiplying a vector by W has the effect of zeroing out some coordinates, its clear that it can not enlarge the norm of a matrix. As such, $\|p\mathsf{BW}\|_2 \leq \|p\mathsf{B}\|_2 \leq \|p\|_2$ by Lemma 21.2.2.

(ii) Write $p = \sum_i \alpha_i v_i$, where $v_1, \ldots, v_n$ is the orthonormal basis of Q (and thus also of B), with eigenvalues $1 = \widehat{\lambda}_1, \ldots, \widehat{\lambda}_n$. We remind the reader that $v_1 = (1, 1, \ldots, 1)/\sqrt{n}$. Since $\overline{\mathsf{W}}$ zeroes out at least $99/100$ of the entries of a vectors it is multiplied by (and copy the rest as they are), we have that $\left\|v_1\overline{\mathsf{W}}\right\| \leq \sqrt{(n/100)(1/\sqrt{n})^2} \leq 1/10 = \|v_1\|/10$. Now, for any $x \in \mathbb{R}^n$, we have $\left\|x\overline{\mathsf{W}}\right\| \leq \|x\|$. As such, we have that

$$\begin{aligned}
\left\|p\mathsf{B}\overline{\mathsf{W}}\right\|_2 &= \left\|\sum_i \alpha_i v_i \mathsf{B}\overline{\mathsf{W}}\right\|_2 \leq \left\|\alpha_1 v_1 \mathsf{B}\overline{\mathsf{W}}\right\| + \left\|\sum_{i=2}^{n} \alpha_i v_i \mathsf{B}\overline{\mathsf{W}}\right\| \\
&\leq \left\|\alpha_1 v_1 \overline{\mathsf{W}}\right\| + \left\|\left(\sum_{i=2}^{n} \alpha_i v_i \widehat{\lambda}_i^{\beta}\right)\overline{\mathsf{W}}\right\| \leq \frac{|\alpha_1|}{10} + \left\|\sum_{i=2}^{n} \alpha_i v_i \widehat{\lambda}_i^{\beta}\right\| \\
&\leq \frac{|\alpha_1|}{10} + \sqrt{\sum_{i=2}^{n}\left(\alpha_i \widehat{\lambda}_i^{\beta}\right)^2} \leq \frac{|\alpha_1|}{10} + \frac{1}{10}\sqrt{\sum_{i=2}^{n}\alpha_i^2} \leq \frac{\|p\|}{10} + \frac{1}{10}\|p\| \leq \frac{\|p\|}{5},
\end{aligned}$$

since $\left|\lambda_i^{\beta}\right| \leq 1/10$, for $i = 2, \ldots, n$. ∎

Consider the strings $r_0, \ldots, r_K$. For each one of these strings, we can write down whether its a "good" string (i.e., **Alg** return the correct result), or a bad string. This results in a binary pattern $b_0, \ldots, b_K$. Given a distribution $p \in \mathbb{R}^n$ on the states of the graph, its natural to ask what is the probability of being in a "good" state. Clearly, this is the quantity $\|p\mathsf{W}\|_1$. Thus, if we are interested in the probability of a specific pattern, then we should start with the initial distribution p^0, truncate away the coordinates that represent an invalid state, apply the transition matrix, again truncate away forbidden coordinates, and repeat in this fashion till we exhaust the pattern. Clearly, the ℓ_1-norm of the resulting vector is the probability of this pattern. To this end, given a pattern $b_0, \ldots, b_K$, let $\mathcal{S} = \langle S_0, \ldots, S_K \rangle$ denote the corresponding sequence of "truncating" matrices (i.e., S_i is either W or $\overline{\mathsf{W}}$). Formally, we set $S_i = \mathsf{W}$ if **Alg**(x, r_i) returns the correct answer, and set $S_i = \overline{\mathsf{W}}$ otherwise.

The above argument implies the following lemma.

Lemma 21.2.4. *For any fixed pattern $b_0, \ldots, b_K$ the probability of the random walk to generate this pattern of random strings is $\left\|p^{(0)} S_0 \mathsf{B} S_1 \ldots \mathsf{B} S_K\right\|_1$, where $\mathcal{S} = \langle S_0, \ldots, S_K \rangle$ is the sequence of $\mathsf{W}, \overline{\mathsf{W}}$ encoded by this pattern.*

Theorem 21.2.5. *The probability that the majority of the outputs **Alg**(x, r_0), **Alg**(x, r_1), $\ldots$, **Alg**(x, r_K) is incorrect is at most $1/2^k$.*

Proof: The majority is wrong, only if (at least) half the elements of the sequence $\mathcal{S} = \langle S_0, \ldots, S_K \rangle$ belong to $\overline{\mathsf{W}}$. Fix such a "bad" sequence $\mathcal{S}$, and observe that the distributions we work with are vectors in $\mathbb{R}^U$. As such, if p^0 is the initial distribution, then we have that

$$\mathbf{Pr}[\mathcal{S}] = \left\|p^{(0)} S_0 \mathsf{B} S_1 \ldots \mathsf{B} S_K\right\|_1 \leq \sqrt{U}\left\|p^{(0)} S_0 \mathsf{B} S_1 \ldots \mathsf{B} S_K\right\|_2 \leq \sqrt{U}\frac{1}{5^{K/2}}\left\|p^{(0)}\right\|_2,$$

by repeatedly applying Lemma 21.2.3, since half of the sequence $\mathcal{S}$ are $\overline{\mathsf{W}}$, and the rest are W. The distribution $p^{(0)}$ was uniform, which implies that $\left\|p^{(0)}\right\|_2 = 1/\sqrt{U}$. As such, we have

$$\mathbf{Pr}[\text{ majority is bad }] \leq 2^K \sqrt{U} \frac{1}{5^{K/2}} \left\|p^{(0)}\right\|_2 = (4/5)^{K/2} = (4/5)^{\alpha k/2} \leq \frac{1}{2^k},$$

for $\alpha = 7$. ■

Chapter 22

The Johnson-Lindenstrauss Lemma

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> Dixon was alive again. Consciousness was upon him before he could get out of the way; not for him the slow, gracious wandering from the halls of sleep, but a summary, forcible ejection. He lay sprawled, too wicked to move, spewed up like a broken spider-crab on the tarry shingle of the morning. The light did him harm, but not as much as looking at things did; he resolved, having done it once, never to move his eyeballs again. A dusty thudding in his head made the scene before him beat like a pulse. His mouth had been used as a latrine by some small creature of the night, and then as its mausoleum. During the night, too, he'd somehow been on a cross-country run and then been expertly beaten up by secret police. He felt bad.
>
> – Lucky Jim, Kingsley Amis.

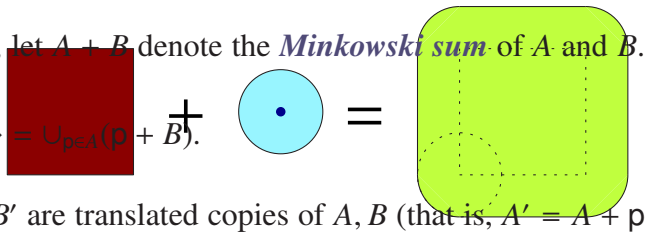
In this chapter, we will prove that given a set P of n points in $\mathbb{R}^d$, one can reduce the dimension of the points to $k = O(\varepsilon^{-2} \log n)$ such that distances are $1 \pm \varepsilon$ preserved. Surprisingly, this reduction is done by randomly picking a subspace of k dimensions and projecting the points into this random subspace. One way of thinking about this result is that we are "compressing" the input of size nd (i.e., n points with d coordinates) into size $O(n\varepsilon^{-2} \log n)$, while (approximately) preserving distances.

22.1 The Brunn-Minkowski inequality

For a set $A \subseteq \mathbb{R}^d$, an a point $\mathsf{p} \in \mathbb{R}^d$, let $A + \mathsf{p}$ denote the translation of A by p. Formally, $A + \mathsf{p} = \left\{ \mathsf{q} + \mathsf{p} \;\middle|\; \mathsf{q} \in A \right\}$.

Definition 22.1.1. For two sets A and B in $\mathbb{R}^n$, let $A + B$ denote the ***Minkowski sum*** of A and B. Formally,

$$A + B = \left\{ a + b \;\middle|\; a \in A, b \in B \right\} = \cup_{\mathsf{p} \in A} (\mathsf{p} + B).$$



Remark 22.1.2. It is easy to verify that if A', B' are translated copies of A, B (that is, $A' = A + \mathsf{p}$ and $B = B + \mathsf{q}$, for some points $\mathsf{p}, \mathsf{q} \in \mathbb{R}^d$), respectively, then $A' + B'$ is a translated copy of $A + B$. In particular, since volume is preserved under translation, we have that $\mathrm{Vol}(A' + B') = \mathrm{Vol}((A + B) + \mathsf{p} + \mathsf{q}) = \mathrm{Vol}(A + B)$.

Our purpose here is to prove the following theorem.

Theorem 22.1.3 (Brunn-Minkowski inequality). *Let A and B be two non-empty compact sets in $\mathbb{R}^n$. Then*

$$\mathrm{Vol}(A+B)^{1/n} \geq \mathrm{Vol}(A)^{1/n} + \mathrm{Vol}(B)^{1/n}.$$

Definition 22.1.4. A set $A \subseteq \mathbb{R}^n$ is a ***brick set*** if it is the union of finitely many (close) axis parallel boxes with disjoint interiors.

It is intuitively clear, by limit arguments, that proving Theorem 22.1.3 for brick sets will imply it for the general case.

Lemma 22.1.5 (Brunn-Minkowski inequality for Brick Sets). *Let A and B be two non-empty brick sets in $\mathbb{R}^n$. Then*

$$\mathrm{Vol}(A+B)^{1/n} \geq \mathrm{Vol}(A)^{1/n} + \mathrm{Vol}(B)^{1/n}$$

Proof: By induction on the number k of bricks in A and B. If $k = 2$ then A and B are just bricks, with dimensions $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$, respectively. In this case, the dimensions of $A + B$ are $a_1+b_1, \ldots, a_n+b_n$, as can be easily verified. Thus, we need to prove that $\left(\prod_{i=1}^n a_i\right)^{1/n} + \left(\prod_{i=1}^n b_i\right)^{1/n} \leq \left(\prod_{i=1}^n (a_i+b_i)\right)^{1/n}$. Dividing the left side by the right side, we have

$$\left(\prod_{i=1}^n \frac{a_i}{a_i+b_i}\right)^{1/n} + \left(\prod_{i=1}^n \frac{b_i}{a_i+b_i}\right)^{1/n} \leq \frac{1}{n}\sum_{i=1}^n \frac{a_i}{a_i+b_i} + \frac{1}{n}\sum_{i=1}^n \frac{b_i}{a_i+b_i} = 1,$$

by the generalized arithmetic-geometric mean inequality[①], and the claim follows for this case.

Now let $k > 2$ and suppose that the Brunn-Minkowski inequality holds for any pair of brick sets with fewer than k bricks (together). Let A and B be a pair of sets having k bricks together, the A has at least two (disjoint) bricks. However, this implies that there is an axis parallel hyperplane h that separates the interior of one brick of A from the interior of another brick of A (the hyperplane h might intersect other bricks of A). Assume that h is the hyperplane $x_1 = 0$ (this can be achieved by translation and renaming of coordinates).

Let $\overline{A^+} = A \cap h^+$ and $\overline{A^-} = A \cap h^-$, where h^+ and h^- are the two open half spaces induced by h. Let A^+ and A^- be the closure of $\overline{A^+}$ and $\overline{A^-}$, respectively. Clearly, A^+ and A^- are both brick sets with (at least) one fewer brick than A.

Next, observe that the claim is translation invariant (see Remark 22.1.2), and as such, let us translate B so that its volume is split by h in the same ratio A's volume is being split. Denote the two parts of B by B^+ and B^-, respectively. Let $\rho = \mathrm{Vol}(A^+)/\mathrm{Vol}(A) = \mathrm{Vol}(B^+)/\mathrm{Vol}(B)$ (if $\mathrm{Vol}(A) = 0$ or $\mathrm{Vol}(B) = 0$ the claim trivially holds).

Observe, that $A^+ + B^+ \subseteq A + B$, and it lies on one side of h (since $h \equiv (x_1 = 0)$), and similarly $A^- + B^- \subseteq A + B$ and it lies on the other side of h. Thus, by induction and since $A^+ + B^+$ and $A^- + B^-$

[①]Here is a proof of the generalized form: Let $x_1, \ldots, x_n$ be n positive real numbers. Consider the quantity $R = x_1 x_2 \cdots x_n$. If we fix the sum of the n numbers to be equal to α, then R is maximized when all the x_is are equal. Thus, $\sqrt[n]{x_1 x_2 \cdots x_n} \leq \sqrt[n]{(\alpha/n)^n} = \alpha/n = (x_1 + \cdots + x_n)/n$.

are interior disjoint, we have

$$\begin{aligned}
\mathrm{Vol}(A+B) &\geq \mathrm{Vol}(A^+ + B^+) + \mathrm{Vol}(A^- + B^-) \\
&\geq \left(\mathrm{Vol}(A^+)^{1/n} + \mathrm{Vol}(B^+)^{1/n}\right)^n + \left(\mathrm{Vol}(A^-)^{1/n} + \mathrm{Vol}(B^-)^{1/n}\right)^n \\
&= \left[\rho^{1/n}\,\mathrm{Vol}(A)^{1/n} + \rho^{1/n}\,\mathrm{Vol}(B)^{1/n}\right]^n \\
&\quad + \left[(1-\rho)^{1/n}\,\mathrm{Vol}(A)^{1/n} + (1-\rho)^{1/n}\,\mathrm{Vol}(B)^{1/n}\right]^n \\
&= (\rho + (1-\rho))\left[\mathrm{Vol}(A)^{1/n} + \mathrm{Vol}(B)^{1/n}\right]^n \\
&= \left[\mathrm{Vol}(A)^{1/n} + \mathrm{Vol}(B)^{1/n}\right]^n,
\end{aligned}$$

establishing the claim. ■

*Proof of **Theorem 22.1.3***: Let $A_1 \subseteq A_2 \subseteq \cdots \subseteq A_i \subseteq \cdots$ be a sequence of finite brick sets, such that $\bigcup_i A_i = A$, and similarly let $B_1 \subseteq B_2 \subseteq \cdots \subseteq B_i \subseteq \cdots$ be a sequence of finite brick sets, such that $\bigcup_i B_i = B$. By the definition of volume[②],we have that $\lim_{i\to\infty} \mathrm{Vol}(A_i) = \mathrm{Vol}(A)$ and $\lim_{i\to\infty} \mathrm{Vol}(B_i) = \mathrm{Vol}(B)$.

We claim that $\lim_{i\to\infty} \mathrm{Vol}(A_i + B_i) = \mathrm{Vol}(A+B)$. Indeed, consider any point $z \in A + B$, and let $u \in A$ and $v \in B$ be such that $u + v = z$. By definition, there exists an i, such that for all $j > i$ we have $u \in A_j$, $v \in B_j$, and as such $z \in A_j + B_j$. Thus, $A + B \subseteq \cup_j (A_j + B_j)$ and $\cup_j (A_j + B_j) \subseteq \cup_j (A + B) \subseteq A + B$; namely, $\cup_j (A_j + B_j) = A + B$.

Furthermore, for any $i > 0$, since A_i and B_i are brick sets, we have

$$\mathrm{Vol}(A_i + B_i)^{1/n} \geq \mathrm{Vol}(A_i)^{1/n} + \mathrm{Vol}(B_i)^{1/n},$$

by Lemma 22.1.5. Thus,

$$\begin{aligned}
\mathrm{Vol}(A+B)^{1/n} &= \lim_{i\to\infty} \mathrm{Vol}(A_i + B_i)^{1/n} \geq \lim_{i\to\infty} \left(\mathrm{Vol}(A_i)^{1/n} + \mathrm{Vol}(B_i)^{1/n}\right) \\
&= \mathrm{Vol}(A)^{1/n} + \mathrm{Vol}(B)^{1/n}.
\end{aligned}$$

■

Theorem 22.1.6 (Brunn-Minkowski for slice volumes.). *Let $\mathcal{P}$ be a convex set in $\mathbb{R}^{n+1}$, and let $A = \mathcal{P} \cap (x_1 = a)$, $B = \mathcal{P} \cap (x_1 = b)$ and $C = \mathcal{P} \cap (x_1 = c)$ be three slices of A, for $a < b < c$. We have* $\mathrm{Vol}(B) \geq \min(\mathrm{Vol}(A), \mathrm{Vol}(C))$.

In fact, consider the function

$$v(t) = (\mathrm{Vol}(\mathcal{P} \cap (x_1 = t)))^{1/n},$$

and let $\mathcal{I} = [t_{min}, t_{max}]$ be the interval where the hyperplane $x_1 = t$ intersects $\mathcal{P}$. Then, $v(t)$ is concave on $\mathcal{I}$.

Proof: If a or c are outside $\mathcal{I}$, then $\mathrm{Vol}(A) = 0$ or $\mathrm{Vol}(C) = 0$, respectively, and then the claim trivially holds.

[②]This is the standard definition in measure theory of volume. The reader unfamiliar with this fanfare can either consult a standard text on the topic, or take it for granted as this is intuitively clear.

Otherwise, let $\alpha = (b-a)/(c-a)$. We have that $b = (1-\alpha)\cdot a + \alpha \cdot c$, and by the convexity of $\mathcal{P}$, we have $(1-\alpha)A + \alpha C \subseteq B$. Thus, by Theorem 22.1.3 we have

$$\begin{aligned} v(b) = \mathrm{Vol}(B)^{1/n} &\geq \mathrm{Vol}((1-\alpha)A + \alpha C)^{1/n} \geq \mathrm{Vol}((1-\alpha)A)^{1/n} + \mathrm{Vol}(\alpha C)^{1/n} \\ &= \left((1-\alpha)^n \mathrm{Vol}(A)\right)^{1/n} + \left(\alpha^n \mathrm{Vol}(C)\right)^{1/n} \\ &= (1-\alpha)\cdot \mathrm{Vol}(A)^{1/n} + \alpha \cdot \mathrm{Vol}(C)^{1/n} \\ &= (1-\alpha)v(a) + \alpha v(c). \end{aligned}$$

Namely, $v(\cdot)$ is concave on $\mathcal{I}$, and in particular $v(b) \geq \min(v(a), v(c))$, which in turn implies that $\mathrm{Vol}(B) = v(b)^n \geq (\min(v(a), v(c)))^n = \min(\mathrm{Vol}(A), \mathrm{Vol}(C))$, as claimed. ■

Corollary 22.1.7. *For A and B compact sets in $\mathbb{R}^n$, the following holds* $\mathrm{Vol}((A+B)/2) \geq \sqrt{\mathrm{Vol}(A)\,\mathrm{Vol}(B)}$.

Proof: We have that $\mathrm{Vol}((A+B)/2)^{1/n} = \mathrm{Vol}(A/2 + B/2)^{1/n} \geq \mathrm{Vol}(A/2)^{1/n} + \mathrm{Vol}(B/2)^{1/n} = (\mathrm{Vol}(A)^{1/n} + \mathrm{Vol}(B)^{1/n})/2 \geq \sqrt{\mathrm{Vol}(A)^{1/n}\,\mathrm{Vol}(B)^{1/n}}$ by Theorem 22.1.3, and since $(a+b)/2 \geq \sqrt{ab}$ for any $a, b \geq 0$. The claim now follows by raising this inequality to the power n. ■

22.1.1 The Isoperimetric Inequality

The following is not used anywhere else and is provided because of its mathematical elegance. The skip-able reader can thus employ their special gift and move on to Section 22.2.

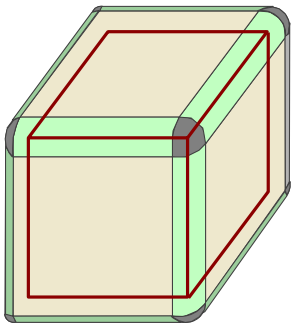
The ***isoperimetric inequality*** states that among all convex bodies of a fixed surface area, the ball has the largest volume (in particular, the unit circle is the largest area planar region with perimeter 2π). This problem can be traced back to antiquity, in particular Zenodorus (200–140 BC) wrote a monograph (which was lost) that seemed to have proved the claim in the plane for some special cases. The first formal proof for the planar case was done by Steiner in 1841. Interestingly, the more general claim is an easy consequence of the Brunn-Minkowski inequality.

Let K be a convex body in $\mathbb{R}^n$ and $\mathbf{b}$ be the n dimensional ball of radius one centered at the origin. Let $\mathsf{S}(X)$ denote the surface area of a compact set $X \subseteq \mathbb{R}^n$. The ***isoperimetric inequality*** states that

$$\left(\frac{\mathrm{Vol}(K)}{\mathrm{Vol}(\mathbf{b})}\right)^{1/n} \leq \left(\frac{\mathsf{S}(K)}{\mathsf{S}(\mathbf{b})}\right)^{1/(n-1)}. \tag{22.1}$$

Namely, the left side is the radius of a ball having the same volume as K, and the right side is the radius of a sphere having the same surface area as K. In particular, if we scale K so that its surface area is the same as $\mathbf{b}$, then the above inequality implies that $\mathrm{Vol}(K) \leq \mathrm{Vol}(\mathbf{b})$.

To prove Eq. (22.1), observe that $\mathrm{Vol}(\mathbf{b}) = \mathsf{S}(\mathbf{b})/n$[3]. Also, observe that $K + \varepsilon\mathbf{b}$ is the body K together with a small "atmosphere" around it of thickness ε. In particular, the volume of this "atmosphere" is (roughly) $\varepsilon\,\mathsf{S}(K)$ (in fact, Minkowski defined the surface area of a convex body to be the limit stated next).



Formally, we have

$$\begin{aligned}\mathsf{S}(K) &= \lim_{\varepsilon\to 0+} \frac{\mathrm{Vol}(K+\varepsilon\mathbf{b}) - \mathrm{Vol}(K)}{\varepsilon} \\ &\geq \lim_{\varepsilon\to 0+} \frac{\left(\mathrm{Vol}(K)^{1/n} + \mathrm{Vol}(\varepsilon\mathbf{b})^{1/n}\right)^n - \mathrm{Vol}(K)}{\varepsilon},\end{aligned}$$

by the Brunn-Minkowski inequality. Now $\mathrm{Vol}(\varepsilon\mathbf{b})^{1/n} = \varepsilon\,\mathrm{Vol}(\mathbf{b})^{1/n}$, and as such

$$\begin{aligned}\mathsf{S}(K) &\geq \lim_{\varepsilon\to 0+} \frac{\mathrm{Vol}(K) + \binom{n}{1}\varepsilon\,\mathrm{Vol}(K)^{(n-1)/n}\,\mathrm{Vol}(\mathbf{b})^{1/n} + \binom{n}{2}\varepsilon^2\langle\cdots\rangle + \cdots + \varepsilon^n\,\mathrm{Vol}(\mathbf{b}) - \mathrm{Vol}(K)}{\varepsilon} \\ &= \lim_{\varepsilon\to 0+} \frac{n\varepsilon\,\mathrm{Vol}(K)^{(n-1)/n}\,\mathrm{Vol}(\mathbf{b})^{1/n}}{\varepsilon} = n\,\mathrm{Vol}(K)^{(n-1)/n}\,\mathrm{Vol}(\mathbf{b})^{1/n}.\end{aligned}$$

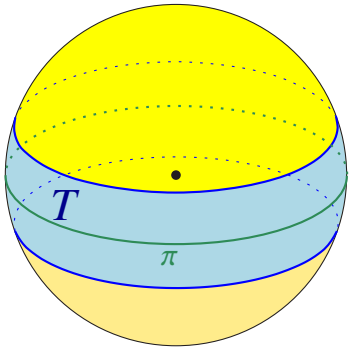
Dividing both sides by $\mathsf{S}(\mathbf{b}) = n\,\mathrm{Vol}(\mathbf{b})$, we have

$$\frac{\mathsf{S}(K)}{\mathsf{S}(\mathbf{b})} \geq \frac{\mathrm{Vol}(K)^{(n-1)/n}}{\mathrm{Vol}(\mathbf{b})^{(n-1)/n}} \quad\Rightarrow\quad \left(\frac{\mathsf{S}(K)}{\mathsf{S}(\mathbf{b})}\right)^{1/(n-1)} \geq \left(\frac{\mathrm{Vol}(K)}{\mathrm{Vol}(\mathbf{b})}\right)^{1/n},$$

establishing the isoperimetric inequality.

22.2 Measure Concentration on the Sphere

Let $\mathbb{S}^{(n-1)}$ be the unit sphere in $\mathbb{R}^n$. We assume there is a uniform probability measure defined over $\mathbb{S}^{(n-1)}$, such that its total measure is 1. Surprisingly, most of the mass of this measure is near the equator. Indeed, consider an arbitrary equator π on $\mathbb{S}^{(n-1)}$ (that it, it is the intersection of the sphere with a hyperplane passing through the center of ball inducing the sphere). Next, consider all the points that are in distance $\approx \ell(n) = c/n^{1/3}$ from π. The question we are interested in is what fraction of the sphere is covered by this strip T (depicted on the right).



Notice, that as the dimension increases the width $\ell(n)$ of this strip decreases. But surprisingly, despite its width becoming smaller, as the dimension increases, this strip contains a larger and larger fraction of the sphere. In particular, the total fraction of the sphere not covered by this (shrinking!) strip converges to zero.

[3] Indeed, $\mathrm{Vol}(\mathbf{b}) = \int_{r=0}^{1} \mathsf{S}(\mathbf{b})\, r^{n-1} dr = \mathsf{S}(\mathbf{b})/n$.

Furthermore, counter intuitively, this is true for *any* equator. We are going to show that even a stronger result holds: The mass of the sphere is concentrated close to the boundary of any set $A \subseteq \mathbb{S}^{(n-1)}$ such that $\mathbf{Pr}[A] = 1/2$.

Before proving this somewhat surprising theorem, we will first try to get an intuition about the behavior of the hypersphere in high dimensions.

22.2.1 The strange and curious life of the hypersphere

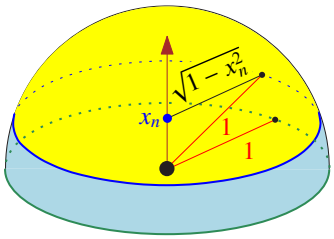
Consider the ball of radius r in $\mathbb{R}^n$ denoted by $r\,\mathbf{b}^n$, where $\mathbf{b}^n$ is the unit radius ball centered at the origin. Clearly, $\mathrm{Vol}(r\,\mathbf{b}^n) = r^n \mathrm{Vol}(\mathbf{b}^n)$. Now, even if r is very close to 1, the quantity r^n might be very close to zero if n is sufficiently large. Indeed, if $r = 1 - \delta$, then $r^n = (1-\delta)^n \leq \exp(-\delta n)$, which is very small if $\delta \gg 1/n$. (Here, we used the fact that $1 - x \leq e^x$, for $x \geq 0$.) Namely, for the ball in high dimensions, its mass is concentrated in a very thin shell close to its surface.

The volume of a ball and the surface area of hypersphere. Let $\mathrm{Vol}(r\mathbf{b}^n)$ denote the volume of the ball of radius r in $\mathbb{R}^n$, and $\mathrm{Area}\left(r\mathbb{S}^{(n-1)}\right)$ denote the surface area of its bounding sphere (i.e., the surface area of $r\mathbb{S}^{(n-1)}$). It is known that

$$\mathrm{Vol}(r\mathbf{b}^n) = \frac{\pi^{n/2} r^n}{\Gamma(n/2+1)} \quad \text{and} \quad \mathrm{Area}\left(r\mathbb{S}^{(n-1)}\right) = \frac{2\pi^{n/2} r^{n-1}}{\Gamma(n/2)},$$

where the gamma function, $\Gamma(\cdot)$, is an extension of the factorial function. Specifically, if n is even then $\Gamma(n/2+1) = (n/2)!$, and for n odd $\Gamma(n/2+1) = \sqrt{\pi}(n!!)/2^{(n+1)/2}$, where $n!! = 1 \cdot 3 \cdot 5 \cdots n$ is the ***double factorial***. The most surprising implication of these two formulas is that, as n increases, the volume of the unit ball first increases (till dimension 5 in fact) and then starts decreasing to zero.

Similarly, the surface area of the unit sphere $\mathbb{S}^{(n-1)}$ in $\mathbb{R}^n$ tends to zero as the dimension increases. To see this, compute the volume of the unit ball using an integral of its slice volume, when it is being sliced by a hyperplanes perpendicular to the nth coordinate.



We have, see figure on the right, that

$$\mathrm{Vol}(\mathbf{b}^n) = \int_{x_n=-1}^{1} \mathrm{Vol}\left(\sqrt{1-x_n^2}\,\mathbf{b}^{n-1}\right) dx_n = \mathrm{Vol}\left(\mathbf{b}^{n-1}\right) \int_{x_n=-1}^{1} \left(1-x_n^2\right)^{(n-1)/2} dx_n,$$

Now, the integral on the right side tends to zero as n increases. In fact, for n very large, the term $\left(1-x_n^2\right)^{(n-1)/2}$ is very close to 0 everywhere except for a small interval around 0. This implies that the main contribution of the volume of the ball happens when we consider slices of the ball by hyperplanes of the form $x_n = \delta$, where δ is small.

If one has to visualize how such a ball in high dimensions looks like, it might be best to think about it as a star-like creature: It has very little mass close to the tips of any set of orthogonal directions we pick, and most of its mass somehow lies on hyperplanes close to its center.[4]

[4] In short, it looks like a Boojum [Car76].

22.2.2 Measure Concentration on the Sphere

Theorem 22.2.1 (Measure concentration on the sphere.). *Let $A \subseteq \mathbb{S}^{(n-1)}$ be a measurable set with $\mathbf{Pr}[A] \geq 1/2$, and let A_t denote the set of points of $\mathbb{S}^{(n-1)}$ in distance at most t from A, where $t \leq 2$. Then $1 - \mathbf{Pr}[A_t] \leq 2\exp\left(-nt^2/2\right)$.*

Proof: We will prove a slightly weaker bound, with $-nt^2/4$ in the exponent. Let $\widehat{A} = T(A)$, where

$$T(X) = \left\{ \alpha x \;\middle|\; x \in X, \alpha \in [0,1] \right\} \subseteq \mathbf{b}^n,$$

and $\mathbf{b}^n$ is the unit ball in $\mathbb{R}^n$. We have that $\mathbf{Pr}[A] = \mu\left(\widehat{A}\right)$, where $\mu\left(\widehat{A}\right) = \mathrm{Vol}\left(\widehat{A}\right) / \mathrm{Vol}(\mathbf{b}^n)$[5]

Let $B = \mathbb{S}^{(n-1)} \setminus A_t$ and $\widehat{B} = T(B)$. We have that $\|a - b\| \geq t$ for all $a \in A$ and $b \in B$. By Lemma 22.2.2 below, the set $\left(\widehat{A} + \widehat{B}\right)/2$ is contained in the ball $r\mathbf{b}^n$ centered at the origin, where $r = 1 - t^2/8$. Observe that $\mu(r\mathbf{b}^n) = \mathrm{Vol}(r\mathbf{b}^n)/\mathrm{Vol}(\mathbf{b}^n) = r^n = \left(1 - t^2/8\right)^n$. As such, applying the Brunn-Minkowski inequality in the form of Corollary 22.1.7, we have

$$\left(1 - \frac{t^2}{8}\right)^n = \mu(r\mathbf{b}^n) \geq \mu\left(\frac{\widehat{A} + \widehat{B}}{2}\right) \geq \sqrt{\mu\left(\widehat{A}\right)\mu\left(\widehat{B}\right)} = \sqrt{\mathbf{Pr}[A]\,\mathbf{Pr}[B]} \geq \sqrt{\mathbf{Pr}[B]\,/2}.$$

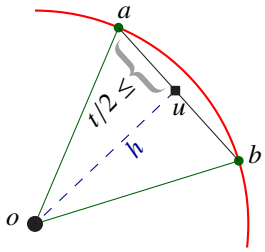
Thus, $\mathbf{Pr}[B] \leq 2(1 - t^2/8)^{2n} \leq 2\exp(-2nt^2/8)$, since $1 - x \leq \exp(-x)$, for $x \geq 0$. ∎

Lemma 22.2.2. *For any $\widehat{a} \in \widehat{A}$ and $\widehat{b} \in \widehat{B}$, we have $\left\|\frac{\widehat{a} + \widehat{b}}{2}\right\| \leq 1 - \frac{t^2}{8}$.*

Proof: Let $\widehat{a} = \alpha a$ and $\widehat{b} = \beta b$, where $a \in A$ and $b \in B$. We have

$$\|u\| = \left\|\frac{a+b}{2}\right\| = \sqrt{1^2 - \left\|\frac{a-b}{2}\right\|^2} \leq \sqrt{1 - \frac{t^2}{4}} \leq 1 - \frac{t^2}{8}, \tag{22.2}$$

since $\|a - b\| \geq t$. As for $\widehat{a}$ and $\widehat{b}$, assume that $\alpha \leq \beta$, and observe that the quantity $\left\|\widehat{a} + \widehat{b}\right\|$ is maximized when $\beta = 1$. As such, by the triangle inequality, we have



$$\begin{aligned}\left\|\frac{\widehat{a} + \widehat{b}}{2}\right\| &= \left\|\frac{\alpha a + b}{2}\right\| \leq \left\|\frac{\alpha(a+b)}{2}\right\| + \left\|(1-\alpha)\frac{b}{2}\right\| \\ &\leq \alpha\left(1 - \frac{t^2}{8}\right) + (1-\alpha)\frac{1}{2} = \tau,\end{aligned}$$

by Eq. (22.2) and since $\|b\| = 1$. Now, τ is a convex combination of the two numbers $1/2$ and $1 - t^2/8$. In particular, we conclude that $\tau \leq \max(1/2, 1 - t^2/8) \leq 1 - t^2/8$, since $t \leq 2$. ∎

[5]This is one of these "trivial" claims that might give the reader a pause, so here is a formal proof. Pick a random point p uniformly inside the ball $\mathbf{b}^n$. Let ψ be the probability that $\mathsf{p} \in \widehat{A}$. Clearly, $\mathrm{Vol}(\widehat{A}) = \psi\,\mathrm{Vol}(\mathbf{b}^n)$. So, consider the normalized point $\mathsf{q} = \mathsf{p}/\|\mathsf{p}\|$. Clearly, $\mathsf{p} \in \widehat{A}$ if and only if $\mathsf{q} \in A$, by the definition of $\widehat{A}$. Thus, $\mu\left(\widehat{A}\right) = \mathrm{Vol}(\widehat{A})/\,\mathrm{Vol}(\mathbf{b}^n) = \psi = \mathbf{Pr}\left[\mathsf{p} \in \widehat{A}\right] = \mathbf{Pr}[\mathsf{q} \in A] = \mathbf{Pr}[A]$, since q has a uniform distribution on the hypersphere by assumption.

22.3 Concentration of Lipschitz Functions

Consider a function $f : \mathbb{S}^{(n-1)} \to \mathbb{R}$, and imagine that we have a probability density function defined over the sphere. Let $\mathbf{Pr}[f \leq t] = \mathbf{Pr}\left[\left\{x \in S^{n-1} \;\middle|\; f(x) \leq t\right\}\right]$. We define the ***median*** of f, denoted by $\mathrm{med}(f)$, to be the $\sup t$, such that $\mathbf{Pr}[f \leq t] \leq 1/2$.

We define $\mathbf{Pr}[f < \mathrm{med}(f)] = \sup_{x<\mathrm{med}(f)} \mathbf{Pr}[f \leq x]$. The following is obvious but (in fact) requires a formal proof.

Lemma 22.3.1. *We have* $\mathbf{Pr}[f < \mathrm{med}(f)] \leq 1/2$ *and* $\mathbf{Pr}[f > \mathrm{med}(f)] \leq 1/2$.

Proof: Since $\bigcup_{k\geq 1}(-\infty, \mathrm{med}(f) - 1/k] = (-\infty, \mathrm{med}(f))$, we have

$$\mathbf{Pr}[f < \mathrm{med}(f)] \quad = \quad \sup_{k\geq 1} \mathbf{Pr}\left[f \leq \mathrm{med}(f) - \frac{1}{k}\right] \leq \sup_{k\geq 1} \frac{1}{2} = \frac{1}{2}.$$

The second claim follows by a symmetric argument. ∎

Definition 22.3.2 (*c-Lipschitz*). A function $f : A \to B$ is ***c-Lipschitz*** if, for any $x, y \in A$, we have $\|f(x) - f(y)\| \leq c\,\|x - y\|$.

Theorem 22.3.3 (Lévy's Lemma.). *Let* $f : \mathbb{S}^{(n-1)} \to \mathbb{R}$ *be* 1*-Lipschitz. Then for all* $t \in [0, 1]$,

$$\mathbf{Pr}[f > \mathrm{med}(f) + t] \leq 2\exp\left(-t^2 n/2\right) \text{ and } \mathbf{Pr}[f < \mathrm{med}(f) - t] \leq 2\exp\left(-t^2 n/2\right).$$

Proof: We prove only the first inequality, the second follows by symmetry. Let

$$A = \left\{x \in \mathbb{S}^{(n-1)} \;\middle|\; f(x) \leq \mathrm{med}(f)\right\}.$$

By Lemma 22.3.1, we have $\mathbf{Pr}[A] \geq 1/2$. Consider a point $x \in A_t$, where A_t is as defined in Theorem 22.2.1. Let $\mathsf{nn}(x)$ be the nearest point in A to x. We have by definition that $\|x - \mathsf{nn}(x)\| \leq t$. As such, since f is 1-Lipschitz and $\mathsf{nn}(x) \in A$, we have that

$$f(x) \leq f(\mathsf{nn}(x)) + \|\mathsf{nn}(x) - x\| \leq \mathrm{med}(f) + t.$$

Thus, by Theorem 22.2.1, we get $\mathbf{Pr}[f > \mathrm{med}(f) + t] \leq 1 - \mathbf{Pr}[A_t] \leq 2\exp\left(-t^2 n/2\right)$. ∎

22.4 The Johnson-Lindenstrauss Lemma

Lemma 22.4.1. *For a unit vector* $x \in \mathbb{S}^{(n-1)}$*, let*

$$f(x) = \sqrt{x_1^2 + x_2^2 + \cdots + x_k^2}$$

be the length of the projection of x into the subspace formed by the first k coordinates. Let x be a vector randomly chosen with uniform distribution from $\mathbb{S}^{(n-1)}$*. Then* $f(x)$ *is sharply concentrated. Namely, there exists* $m = m(n, k)$ *such that*

$$\mathbf{Pr}[f(x) \geq m + t] \leq 2\exp(-t^2 n/2) \quad \text{and} \quad \mathbf{Pr}[f(x) \leq m - t] \leq 2\exp(-t^2 n/2),$$

for any $t \in [0, 1]$*. Furthermore, for* $k \geq 10 \ln n$*, we have* $m \geq \frac{1}{2}\sqrt{k/n}$*.*

Proof: The orthogonal projection $p : \mathbb{R}^n \to \mathbb{R}^k$ given by $p(x_1, \ldots, x_n) = (x_1, \ldots, x_k)$ is 1-Lipschitz (since projections can only shrink distances, see Exercise 22.6.4). As such, $f(x) = \|p(x)\|$ is 1-Lipschitz, since for any x, y we have

$$|f(x) - f(y)| = \Big| \|p(x)\| - \|p(y)\| \Big| \leq \|p(x) - p(y)\| \leq \|x - y\|,$$

by the triangle inequality and since p is 1-Lipschitz. Theorem 22.3.3 (i.e., Lévy's lemma) gives the required tail estimate with $m = \operatorname{med}(f)$.

Thus, we only need to prove the lower bound on m. For a random $x = (x_1, \ldots, x_n) \in \mathbb{S}^{(n-1)}$, we have $\mathbf{E}\Big[\|x\|^2\Big] = 1$. By linearity of expectations, and symmetry, we have $1 = \mathbf{E}\Big[\|x\|^2\Big] = \mathbf{E}\Big[\sum_{i=1}^n x_i^2\Big] = \sum_{i=1}^n \mathbf{E}\Big[x_i^2\Big] = n\,\mathbf{E}\Big[x_j^2\Big]$, for any $1 \leq j \leq n$. Thus, $\mathbf{E}\Big[x_j^2\Big] = 1/n$, for $j = 1, \ldots, n$. Thus,

$$\mathbf{E}\Big[(f(x))^2\Big] = \mathbf{E}\left[\sum_{i=1}^k x_i^2\right] = \sum_{i=1}^k \mathbf{E}[x_i] = \frac{k}{n},$$

by linearity of expectation.

We next use that f is concentrated, to show that f^2 is also relatively concentrated. For any $t \geq 0$, we have

$$\frac{k}{n} = \mathbf{E}\Big[f^2\Big] \leq \mathbf{Pr}\Big[f \leq m + t\Big](m + t)^2 + \mathbf{Pr}\Big[f \geq m + t\Big] \cdot 1 \leq 1 \cdot (m + t)^2 + 2\exp(-t^2 n/2),$$

since $f(x) \leq 1$, for any $x \in \mathbb{S}^{(n-1)}$. Let $t = \sqrt{k/5n}$. Since $k \geq 10 \ln n$, we have that $2\exp(-t^2 n/2) \leq 2/n$. We get that

$$\frac{k}{n} \leq \Big(m + \sqrt{k/5n}\Big)^2 + 2/n.$$

Implying that $\sqrt{(k-2)/n} \leq m + \sqrt{k/5n}$, which in turn implies that $m \geq \sqrt{(k-2)/n} - \sqrt{k/5n} \geq \frac{1}{2}\sqrt{k/n}$. ∎

Next, we would like to argue that given a fixed vector, projecting it down into a random k-dimensional subspace results in a random vector such that its length is highly concentrated. This would imply that we can do dimension reduction and still preserve distances between points that we care about.

To this end, we would like to flip Lemma 22.4.1 around. Instead of randomly picking a point and projecting it down to the first k-dimensional space, we would like x to be fixed, and randomly pick the k-dimensional subspace we project into. However, we need to pick this random k-dimensional space carefully. Indeed, if we rotate this random subspace, by a transformation T, so that it occupies the first k dimensions, then the point $T(x)$ needs to be uniformly distributed on the hypersphere if we want to use Lemma 22.4.1.

As such, we would like to randomly pick a rotation of $\mathbb{R}^n$. This maps the standard orthonormal basis into a randomly rotated orthonormal space. Taking the subspace spanned by the first k vectors of the rotated basis results in a k-dimensional random subspace. Such a rotation is an orthonormal matrix with determinant 1. We can generate such a matrix, by randomly picking a vector $e_1 \in \mathbb{S}^{(n-1)}$. Next, we set e_1 as the first column of our rotation matrix, and generate the other $n - 1$ columns, by generating recursively $n - 1$ orthonormal vectors in the space orthogonal to e_1.

Remark 22.4.2 (Generating a random point on the sphere.). At this point, the reader might wonder how do we pick a point uniformly from the unit hypersphere. The idea is to pick a point from the multi-dimensional normal distribution $N^n(0,1)$, and normalizing it to have length 1. Since the multi-dimensional normal distribution has the density function

$$(2\pi)^{-n/2} \exp\Big(-(x_1^2 + x_2^2 + \cdots + x_n^2)/2\Big),$$

which is symmetric (i.e., all the points in distance r from the origin have the same distribution), it follows that this indeed generates a point randomly and uniformly on $\mathbb{S}^{(n-1)}$.

Generating a vector with multi-dimensional normal distribution, is no more than picking each coordinate according to the normal distribution, see Lemma 22.7.1$_{\text{p129}}$. Given a source of random numbers according to the uniform distribution, this can be done using a $O(1)$ computations per coordinate, using the Box-Muller transformation [BM58]. Overall, each random vector can be generated in $O(n)$ time.

Since projecting down n-dimensional normal distribution to the lower dimensional space yields a normal distribution, it follows that generating a random projection, is no more than randomly picking n vectors according to the multidimensional normal distribution $v_1, \ldots, v_n$. Then, we orthonormalize them, using Graham-Schmidt, where $\widehat{v_1} = v_1 / \|v_1\|$, and $\widehat{v_i}$ is the normalized vector of $v_i - w_i$, where w_i is the projection of v_i to the space spanned by $v_1, \ldots, v_{i-1}$.

Taking those vectors as columns of a matrix, generates a matrix A, with determinant either 1 or -1. We multiply one of the vectors by -1 if the determinant is -1. The resulting matrix is a random rotation matrix.

We can now restate Lemma 22.4.1 in the setting where the vector is fixed and the projection is into a random subspace.

Lemma 22.4.3. *Let $x \in \mathbb{S}^{(n-1)}$ be an arbitrary unit vector, and consider a random k dimensional subspace $\mathcal{F}$, and let $f(x)$ be the length of the projection of x into $\mathcal{F}$. Then, there exists $m = m(n,k)$ such that*

$$\mathbf{Pr}[f(x) \geq m + t] \leq 2\exp(-t^2 n/2) \quad \text{and} \quad \mathbf{Pr}[f(x) \leq m - t] \leq 2\exp(-t^2 n/2),$$

for any $t \in [0,1]$. Furthermore, for $k \geq 10 \ln n$, we have $m \geq \frac{1}{2}\sqrt{k/n}$.

Proof: Let v_i be the ith orthonormal vector having 1 at the ith coordinate. Let M be a random translation of space generated as described above. Clearly, for arbitrary fixed unit vector x, the vector $\mathsf{M}x$ is distributed uniformly on the sphere. Now, the ith column of the matrix M is the random vector e_i, and $\mathsf{M}^T v_i = e_i$. As such, we have

$$\langle \mathsf{M}x, v_i \rangle = (\mathsf{M}x)^T v_i = x^T \mathsf{M}^T v_i = x^T e_i = \langle x, e_i \rangle .$$

In particular, treating $\mathsf{M}x$ as a random vector, and projecting it on the first k coordinates, we have that

$$f(x) = \sqrt{\sum_{i=1}^{k} \langle \mathsf{M}x, v_i \rangle^2} = \sqrt{\sum_{i=1}^{k} \langle x, e_i \rangle^2}.$$

But $e_1, \ldots, e_k$ is just an orthonormal basis of a random k-dimensional subspace. As such, the expression on the right is the length of the projection of x into a k-dimensional random subspace.

As such, the length of the projection of x into a random k-dimensional subspace has exactly the same distribution as the length of the projection of a random vector into the first k coordinates. The claim now follows by Lemma 22.4.1. ■

Definition 22.4.4. The mapping $f : \mathbb{R}^n \to \mathbb{R}^k$ is called ***K-bi-Lipschitz*** for a subset $X \subseteq \mathbb{R}^n$ if there exists a constant $c > 0$ such that

$$cK^{-1} \cdot \|\mathsf{p} - \mathsf{q}\| \leq \|f(\mathsf{p}) - f(\mathsf{q})\| \leq c \cdot \|\mathsf{p} - \mathsf{q}\|,$$

for all $\mathsf{p}, \mathsf{q} \in X$.

The least K for which f is K-bi-Lipschitz is called the ***distortion*** of f, and is denoted dist(f). We will refer to f as a ***K-embedding*** of X.

Remark 22.4.5. Let $X \subseteq \mathbb{R}^m$ be a set of n points, where m potentially might be much larger than n. Observe, that in this case, since we only care about the inter-point distances of points in X, we can consider X to be a set of points lying in the affine subspace $\mathcal{F}$ spanned by the points of X. Note, that this subspace has dimension $n-1$. As such, each point of X be interpreted as $n-1$ dimensional point in $\mathcal{F}$. Namely, we can assume, for our purposes, that the set of n points in Euclidean space we care about lies in $\mathbb{R}^n$ (in fact, $\mathbb{R}^{n-1}$).

Note, that if $m < n$ we can always pad all the coordinates of the points of X by zeros, such that the resulting point set lies in $\mathbb{R}^n$.

Theorem 22.4.6 (Johnson-Lindenstrauss lemma.). *Let X be an n-point set in a Euclidean space, and let $\varepsilon \in (0, 1]$ be given. Then there exists a $(1+\varepsilon)$-embedding of X into $\mathbb{R}^k$, where $k = O(\varepsilon^{-2} \log n)$.*

Proof: By Remark 22.4.5, we can assume that $X \subseteq \mathbb{R}^n$. Let $k = 200\varepsilon^{-2} \ln n$. Assume $k < n$, and let $\mathcal{F}$ be a random k-dimensional linear subspace of $\mathbb{R}^n$. Let $P_{\mathcal{F}} : \mathbb{R}^n \to \mathcal{F}$ be the orthogonal projection operator of $\mathbb{R}^n$ into $\mathcal{F}$. Let m be the number around which $\|P_{\mathcal{F}}(x)\|$ is concentrated, for $x \in \mathbb{S}^{(n-1)}$, as in Lemma 22.4.3.

Fix two points $x, y \in \mathbb{R}^n$, we prove that

$$\left(1 - \frac{\varepsilon}{3}\right) m \|x - y\| \leq \|P_{\mathcal{F}}(x) - P_{\mathcal{F}}(y)\| \leq \left(1 + \frac{\varepsilon}{3}\right) m \|x - y\|$$

holds with probability $\geq 1 - n^{-2}$. Since there are $\binom{n}{2}$ pairs of points in X, it follows that with constant probability (say $> 1/3$) this holds for all pairs of points of X. In such a case, the mapping p is D-embedding of X into $\mathbb{R}^k$ with $D \leq \frac{1+\varepsilon/3}{1-\varepsilon/3} \leq 1 + \varepsilon$, for $\varepsilon \leq 1$.

Let $u = x - y$, we have $P_{\mathcal{F}}(u) = P_{\mathcal{F}}(x) - P_{\mathcal{F}}(y)$ since $P_{\mathcal{F}}(\cdot)$ is a linear operator. Thus, the condition becomes $\left(1 - \frac{\varepsilon}{3}\right) m \|u\| \leq \|P_{\mathcal{F}}(u)\| \leq \left(1 + \frac{\varepsilon}{3}\right) m \|u\|$. Again, since projection is a linear operator, for any $\alpha > 0$, the condition is equivalent to

$$\left(1 - \tfrac{\varepsilon}{3}\right) m \|\alpha u\| \leq \|P_{\mathcal{F}}(\alpha u)\| \leq \left(1 + \tfrac{\varepsilon}{3}\right) m \|\alpha u\|.$$

As such, we can assume that $\|u\| = 1$ by picking $\alpha = 1/\|u\|$. Namely, we need to show that

$$\left| \|P_{\mathcal{F}}(u)\| - m \right| \leq \frac{\varepsilon}{3} m.$$

Let $f(u) = \|P_{\mathcal{F}}(u)\|$. By Lemma 22.4.1 (exchanging the random space with the random vector), for $t = \varepsilon m/3$, we have that the probability that this does not hold is bounded by

$$\mathbf{Pr}[|f(u) - m| \geq t] \leq 4\exp\left(-\frac{t^2 n}{2}\right) = 4\exp\left(\frac{-\varepsilon^2 m^2 n}{18}\right) \leq 4\exp\left(-\frac{\varepsilon^2 k}{72}\right) < n^{-2},$$

since $m \geq \frac{1}{2}\sqrt{k/n}$ and $k = 200\varepsilon^{-2}\ln n$. ∎

22.5 Bibliographical notes

Our presentation follows Matoušek [Mat02]. The Brunn-Minkowski inequality is a powerful inequality which is widely used in mathematics. A nice survey of this inequality and its applications is provided by Gardner [Gar02]. Gardner says: "In a sea of mathematics, the Brunn-Minkowski inequality appears like an octopus, tentacles reaching far and wide, its shape and color changing as it roams from one area to the next." However, Gardner is careful in claiming that the Brunn-Minkowski inequality is one of the most powerful inequalities in mathematics since as a wit put it "the most powerful inequality is $x^2 \geq 0$, since all inequalities are in some sense equivalent to it."

A striking application of the Brunn-Minkowski inequality is the proof that in any partial ordering of n elements, there is a single comparison that knowing its result, reduces the number of linear extensions that are consistent with the partial ordering, by a constant fraction. This immediately implies (the uninteresting result) that one can sort n elements in $O(n \log n)$ comparisons. More interestingly, it implies that if there are m linear extensions of the current partial ordering, we can *always* sort it using $O(\log m)$ comparisons. A nice exposition of this surprising result is provided by Matoušek [Mat02, Section 12.3].

There are several alternative proofs of the JL lemma, see [IM98] and[DG03]. Interestingly, it is enough to pick each entry in the dimension reducing matrix randomly out of $-1, 0, 1$. This requires a more involved proof [Ach01]. This is useful when one cares about storing this dimension reduction transformation efficiently.

Magen [Mag07] observed that the JL lemma preserves angles, and in fact can be used to preserve any "k dimensional angle", by projecting down to dimension $O(k\varepsilon^{-2}\log n)$. In particular, Exercise 22.6.5 is taken from there.

In fact, the random embedding preserves much more structure than just distances between points. It preserves the structure and distances of surfaces as long as they are low dimensional and "well behaved", see [AHY07] for some results in this direction.

Dimension reduction is crucial in learning, AI, databases, etc. One common technique that is being used in practice is to do PCA (i.e., principal component analysis) and take the first few main axes. Other techniques include independent component analysis, and MDS (multidimensional scaling). MDS tries to embed points from high dimensions into low dimension ($d = 2$ or 3), while preserving some properties. Theoretically, dimension reduction into really low dimensions is hopeless, as the distortion in the worst case is $\Omega(n^{1/(k-1)})$, if k is the target dimension [Mat90].

22.6 Exercises

Exercise 22.6.1 (Boxes can be separated.). (Easy.) Let A and B be two axis-parallel boxes that are interior disjoint. Prove that there is always an axis-parallel hyperplane that separates the interior

of the two boxes.

Exercise 22.6.2 (Brunn-Minkowski inequality slight extension.).
Corollary 22.6.3. *For A and B compact sets in $\mathbb{R}^n$, we have for any $\lambda \in [0,1]$ that* $\mathrm{Vol}(\lambda A + (1-\lambda)B) \geq \mathrm{Vol}(A)^{\lambda} \mathrm{Vol}(B)^{1-\lambda}$.

Exercise 22.6.4 (Projections are contractions.). (Easy.) Let $\mathcal{F}$ be a k-dimensional affine subspace, and let $P_{\mathcal{F}} : \mathbb{R}^d \to \mathcal{F}$ be the projection that maps every point $x \in \mathbb{R}^d$ to its nearest neighbor on $\mathcal{F}$. Prove that p is a contraction (i.e., 1-Lipschitz). Namely, for any $\mathsf{p}, \mathsf{q} \in \mathbb{R}^d$, it holds that $\|P_{\mathcal{F}}(\mathsf{p}) - P_{\mathcal{F}}(\mathsf{q})\| \leq \|\mathsf{p} - \mathsf{q}\|$.

Exercise 22.6.5 (JL Lemma works for angles.). Show that the Johnson-Lindenstrauss lemma also $(1 \pm \varepsilon)$-preserves angles among triples of points of P (you might need to increase the target dimension however by a constant factor). [**Hint:** For every angle, construct a equilateral triangle that its edges are being preserved by the projection (add the vertices of those triangles [conceptually] to the point set being embedded). Argue, that this implies that the angle is being preserved.**]**

22.7 Miscellaneous

Lemma 22.7.1. *(A) The multidimensional normal distribution is symmetric; that is, for any two points $\mathsf{p}, \mathsf{q} \in \mathbb{R}^d$ such that $\|\mathsf{p}\| = \|\mathsf{q}\|$ we have that $g(\mathsf{p}) = g(\mathsf{q})$, where $g(\cdot)$ is the density function of the multidimensional normal distribution $\mathbf{N}^d$.*

(B) The projection of the normal distribution on any direction is a one dimensional normal distribution.

(C) Picking d variables $X_1, \ldots, X_d$ using one dimensional normal distribution $\mathbf{N}$ results in a point $(X_1, \ldots, X_d)$ that has multidimensional normal distribution $\mathbf{N}^d$.

Chapter 23

Finite Metric Spaces and Partitions

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

23.1 Finite Metric Spaces

Definition 23.1.1. A ***metric space*** is a pair $(\mathcal{X}, \mathsf{d})$ where $\mathcal{X}$ is a set and $\mathsf{d} : \mathcal{X} \times \mathcal{X} \to [0, \infty)$ is a ***metric***, satisfying the following axioms: (i) $\mathsf{d}(x, y) = 0$ iff $x = y$, (ii) $\mathsf{d}(x, y) = \mathsf{d}(y, x)$, and (iii) $\mathsf{d}(x, y) + \mathsf{d}(y, z) \geq \mathsf{d}(x, z)$ (triangle inequality).

For example, $\mathbb{R}^2$ with the regular Euclidean distance is a metric space.

It is usually of interest to consider the finite case, where $\mathcal{X}$ is an n-point set. Then, the function d can be specified by $\binom{n}{2}$ real numbers. Alternatively, one can think about $(\mathcal{X}, \mathsf{d})$ is a weighted complete graph, where we specify positive weights on the edges, and the resulting weights on the edges comply with the triangle inequality.

In fact, finite metric spaces rise naturally from (sparser) graphs. Indeed, let $G = (\mathcal{X}, E)$ be an undirected weighted graph defined over $\mathcal{X}$, and let $\mathsf{d}_G(x, y)$ be the length of the shortest path between x and y in G. It is easy to verify that $(\mathcal{X}, \mathsf{d}_G)$ is a finite metric space. As such if the graph G is sparse, it provides a compact representation to the finite space $(\mathcal{X}, \mathsf{d}_G)$.

Definition 23.1.2. Let $(\mathcal{X}, d)$ be an n-point metric space. We denote the ***open ball*** of radius r about $x \in \mathcal{X}$, by $\mathbf{b}(x, r) = \left\{ y \in \mathcal{X} \;\middle|\; \mathsf{d}(x, y) < r \right\}$.

Underling our discussion of metric spaces are algorithmic applications. The hardness of various computational problems depends heavily on the structure of the finite metric space. Thus, given a finite metric space, and a computational task, it is natural to try to map the given metric space into a new metric where the task at hand becomes easy.

Example 23.1.3. For example, computing the diameter is not trivial in two dimensions, but is easy in one dimension. Thus, if we could map points in two dimensions into points in one dimension, such that the diameter is preserved, then computing the diameter becomes easy. In fact, this approach yields an efficient approximation algorithm, see Exercise 23.7.3 below.

Of course, this mapping from one metric space to another, is going to introduce error. We would be interested in minimizing the error introduced by such a mapping.

Definition 23.1.4. Let $(\mathcal{X}, \mathsf{d}_\mathcal{X})$ and (Y, d_Y) be metric spaces. A mapping $f : \mathcal{X} \to Y$ is called an ***embedding***, and is ***C-Lipschitz*** if $\mathsf{d}_Y(f(x), f(y)) \leq C \cdot \mathsf{d}_\mathcal{X}(x, y)$ for all $x, y \in \mathcal{X}$. The mapping f is called ***K-bi-Lipschitz*** if there exists a $C > 0$ such that

$$CK^{-1} \cdot \mathsf{d}_\mathcal{X}(x, y) \leq \mathsf{d}_Y(f(x), f(y)) \leq C \cdot \mathsf{d}_\mathcal{X}(x, y),$$

for all $x, y \in \mathcal{X}$.

The least K for which f is K-bi-Lipschitz is called the ***distortion*** of f, and is denoted $\operatorname{dist}(f)$. The least distortion with which $\mathcal{X}$ may be embedded in Y is denoted $c_Y(\mathcal{X})$.

There are several powerful results in this vain, that show the existence of embeddings with low distortion that would be presented:

1. Probabilistic trees - every finite metric can be randomly embedded into a tree such that the "expected" distortion for a specific pair of points is $O(\log n)$.

2. Bourgain embedding - shows that any n-point metric space can be embedded into (finite dimensional) metric space with $O(\log n)$ distortion.

3. Johnson-Lindenstrauss lemma - shows that any n-point set in Euclidean space with the regular Euclidean distance can be embedded into $\mathbb{R}^k$ with distortion $(1 + \varepsilon)$, where $k = O(\varepsilon^{-2} \log n)$.

23.2 Examples

What is distortion? When considering a mapping $f : \mathcal{X} \to \mathbb{R}^d$ of a metric space $(\mathcal{X}, \mathsf{d})$ to $\mathbb{R}^d$, it would useful to observe that since $\mathbb{R}^d$ can be scaled, we can consider f to be an an expansion (i.e., no distances shrink). Furthermore, we can in fact assume that there is at least one pair of points $x, y \in \mathcal{X}$, such that $\mathsf{d}(x, y) = \|x - y\|$. As such, we have $\operatorname{dist}(f) = \max_{x,y} \frac{\|x-y\|}{\mathsf{d}(x,y)}$.

Why distortion is necessary? Consider the a graph $G = (V, E)$ with one vertex s connected to three other vertices a, b, c, where the weights on the edges are all one (i.e., G is the star graph with three leafs). We claim that G can not be embedded into Euclidean space with distortion $\leq \sqrt{2}$. Indeed, consider the associated metric space (V, d_G) and an (expansive) embedding $f : V \to \mathbb{R}^d$.

Consider the triangle formed by $\triangle = a'b'c'$, where $a' = f(a), b' = f(b)$ and $c' = f(c)$. Next, consider the following quantity $\max(\|a' - s'\|, \|b' - s'\|, \|c' - s'\|)$ which lower bounds the distortion of f. This quantity is minimized when $r = \|a' - s'\| = \|b' - s'\| = \|c' - s'\|$. Namely, s' is the center of the smallest enclosing circle of $\triangle$. However, r is minimize when all the edges of $\triangle$ are of equal length, and are in fact of length $\mathsf{d}_G(a, b) = 2$. It follows that $\operatorname{dist}(f) \geq r \geq 2/\sqrt{3}$.

It is known that $\Omega(\log n)$ distortion is necessary in the worst case. This is shown using expanders [Mat02].

23.2.1 Hierarchical Tree Metrics

The following metric is quite useful in practice, and nicely demonstrate why algorithmically finite metric spaces are useful.

Definition 23.2.1. ***Hierarchically well-separated tree*** (HST) is a metric space defined on the leaves of a rooted tree T. To each vertex $u \in T$ there is associated a label $\Delta_u \geq 0$ such that $\Delta_u = 0$ if and only if u is a leaf of T. The labels are such that if a vertex u is a child of a vertex v then $\Delta_u \leq \Delta_v$. The distance between two leaves $x, y \in T$ is defined as $\Delta_{\mathrm{lca}(x,y)}$, where $\mathrm{lca}(x, y)$ is the least common ancestor of x and y in T.

A HST T is a ***k-HST*** if for a vertex $v \in T$, we have that $\Delta_v \leq \Delta_{\overline{\mathrm{p}}(v)}/k$, where $\overline{\mathrm{p}}(v)$ is the parent of v in T.

Note that a HST is a very limited metric. For example, consider the cycle $G = C_n$ of n vertices, with weight one on the edges, and consider an expansive embedding f of G into a HST HST. It is easy to verify, that there must be two consecutive nodes of the cycle, which are mapped to two different subtrees of the root r of HST. Since HST is expansive, it follows that $\Delta_r \geq n/2$. As such, $\mathrm{dist}(f) \geq n/2$. Namely, HSTs fail to faithfully represent even very simple metrics.

23.2.2 Clustering

One natural problem we might want to solve on a graph (i.e., finite metric space) $(\mathcal{X}, \mathsf{d})$ is to partition it into clusters. One such natural clustering is the ***k-median clustering***, where we would like to choose a set $C \subseteq \mathcal{X}$ of k centers, such that $\nu_C(\mathcal{X}, \mathsf{d}) = \sum_{\mathsf{q} \in \mathcal{X}} \mathsf{d}(\mathsf{q}, C)$ is minimized, where $\mathsf{d}(\mathsf{q}, C) = \min_{c \in C} \mathsf{d}(\mathsf{q}, c)$ is the distance of q to its closest center in C.

It is known that finding the optimal k-median clustering in a (general weighted) graph is NP-complete. As such, the best we can hope for is an approximation algorithm. However, if the structure of the finite metric space $(\mathcal{X}, \mathsf{d})$ is simple, then the problem can be solved efficiently. For example, if the points of $\mathcal{X}$ are on the real line (and the distance between a and b is just $|a - b|$), then k-median can be solved using dynamic programming.

Another interesting case is when the metric space $(\mathcal{X}, \mathsf{d})$ is a HST. Is not too hard to prove the following lemma. See Exercise 23.7.1.

Lemma 23.2.2. *Let $(\mathcal{X}, \mathsf{d})$ be a HST defined over n points, and let $k > 0$ be an integer. One can compute the optimal k-median clustering of $\mathcal{X}$ in $O(k^2 n)$ time.*

Thus, if we can embed a general graph G into a HST HST, with low distortion, then we could approximate the k-median clustering on G by clustering the resulting HST, and "importing" the resulting partition to the original space. The quality of approximation, would be bounded by the distortion of the embedding of G into HST.

23.3 Random Partitions

Let $(\mathcal{X}, d)$ be a finite metric space. Given a partition $P = \{C_1, \ldots, C_m\}$ of $\mathcal{X}$, we refer to the sets C_i as ***clusters***. We write $\mathcal{P}_{\mathcal{X}}$ for the set of all partitions of $\mathcal{X}$. For $x \in \mathcal{X}$ and a partition $P \in \mathcal{P}_{\mathcal{X}}$ we denote by $P(x)$ the unique cluster of P containing x. Finally, the set of all probability distributions on $\mathcal{P}_{\mathcal{X}}$ is denoted $\mathcal{D}_{\mathcal{X}}$.

23.3.1 Constructing the partition

Let $\Delta = 2^u$ be a prescribed parameter, which is the required diameter of the resulting clusters. Choose, uniformly at random, a permutation π of $\mathcal{X}$ and a random value $\alpha \in [1/4, 1/2]$. Let $R = \alpha\Delta$, and observe that it is uniformly distributed in the interval $[\Delta/4, \Delta/2]$.

The partition is now defined as follows: A point $x \in \mathcal{X}$ is assigned to the cluster C_y of y, where y is the first point in the permutation in distance $\leq R$ from x. Formally,

$$C_y = \left\{ x \in \mathcal{X} \;\middle|\; x \in \mathbf{b}(y, R) \text{ and } \pi(y) \leq \pi(z) \text{ for all } z \in \mathcal{X} \text{ with } x \in \mathbf{b}(z, R) \right\}.$$

Let $P = \{C_y\}_{y \in \mathcal{X}}$ denote the resulting partition.

Here is a somewhat more intuitive explanation: Once we fix the radius of the clusters R, we start scooping out balls of radius R centered at the points of the random permutation π. At the ith stage, we scoop out only the remaining mass at the ball centered at x_i of radius r, where x_i is the ith point in the random permutation.

23.3.2 Properties

Lemma 23.3.1. *Let $(\mathcal{X}, d)$ be a finite metric space, $\Delta = 2^u$ a prescribed parameter, and let P be the partition of $\mathcal{X}$ generated by the above random partition. Then the following holds:*

(i) For any $C \in P$, we have $\operatorname{diam}(C) \leq \Delta$.

(ii) Let x be any point of $\mathcal{X}$, and t a parameter $\leq \Delta/8$. Then,

$$\mathbf{Pr}[\mathbf{b}(x, t) \not\subseteq P(x)] \leq \frac{8t}{\Delta} \ln \frac{b}{a},$$

where $a = |\mathbf{b}(x, \Delta/8)|$, $b = |\mathbf{b}(x, \Delta)|$.

Proof: Since $C_y \subseteq \mathbf{b}(y, R)$, we have that $\operatorname{diam}(C_y) \leq \Delta$, and thus the first claim holds.

Let U be the set of points of $\mathbf{b}(x, \Delta)$, such that $w \in U$ iff $\mathbf{b}(w, R) \cap \mathbf{b}(x, t) \neq \emptyset$. Arrange the points of U in increasing distance from x, and let $w_1, \ldots, w_{b'}$ denote the resulting order, where $b' = |U|$. Let $I_k = [d(x, w_k) - t, d(x, w_k) + t]$ and write $\mathcal{E}_k$ for the event that w_k is the first point in π such that $\mathbf{b}(x, t) \cap C_{w_k} \neq \emptyset$, and yet $\mathbf{b}(x, t) \not\subseteq C_{w_k}$. Note that if $w_k \in \mathbf{b}(x, \Delta/8)$, then $\mathbf{Pr}[\mathcal{E}_k] = 0$ since $\mathbf{b}(x, t) \subseteq \mathbf{b}(x, \Delta/8) \subseteq \mathbf{b}(w_k, \Delta/4) \subseteq \mathbf{b}(w_k, R)$.

In particular, $w_1, \ldots, w_a \in \mathbf{b}(x, \Delta/8)$ and as such $\mathbf{Pr}[\mathcal{E}_1] = \cdots = \mathbf{Pr}[\mathcal{E}_a] = 0$. Also, note that if $\mathsf{d}(x, w_k) < R - t$ then $\mathbf{b}(w_k, R)$ contains $\mathbf{b}(x, t)$ and as such $\mathcal{E}_k$ can not happen. Similarly, if $\mathsf{d}(x, w_k) > R + t$ then $\mathbf{b}(w_k, R) \cap \mathbf{b}(x, t) = \emptyset$ and $\mathcal{E}_k$ can not happen. As such, if $\mathcal{E}_k$ happen then $R - t \leq \mathsf{d}(x, w_k) \leq R + t$. Namely, if $\mathcal{E}_k$ happen then $R \in I_k$. Namely, $\mathbf{Pr}[\mathcal{E}_k] = \mathbf{Pr}[\mathcal{E}_k \cap (R \in I_k)] = \mathbf{Pr}[R \in I_k] \cdot \mathbf{Pr}[\mathcal{E}_k \,|\, R \in I_k]$. Now, R is uniformly distributed in the interval $[\Delta/4, \Delta/2]$, and I_k is an interval of length $2t$. Thus, $\mathbf{Pr}[R \in I_k] \leq 2t/(\Delta/4) = 8t/\Delta$.

Next, to bound $\mathbf{Pr}[\mathcal{E}_k \,|\, R \in I_k]$, we observe that $w_1, \ldots, w_{k-1}$ are closer to x than w_k and their distance to $\mathbf{b}(x, t)$ is smaller than R. Thus, if any of them appear before w_k in π then $\mathcal{E}_k$ does not happen. Thus, $\mathbf{Pr}[\mathcal{E}_k \,|\, R \in I_k]$ is bounded by the probability that w_k is the first to appear in π out of $w_1, \ldots, w_k$. But this probability is $1/k$, and thus $\mathbf{Pr}[\mathcal{E}_k \,|\, R \in I_k] \leq 1/k$.

We are now ready for the kill. Indeed,

$$\begin{aligned}\mathbf{Pr}[\mathbf{b}(x,t) \not\subseteq P(x)] &= \sum_{k=1}^{b'} \mathbf{Pr}[\mathcal{E}_k] = \sum_{k=a+1}^{b'} \mathbf{Pr}[\mathcal{E}_k] = \sum_{k=a+1}^{b'} \mathbf{Pr}[R \in I_k] \cdot \mathbf{Pr}[\mathcal{E}_k \,|\, R \in I_k] \\ &\leq \sum_{k=a+1}^{b'} \frac{8t}{\Delta} \cdot \frac{1}{k} \leq \frac{8t}{\Delta} \ln \frac{b'}{a} \leq \frac{8t}{\Delta} \ln \frac{b}{a},\end{aligned}$$

since $\sum_{k=a+1}^{b} \frac{1}{k} \leq \int_a^b \frac{dx}{x} = \ln \frac{b}{a}$ and $b' \leq b$. ∎

23.4 Probabilistic embedding into trees

In this section, given n-point finite metric $(\mathcal{X}, \mathsf{d})$. we would like to embed it into a HST. As mentioned above, one can verify that for any embedding into HST, the distortion in the worst case is $\Omega(n)$. Thus, we define a randomized algorithm that embed $(\mathcal{X}, d)$ into a tree. Let T be the resulting tree, and consider two points $x, y \in \mathcal{X}$. Consider the ***random variable*** $\mathsf{d}_T(x,y)$. We constructed the tree T such that distances never shrink; i.e. $\mathsf{d}(x,y) \leq \mathsf{d}_T(x,y)$. The ***probabilistic distortion*** of this embedding is $\max_{x,y} \mathbf{E}\left[\frac{\mathsf{d}_T(x,y)}{\mathsf{d}(x,y)}\right]$. Somewhat surprisingly, one can find such an embedding with logarithmic probabilistic distortion.

Theorem 23.4.1. *Given n-point metric $(\mathcal{X}, d)$ one can randomly embed it into a 2-HST with probabilistic distortion $\leq 24 \ln n$.*

Proof: The construction is recursive. Let $\operatorname{diam}(P)$, and compute a random partition of $\mathcal{X}$ with cluster diameter $\operatorname{diam}(P)/2$, using the construction of Section 23.3.1. We recursively construct a 2-HST for each cluster, and hang the resulting clusters on the root node v, which is marked by $\Delta_v = \operatorname{diam}(P)$. Clearly, the resulting tree is a 2-HST.

For a node $v \in T$, let $\mathcal{X}(v)$ be the set of points of $\mathcal{X}$ contained in the subtree of v.

For the analysis, assume $\operatorname{diam}(P) = 1$, and consider two points $x, y \in \mathcal{X}$. We consider a node $v \in T$ to be in level i if $\operatorname{level}(v) = \lceil \lg \Delta_v \rceil = i$. The two points x and y correspond to two leaves in T, and let $\widehat{u}$ be the least common ancestor of x and y in t. We have $\mathsf{d}_T(x,y) \leq 2^{\operatorname{level}(v)}$. Furthermore, note that along a path the levels are strictly monotonically increasing.

In fact, we are going to be conservative, and let w be the first ancestor of x, such that $\mathbf{b} = \mathbf{b}(x, \mathsf{d}(x,y))$ is not completely contained in $\mathcal{X}(u_1), \ldots, \mathcal{X}(u_m)$, where $u_1, \ldots, u_m$ are the children of w. Clearly, $\operatorname{level}(w) > \operatorname{level}(\widehat{u})$. Thus, $\mathsf{d}_T(x,y) \leq 2^{\operatorname{level}(w)}$.

Consider the path σ from the root of T to x, and let $\mathcal{E}_i$ be the event that $\mathbf{b}$ is not fully contained in $\mathcal{X}(v_i)$, where v_i is the node of σ of level i (if such a node exists). Furthermore, let Y_i be the indicator variable which is 1 if $\mathcal{E}_i$ is the first to happened out of the sequence of events $\mathcal{E}_0, \mathcal{E}_{-1}, \ldots$. Clearly, $\mathsf{d}_T(x,y) \leq \sum Y_i 2^i$.

Let $t = \mathsf{d}(x,y)$ and $j = \lfloor \lg \mathsf{d}(x,y) \rfloor$, and $n_i = \left|\mathbf{b}(x, 2^i)\right|$ for $i = 0, \ldots, -\infty$. We have

$$\mathbf{E}[\mathsf{d}_T(x,y)] \leq \sum_{i=j}^{0} \mathbf{E}[Y_i]\, 2^i \leq \sum_{i=j}^{0} 2^i \,\mathbf{Pr}\left[\mathcal{E}_i \cap \overline{\mathcal{E}_{i-1}} \cap \overline{\mathcal{E}_{i-1}} \cdots \overline{\mathcal{E}_0}\right] \leq \sum_{i=j}^{0} 2^i \cdot \frac{8t}{2^i} \ln \frac{n_i}{n_{i-3}},$$

by Lemma 23.3.1. Thus,

$$\mathbf{E}[\mathsf{d}_T(x,y)] \leq 8t \ln\left(\prod_{i=j}^{0} \frac{n_i}{n_{i-3}}\right) \leq 8t \ln(n_0 \cdot n_1 \cdot n_2) \leq 24t \ln n.$$

It thus follows, that the expected distortion for x and y is $\leq 24 \ln n$. ■

23.4.1 Application: approximation algorithm for k-median clustering

Let $(\mathcal{X}, \mathsf{d})$ be a n-point metric space, and let k be an integer number. We would like to compute the optimal k-median clustering. Number, find a subset $C_{\text{opt}} \subseteq \mathcal{X}$, such that $\nu_{C_{\text{opt}}}(\mathcal{X}, \mathsf{d})$ is minimized, see Section 23.2.2. To this end, we randomly embed $(\mathcal{X}, \mathsf{d})$ into a HST HST using Theorem 23.4.1. Next, using Lemma 23.2.2, we compute the optimal k-median clustering of HST. Let C be the set of centers computed. We return C together with the partition of $\mathcal{X}$ it induces as the required clustering.

Theorem 23.4.2. *Let $(\mathcal{X}, \mathsf{d})$ be a n-point metric space. One can compute in polynomial time a k-median clustering of $\mathcal{X}$ which has expected price $O(\alpha \log n)$, where α is the price of the optimal k-median clustering of $(\mathcal{X}, \mathsf{d})$.*

Proof: The algorithm is described above, and the fact that its running time is polynomial can be easily be verified. To prove the bound on the quality of the clustering, for any point $p \in \mathcal{X}$, let $center(p)$ denote the closest point in C_{opt} to p according to d, where C_{opt} is the set of k-medians in the optimal clustering. Let C be the set of k-medians returned by the algorithm, and let HST be the HST used by the algorithm. We have

$$\beta = \nu_C(\mathcal{X}, \mathsf{d}) \leq \nu_C(\mathcal{X}, \mathsf{d}_{\mathsf{HST}}) \leq \nu_{C_{\text{opt}}}(\mathcal{X}, \mathsf{d}_{\mathsf{HST}}) \leq \sum_{p \in \mathcal{X}} \mathsf{d}_{\mathsf{HST}}(p, C_{\text{opt}}) \leq \sum_{p \in \mathcal{X}} \mathsf{d}_{\mathsf{HST}}(p, center(p)).$$

Thus, in expectation we have

$$\begin{aligned}
\mathbf{E}[\beta] &= \mathbf{E}\left[\sum_{p \in \mathcal{X}} \mathsf{d}_{\mathsf{HST}}(p, center(p))\right] = \sum_{p \in \mathcal{X}} \mathbf{E}[\mathsf{d}_{\mathsf{HST}}(p, center(p))] = \sum_{p \in \mathcal{X}} O(\mathsf{d}(p, center(p)) \log n) \\
&= O\left((\log n) \sum_{p \in \mathcal{X}} \mathsf{d}(p, center(p))\right) = O\Big(\nu_{C_{\text{opt}}}(\mathcal{X}, \mathsf{d}) \log n\Big),
\end{aligned}$$

by linearity of expectation and Theorem 23.4.1. ■

23.5 Embedding any metric space into Euclidean space

Lemma 23.5.1. *Let $(\mathcal{X}, \mathsf{d})$ be a metric, and let $Y \subset \mathcal{X}$. Consider the mapping $f : \mathcal{X} \to \mathbb{R}$, where $f(x) = \mathsf{d}(x, Y) = \min_{y \in Y} \mathsf{d}(x, y)$. Then for any $x, y \in \mathcal{X}$, we have $|f(x) - f(y)| \leq \mathsf{d}(x, y)$. Namely f is nonexpansive.*

Proof: Indeed, let x' and y' be the closet points of Y, to x and y, respectively. Observe that $f(x) = \mathsf{d}(x, x') \leq \mathsf{d}(x, y') \leq \mathsf{d}(x, y) + \mathsf{d}(y, y') = \mathsf{d}(x, y) + f(y)$ by the triangle inequality. Thus, $f(x) - f(y) \leq \mathsf{d}(x, y)$. By symmetry, we have $f(y) - f(x) \leq \mathsf{d}(x, y)$. Thus, $|f(x) - f(y)| \leq \mathsf{d}(x, y)$. ■

23.5.1 The bounded spread case

Let $(\mathcal{X}, \mathsf{d})$ be a n-point metric. The ***spread*** of $\mathcal{X}$, denoted by $\Phi(\mathcal{X}) = \frac{\operatorname{diam}(\mathcal{X})}{\min_{x,y\in\mathcal{X}, x\neq y} \mathsf{d}(x,y)}$, is the ratio between the diameter of $\mathcal{X}$ and the distance between the closest pair of points.

Theorem 23.5.2. *Given a n-point metric $\mathcal{Y} = (\mathcal{X}, d)$, with spread Φ, one can embed it into Euclidean space $\mathbb{R}^k$ with distortion $O(\sqrt{\ln \Phi} \ln n)$, where $k = O(\ln \Phi \ln n)$.*

Proof: Assume that $\operatorname{diam}(\mathcal{Y}) = \Phi$ (i.e., the smallest distance in $\mathcal{Y}$ is 1), and let $r_i = 2^{i-2}$, for $i = 1, \ldots, \alpha$, where $\alpha = \lceil \lg \Phi \rceil$. Let $P_{i,j}$ be a random partition of P with diameter r_i, using Theorem 23.4.1, for $i = 1, \ldots, \alpha$ and $j = 1, \ldots, \beta$, where $\beta = \lceil c \log n \rceil$ and c is a large enough constant to be determined shortly.

For each cluster of $P_{i,j}$ randomly toss a coin, and let $V_{i,j}$ be the all the points of $\mathcal{X}$ that belong to clusters in $P_{i,j}$ that got 'T' in their coin toss. For a point $u \in x$, let $f_{i,j}(x) = \mathsf{d}(x, \mathcal{X} \setminus V_{i,j}) = \min_{v\in\mathcal{X}\setminus V_{i,j}} \mathsf{d}(x,v)$, for $i = 0, \ldots, m$ and $j = 1, \ldots, \beta$. Let $F : \mathcal{X} \to \mathbb{R}^{(m+1)\cdot\beta}$ be the embedding, such that $F(x) = (f_{0,1}(x), f_{0,2}(x), \ldots, f_{0,\beta}(x), f_{1,1}(x), f_{0,2}(x), \ldots, f_{1,\beta}(x), \ldots, f_{m,1}(x), f_{m,2}(x), \ldots, f_{m,\beta}(x))$.

Next, consider two points $x, y \in \mathcal{X}$, with distance $\phi = \mathsf{d}(x,y)$. Let k be an integer such that $r_u \le \phi/2 \le r_{u+1}$. Clearly, in any partition of $P_{u,1}, \ldots, P_{u,\beta}$ the points x and y belong to different clusters. Furthermore, with probability half $x \in V_{u,j}$ and $y \notin V_{u,j}$ or $x \notin V_{u,j}$ and $y \in V_{u,j}$, for $1 \le j \le \beta$.

Let $\mathcal{E}_j$ denote the event that $\mathbf{b}(x,\rho) \subseteq V_{u,j}$ and $y \notin V_{u,j}$, for $j = 1, \ldots, \beta$, where $\rho = \phi/(64 \ln n)$. By Lemma 23.3.1, we have

$$\mathbf{Pr}\Big[\mathbf{b}(x,\rho) \nsubseteq P_{u,j}(x)\Big] \le \frac{8\rho}{r_u} \ln n \le \frac{\phi}{8 r_u} \le 1/2.$$

Thus,

$$\begin{aligned}\mathbf{Pr}\Big[\mathcal{E}_j\Big] &= \mathbf{Pr}\Big[\Big(\mathbf{b}(x,\rho) \subseteq P_{u,j}(x)\Big) \cap \Big(x \in V_{u,j}\Big) \cap \Big(y \notin V_{u,j}\Big)\Big] \\ &= \mathbf{Pr}\Big[\mathbf{b}(x,\rho) \subseteq P_{u,j}(x)\Big] \cdot \mathbf{Pr}\Big[x \in V_{u,j}\Big] \cdot \mathbf{Pr}\Big[y \notin V_{u,j}\Big] \ge 1/8,\end{aligned}$$

since those three events are independent. Notice, that if $\mathcal{E}_j$ happens, than $f_{u,j}(x) \ge \rho$ and $f_{u,j}(y) = 0$.

Let X_j be an indicator variable which is 1 if $\mathcal{E}_i$ happens, for $j = 1, \ldots, \beta$. Let $Z = \sum_j X_j$, and we have $\mu = \mathbf{E}[Z] = \mathbf{E}\Big[\sum_j X_j\Big] \ge \beta/8$. Thus, the probability that only $\beta/16$ of $\mathcal{E}_1, \ldots, \mathcal{E}_\beta$ happens, is $\mathbf{Pr}[Z < (1 - 1/2)\,\mathbf{E}[Z]]$. By the Chernoff inequality, we have $\mathbf{Pr}[Z < (1 - 1/2)\,\mathbf{E}[Z]] \le \exp\Big(-\mu 1/(2 \cdot 2^2)\Big) = \exp(-\beta/64) \le 1/n^{10}$, if we set $c = 640$.

Thus, with high probability

$$\|F(x) - F(y)\| \ge \sqrt{\sum_{j=1}^{\beta} \Big(f_{u,j}(x) - f_{u,j}(y)\Big)^2} \ge \sqrt{\rho^2 \frac{\beta}{16}} = \sqrt{\beta}\frac{\rho}{4} = \phi \cdot \frac{\sqrt{\beta}}{256 \ln n}.$$

On the other hand, $\Big|f_{i,j}(x) - f_{i,j}(y)\Big| \le \mathsf{d}(x,y) = \phi \le 64\rho \ln n$. Thus,

$$\|F(x) - F(y)\| \le \sqrt{\alpha\beta(64\rho \ln n)^2} \le 64\sqrt{\alpha\beta}\rho \ln n = \sqrt{\alpha\beta} \cdot \phi.$$

Thus, setting $G(x) = F(x)\frac{256 \ln n}{\sqrt{\beta}}$, we get a mapping that maps two points of distance ϕ from each other to two points with distance in the range $\left[\phi, \phi \cdot \sqrt{\alpha\beta} \cdot \frac{256 \ln n}{\sqrt{\beta}}\right]$. Namely, $G(\cdot)$ is an embedding with distortion $O(\sqrt{\alpha} \ln n) = O(\sqrt{\ln \Phi} \ln n)$.

The probability that G fails on one of the pairs, is smaller than $(1/n^{10}) \cdot \binom{n}{2} < 1/n^8$. In particular, we can check the distortion of G for all $\binom{n}{2}$ pairs, and if any of them fail (i.e., the distortion is too big), we restart the process. ∎

23.5.2 The unbounded spread case

Our next task, is to extend Theorem 23.5.2 to the case of unbounded spread. Indeed, let $(\mathcal{X}, d)$ be a n-point metric, such that $\operatorname{diam}(\mathcal{X}) \leq 1/2$. Again, we look on the different resolutions $r_1, r_2, \ldots$, where $r_i = 1/2^{i-1}$. For each one of those resolutions r_i, we can embed this resolution into β coordinates, as done for the bounded case. Then we concatenate the coordinates together.

There are two problems with this approach: (i) the number of resulting coordinates is infinite, and (ii) a pair x, y, might be distorted a "lot" because it contributes to all resolutions, not only to its "relevant" resolutions.

Both problems can be overcome with careful tinkering. Indeed, for a resolution r_i, we are going to modify the metric, so that it ignores short distances (i.e., distances $\leq r_i/n^2$). Formally, for each resolution r_i, let $G_i = (\mathcal{X}, \widehat{E_i})$ be the graph where two points x and y are connected if $\mathsf{d}(x, y) \leq r_i/n^2$. Consider a connected component $C \in G_i$. For any two points $x, y \in C$, we have $\mathsf{d}(x, y) \leq n(r_i/n^2) \leq r_i/n$. Let $\mathcal{X}_i$ be the set of connected components of G_i, and define the distances between two connected components $C, C' \in \mathcal{X}_i$, to be $\mathsf{d}_i(C, C') = \mathsf{d}(C, C') = \min_{c \in C, c' \in C'} \mathsf{d}(c, c')$.

It is easy to verify that $(\mathcal{X}_i, \mathsf{d}_i)$ is a metric space (see Exercise 23.7.2). Furthermore, we can naturally embed $(\mathcal{X}, \mathsf{d})$ into $(\mathcal{X}_i, \mathsf{d}_i)$ by mapping a point $x \in \mathcal{X}$ to its connected components in $\mathcal{X}_i$. Essentially $(\mathcal{X}_i, \mathsf{d}_i)$ is a snapped version of the metric $(\mathcal{X}, d)$, with the advantage that $\Phi((\mathcal{X}, \mathsf{d}_i)) = O(n^2)$. We now embed $\mathcal{X}_i$ into $\beta = O(\log n)$ coordinates. Next, for any point of $\mathcal{X}$ we embed it into those β coordinates, by using the embedding of its connected component in $\mathcal{X}_i$. Let E_i be the embedding for resolution r_i. Namely, $E_i(x) = (f_{i,1}(x), f_{i,2}(x), \ldots, f_{i,\beta}(x))$, where $f_{i,j}(x) = \min(\mathsf{d}_i(x, \mathcal{X} \setminus V_{i,j}), 2r_i)$. The resulting embedding is $F(x) = \oplus E_i(x) = (E_1(x), E_2(x), \ldots,)$.

Since we slightly modified the definition of $f_{i,j}(\cdot)$, we have to show that $f_{i,j}(\cdot)$ is nonexpansive. Indeed, consider two points $x, y \in \mathcal{X}_i$, and observe that

$$\left|f_{i,j}(x) - f_{i,j}(y)\right| \leq \left|\mathsf{d}_i(x, V_{i,j}) - \mathsf{d}_i(y, V_{i,j})\right| \leq \mathsf{d}_i(x, y) \leq \mathsf{d}(x, y),$$

as a simple case analysis[①] shows.

For a pair $x, y \in \mathcal{X}$, and let $\phi = \mathsf{d}(x, y)$. To see that $F(\cdot)$ is the required embedding (up to scaling), observe that, by the same argumentation of Theorem 23.5.2, we have that with high probability

$$\|F(x) - F(y)\| \geq \phi \cdot \frac{\sqrt{\beta}}{256 \ln n}.$$

[①]Indeed, if $f_{i,j}(x) < \mathsf{d}_i(x, V_{i,j})$ and $f_{i,j}(y) < \mathsf{d}_i(x, V_{i,j})$ then $f_{i,j}(x) = 2r_i$ and $f_{i,j}(y) = 2r_i$, which implies the above inequality. If $f_{i,j}(x) = \mathsf{d}_i(x, V_{i,j})$ and $f_{i,j}(y) = \mathsf{d}_i(x, V_{i,j})$ then the inequality trivially holds. The other option is handled in a similar fashion.

To get an upper bound on this distance, observe that for i such that $r_i > \phi n^2$, we have $E_i(x) = E_i(y)$. Thus,

$$\begin{aligned}
\|F(x) - F(y)\|^2 &= \sum_i \|E_i(x) - E_i(y)\|^2 = \sum_{i, r_i < \phi n^2} \|E_i(x) - E_i(y)\|^2 \\
&= \sum_{i, \phi/n^2 < r_i < \phi n^2} \|E_i(x) - E_i(y)\|^2 + \sum_{i, r_i < \phi/n^2} \|E_i(x) - E_i(y)\|^2 \\
&= \beta\phi^2 \lg\left(n^4\right) + \sum_{i, r_i < \phi/n^2} (2r_i)^2\beta \leq 4\beta\phi^2 \lg n + \frac{4\phi^2\beta}{n^4} \leq 5\beta\phi^2 \lg n.
\end{aligned}$$

Thus, $\|F(x) - F(y)\| \leq \phi\sqrt{5\beta \lg n}$. We conclude, that with high probability, $F(\cdot)$ is an embedding of $\mathcal{X}$ into Euclidean space with distortion $\left(\phi\sqrt{5\beta \lg n}\right) / \left(\phi \cdot \frac{\sqrt{\beta}}{256 \ln n}\right) = O(\log^{3/2} n)$.

We still have to handle the infinite number of coordinates problem. However, the above proof shows that we care about a resolution r_i (i.e., it contributes to the estimates in the above proof) only if there is a pair x and y such that $r_i/n^2 \leq \mathsf{d}(x, y) \leq r_i n^2$. Thus, for every pair of distances there are $O(\log n)$ relevant resolutions. Thus, there are at most $\eta = O(n^2\beta \log n) = O(n^2 \log^2 n)$ relevant coordinates, and we can ignore all the other coordinates. Next, consider the affine subspace h that spans $F(P)$. Clearly, it is $n - 1$ dimensional, and consider the projection $G : \mathbb{R}^\eta \to \mathbb{R}^{n-1}$ that projects a point to its closest point in h. Clearly, $G(F(\cdot))$ is an embedding with the same distortion for P, and the target space is of dimension $n - 1$.

Note, that all this process succeeds with high probability. If it fails, we try again. We conclude:

Theorem 23.5.3 (Low quality Bourgain theorem.). *Given a n-point metric M, one can embed it into Euclidean space of dimension $n - 1$, such that the distortion of the embedding is at most $O(\log^{3/2} n)$.*

Using the Johnson-Lindenstrauss lemma, the dimension can be further reduced to $O(\log n)$. In fact, being more careful in the proof, it is possible to reduce the dimension to $O(\log n)$ directly.

23.6 Bibliographical notes

The partitions we use are due to Calinescu *et al.* [CKR01]. The idea of embedding into spanning trees is due to Alon *et al.* [AKPW95], which showed that one can get a probabilistic distortion of $2^{O\left(\sqrt{\log n \log\log n}\right)}$. Yair Bartal realized that by allowing trees with additional vertices, one can get a considerably better result. In particular, he showed [Bar96] that probabilistic embedding into trees can be done with polylogarithmic average distortion. He later improved the distortion to $O(\log n \log\log n)$ in [Bar98]. Improving this result was an open question, culminating in the work of Fakcharoenphol *et al.* [FRT03] which achieve the optimal $O(\log n)$ distortion.

Interestingly, if one does not care about the optimal distortion, one can get similar result (for embedding into probabilistic trees), by first embedding the metric into Euclidean space, then reduce the dimension by the Johnson-Lindenstrauss lemma, and finally, construct an HST by constructing a quadtree over the points. The "trick" is to randomly translate the quadtree. It is easy to verify that this yields $O(\log^4 n)$ distortion. See the survey by Indyk [Ind01] for more details. This

random shifting of quadtrees is a powerful technique that was used in getting several result, and it is a crucial ingredient in Arora [Aro98] approximation algorithm for Euclidean TSP.

Our proof of Lemma 23.3.1 (which is originally from [FRT03]) is taken from [KLMN05]. The proof of Theorem 23.5.3 is by Gupta [Gup00].

A good exposition of metric spaces is available in Matoušek [Mat02].

23.7 Exercises

Exercise 23.7.1 (Clustering for HST.). Let $(\mathcal{X}, \mathsf{d})$ be a HST defined over n points, and let $k > 0$ be an integer. Provide an algorithm that computes the optimal k-median clustering of $\mathcal{X}$ in $O(k^2 n)$ time.

[**Hint:** Transform the HST into a tree where every node has only two children. Next, run a dynamic programming algorithm on this tree.]

Exercise 23.7.2 (Partition induced metric.).

(a) Give a counter example to the following claim: Let $(\mathcal{X}, \mathsf{d})$ be a metric space, and let P be a partition of $\mathcal{X}$. Then, the pair (P, d') is a metric, where $\mathsf{d}'(C, C') = \mathsf{d}(C, C') = \min_{x \in C, y \in C'} \mathsf{d}(x, y)$ and $C, C' \in P$.

(b) Let $(\mathcal{X}, \mathsf{d})$ be a n-point metric space, and consider the set $U = \left\{ i \;\middle|\; 2^i \le \mathsf{d}(x, y) \le 2^{i+1}, \text{ for } x, y \in \mathcal{X} \right\}$. Prove that $|U| = O(n)$. Namely, there are only n different resolutions that "matter" for a finite metric space.

Exercise 23.7.3 (Computing the diameter via embeddings.).

(a) (h:1) Let ℓ be a line in the plane, and consider the embedding $f : \mathbb{R}^2 \to \ell$, which is the projection of the plane into ℓ. Prove that f is 1-Lipschitz, but it is not K-bi-Lipschitz for any constant K.

(b) (h:3) Prove that one can find a family of projections $\mathcal{F}$ of size $O(1/\sqrt{\varepsilon})$, such that for any two points $x, y \in \mathbb{R}^2$, for one of the projections $f \in \mathcal{F}$ we have $\mathsf{d}(f(x), f(y)) \ge (1 - \varepsilon)\mathsf{d}(x, y)$.

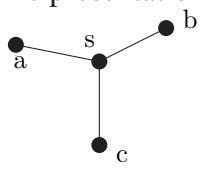
(c) (h:1) Given a set P of n in the plane, given a $O(n/\sqrt{\varepsilon})$ time algorithm that outputs two points $x, y \in P$, such that $\mathsf{d}(x, y) \ge (1 - \varepsilon)\mathrm{diam}(P)$, where $\mathrm{diam}(P) = \max_{z,w \in P} \mathsf{d}(z, w)$ is the diameter of P.

(d) (h:2) Given P, show how to extract, in $O(n)$ time, a set $Q \subseteq P$ of size $O(\varepsilon^{-2})$, such that $\mathrm{diam}(Q) \ge (1 - \varepsilon/2)\mathrm{diam}(P)$. (Hint: Construct a grid of appropriate resolution.)

In particular, give an $(1 - \varepsilon)$-approximation algorithm to the diameter of P that works in $O(n + \varepsilon^{-2.5})$ time. (There are slightly faster approximation algorithms known for approximating the diameter.)

Acknowledgments

The presentation in this write-up follows closely the insightful suggestions of Manor Mendel.



Chapter 24

VC Dimension, ε-nets and ε-approximation

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> "I've never touched the hard stuff, only smoked grass a few times with the boys to be polite, and that's all, though ten is the age when the big guys come around teaching you all sorts to things. But happiness doesn't mean much to me, I still think life is better. Happiness is a mean son of a bitch and needs to be put in his place. Him and me aren't on the same team, and I'm cutting him dead. I've never gone in for politics, because somebody always stand to gain by it, but happiness is an even crummier racket, and their ought to be laws to put it out of business."
>
> – Emile Ajar, Momo.

In this lecture, we would be interested in using sampling to capture or learn a concept. For example, consider an algorithm that tries to learn a classifier, that given positive and negative examples, construct a model of the universe. For example, the inputs are records of clients, and we would like to predict whether or not one should give them a loan.

Clearly, we are trying to approximate a function. The natural question to ask, is how many samples one needs to learn a concept reliably? It turns out that this very fundamental question has a (partial) answer, which is very useful in the developement of algorithms.

24.1 VC Dimension

Definition 24.1.1. A *range space* S is a pair $(X, \mathcal{R})$, where X is a (finite or infinite) set and $\mathcal{R}$ is a (finite or infinite) family of subsets of X. The elements of X are *points* and the elements of $\mathcal{R}$ are *ranges*. For $A \subseteq X$, $P_{\mathcal{R}}(A) = \left\{ r \cap A \;\middle|\; r \in \mathcal{R} \right\}$ is the *projection* of $\mathcal{R}$ on A.

If $P_{\mathcal{R}}(A)$ contains all subsets of A (i.e., if A is finite, we have $|P_{\mathcal{R}}(A)| = 2^{|A|}$) then A is *shattered* by $\mathcal{R}$.

The *Vapnik-Chervonenkis* dimension (or VC-dimension) of S, denoted by $\mathsf{VC}(S)$, is the maximum cardinality of a shattered subset of X. It there are arbitrarily large shattered subsets then $\mathsf{VC}(S) = \infty$.

24.1.1 Examples

Example. Let $X = \mathbb{R}^2$, and let $\mathcal{R}$ be the set of disks in the plane. Clearly, for three points in the plane $1, 2, 3$, one can find 8 disks that realize all possible 2^3 different subsets.

But can disks shatter a set with four points? Consider such a set P of four points, and there are two possible options. Either the convex-hull of P has three points on its boundary, and in this case, the subset having those vertices in the subset but not including the middle point is impossible, by convexity. Alternatively, if all four points are vertices of the convex hull, and they are p_1, p_2, p_3, p_4 along the boundary of the convex hull, either the set $\{p_1, p_3\}$ or the set $\{p_2, p_4\}$ is not realizable. Indeed, if both options are realizable, then consider the two disks D_1, D_2 the realizes those assignments. Clearly, D_1 and D_2 must intersect in four points, but this is not possible, since two disks have at most two intersection points. See Figure 24.1 (b).

Example. Consider the range space $S = (\mathbb{R}^2, \mathcal{R})$, where $\mathcal{R}$ is the set of all (closed) convex sets in the plane. We claim that the $\mathsf{VC}(S) = \infty$. Indeed, consider a set U of n points $p_1, \ldots, p_n$ all lying on the boundary of the unit circle in the plane. Let V be any subset of U, and consider the convex-hull $\mathcal{CH}(V)$. Clearly, $\mathcal{CH}(V) \in \mathcal{R}$, and furthermore, $\mathcal{CH}(V) \cap U = V$. Namely, any subset of U is realizable by S. Thus, S can shatter sets of arbitrary size, and its VC dimension is unbounded.

Example 24.1.2. Let $S = (X, R)$, where $X = \mathbb{R}^d$ and R is the set of all (closed) halfspaces in $\mathbb{R}^d$. To see what is the VC dimension of S, we nee the following result of Radon:

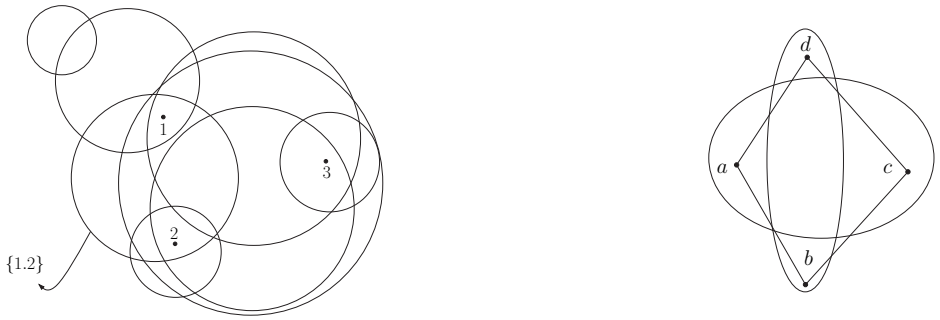
> **Theorem 24.1.3 (Randon's Lemma).** *Let A be a set of $d+2$ points in $\mathbb{R}^d$. Then, there exists two disjoint subsets C, D of A, such that $\mathcal{CH}(C) \cap \mathcal{CH}(D) \neq \emptyset$.*
>
> *Proof*: The points $p_1, \ldots, p_{d+2}$ of A are linearly dependent. As such, there exists $\beta_1, \ldots, \beta_{d+2}$, not all of them zero, such that $\sum_i \beta_i p_i = 0$ and $\sum_i \beta_i = 0$ (to see that, remember that the affine subspace spanned by $p_1, \ldots, p_{d+2}$ is induced by all points that can be represented as $p_1 + \sum_{i=2}^{d+2} \alpha_i (p_i - p_1)$ where $\sum_i \alpha_i = 0$). Assume, for the sake of simplicity of exposition, that the $\beta_1, \ldots \beta_k \geq 0$ and $\beta_{k+1}, \ldots, \beta_{d+2} < 0$. Furthermore, let $\mu = \sum_{i=1}^{k} \beta_i$. We have that
>
> $$\sum_{i=0}^{k} \beta_i p_i = - \sum_{i=k+1}^{d+2} \beta_i p_i.$$
>
> In particular, $v = \sum_{i=0}^{k} (\beta_i/\mu) p_i$ is a point in the $\mathcal{CH}(\{p_1, \ldots, p_k\})$ and $\sum_{i=k+1}^{d+2} -(\beta_i/\mu) p_i \in \mathcal{CH}(\{p_{k+1}, \ldots, p_{d+2}\})$. We conclude that v is in the intersection of the two convex hulls, as required. ■

In particular, this implies that if a set Q of $d+2$ points is being shattered by S, we can partition this set Q into two disjoint sets A and B such that $\mathcal{CH}(A) \cap \mathcal{CH}(B) \neq \emptyset$. It should now be clear that any halfspace h^+ containing all the points of A, must also contain a point of the $\mathcal{CH}(B)$. But this implies that a point of B must be in h^+. Namely, the subset A can not be realized by a halfspace, which implies that Q can not be shattered. Thus $\mathsf{VC}(S) < d+2$. It is also easy to verify that the regular simplex with $d+1$ vertices is being shattered by S. Thus, $\mathsf{VC}(S) = d+1$.

24.2 VC-Dimensions and the number of different ranges

Let

$$g(d, n) = \sum_{i=0}^{d} \binom{n}{i}.$$



(a) (b)

Figure 24.1: Disks in the plane can shatter three points, but not four.

Note that for all $n, d \geq 1$, $g(d, n) = g(d, n-1) + g(d-1, n-1)$

Lemma 24.2.1 (Sauer's Lemma). *If (X, R) is a range space of* VC*-dimension d with $|X| = n$ points then $|R| \leq g(d, n)$.*

Proof: The claim trivially holds for $d = 0$ or $n = 0$.

Let x be any element of X, and consider the sets

$$R_x = \left\{ r \setminus \{x\} \;\middle|\; x \in r, r \in R, r \setminus \{x\} \in R \right\}$$

and

$$R \setminus x = \left\{ r \setminus \{x\} \;\middle|\; r \in R \right\}.$$

Observe that $|R| = |R_x| + |R \setminus x|$ (Indeed, if r does not contain x than it is counted in R_x, if does contain x but $r \setminus x \notin R$, then it is also counted in R_x. The only remaining case is when both $r \setminus \{x\}$ and $r \cup \{x\}$ are in R, but then it is being counted once in R_x and once in $R \setminus x$.)

Observe that R_x has VC dimension $d-1$, as the largest set that can be shattered is of size $d-1$. Indeed, any set $A \subset X$ shattered by R_x, implies that $A \cup \{x\}$ is shattered in R.

Thus,

$$|R| = |R_x| + |R \setminus x| = g(n-1, d-1) + g(n-1, d) = g(d, n),$$

by induction. ∎

By applying Lemma 24.2.1, to a finite subset of X, we get:

Corollary 24.2.2. *If $(X, \mathcal{R})$ is a range space of* VC*-dimension d then for every finite subset A of X, we have $|P_{\mathcal{R}}(A)| \leq g(d, |A|)$.*

Lemma 24.2.3. *Let $S = (X, \mathcal{R})$ and $S' = (X, \mathcal{R}')$ be two range spaces of dimension d and d', respectively, where $d, d' > 1$. Let $\widehat{\mathcal{R}} = \left\{ r \cup r' \;\middle|\; r \in \mathcal{R}, r' \in \mathcal{R}' \right\}$. Then, for the range space $\widehat{S} = (X, \widehat{\mathcal{R}})$, we have that $\mathsf{VC}\left(\widehat{S}\right) = O((d + d') \log(d + d'))$*

Proof: Let A be a set of n points in X that are being shattered by $\widehat{S}$. There are $g(n, d)$ and $g(n, d')$ different assignments for the elements of A by ranges of $\mathcal{R}$ and $\mathcal{R}'$, respectively. Every subset C of A realized by $\widehat{r} \in \widehat{\mathcal{R}}$, is a union of two subsets $A \cap r$ and $A \cap r'$ where $r \in \mathcal{R}$ and $r' \in \mathcal{R}'$. Thus, the number of different subsets of A realized by $\widehat{S}$ is bounded by $g(n, d)g(n, d')$. Thus, $2^n \leq n^d n^{d'}$, for $d, d' > 1$. We conclude $n \leq (d + d') \lg n$, which implies that $n \leq O((d + d') \log(d + d'))$. ∎

24.3 On ε-nets and ε-sampling

Definition 24.3.1. Let (X, R) be a range space, and let A be a finite subset of X. For $0 \leq \varepsilon \leq 1$, a subset $B \subseteq A$, is an *ε-sample* for A if for any range $r \in R$, we have

$$\left| \frac{|A \cap r|}{|A|} - \frac{|B \cap r|}{|B|} \right| \leq \varepsilon.$$

Similarly, $N \subseteq A$ is an *ε-net* for A, if for any range $r \in R$, if $|r \cap A| \geq \varepsilon |A|$ implies that r contains at least one point of N (i.e., $r \cap N \neq \emptyset$).

Theorem 24.3.2. *There is a positive constant c such that if (X, R) is any range space of* VC*-dimension at most d, $A \subseteq X$ is a finite subset and $\varepsilon, \delta > 0$, then a random subset B of cardinality s of A where s is at least the minimum between $|A|$ and*

$$\frac{c}{\varepsilon^2}\left(d \log \frac{d}{\varepsilon} + \log \frac{1}{\delta} \right)$$

is an ε-sample for A with probability at least $1 - \delta$.

Theorem 24.3.3 (ε-net Theorem). *Let (X, R) be a range space of* VC*-dimension d, let A be a finite subset of X and suppose $0 < \varepsilon, \delta < 1$. Let N be a set obtained by m random independent draws from A, where*

$$m \geq \max\left(\frac{4}{\varepsilon} \log \frac{2}{\delta}, \frac{8d}{\varepsilon} \log \frac{8d}{\varepsilon} \right). \tag{24.1}$$

Then N is an ε-net for A with probability at least $1 - \delta$.

24.4 Proof of the ε-net Theorem

Let (X, R) be a range space of VC-dimension d, and let A be a subset of X of cardinality n. Suppose that m satisfiers Eq. (24.1). Let $N = (x_1, \ldots, x_m)$ be the sample obtained by m independent samples from A (the elements of N are not necessarily distinct, and thats why we treat N as a ordered set). Let E_1 be the probability that N fails to be an ε-net. Namely,

$$E_1 = \left\{ \exists r \in R \;\middle|\; |r \cap A| \geq \varepsilon n, r \cap N = \emptyset \right\}.$$

(Namely, there exists a "heavy" range r that does not contain any point of N.) To complete the proof, we must show that $\mathbf{Pr}[E_1] \leq \delta$. Let $T = (y_1, \ldots, y_m)$ be another random sample generated in a similar fashion to N. Let E_2 be the event that N fails, but T "works", formally

$$E_2 = \left\{ \exists r \in R \;\middle|\; |r \cap A| \geq \varepsilon n, r \cap N = \emptyset, |r \cap T| \geq \frac{\varepsilon m}{2} \right\}.$$

Intuitively, since $E_T\Big[|r \cap T|\Big] \geq \varepsilon m$, then for the range r that N fails for, we have with "good" probability that $|r \cap T| \geq \frac{\varepsilon n}{2}$. Namely, E_1 and E_2 have more or less the same probability.

Claim 24.4.1. $\mathbf{Pr}[E_2] \leq \mathbf{Pr}[E_1] \leq 2\,\mathbf{Pr}[E_2]$.

Proof: Clearly, $E_2 \subseteq E_1$, and thus $\mathbf{Pr}[E_2] \leq \mathbf{Pr}[E_1]$. As for the other part, note that $\mathbf{Pr}\left[E_2 \;\middle|\; E_1\right] = \mathbf{Pr}[E_2 \cap E_1] \,/\, \mathbf{Pr}[E_1] = \mathbf{Pr}[E_2] \,/\, \mathbf{Pr}[E_1]$. It is thus enough to show that $\mathbf{Pr}\left[E_2 \;\middle|\; E_1\right] \geq 1/2$.

Assume that E_1 occur. There is $r \in R$, such that $|r \cap A| > \varepsilon n$ and $r \cap N = \emptyset$. The required probability is at least the probability that for this specific r, we have $|r \cap T| \geq \frac{\varepsilon n}{2}$. However, $|r \cap T|$ is a binomial variable with expectation εm, and variance $\varepsilon(1-\varepsilon)m \leq \varepsilon m$. Thus, by Chebychev inequality (Theorem 4.3.3),

$$\mathbf{Pr}\left[|r \cap T| < \frac{\varepsilon m}{2}\right] \leq \mathbf{Pr}\left[\left||r \cap T| - \varepsilon m\right| > \frac{\varepsilon m}{2}\right] \mathbf{Pr}\left[\left||r \cap T| - \varepsilon m\right| > \frac{\sqrt{\varepsilon m}}{2}\sqrt{\varepsilon m}\right] \leq \frac{4}{\varepsilon m} \leq \frac{1}{2},$$

by Eq. (24.1). Thus, $\mathbf{Pr}[E_2] \,/\, \mathbf{Pr}[E_1] = \mathbf{Pr}\left[|r \cap T| \geq \frac{\varepsilon n}{2}\right] = 1 - \mathbf{Pr}\left[|r \cap T| < \frac{\varepsilon m}{2}\right] \geq \frac{1}{2}$. ∎

Thus, it is enough to bound the probability of E_2. Let

$$E_2' = \left\{\exists r \in R \;\middle|\; r \cap N = \emptyset, |r \cap T| \geq \frac{\varepsilon m}{2}\right\},$$

Clearly, $E_2 \subseteq E_2'$. Thus, bounding the probability of E_2' is enough to prove the theorem. Note however, that a shocking thing happened! We no longer have A as participating in our event. Namely, we turned bounding an event that depends on a global quantity, into bounding a quantity that depends only on local quantity/experiment. This is the crucial idea in this proof.

Claim 24.4.2. $\mathbf{Pr}[E_2] \leq \mathbf{Pr}\left[E_2'\right] \leq g(d, 2m)2^{-\varepsilon m/2}$.

Proof: We imagine that we sample the elements of $N \cup T$ together, by picking a set $Z = (z_1, \ldots, z_{2m})$ from A, by picking each element independently from A. Next, we randomly decide which of the m elements of Z form N, and remaining elements from T. Clearly,

$$\mathbf{Pr}[E_2'] = \sum_Z \mathbf{Pr}\left[E_2' \;\middle|\; Z\right] \mathbf{Pr}[Z].$$

Thus, from this point on, we fix the set Z, and we bound $\mathbf{Pr}\left[E_2' \;\middle|\; Z\right]$.

It is now enough to consider the ranges in the projection space $P_R(Z)$. By Lemma 24.2.1, we have $|P_R(Z)| \leq g(d, 2m)$.

Let us fix any $r \in P_R(Z)$, and consider the event

$$E_r = \left\{r \cap N = \emptyset \text{ and } |r \cap T| > \frac{\varepsilon m}{2}\right\}.$$

For $k = |r \cap (N \cup T)|$, we have

$$\begin{aligned}
\mathbf{Pr}[E_r] &\leq \mathbf{Pr}\left[r \cap N = \emptyset \;\middle|\; |r \cap (N \cup T)| > \frac{\varepsilon m}{2}\right] = \frac{\binom{2m-k}{m}}{\binom{2m}{m}} \\
&= \frac{(2m-k)(2m-k-1)\cdots(m-k+1)}{2m(2m-1)\cdots(m+1)} \\
&= \frac{m(m-1)\cdots(m-k+1)}{2m(2m-1)\cdots(2m-k+1)} \leq 2^{-k} \leq 2^{-\varepsilon m/2}.
\end{aligned}$$

Thus,

$$\mathbf{Pr}\Big[E_2' \;\Big|\; Z\Big] \leq \sum_{r \in P_R(Z)} \mathbf{Pr}[E_r] \leq |P_R(Z)|\, 2^{-\varepsilon m/2} = g(d, 2m) 2^{-\varepsilon m/2},$$

implying that $\mathbf{Pr}\Big[E_2'\Big] \leq g(d, 2m) 2^{-\varepsilon m/2}$. ∎

Proof of Theorem 24.3.3: By Lemma 24.4.1 and Lemma 24.4.2, we have $\mathbf{Pr}[E_1] \leq 2g(d, 2m)2^{-\varepsilon m/2}$. It is thus remains to verify that if m satisfies Eq. (24.1), then $2g(d, 2m)2^{-\varepsilon m/2} \leq \delta$. One can verify that this inequality is implied by Eq. (24.1).

Indeed, we know that $2m \geq 8d$ and as such $g(d, 2m) = \sum_{i=0}^{d} \binom{2m}{i} \leq \sum_{i=0}^{d} \frac{(2m)^i}{i!} \leq (2m)^d$, for $d > 1$. Thus, it is sufficient to show that the inequality $2(2m)^d 2^{-\varepsilon m/2} \leq \delta$ holds. By taking lg of both sides and rearranging, we have that this is equivalent to

$$\frac{\varepsilon m}{2} \geq d \lg(2m) + \lg \frac{2}{\delta}.$$

By our choice of m (see Eq. (24.1)), we have that $\varepsilon m/4 \geq \lg(2/\delta)$. Thus, we need to show that

$$\frac{\varepsilon m}{4} \geq d \lg(2m).$$

We verify this inequality for $m = \frac{8d}{\varepsilon} \lg \frac{8d}{\varepsilon}$, indeed

$$2d \lg \frac{8d}{\varepsilon} \geq d \lg\left(\frac{16d}{\varepsilon} \lg \frac{8d}{\varepsilon}\right).$$

This is equivalent to $\left(\frac{8d}{\varepsilon}\right)^2 \geq \frac{16d}{\varepsilon} \lg \frac{8d}{\varepsilon}$. Which is equivalent to $\frac{4d}{\varepsilon} \geq \lg \frac{8d}{\varepsilon}$, which is certainly true for $0 \leq \varepsilon \leq 1$ and $d > 1$. Note that it is easy to verify that the inequality holds for $m \geq \frac{8d}{\varepsilon} \lg \frac{8d}{\varepsilon}$, by deriving both sides of the inequality.

This completes the proof of the theorem. ∎

24.5 Exercises

Exercise 24.5.1 (Flip and Flop.). (A) [**5 points**] Let $b_1, \ldots, b_{2m}$ be m binary bits. Let Ψ be the set of all permutations of $1, \ldots, 2m$, such that for any $\sigma \in \Psi$, we have $\sigma(i) = i$ or $\sigma(i) = m + i$, for $1 \leq i \leq m$, and similarly, $\sigma(m + i) = i$ or $\sigma(m + i) = m + i$. Namely, $\sigma \in \Psi$ either leave the pair $i, i + m$ in their positions, or it exchange them, for $1 \leq i \leq m$. As such $|\Psi| = 2^m$.

Prove that for a random $\sigma \in \Psi$, we have

$$\mathbf{Pr}\left[\left|\frac{\sum_{i=1}^{m} b_{\sigma(i)}}{m} - \frac{\sum_{i=1}^{m} b_{\sigma(i+m)}}{m}\right| \geq \varepsilon\right] \leq 2e^{-\varepsilon^2 m/2}.$$

(B) [**5 points**] Let Ψ' be the set of all permutations of $1, \ldots, 2m$. Prove that for a random $\sigma \in \Psi'$, we have

$$\mathbf{Pr}\left[\left|\frac{\sum_{i=1}^{m} b_{\sigma(i)}}{m} - \frac{\sum_{i=1}^{m} b_{\sigma(i+m)}}{m}\right| \geq \varepsilon\right] \leq 2e^{-C\varepsilon^2 m/2},$$

where C is an appropriate constant. [**Hint:** Use (A), but be careful.]

(C) [**10 points**] Prove Theorem 24.3.2 using (B).

Exercise 24.5.2 (Dual VC dimension.). Let $(X, \mathcal{R})$ be a range space with VC dimension d, and let $A \subseteq X$ be a finite set. Consider the induced range space $\mathsf{S} = (A, P_R(A))$.

Next, for a point $\mathsf{q} \in A$, let $\mathcal{R}(\mathsf{q})$ denote the set of all the ranges of $P_R(A)$ that contains is, and consider the *dual range space* $\mathsf{D} = \left(P_R(A), \left\{ \mathcal{R}(\mathsf{q}) \;\middle|\; \mathsf{q} \in A \right\}\right)$.

Prove that the VC dimension of D is at most 2^d.

Exercise 24.5.3 (On VC dimension.). (A) Prove directly a bound on the VC dimension of the range space of ellipses in two dimensions (i.e., the ranges are the interior of ellipses). Show a matching lower bound (or as matching as you can).

(B) Prove that the VC dimension of regions defined by a polynomial of degree at most s in d dimensions is bounded. Such an inequality might be for example $ax^2 + bxy + y^3 - x^2y^2 \leq 3$ ($s = 2 + 2 = 4$ in this example), and the region it defines is all the points that comply with this inequality.

[**Hint:** Consider a mapping of $\mathbb{R}^d$ into $\mathbb{R}^k$, such that all polynomials of degree s correspond to linear inequalities.**]**

Exercise 24.5.4 (Dual VC dimension.). Let $(X, \mathcal{R})$ be a range space with VC dimension d, and let $A \subseteq X$ be a finite set. Consider the induced range space $\mathsf{S} = (A, P_R(A))$.

Next, for a point $\mathsf{q} \in A$, let $\mathcal{R}(\mathsf{q})$ denote the set of all the ranges of $P_R(A)$ that contains is, and consider the *dual range space* $\mathsf{D} = \left(P_R(A), \left\{ \mathcal{R}(\mathsf{q}) \;\middle|\; \mathsf{q} \in A \right\}\right)$.

Prove that the VC dimension of D is at most 2^d.

Exercise 24.5.5 (Improved Hitting Set.). Let $(X, \mathcal{R})$ be a range space with constant VC dimension d. Furthermore, assume that you have access to an oracle, such that given a finite set $A \subseteq X$ of n elements, it computes the range space $\mathsf{S} = (A, P_{\mathcal{R}}(A))$ in time $O(|A| + |P_{\mathcal{R}}(A)|)$.

(A) Assume, that ever element of $\mathsf{q} \in A$ has an associated weight w_{q}, where the weight is a positive integer number. Show, how to compute ε-net efficiently so that it is an ε-net for the weighted points.

(B) In fact, the computation in the previous part would be slow if the weights are very large integers. To make things easier, assume very weight w_{q} is of the form 2^j, where j is a non-negative integer bounded by a parameter M. Show how to compute efficiently an ε-net in this case. (You can assume that computations on integers smaller than $M^O(1)$ can be performed in constant time.)

(C) Prove the following theorem:

Theorem 24.5.6. *Let $(X, \mathcal{R})$ be a range space with constant VC dimension d. Let A be subset of X with n elements. Furthermore, assume that there is a* hitting set *$H \subseteq A$ of size k for $(A, P_{\mathcal{R}}(A))$. Namely, any range $\mathbf{r}$ of $P_{\mathcal{R}}(A)$ contains a point of H.*

Then one can compute in polynomial time, a set U of $O(dk \log(dk))$ points of X, such that U is a hitting set for $\mathsf{S} = (A, P_{\mathcal{R}}(A))$.

To this end, assign weight 1 to all the points of A. Next, consider an δ-net for S, for the appropriate δ. If it is the required hitting set, then we are done. Otherwise, consider a "light" range (which is not being hit) and double the weight of its elements. Repeat. Argue that this algorithm terminates (by comparing the weight of H to the weight of the whole set A). What is the number of iterations of the algorithm being performed? What is the required value of δ? What is the exact size of the generated hitting set.

(D) Show a polynomial time algorithm that compute a hitting set of the range space $\mathsf{S} = (A, P_{\mathcal{R}}(A))$, of size $O(kd \log(kd))$, where d is the VC dimension of S, $n = |A|$, and k is the smallest hitting set of S. What is the expected running time of your algorithm?

(This is interesting because in general the smallest hitting set of a range space can not be approximated within a factor better than $\Omega(\log n)$ unless $P = NP$.)

24.6 Bibliographical notes

The exposition here is based on [AS00]. The usual exposition of the ε-net/ε-sample tend to be long and tedious in the learning literature. The proof of the ε-net theorem is due Haussler and Welzl [HW87]. The proof of the ε-sample theorem is due to Vapnik and Chervonenkis [VC71]. However, the importance of Vapnik and Chervonenkis result was not realized at the time, and only in the late eighties the strong connection to learning was established.

An alternative proof of both theorems exists via the usage of discrepancy. Using discrepancy, one can compute ε-samples and ε-nets deterministically. In fact, in some geometric cases, discrepancy yields better results than the ε-net and ε-sample theorem. See [Mat99, Cha01] for more details.

Exercise 24.5.1 is from Anthony and Bartlett [AB99].

Chapter 25

Approximate Max Cut

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

25.1 Problem Statement

Given an undirected graph $G = (V, E)$ and nonnegative weights w_{ij} on the edge $ij \in E$, the *maximum cut problem* (MAX CUT) is that of finding the set of vertices S that maximizes the weight of the edges in the cut $(S, \overline{S})$; that is, the weight of the edges with one endpoint in S and the other in $\overline{S}$. For simplicity, we usually set $w_{ij} = O$ for $ij \notin E$ and denote the weight of a cut $(S, \overline{S})$ by $w(S, \overline{S}) = \sum_{i \in S, j \in \overline{j}} w_{ij}$.

This problem is NP-Complete, and hard to approximate within a certain constant.

Given a graph with vertex set $V = 1, \ldots, n$ and nonnegative weights W_{ij}, the weight of the maximum cut $w(S, \overline{S})$ N given by the following integer quadratic program:

$$\begin{aligned} \text{(Q)} \quad \text{Maximize} \quad & \frac{1}{2} \sum_{i<j}, w_{ij}(1 - y_i y_j) \\ \text{subject to: } & y_i \in \{-1, 1\} \qquad \forall i \in V. \end{aligned}$$

Indeed, set $S = \left\{ i \;\middle|\; y_i = 1 \right\}$. Clearly, $w(S, \overline{S}) = \frac{1}{2} \sum_{i<j}, w_{ij}(1 - y_i y_j)$.

Solving quadratic integer programming is of course NP-Hard. Thus, we we will relax it, by thinking about the the numbers y_i as unit vectors in higher dimensional space. If so, the multiplication of the two vectors, is now replaced by dot product. We have:

$$\begin{aligned} (P) \quad \text{Maximize} \quad & \tfrac{1}{2} \textstyle\sum_{i<j} w_{ij}(1 - \left\langle v_i, v_j \right\rangle) \\ \text{subject to:} \quad & v_i \in \mathbb{S}^{(n)} \qquad \forall i \in V, \end{aligned}$$

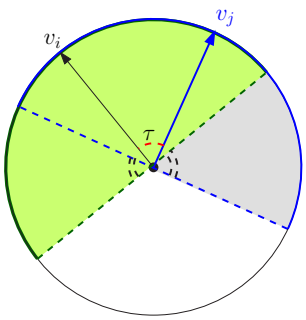
where $\mathbb{S}^{(n)}$ is the n dimensional unit sphere in $\mathbb{R}^{n+1}$. This is an instance of semi-definite programming, which is a special case of convex programming, which can be solved in polynomial time (solved here means approximated within arbitrary constant in polynomial time). Observe that (P) is a relaxation of (Q), and as such the optimal solution of (P) has value larger than the optimal value of (Q).

The intuition is that vectors that correspond to vertices that should be on one side of the cut, and vertices on the other sides, would have vectors which are faraway from each other in (P). Thus, we compute the optimal solution for (P), and we uniformly generate a random vector $\vec{r}$ on the unit sphere $\mathbb{S}^{(n)}$. This induces a hyperplane h which passes through the origin and is orthogonal to $\vec{r}$. We next assign all the vectors that are on one side of h to S, and the rest to $\overline{S}$.

25.1.1 Analysis

The intuition of the above rounding procedure, is that with good probability, vectors that have big angle between them would be separated by this cut.

Lemma 25.1.1. *We have* $\mathbf{Pr}\Big[\operatorname{sign}(\langle v_i, \vec{r}\,\rangle) \neq \operatorname{sign}\Big(\big\langle v_j, \vec{r}\,\big\rangle\Big)\Big] = \dfrac{1}{\pi}\arccos\Big(\big\langle v_i, v_j\big\rangle\Big)$.



Proof: Let us think about the vectors v_i, v_j and $\vec{r}$ as being in the plane. To see why this is a reasonable assumption, consider the plane g spanned by v_i and v_j, and observe that for the random events we consider, only the direction of $\vec{r}$ matter, which can be decided by projecting $\vec{r}$ on g, and normalizing it to have length 1. Now, the sphere is symmetric, and as such, sampling $\vec{r}$ randomly from $\mathbb{S}^{(n)}$, projecting it down to g, and then normalizing it, is equivalent to just choosing uniformly a vector from the unit circle.

Now, $\operatorname{sign}(\langle v_i, \vec{r}\,\rangle) \neq \operatorname{sign}\Big(\big\langle v_j, \vec{r}\,\big\rangle\Big)$ happens only if $\vec{r}$ falls in the double wedge formed by the lines perpendicular to v_i and v_j. The angle of this double wedge is exactly the angle between v_i and v_j. Now, since v_i and v_j are unit vectors, we have $\big\langle v_i, v_j\big\rangle = \cos(\tau)$, where $\tau = \angle v_i v_j$. Thus, $\mathbf{Pr}\Big[\operatorname{sign}(\langle v_i, \vec{r}\,\rangle) \neq \operatorname{sign}\Big(\big\langle v_j, \vec{r}\,\big\rangle\Big)\Big] = 2\tau/(2\pi) = \frac{1}{\pi}\cdot\arccos\Big(\big\langle v_i, v_j\big\rangle\Big)$, as claimed. ■

Theorem 25.1.2. *Let W be the random variable which is the weight of the cut generated by the algorithm. We have*

$$\mathbf{E}[W] = \frac{1}{\pi}\sum_{i<j} w_{ij}\arccos\Big(\big\langle v_i, v_j\big\rangle\Big).$$

Proof: Let X_{ij} be an indicator variable which is 1 if ij is in the cut. We have $\mathbf{E}\Big[X_{ij}\Big] = \mathbf{Pr}\Big[\operatorname{sign}(\langle v_i, \vec{r}\,\rangle) \neq \operatorname{sign}\Big(\big\langle v_j, \vec{r}$ $\frac{1}{\pi}\arccos\Big(\big\langle v_i, v_j\big\rangle\Big)$, by Lemma 25.1.1.

Clearly, $W = \sum_{i<j} w_{ij}X_{ij}$, and by linearity of expectation, we have

$$\mathbf{E}[W] = \sum_{i<j} w_{ij}\,\mathbf{E}\Big[X_{ij}\Big] = \sum_{i<j}\arccos\Big(\big\langle v_i, v_j\big\rangle\Big).$$

■

Lemma 25.1.3. *For $-1 \le y \le 1$, we have* $\dfrac{\arccos(y)}{\pi} \ge \alpha\cdot\dfrac{1}{2}(1-y)$, *where* $\alpha = \min_{0\le\psi\le\pi}\dfrac{2}{\pi}\dfrac{\psi}{1-\cos(\psi)}$.

Proof: Set $y = \cos(\psi)$. The inequality now becomes $\frac{\psi}{\pi} \ge \alpha\frac{1}{2}(1-\cos\psi)$. Reorganizing, the inequality becomes $\frac{2}{\pi}\frac{\psi}{1-\cos\psi} \ge \alpha$, which trivially holds by the definition of α. ■

Lemma 25.1.4. $\alpha > 0.87856$.

Proof: Using simple calculus, one can see that α achieves its value for $\psi = 2.331122...$, the nonzero root of $cos\psi + \psi \sin\psi = 1$. ∎

Theorem 25.1.5. *The above algorithm computes in expectation a cut of size* $\alpha\text{Opt} \geq 0.87856\text{Opt}$*, where* Opt *is the weight of the maximal cut.*

Proof: Consider the optimal solution to (P), and lets its value be $\gamma \geq \text{Opt}$. We have

$$\mathbf{E}[W] = \frac{1}{\pi}\sum_{i<j} w_{ij} \arccos\Big(\big\langle v_i, v_j \big\rangle\Big) \geq \sum_{i<j} w_{ij}\alpha\frac{1}{2}\Big(1 - \big\langle v_i, v_j \big\rangle\Big) = \alpha\gamma \geq \alpha\text{Opt},$$

by Lemma 25.1.3. ∎

25.2 Semi-definite programming

Let us define a variable $x_{ij} = \big\langle v_i, v_j \big\rangle$, and consider the n by n matrix M formed by those variables, where $x_{ii} = 1$ for $i = 1, \ldots, n$. Let V be the matrix having $v_1, \ldots, v_n$ as its columns. Clearly, $M = V^T V$. In particular, this implies that for any non-zero vector $v \in \mathbb{R}^n$, we have $v^T M v = v^T A^T A v = (Av)^T (Av) \geq 0$. A matrix that has this property, is called ***semidefinite***. The interesting thing is that any semi-definite matrix P can be represented as a product of a matrix with its transpose; namely, $P = B^T B$. It is easy to observe that if this semi-definite matrix has a diagonal one, then B has rows which are unit vectors. Thus, if we solve (P) and get back a semi-definite matrix, then we can recover the vectors realizing the solution, and use them for the rounding.

In particular, (P) can now be restated as

$$\begin{aligned} (SD) \quad \text{Maximize} \quad & \tfrac{1}{2}\textstyle\sum_{i<j} w_{ij}(1 - x_{ij}) \\ x_{ii} = 1 \quad & \text{for } i = 1, \ldots, n \\ \text{subject to:} \quad & \big(x_{ij}\big)_{i=1,\ldots,n, j=1,\ldots,n} \text{ is semi-definite.} \end{aligned}$$

We are trying to find the optimal value of a linear function over a set which is the intersection of linear constraints and the set of semi-definite matrices.

Lemma 25.2.1. *Let* $\mathcal{U}$ *be the set of* $n \times n$ *semidefinite matrices. The set* $\mathcal{U}$ *is convex.*

Proof: Consider $A, B \in \mathcal{U}$, and observe that for any $t \in [0, 1]$, and vector $v \in \mathbb{R}^n$, we have: $v^T(tA + (1-t)B)v = tv^T Av + (1-t)v^T Bv \geq 0 + 0 \geq 0$, since A and B are semidefinite. ∎

Positive semidefinite matrices corresponds to ellipsoids. Indeed, consider the set $x^T Ax = 1$: the set of vectors that solve this equation is an ellipsoid. Also, the eigenvalues of a positive semidefinite matrix are all non-negative real numbers. Thus, given a matrix, we can in polynomial time decide if it is positive semidefinite or not.

Thus, we are trying to optimize a linear function over a convex domain. There is by now machinery to approximately solve those problems to within any additive error in polynomial time. This is done by using interior point method, or the ellipsoid method. See [BV04, GLS93] for more details.

25.3 Bibliographical Notes

The approximation algorithm presented is from the work of Goemans and Williamson [GW95]. Håstad [Hås01] showed that MAX CUT can not be approximated within a factor of $16/17 \approx 0.941176$. Recently, Khot et al. [KKMO04] showed a hardness result that matches the constant of Goemans and Williamson (i.e., one can not approximate it better than ϕ, unless **P** = **NP**). However, this relies on two conjectures, the first one is the "Unique Games Conjecture", and the other one is "Majority is Stablest". The "Majority is Stablest" conjecture was recently proved by Mossel *et al.* [MOO05]. However, it is not clear if the "Unique Games Conjecture" is true, see the discussion in [KKMO04].

The work of Goemans and Williamson was very influential and spurred wide research on using SDP for approximation algorithms. For an extension of the MAX CUT problem where negative weights are allowed and relevant references, see the work by Alon and Naor [AN04].

Chapter 26

Entropy, Randomness, and Information

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> "If only once - only once - no matter where, no matter before what audience - I could better the record of the great Rastelli and juggle with thirteen balls, instead of my usual twelve, I would feel that I had truly accomplished something for my country. But I am not getting any younger, and although I am still at the peak of my powers there are moments - why deny it? - when I begin to doubt - and there is a time limit on all of us."
>
> – –Romain Gary, The talent scout..

26.1 Entropy

Definition 26.1.1. The ***entropy*** in bits of a discrete random variable X is given by

$$\mathbb{H}(X) = -\sum_x \mathbf{Pr}[X = x] \lg \mathbf{Pr}[X = x] .$$

Equivalently, $\mathbb{H}(X) = \mathbf{E}\left[\lg \frac{1}{\mathbf{Pr}[X]}\right]$.

The ***binary entropy*** function $\mathbb{H}(p)$ for a random binary variable that is 1 with probability p, is $\mathbb{H}(p) = -p \lg p - (1-p)\lg(1-p)$. We define $\mathbb{H}(0) = \mathbb{H}(1) = 0$.

The function $\mathbb{H}(p)$ is a concave symmetric around $1/2$ on the interval $[0, 1]$ and achieves its maximum at $1/2$. For a concrete example, consider $\mathbb{H}(3/4) \approx 0.8113$ and $\mathbb{H}(7/8) \approx 0.5436$. Namely, a coin that has $3/4$ probably to be heads have higher amount of "randomness" in it than a coin that has probability $7/8$ for heads.

We have that

$$\mathbb{H}(p) = \frac{1}{\ln 2}\Big(- p \ln p - (1-p)\ln(1-p)\Big)$$

$$\text{and} \quad \mathbb{H}'(p) = \frac{1}{\ln 2}\left(- \ln p - \frac{p}{p} - (-1)\ln(1-p) - \frac{1-p}{1-p}(-1)\right) = \lg \frac{1-p}{p}.$$

Deploying our amazing ability to compute derivative of simple functions once more, we get that

$$\mathbb{H}''(p) = \frac{1}{\ln 2}\frac{p}{1-p}\left(\frac{p(-1)-(1-p)}{p^2}\right) = -\frac{1}{p(1-p)\ln 2}.$$

Since $\ln 2 \approx 0.693$, we have that $\mathbb{H}''(p) \leq 0$, for all $p \in (0,1)$, and the $\mathbb{H}(\cdot)$ is concave in this range. Also, $\mathbb{H}'(1/2) = 0$, which implies that $\mathbb{H}(1/2) = 1$ is a maximum of the binary entropy. Namely, a balanced coin has the largest amount of randomness in it.

Example 26.1.2. A random variable X that has probability $1/n$ to be i, for $i = 1, \ldots, n$, has entropy $\mathbb{H}(X) = -\sum_{i=1}^{n} \frac{1}{n} \lg \frac{1}{n} = \lg n$.

Note, that the entropy is oblivious to the exact values that the random variable can have, and it is sensitive only to the probability distribution. Thus, a random variables that accepts $-1, +1$ with equal probability has the same entropy (i.e., 1) as a fair coin.

Lemma 26.1.3. *Let X and Y be two independent random variables, and let Z be the random variable (X, T). Then $\mathbb{H}(Z) = \mathbb{H}(X) + \mathbb{H}(Y)$.*

Proof: In the following, summation are over all possible values that the variables can have. By the independence of X and Y we have

$$\begin{aligned}
\mathbb{H}(Z) &= \sum_{x,y} \mathbf{Pr}[(X,Y) = (x,y)] \lg \frac{1}{\mathbf{Pr}[(X,Y) = (x,y)]} \\
&= \sum_{x,y} \mathbf{Pr}[X = x]\,\mathbf{Pr}[Y = y] \lg \frac{1}{\mathbf{Pr}[X = x]\,\mathbf{Pr}[Y = y]} \\
&= \sum_{x} \sum_{y} \mathbf{Pr}[X = x]\,\mathbf{Pr}[Y = y] \lg \frac{1}{\mathbf{Pr}[X = x]} \\
&\quad + \sum_{y} \sum_{x} \mathbf{Pr}[X = x]\,\mathbf{Pr}[Y = y] \lg \frac{1}{\mathbf{Pr}[Y = y]} \\
&= \sum_{x} \mathbf{Pr}[X = x] \lg \frac{1}{\mathbf{Pr}[X = x]} + \sum_{y} \mathbf{Pr}[Y = y] \lg \frac{1}{\mathbf{Pr}[Y = y]} = \mathbb{H}(X) + \mathbb{H}(Y).
\end{aligned}$$

■

Lemma 26.1.4. *Suppose that nq is integer in the range $[0, n]$. Then $\dfrac{2^{n\mathbb{H}(q)}}{n+1} \leq \binom{n}{nq} \leq 2^{n\mathbb{H}(q)}$.*

Proof: This trivially holds if $q = 0$ or $q = 1$, so assume $0 < q < 1$. We know that $\binom{n}{nq} q^{nq}(1-q)^{n-nq} \leq (q + (1-q))^n = 1$. As such, since $q^{-nq}(1-q)^{-(1-q)n} = 2^{n\left(-q\lg q - (1-q)\lg(1-q)\right)} = 2^{n\mathbb{H}(q)}$, we have

$$\binom{n}{nq} \leq q^{-nq}(1-q)^{-(1-q)n} = 2^{n\mathbb{H}(q)}.$$

As for the other direction, we claim that $\mu(nq) = \binom{n}{nq} q^{nq}(1-q)^{n-nq}$ is the largest term in $\sum_{k=0}^{n} \mu(k) = 1$, where $\mu(k) = \binom{n}{k} q^k (1-q)^{n-k}$. Indeed,

$$\Delta_k = \mu(k) - \mu(k+1) = \binom{n}{k} q^k (1-q)^{n-k} \left(1 - \frac{n-k}{k+1} \frac{q}{1-q}\right),$$

and the sign of this quantity is the sign of $(k+1)(1-q) - (n-k)q = k+1-kq-q-nq+kq = 1+k-q-nq$. Namely, $\Delta_k \geq 0$ when $k \geq nq + q - 1$, and $\Delta_k < 0$ otherwise. Namely, $\mu(k) < \mu(k+1)$, for $k < nq$, and $\mu(k) \geq \mu(k+1)$ for $k \geq nq$. Namely, $\mu(nq)$ is the largest term in $\sum_{k=0}^{n} \mu(k) = 1$, and as such it is larger than the average. We have $\mu(nq) = \binom{n}{nq} q^{nq}(1-q)^{n-nq} \geq \frac{1}{n+1}$, which implies

$$\binom{n}{nq} \geq \frac{1}{n+1} q^{-nq} (1-q)^{-(n-nq)} = \frac{1}{n+1} 2^{n\mathbb{H}(q)}.$$

■

Lemma 26.1.4 can be extended to handle non-integer values of q. This is straightforward, and we omit the easy details.

Corollary 26.1.5. *We have:* *(i)* $q \in [0, 1/2] \Rightarrow \binom{n}{\lfloor nq \rfloor} \leq 2^{n\mathbb{H}(q)}$. *(ii)* $q \in [1/2, 1]$ $\binom{n}{\lceil nq \rceil} \leq 2^{n\mathbb{H}(q)}$. *(iii)* $q \in [1/2, 1] \Rightarrow \frac{2^{n\mathbb{H}(q)}}{n+1} \leq \binom{n}{\lfloor nq \rfloor}$. *(iv)* $q \in [0, 1/2] \Rightarrow \frac{2^{n\mathbb{H}(q)}}{n+1} \leq \binom{n}{\lceil nq \rceil}$.

The bounds of Lemma 26.1.4 and Corollary 26.1.5 are loose but sufficient for our purposes. As a sanity check, consider the case when we generate a sequence of n bits using a coin with probability q for head, then by the Chernoff inequality, we will get roughly nq heads in this sequence. As such, the generated sequence Y belongs to $\binom{n}{nq} \approx 2^{n\mathbb{H}(q)}$ possible sequences that have similar probability. As such, $\mathbb{H}(Y) \approx \lg \binom{n}{nq} = n\mathbb{H}(q)$, by Example 26.1.2, a fact that we already know from Lemma 26.1.3.

26.1.1 Extracting randomness

Entropy can be interpreted as the amount of unbiased random coin flips can be extracted from a random variable.

Definition 26.1.6. An extraction function $\texttt{Ext}$ takes as input the value of a random variable X and outputs a sequence of bits y, such that $\mathbf{Pr}\left[\texttt{Ext}(X) = y \;\middle|\; |y| = k\right] = \frac{1}{2^k}$, whenever $\mathbf{Pr}[|y| = k] \geq 0$, where $|y|$ denotes the length of y.

As a concrete (easy) example, consider X to be a uniform random integer variable out of $0, \ldots, 7$. All that $\texttt{Ext}(x)$ has to do in this case, is just to compute the binary representation of x. However, note that Definition 26.1.6 is somewhat more subtle, as it requires that all extracted sequence of the same length would have the same probability.

Thus, for X a uniform random integer variable in the range $0, \ldots, 11$, the function $\texttt{Ext}(x)$ can output the binary representation for x if $0 \leq x \leq 7$. However, what do we do if x is between 8 and 11? The idea is to output the binary representation of $x - 8$ as a two bit number. Clearly, Definition 26.1.6 holds for this extraction function, since $\mathbf{Pr}\left[\texttt{Ext}(X) = 00 \;\middle|\; |\texttt{Ext}(X)| = 2\right] = \frac{1}{4}$, as required. This scheme can be of course extracted for any range.

Theorem 26.1.7. *Suppose that the value of a random variable X is chosen uniformly at random from the integers $\{0, \ldots, m-1\}$. Then there is an extraction function for X that outputs on average (i.e., in expectation) at least $\lfloor \lg m \rfloor - 1 = \lfloor \mathbb{H}(X) \rfloor - 1$ independent and unbiased bits.*

Proof: We represent m as a sum of unique powers of 2, namely $m = \sum_i a_i 2^i$, where $a_i \in \{0, 1\}$. Thus, we decomposed $\{0, \ldots, m-1\}$ into a disjoint union of blocks that have sizes which are distinct powers of 2. If a number falls inside such a block, we output its relative location in the block, using binary representation of the appropriate length (i.e., k if the block is of size 2^k). The fact that this is an extraction function, fulfilling Definition 26.1.6, is obvious.

Now, observe that the claim holds trivially if m is a power of two. Thus, if m is not a power of 2, then in the decomposition if there is a block of size 2^k, and the X falls inside this block, then the entropy is k. Thus, for the inductive proof, assume that are looking at the largest block in the decomposition, that is $m < 2^{k+1}$, and let $u = \left\lfloor \lg(m - 2^k) \right\rfloor < k$. It is easy to verify that, for any integer $\alpha > 2^k$, we have $\frac{\alpha - 2^k}{\alpha} \leq \frac{\alpha + 1 - 2^k}{\alpha + 1}$. Furthermore, $m \leq 2^{u+1} + 2^k$. As such, $\frac{m - 2^k}{m} \leq \frac{2^{u+1}}{2^{u+1} + 2^k}$.

Let Y be the random variable which is the number of random bits extract. We have that

$$\begin{aligned} \mathbf{E}[Y] &\geq \frac{2^k}{m} k + \frac{m - 2^k}{m} \Big(\left\lfloor \lg(m - 2^k) \right\rfloor - 1 \Big) = k + \frac{m - 2^k}{m} (u - k - 1) \\ &\geq k + \frac{2^{u+1}}{2^{u+1} + 2^k} (u - k - 1) = k - \frac{2^{u+1}}{2^{u+1} + 2^k} (1 + k - u) . \end{aligned}$$

If $u = k - 1$, then $\mathbb{H}(X) \geq k - \frac{1}{2} \cdot 2 = k - 1$, as required. If $u = k - 2$ then $\mathbb{H}(X) \geq k - \frac{1}{3} \cdot 3 = k - 1$. Finally, if $u < k - 2$ then

$$\mathbf{E}[Y] \geq k - \frac{2^{u+1}}{2^k} (1 + k - u) \geq k - \frac{k - u + 1}{2^{k-u-1}} \geq k - 1,$$

since $\frac{2+i}{2^i} \leq 1$ for $i \geq 2$. ∎

Theorem 26.1.8. *Consider a coin that comes up heads with probability $p > 1/2$. For any constant $\delta > 0$ and for n sufficiently large:*

1. *One can extract, from an input of a sequence of n flips, an output sequence of $(1 - \delta) n \mathbb{H}(p)$ (unbiased) independent random bits.*
2. *One can not extract more than $n\mathbb{H}(p)$ bits from such a sequence.*

Proof: There are $\binom{n}{j}$ input sequences with exactly j heads, and each has probability $p^j (1 - p)^{n-j}$. We map this sequence to the corresponding number in the set $\left\{0, \ldots, \binom{n}{j} - 1\right\}$. Note, that this, conditional distribution on j, is uniform on this set, and we can apply the extraction algorithm of Theorem 26.1.7. Let Z be the random variables which is the number of heads in the input, and let B be the number of random bits extracted. We have

$$\mathbf{E}[B] = \sum_{k=0}^{n} \mathbf{Pr}[Z = k] \, \mathbf{E}\Big[B \;\Big|\; Z = k \Big],$$

and by Theorem 26.1.7, we have $\mathbf{E}\Big[B \;\Big|\; Z = k\Big] \geq \left\lfloor \lg \binom{n}{k} \right\rfloor - 1$. Let $\varepsilon < p - 1/2$ be a constant to be determined shortly. For $n(p-\varepsilon) \leq k \leq n(p+\varepsilon)$, we have

$$\binom{n}{k} \geq \binom{n}{\lfloor n(p+\varepsilon) \rfloor} \geq \frac{2^{n\mathbb{H}(p+\varepsilon)}}{n+1},$$

by Corollary 26.1.5 (iii). We have

$$\begin{aligned}
\mathbf{E}[B] &\geq \sum_{k=\lfloor n(p-\varepsilon)\rfloor}^{\lceil n(p-\varepsilon)\rceil} \mathbf{Pr}[Z=k]\,\mathbf{E}\Big[B \;\Big|\; Z=k\Big] \geq \sum_{k=\lfloor n(p-\varepsilon)\rfloor}^{\lceil n(p-\varepsilon)\rceil} \mathbf{Pr}[Z=k]\left(\left\lfloor \lg \binom{n}{k}\right\rfloor - 1\right) \\
&\geq \sum_{k=\lfloor n(p-\varepsilon)\rfloor}^{\lceil n(p-\varepsilon)\rceil} \mathbf{Pr}[Z=k]\left(\lg \frac{2^{n\mathbb{H}(p+\varepsilon)}}{n+1} - 2\right) \\
&= (n\mathbb{H}(p+\varepsilon) - \lg(n+1))\,\mathbf{Pr}[|Z-np| \leq \varepsilon n] \\
&\geq (n\mathbb{H}(p+\varepsilon) - \lg(n+1))\left(1 - 2\exp\left(-\frac{n\varepsilon^2}{4p}\right)\right),
\end{aligned}$$

since $\mu = \mathbf{E}[Z] = np$ and $\mathbf{Pr}\Big[|Z-np| \geq \frac{\varepsilon}{p}pn\Big] \leq 2\exp\Big(-\frac{np}{4}\Big(\frac{\varepsilon}{p}\Big)^2\Big) = 2\exp\Big(-\frac{n\varepsilon^2}{4p}\Big)$, by the Chernoff inequality. In particular, fix $\varepsilon > 0$, such that $\mathbb{H}(p+\varepsilon) > (1-\delta/4)\mathbb{H}(p)$, and since p is fixed $n\mathbb{H}(p) = \Omega(n)$, in particular, for n sufficiently large, we have $-\lg(n+1) \geq -\frac{\delta}{10}n\mathbb{H}(p)$. Also, for n sufficiently large, we have $2\exp\Big(-\frac{n\varepsilon^2}{4p}\Big) \leq \frac{\delta}{10}$. Putting it together, we have that for n large enough, we have

$$\mathbf{E}[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right)n\mathbb{H}(p)\left(1 - \frac{\delta}{10}\right) \geq (1-\delta)\,n\mathbb{H}(p),$$

as claimed.

As for the upper bound, observe that if an input sequence x has probability q, then the output sequence $y = \mathtt{Ext}(x)$ has probability to be generated which is at least q. Now, all sequences of length $|y|$ have equal probability to be generated. Thus, we have the following (trivial) inequality $2^{|\mathtt{Ext}(x)|}q \leq 2^{|\mathtt{Ext}(x)|}\,\mathbf{Pr}[y = \mathtt{Ext}(X)] \leq 1$, implying that $|\mathtt{Ext}(x)| \leq \lg(1/q)$. Thus,

$$\mathbf{E}[B] = \sum_x \mathbf{Pr}[X=x]\,|\mathtt{Ext}(x)| \leq \sum_x \mathbf{Pr}[X=x] \lg \frac{1}{\mathbf{Pr}[X=x]} = \mathbb{H}(X).$$

■

26.2 Bibliographical Notes

The presentation here follows [MU05, Sec. 9.1-Sec 9.3].

Chapter 27

Entropy II

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> The memory of my father is wrapped up in white paper, like sandwiches taken for a day at work. Just as a magician takes towers and rabbits out of his hat, he drew love from his small body, and the rivers of his hands overflowed with good deeds.
>
> – – Yehuda Amichai, My Father..

27.1 Compression

In this section, we will consider the problem of how to compress a binary string. We will map each binary string, into a new string (which is hopefully shorter). In general, by using a simple counting argument, one can show that no such mapping can achieve real compression (when the inputs are adversarial). However, the hope is that there is an underling distribution on the inputs, such that some strings are considerably more common than others.

Definition 27.1.1. A compression function `Compress` takes as input a sequence of n coin flips, given as an element of $\{H, T\}^n$, and outputs a sequence of bits such that each input sequence of n flips yields a distinct output sequence.

The following is easy to verify.

Lemma 27.1.2. *If a sequence S_1 is more likely than S_2 then the compression function that minimizes the expected number of bits in the output assigns a bit sequence to S_2 which is at least as long as S_1.*

Note, that this is very weak. Usually, we would like the function to output a prefix code, like the Huffman code.

Theorem 27.1.3. *Consider a coin that comes up heads with probability $p > 1/2$. For any constant $\delta > 0$, when n is sufficiently large, the following holds.*

(i) There exists a compression function `Compress` *such that the expected number of bits output by* `Compress` *on an input sequence of n independent coin flips (each flip gets heads with probability p) is at most $(1 + \delta)n\mathbb{H}(p)$; and*

(ii) The expected number of bits output by any compression function on an input sequence of n independent coin flips is at least $(1-\delta)n\mathbb{H}(p)$.

Proof: Let $\varepsilon > 0$ be a constant such that $p - \varepsilon > 1/2$. The first bit output by the compression procedure is '1' if the output string is just a copy of the input (using $n+1$ bits overall in the output), and '0' if it is compressed. We compress only if the number of ones in the input sequence, denoted by X is larger than $(p-\varepsilon)n$. By the Chernoff inequality, we know that $\mathbf{Pr}[X < (p-\varepsilon)n] \leq \exp\left(-n\varepsilon^2/2p\right)$.

If there are more than $(p-\varepsilon)n$ ones in the input, and since $p - \varepsilon > 1/2$, we have that

$$\sum_{j=\lceil n(p-\varepsilon)\rceil}^{n} \binom{n}{j} \leq \sum_{j=\lceil n(p-\varepsilon)\rceil}^{n} \binom{n}{\lceil n(p-\varepsilon)\rceil} \leq \frac{n}{2} 2^{n\mathbb{H}(p-\varepsilon)},$$

by Corollary 26.1.5. As such, we can assign each such input sequence a number in the range $0 \ldots \frac{n}{2} 2^{n\mathbb{H}(p-\varepsilon)}$, and this requires (with the flag bit) $1 + \lfloor \lg n + n\mathbb{H}(p-\varepsilon) \rfloor$ random bits.

Thus, the expected number of bits output is bounded by

$$(n+1)\exp\left(-n\varepsilon^2/2p\right) + (1 + \lfloor \lg n + n\mathbb{H}(p-\varepsilon)\rfloor) \leq (1+\delta)n\mathbb{H}(p)\,,$$

by carefully setting ε and n being sufficiently large. Establishing the upper bound.

As for the lower bound, observe that at least one of the sequences having exactly $\tau = \lfloor (p+\varepsilon)n \rfloor$ heads, must be compressed into a sequence having

$$\lg \binom{n}{\lfloor (p+\varepsilon)n \rfloor} - 1 \geq \lg \frac{2^{n\mathbb{H}(p+\varepsilon)}}{n+1} - 1 = n\mathbb{H}(p-\varepsilon) - \lg(n+1) - 1 = \mu,$$

by Corollary 26.1.5. Now, any input string with less than τ heads has lower probability to be generated. Indeed, for a specific strings with $\alpha < \tau$ ones the probability to generate them is $p^\alpha (1-p)^{n-\alpha}$ and $p^\tau (1-p)^{n-\tau}$, respectively. Now, observe that

$$p^\alpha (1-p)^{n-\alpha} = p^\tau (1-p)^{n-\tau} \cdot \frac{(1-p)^{\tau-\alpha}}{p^{\tau-\alpha}} = p^\tau (1-p)^{n-\tau} \left(\frac{1-p}{p}\right)^{\tau-\alpha} < p^\tau (1-p)^{n-\tau},$$

as $1 - p < 1/2 < p$ implies that $(1-p)/p < 1$.

As such, Lemma 27.1.2 implies that all the input strings with less than τ ones, must be compressed into strings of length at least μ, by an optimal compresser. Now, the Chenroff inequality implies that $\mathbf{Pr}[X \leq \tau] \geq 1 - \exp\left(-n\varepsilon^2/12p\right)$. Implying that an optimal compresser outputs on average at least $\left(1 - \exp\left(-n\varepsilon^2/12p\right)\right)\mu$. Again, by carefully choosing ε and n sufficiently large, we have that the average output length of an optimal compressor is at least $(1-\delta)n\mathbb{H}(p)$. ∎

27.2 Bibliographical Notes

The presentation here follows [MU05, Sec. 9.1-Sec 9.3].

Chapter 28

Entropy III - Shannon's Theorem

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> The memory of my father is wrapped up in
> white paper, like sandwiches taken for a day at work.
>
> Just as a magician takes towers and rabbits
> out of his hat, he drew love from his small body,
>
> and the rivers of his hands
> overflowed with good deeds.
> – – Yehuda Amichai, My Father..

28.1 Coding: Shannon's Theorem

We are interested in the problem sending messages over a noisy channel. We will assume that the channel noise is "nicely" behaved.

Definition 28.1.1. The input to a ***binary symmetric channel*** with parameter p is a sequence of bits $x_1, x_2, \ldots$, and the output is a sequence of bits $y_1, y_2, \ldots$, such that $\mathbf{Pr}[x_i = y_i] = 1-p$ independently for each i.

Translation: Every bit transmitted have the same probability to be flipped by the channel. The question is how much information can we send on the channel with this level of noise. Naturally, a channel would have some capacity constraints (say, at most 4,000 bits per second can be sent on the channel), and the question is how to send the largest amount of information, so that the receiver can recover the original information sent.

Now, its important to realize that noise handling is unavoidable in the real world. Furthermore, there are tradeoffs between channel capacity and noise levels (i.e., we might be able to send considerably more bits on the channel but the probability of flipping (i.e., p) might be much larger). In designing a communication protocol over this channel, we need to figure out where is the optimal choice as far as the amount of information sent.

Definition 28.1.2. A (k, n) ***encoding function*** $\texttt{Enc} : \{0,1\}^k \to \{0,1\}^n$ takes as input a sequence of k bits and outputs a sequence of n bits. A (k, n) ***decoding function*** $\texttt{Dec} : \{0,1\}^n \to \{0,1\}^k$ takes as input a sequence of n bits and outputs a sequence of k bits.

Thus, the sender would use the encoding function to send its message, and the decoder would use the received string (with the noise in it), to recover the sent message. Thus, the sender starts with a message with k bits, it blow it up to n bits, using the encoding function, to get some robustness to noise, it send it over the (noisy) channel to the receiver. The receiver, takes the given (noisy) message with n bits, and use the decoding function to recover the original k bits of the message.

Naturally, we would like k to be as large as possible (for a fixed n), so that we can send as much information as possible on the channel. Naturally, there might be some failure probability; that is, the receiver might be unable to recover the original string, or recover an incorrect string.

The following celebrated result of Shannon[①] in 1948 states exactly how much information can be sent on such a channel.

Theorem 28.1.3 (Shannon's theorem.). *For a binary symmetric channel with parameter $p < 1/2$ and for any constants $\delta, \gamma > 0$, where n is sufficiently large, the following holds:*

(i) For an $k \leq n(1 - \mathbb{H}(p) - \delta)$ there exists (k, n) encoding and decoding functions such that the probability the receiver fails to obtain the correct message is at most γ for every possible k-bit input messages.

(ii) There are no (k, n) encoding and decoding functions with $k \geq n(1 - \mathbb{H}(p) + \delta)$ such that the probability of decoding correctly is at least γ for a k-bit input message chosen uniformly at random.

28.2 Proof of Shannon's theorem

The proof is not hard, but requires some care, and we will break it into parts.

28.2.1 How to encode and decode efficiently

28.2.1.1 The scheme

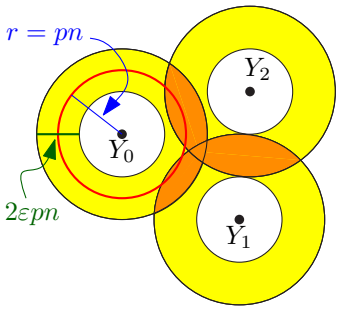
Our scheme would be simple. Pick $k \leq n(1 - \mathbb{H}(p) - \delta)$. For any number $i = 0, \ldots, \widehat{K} = 2^{k+1} - 1$, randomly generate a binary string Y_i made out of n bits, each one chosen independently and uniformly. Let $Y_0, \ldots, Y_{\widehat{K}}$ denote these codewords.

For each of these codewords we will compute the probability that if we send this codeword, the receiver would fail. Let $X_0, \ldots, X_K$, where $K = 2^k - 1$, be the K codewords with the lowest probability of failure. We assign these words to the 2^k messages we need to encode in an arbitrary fashion. Specifically, for $i = 0, \ldots, 2^k - 1$, we encode i as the string X_i.

The decoding of a message w is done by going over all the codewords, and finding all the codewords that are in (Hamming) distance in the range $[p(1 - \varepsilon)n, p(1 + \varepsilon)n]$ from w. If there is only a single word X_i with this property, we return i as the decoded word. Otherwise, if there are no such word or there is more than one word then the decoder stops and report an error.

[①] Claude Elwood Shannon (April 30, 1916 - February 24, 2001), an American electrical engineer and mathematician, has been called "the father of information theory".

28.2.1.2 The proof



Intuition. Each code Y_i corresponds to a region that looks like a ring. The "ring" for Y_i is all the strings in Hamming distance between $(1-\varepsilon)r$ and $(1+\varepsilon)r$ from Y_i, where $r = pn$. Clearly, if we transmit a string Y_i, and the receiver gets a string inside the ring of Y_i, it is natural to try to recover the received string to the original code corresponding to Y_i. Naturally, there are two possible bad events here:

(A) The received string is outside the ring of Y_i, and

(B) The received string is contained in several rings of different Ys, and it is not clear which one should the receiver decode the string to. These bad regions are depicted as the darker regions in the figure on the right.

Let $S_i = \mathcal{S}(Y_i)$ be all the binary strings (of length n) such that if the receiver gets this word, it would decipher it to be the original string assigned to Y_i (here are still using the extended set of codewords $Y_0, \ldots, Y_{\widehat{K}}$). Note, that if we remove some codewords from consideration, the set $\mathcal{S}(Y_i)$ just increases in size (i.e., the bad region in the ring of Y_i that is covered multiple times shrinks). Let W_i be the probability that Y_i was sent, but it was not deciphered correctly. Formally, let r denote the received word. We have that

$$W_i = \sum_{r \notin S_i} \mathbf{Pr}\Big[r \text{ was received when } Y_i \text{ was sent}\Big]. \tag{28.1}$$

To bound this quantity, let $\Delta(x, y)$ denote the Hamming distance between the binary strings x and y. Clearly, if x was sent the probability that y was received is

$$w(x, y) = p^{\Delta(x,y)}(1-p)^{n-\Delta(x,y)}.$$

As such, we have

$$\mathbf{Pr}\Big[r \text{ received when } Y_i \text{ was sent}\Big] = w(Y_i, r).$$

Let $\overline{S_{i,r}}$ be an indicator variable which is 1 if $r \notin S_i$. We have that

$$W_i = \sum_{r \notin S_i} \mathbf{Pr}\Big[r \text{ received when } Y_i \text{ was sent}\Big] = \sum_{r \notin S_i} w(Y_i, r) = \sum_r \overline{S_{i,r}} w(Y_i, r). \tag{28.2}$$

The value of W_i is a random variable over the choice of $Y_0, \ldots, Y_{\widehat{K}}$. As such, its natural to ask what is the expected value of W_i.

Consider the ring

$$\text{ring}(r) = \left\{ x \in \{0,1\}^n \;\middle|\; (1-\varepsilon)np \le \Delta(x, r) \le (1+\varepsilon)np \right\},$$

where $\varepsilon > 0$ is a small enough constant. Observe that $x \in \text{ring}(y)$ if and only if $y \in \text{ring}(x)$. Suppose, that the code word Y_i was sent, and r was received. The decoder returns the original code associated with Y_i, if Y_i is the only codeword that falls inside $\text{ring}(r)$.

Lemma 28.2.1. *Given that Y_i was sent, and r was received and furthermore $r \in \text{ring}(Y_i)$, then the probability of the decoder failing, is*

$$\tau = \mathbf{Pr}\Big[r \notin S_i \;\Big|\; r \in \text{ring}(Y_i)\Big] \le \frac{\gamma}{8},$$

where γ is the parameter of Theorem 28.1.3.

Proof: The decoder fails here, only if ring(r) contains some other codeword Y_j ($j \neq i$) in it. As such,

$$\tau = \mathbf{Pr}\Big[r \notin S_i \;\Big|\; r \in \mathrm{ring}(Y_i)\Big] \leq \mathbf{Pr}\Big[Y_j \in \mathrm{ring}(r)\,,\ \text{for any } j \neq i\Big] \leq \sum_{j \neq i} \mathbf{Pr}\Big[Y_j \in \mathrm{ring}(r)\Big].$$

Now, we remind the reader that the Y_js are generated by picking each bit randomly and independently, with probability $1/2$. As such, we have

$$\mathbf{Pr}\Big[Y_j \in \mathrm{ring}(r)\Big] = \frac{\big|\mathrm{ring}\,(r)\big|}{\big|\{0,1\}^n\big|} = \sum_{m=(1-\varepsilon)np}^{(1+\varepsilon)np} \frac{\binom{n}{m}}{2^n} \leq \frac{n}{2^n}\binom{n}{\lfloor (1+\varepsilon)np \rfloor},$$

since $(1+\varepsilon)p < 1/2$ (for ε sufficiently small), and as such the last binomial coefficient in this summation is the largest. By Corollary 26.1.5 (i), we have

$$\mathbf{Pr}\Big[Y_j \in \mathrm{ring}(r)\Big] \leq \frac{n}{2^n}\binom{n}{\lfloor (1+\varepsilon)np \rfloor} \leq \frac{n}{2^n} 2^{n\mathbb{H}\left((1+\varepsilon)p\right)} = n2^{n\left(\mathbb{H}\left((1+\varepsilon)p\right)-1\right)}.$$

As such, we have

$$\begin{aligned}\tau = \mathbf{Pr}\Big[r \notin S_i \;\Big|\; r \in \mathrm{ring}(Y_i)\Big] &\leq \sum_{j \neq i} \mathbf{Pr}\Big[Y_j \in \mathrm{ring}(r)\Big] \leq \widehat{K}\,\mathbf{Pr}[Y_1 \in \mathrm{ring}(r)] \leq 2^{k+1} n 2^{n\left(\mathbb{H}\left((1+\varepsilon)p\right)-1\right)} \\ &\leq n2^{n\left(1-\mathbb{H}(p)-\delta\right)+1+n\left(\mathbb{H}\left((1+\varepsilon)p\right)-1\right)} \leq n2^{n\left(\mathbb{H}\left((1+\varepsilon)p\right)-\mathbb{H}(p)-\delta\right)+1}\end{aligned}$$

since $k \leq n(1-\mathbb{H}(p)-\delta)$. Now, we choose ε to be a small enough constant, so that the quantity $\mathbb{H}\Big((1+\varepsilon)p\Big) - \mathbb{H}\Big(p\Big) - \delta$ is equal to some (absolute) negative (constant), say $-\beta$, where $\beta > 0$. Then, $\tau \leq n2^{-\beta n+1}$, and choosing n large enough, we can make τ smaller than $\gamma/8$, as desired. As such, we just proved that

$$\tau = \mathbf{Pr}\Big[r \notin S_i \;\Big|\; r \in \mathrm{ring}(Y_i)\Big] \leq \frac{\gamma}{8}.$$

■

Lemma 28.2.2. *Consider the situation where Y_i is sent, and the received string is r. We have that*

$$\mathbf{Pr}[r \notin \mathrm{ring}(Y_i)] = \sum_{r \notin \mathrm{ring}(Y_i)} w(Y_i, r) \leq \frac{\gamma}{8},$$

where γ is the parameter of Theorem 28.1.3.

Proof: This quantity, is the probability of sending Y_i when every bit is flipped with probability p, and receiving a string r such that more than $pn + \varepsilon pn$ bits where flipped (or less than $pn - \varepsilon pn$). But this quantity can be bounded using the Chernoff inequality. Indeed, let $Z = \Delta(Y_i, r)$, and observe that $\mathbf{E}[Z] = pn$, and it is the sum of n independent indicator variables. As such

$$\sum_{r \notin \mathrm{ring}(Y_i)} w(Y_i, r) = \mathbf{Pr}\Big[\,|Z - \mathbf{E}[Z]| > \varepsilon pn\Big] \leq 2\exp\left(-\frac{\varepsilon^2}{4} pn\right) < \frac{\gamma}{4},$$

since ε is a constant, and for n sufficiently large. ■

Lemma 28.2.3. *We have that* $f(Y_i) = \sum_{r \notin \text{ring}(Y_i)} \mathbf{E}\left[\overline{S_{i,r}} w(Y_i, r)\right] \leq \gamma/8$ *(the expectation is over all the choices of the Ys excluding* Y_i*).*

Proof: Observe that $\overline{S_{i,r}} w(Y_i, r) \leq w(Y_i, r)$ and for fixed Y_i and r we have that $\mathbf{E}[w(Y_i, r)] = w(Y_i, r)$. As such, we have that

$$f(Y_i) = \sum_{r \notin \text{ring}(Y_i)} \mathbf{E}\left[\overline{S_{i,r}} w(Y_i, r)\right] \leq \sum_{r \notin \text{ring}(Y_i)} \mathbf{E}[w(Y_i, r)] = \sum_{r \notin \text{ring}(Y_i)} w(Y_i, r) \leq \frac{\gamma}{8},$$

by Lemma 28.2.2. ∎

Lemma 28.2.4. *We have that* $g(Y_i) = \sum_{r \in \text{ring}(Y_i)} \mathbf{E}\left[\overline{S_{i,r}} w(Y_i, r)\right] \leq \gamma/8$ *(the expectation is over all the choices of the Ys excluding* Y_i*).*

Proof: We have that $\overline{S_{i,r}} w(Y_i, r) \leq \overline{S_{i,r}}$, as $0 \leq w(Y_i, r) \leq 1$. As such, we have that

$$\begin{aligned}
g(Y_i) &= \sum_{r \in \text{ring}(Y_i)} \mathbf{E}\left[\overline{S_{i,r}} w(Y_i, r)\right] \leq \sum_{r \in \text{ring}(Y_i)} \mathbf{E}\left[\overline{S_{i,r}}\right] = \sum_{r \in \text{ring}(Y_i)} \mathbf{Pr}[r \notin S_i] \\
&= \sum_r \mathbf{Pr}[r \notin S_i \cap (r \in \text{ring}(Y_i))] \\
&= \sum_r \mathbf{Pr}\left[r \notin S_i \;\middle|\; r \in \text{ring}(Y_i)\right] \mathbf{Pr}[r \in \text{ring}(Y_i)] \\
&\leq \sum_r \frac{\gamma}{8} \mathbf{Pr}[r \in \text{ring}(Y_i)] \leq \frac{\gamma}{8},
\end{aligned}$$

by Lemma 28.2.1. ∎

Lemma 28.2.5. *For any i, we have* $\mu = \mathbf{E}[W_i] \leq \gamma/4$*, where* γ *is the parameter of Theorem 28.1.3, where* W_i *is the probability of failure to recover* Y_i *if it was sent, see Eq. (28.1).*

Proof: We have by Eq. (28.2) that $W_i = \sum_r \overline{S_{i,r}} w(Y_i, r)$. For a fixed value of Y_i, we have by linearity of expectation, that

$$\begin{aligned}
\mathbf{E}\left[W_i \;\middle|\; Y_i\right] &= \mathbf{E}\left[\sum_r \overline{S_{i,r}} w(Y_i, r) \;\middle|\; Y_i\right] = \sum_r \mathbf{E}\left[\overline{S_{i,r}} w(Y_i, r) \;\middle|\; Y_i\right] \\
&= \sum_{r \in \text{ring}(Y_i)} \mathbf{E}\left[\overline{S_{i,r}} w(Y_i, r) \;\middle|\; Y_i\right] + \sum_{r \notin \text{ring}(Y_i)} \mathbf{E}\left[\overline{S_{i,r}} w(Y_i, r) \;\middle|\; Y_i\right] = g(Y_i) + f(Y_i) \leq \frac{\gamma}{8} + \frac{\gamma}{8} = \frac{\gamma}{4},
\end{aligned}$$

by Lemma 28.2.3 and Lemma 28.2.4. Now $\mathbf{E}[W_i] = \mathbf{E}\left[\mathbf{E}\left[W_i \;\middle|\; Y_i\right]\right] \leq \mathbf{E}[\gamma/4] \leq \gamma/4$. ∎

In the following, we need the following trivial (but surprisingly deep) observation.

Observation 28.2.6. *For a random variable X, if* $\mathbf{E}[X] \leq \psi$*, then there exists an event in the probability space, that assigns X a value* $\leq \psi$*.*

Lemma 28.2.7. *For the codewords $X_0, \ldots, X_K$, the probability of failure in recovering them when sending them over the noisy channel is at most γ.*

Proof: We just proved that when using $Y_0, \ldots, Y_{\widehat{K}}$, the expected probability of failure when sending Y_i, is $\mathbf{E}[W_i] \leq \gamma/4$, where $\widehat{K} = 2^{k+1} - 1$. As such, the expected total probability of failure is

$$\mathbf{E}\left[\sum_{i=0}^{\widehat{K}} W_i\right] = \sum_{i=0}^{\widehat{K}} \mathbf{E}[W_i] \leq \frac{\gamma}{4} 2^{k+1} \leq \gamma 2^k,$$

by Lemma 28.2.5. As such, by Observation 28.2.6, there exist a choice of Y_is, such that

$$\sum_{i=0}^{\widehat{K}} W_i \leq 2^k \gamma.$$

Now, we use a similar argument used in proving Markov's inequality. Indeed, the W_i are always positive, and it can not be that 2^k of them have value larger than γ, because in the summation, we will get that

$$\sum_{i=0}^{\widehat{K}} W_i > 2^k \gamma.$$

Which is a contradiction. As such, there are 2^k codewords with failure probability smaller than γ. We set the 2^k codewords $X_0, \ldots, X_K$ to be these words, where $K = 2^k - 1$. Since we picked only a subset of the codewords for our code, the probability of failure for each codeword shrinks, and is at most γ. ■

Lemma 28.2.7 concludes the proof of the constructive part of Shannon's theorem.

28.2.2 Lower bound on the message size

We omit the proof of this part. It follows similar argumentation showing that for every ring associated with a codewords it must be that most of it is covered only by this ring (otherwise, there is no hope for recovery). Then an easy packing argument implies the claim.

28.3 Bibliographical Notes

The presentation here follows [MU05, Sec. 9.1-Sec 9.3].

Chapter 29

Low Dimensional Linear Programming

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
May 29, 2013

> "Napoleon has not been conquered by man. He was greater than all of us. But god punished him because he relied on his own intelligence alone, until that prodigious instrument was strained to breaking point. Everything breaks in the end."
>
> – Carl XIV Johan, King of Sweden.

29.1 Linear programming in constant dimension ($d > 2$)

Let assume that we have a set H of n linear inequalities defined over d (d is a small constant) variables. Every inequality in H defines a closed half space in $\mathbb{R}^d$. Given a vector $\overrightarrow{c} = (c_1, \ldots, c_d)$ we want to find $p = (p_1, \ldots, p_d) \in \mathbb{R}^d$ which is in all the half spaces $h \in H$ and $f(p) = \sum_i c_i p_i$ is maximized. Formally:

LP in d dimensions:$(H, \overrightarrow{c})$ H - set of n closed half spaces in $\mathbb{R}^d$ $\overrightarrow{c}$ - vector in d dimensions Find $p \in \mathbb{R}^d$ s.t. $\forall h \in H$ we have $p \in h$ and $f(p)$ is maximized. Where $f(p) = \left\langle p, \overrightarrow{c} \right\rangle$.

A closed half space in d dimensions is defined by an inequality of the form

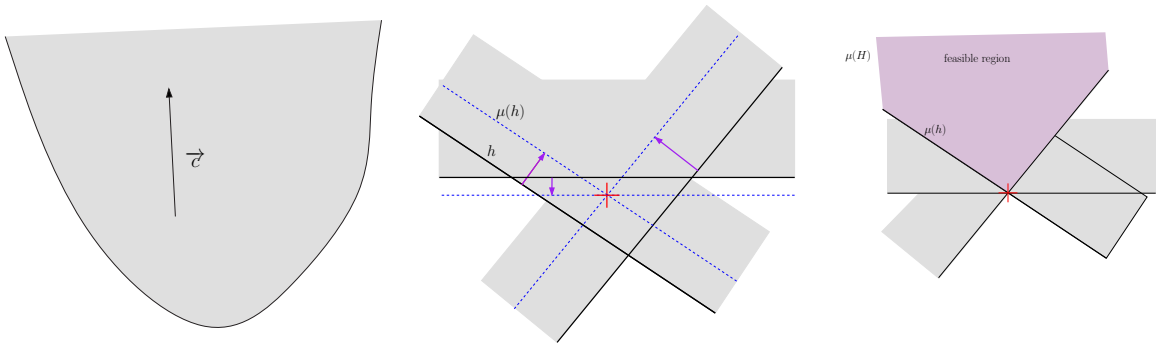
$$a_1 x_1 + a_2 x_2 + \cdots + a_n x_n \leq b_n.$$

One difficulty that we ignored earlier, is that the optimal solution for the LP might be unbounded, see Figure 29.1.

Namely, we can find a solution with value ∞ to the target function.

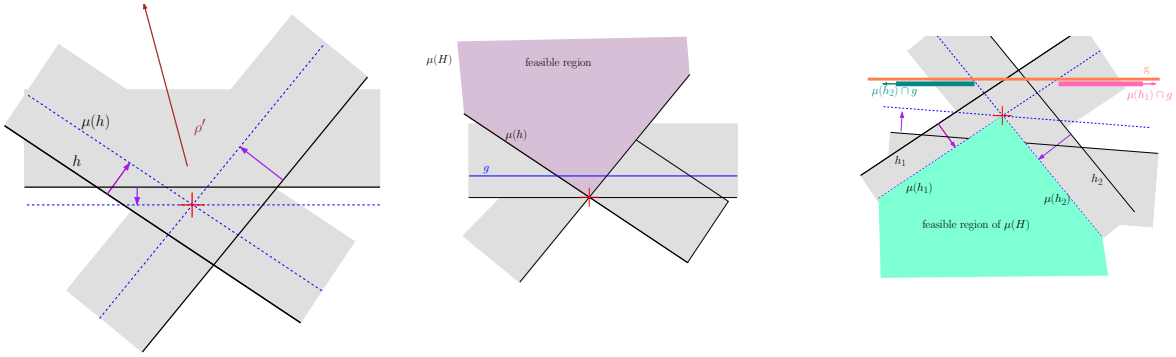
For a half space h let $\eta(h)$ denote the normal of h directed into the feasible region. Let $\mu(h)$ denote the closed half space, resulting from h by translating it so that it passes through the origin. Let $\mu(H)$ be the resulting set of half spaces from H. See Figure 29.1 (b).

The new set of constraints $\mu(H)$ is depicted in Figure 29.1 (c).



(a) (b) (c)

Figure 29.1: (a) Unbounded LP. (b). (c).



(a) (b) (c)

Figure 29.2: (a). (b). (c).

Lemma 29.1.1. *$(H, \overrightarrow{c})$ is unbounded if and only if $(\mu(H), \overrightarrow{c})$ is unbounded.*

Proof: Consider the ρ' the unbounded ray in the feasible region of $(H, \overrightarrow{c})$ such that the line that contain it passes through the origin. Clearly, ρ' is unbounded also in $(H, \overrightarrow{c})$, and this is if and only if. See Figure 29.2 (a). ■

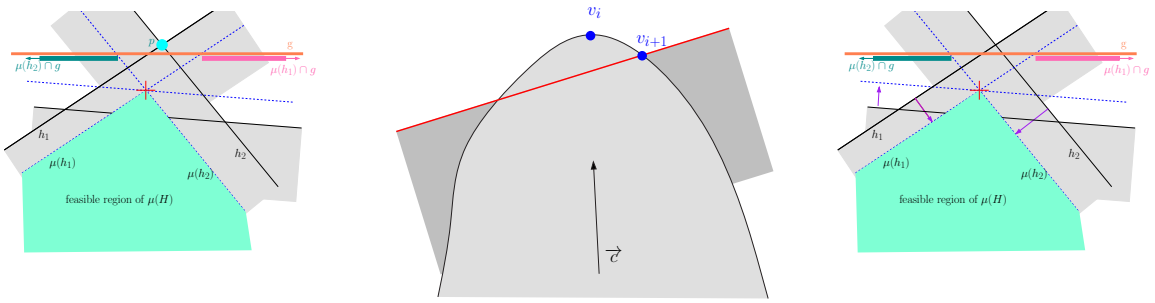
Lemma 29.1.2. *Deciding if $(\mu(H), \overrightarrow{c})$ is bounded can be done by solving a $d-1$ dimensional LP. Furthermore, if it is bounded, then we have a set of d constraints, such that their intersection prove this.*

Furthermore, the corresponding set of d constraints in H testify that $(H, \overrightarrow{c})$ is bounded.

Proof: Rotate space, such that $\overrightarrow{c}$ is the vector $(0, 0, \ldots, 0, 1)$. And consider the hyperplane $g \equiv x_d = 1$. Clearly, $(\mu(H), \overrightarrow{c})$ is unbounded if and only if the region $g \cap \bigcap_{h \in \mu(H)} h$ is non-empty. By deciding if this region is unbounded, is equivalent to solving the following LP: $L' = (H', (1, 0, \ldots, 0))$ where

$$H' = \left\{ g \cap h \;\middle|\; h \in \mu(H) \right\}.$$

Let $h \equiv a_1 x_1 + \ldots + a_d x_d \leq 0$, the region corresponding to $g \cap h$ is $a_1 x_1 + \cdots + a_{d-1} x_{d-1} \leq -a_d$ which is a $d-1$ dimensional hyperplane. See Figure 29.2 (b).



(a) (b) (c)

Figure 29.3: (a). (b). (c).

But this is a $d-1$ dimensional LP, because everything happens on the hyperplane $x_d = 1$.

Notice that if $(\mu(H), \overrightarrow{c})$ is bounded (which happens if and only if $(H, \overrightarrow{c})$ is bounded), then L' is infeasible, and the LP L' would return us a set d constraints that their intersection is empty. Interpreting those constraints in the original LP, results in a set of constraints that their intersection is bounded in the direction of $\overrightarrow{c}$. See Figure 29.2 (c).

(In the above example, $\mu(H) \cap g$ is infeasible because the intersection of $\mu(h_2) \cap g$ and $\mu(h_1) \cap g$ is empty, which implies that $h_1 \cap h_2$ is bounded in the direction $\overrightarrow{c}$ which we care about. The positive y direction in this figure.) ■

We are now ready to show the algorithm for the LP for $L = (H, \overrightarrow{c})$. By solving a $d-1$ dimensional LP we decide whether L is unbounded. If it is unbounded, we are done (we also found the unbounded solution, if you go carefully through the details).

See Figure 29.3 (a).

(in the above figure, we computed p.)

In fact, we just computed a set $h_1, \ldots, h_d$ s.t. their intersection is bounded in the direction of $\overrightarrow{c}$ (thats what the boundness check returned).

Let us randomly permute the remaining half spaces of H, and let $h_1, h_2, \ldots, h_d, h_{d+1}, \ldots, h_n$ be the resulting permutation.

Let v_i be the vertex realizing the optimal solution for the LP:

$$L_i = \left(\{h_1, \ldots, h_i\}, \overrightarrow{c}\right)$$

There are two possibilities:

1. $v_i = v_{i+1}$. This means that $v_i \in h_{i+1}$ and it can be checked in constant time.
2. $v_i \neq v_{i+1}$. It must be that $v_i \notin h_{i+1}$ but then, we must have... What is depicted in Figure 29.3 (b).

B - the set of d constraints that define v_{i+1}. If $h_{i+1} \notin B$ then $v_i = v_{i+1}$. As such, the probability of $v_i \neq v_{i+1}$ is roughly d/i because this is the probability that one of the elements of B is h_{i+1}. Indeed, fix the first $i+1$ elements, and observe that there are d elements that are marked (those are the

elements of B). Thus, we are asking what is the probability of one of d marked elements to be the last one in a random permutation of $h_{d+1}, \ldots, h_{i+1}$, which is exactly $d/(i+1-d)$.

Note that if some of the elements of B is $h_1, \ldots, h_d$ than the above expression just decreases (as there are less marked elements).

Well, let us restrict our attention to ∂h_{i+1}. Clearly, the optimal solution to L_{i+1} on h_{i+1} is the required v_{i+1}. Namely, we solve the LP $L_{i+1} \cap h_{i+1}$ using recursion.

This takes $T(i+1, d-1)$ time. What is the probability that $v_{i+1} \neq v_i$?

Well, one of the d constraints defining v_{i+1} has to be h_{i+1}.The probability for that is ≤ 1 for $i \leq 2d-1$, and it is

$$\leq \frac{d}{i+1-d},$$

otherwise.

Summarizing everything, we have:

$$\begin{aligned} T(n,d) &= O(n) + T(n,d-1) + \sum_{i=d+1}^{2d} T(i,d-1) \\ &+ \sum_{i=2d+1}^{n} \frac{d}{i+1-d} T(i,d-1) \end{aligned}$$

What is the solution of this monster? Well, one essentially to guess the solution and verify it. To guess solution, let us "simplify" (incorrectly) the recursion to :

$$T(n,d) = O(n) + T(n,d-1) + d \sum_{i=2d+1}^{n} \frac{T(i,d-1)}{i+1-d}$$

So think about the recursion tree. Now, every element in the sum is going to contribute a near constant factor, because we divide it by (roughly) $i+1-d$ and also, we are guessing the the optimal solution is linear/near linear.

In every level of the recursion we are going to penalized by a multiplicative factor of d. Thus, it is natural, to conjecture that $T(n,d) \leq (3d)^{3d} n$.

Which can be verified by tedious substitution into the recurrence, and is left as exercise.

Theorem 29.1.3. *Given an d dimensional LP $(H, \overrightarrow{c})$,it can be solved in expected $O\Big((3d)^{3d} n\Big)$ time (the constant in the O is dim independent).*

BTW, we are being a bit conservative about the constant. In fact, one can prove that the running time is $d!n$. Which is still exponential in d.

```
SolveLP((H, c⃗))
    /* initialization */
    Rotate (H, c⃗) s.t. c⃗ = (0, ..., 1)
    Solve recursively the d − 1 dim LP:
                L′ ≡ μ(H) ∩ (x_d = 1)
    if L′ has a solution then
            return "Unbounded"

    Let g_1, ..., g_d be the set of constraints of L′ that testifies that L′ is infeasible
    Let h_1, ..., h_d be the hyperplanes of H corresponding to g_1, ..., g_d
    Permute H s.t. h_1, ..., h_d are first.
    v_d = ∂h_1 ∩ ∂h_2 ∩ ··· ∩ ∂h_d
    /*v_d is a vertex that testifies that (H, c⃗) is bounded */

    /* the algorithm itself */
    for i ← d + 1 to n do
        if v_{i−1} ∈ h_i then
            v_i ← v_{i−1}
        else
            v_i ← SolveLP((H_{i−1} ∩ ∂h_i, c⃗))      (*)
              where H_{i−1} = {h_1, ..., h_{i−1}}

    return v_n
```

29.2 Handling Infeasible Linear Programs

In the above discussion, we glossed over the question of how to handle LPs which are infeasible. This requires slightly modifying our algorithm to handle this case, and I am only describing the required modifications.

First, the simplest case, where we are given an LP L which is one dimensional (i.e., defined over one variable). Clearly, we can solve this LP in linear time (verify!), and furthermore, if there is no solution, we can return two input inequality $ax \leq b$ and $cx \geq d$ for which there is no solution together (i.e., those two inequalities [i.e., constraints] testifies that the LP is not satisfiable).

Next, assume that the algorithm `SolveLP` when called on a $d-1$ dimensional LP L', if L' is not feasible it return the d constraints of L' that together have non-empty intersection. Namely, those constraints are the witnesses that L' is infeasible.

So the only place, where we can get such answer, is when computing v_i (in the (*) line in the algorithm). Let $h'_1, \ldots, h'_d$ be the corresponding set of d constraints of H_{i-1} that testifies that $(H_{i-1} \cap \partial h_i, \overrightarrow{c})$ is an infeasible LP. Clearly, $h'_1, \ldots, h'_d, h_i$ must be a set of $d+1$ constraints that are together are infeasible, and that is what `SolveLP` returns.

29.3 References

The description in this class notes is loosely based on the description of low dimensional LP in the book of de Berg *et al.* [dBCvKO08].

Bibliography

[AB99] M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge, 1999.

[ABKU00] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal. Balanced allocations. *SIAM J. Comput.*, 29(1):180–200, 2000.

[Ach01] D. Achlioptas. Database-friendly random projections. In *Proc. 20th ACM Sympos. Principles Database Syst.*, pages 274–281, 2001.

[AHY07] P. Agarwal, S. Har-Peled, and H. Yu. Embeddings of surfaces, curves, and moving points in Euclidean space. In *Proc. 23rd Annu. ACM Sympos. Comput. Geom.*, pages 381–389, 2007.

[AKPW95] N. Alon, R. M. Karp, D. Peleg, and D. West. A graph-theoretic game and its application to the k-server problem. *SIAM J. Comput.*, 24(1):78–100, February 1995.

[AN04] N. Alon and A. Naor. Approximating the cut-norm via grothendieck's inequality. In *Proc. 36th Annu. ACM Sympos. Theory Comput.*, pages 72–80, 2004.

[AR94] N. Alon and Y. Roichman. Random cayley graphs and expanders. *Random Struct. Algorithms*, 5(2):271–285, 1994.

[Aro98] S. Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. *J. Assoc. Comput. Mach.*, 45(5):753–782, Sept. 1998.

[AS00] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley InterScience, 2nd edition, 2000.

[ASS08] N. Alon, O. Schwartz, and A. Shapira. An elementary construction of constant-degree expanders. *Combin. Probab. Comput.*, 17(3):319–327, 2008.

[Bar96] Y. Bartal. Probabilistic approximations of metric space and its algorithmic application. In *Proc. 37th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 183–193, October 1996.

[Bar98] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *Proc. 30th Annu. ACM Sympos. Theory Comput.*, pages 161–168, 1998.

[BM58] G. E.P. Box and M. E. Muller. A note on the generation of random normal deviates. *Ann. Math. Stat.*, 28:610–611, 1958.

[Bol98] B. Bollobas. *Modern Graph Theory*. Springer-Verlag, 1998.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge, 2004.

[Car76] L. Carroll. The hunting of the snark, 1876.

[Cha01] B. Chazelle. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, New York, 2001.

[CKR01] G. Calinescu, H. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension problem. In *Proc. 12th ACM-SIAM Sympos. Discrete Algs.*, pages 8–16, 2001.

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press / McGraw-Hill, 2001.

[dBCvKO08] M. de Berg, O. Cheong, M. van Kreveld, and M. H. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 3rd edition, 2008.

[DG03] S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Rand. Struct. Alg.*, 22(3):60–65, 2003.

[FRT03] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proc. 35th Annu. ACM Sympos. Theory Comput.*, pages 448–455, 2003.

[Gar02] R. J. Gardner. The Brunn-Minkowski inequality. *Bull. Amer. Math. Soc.*, 39:355–405, 2002.

[GLS93] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin Heidelberg, 2nd edition, 1993.

[GRSS95] M. Golin, R. Raman, C. Schwarz, and M. Smid. Simple randomized algorithms for closest pair problems. *Nordic J. Comput.*, 2:3–27, 1995.

[Gup00] A. Gupta. *Embeddings of Finite Metrics*. PhD thesis, University of California, Berkeley, 2000.

[GW95] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, 42(6):1115–1145, November 1995.

[Hås01] J. Håstad. Some optimal inapproximability results. *J. Assoc. Comput. Mach.*, 48(4):798–859, 2001.

[HLW06] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bulletin Amer. Math. Soc.*, 43:439–561, 2006.

[HW87] D. Haussler and E. Welzl. ε-nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.

[IM98] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. 30th Annu. ACM Sympos. Theory Comput.*, pages 604–613, 1998.

[Ind01] P. Indyk. Algorithmic applications of low-distortion geometric embeddings. In *Proc. 42nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 10–31, 2001. Tutorial.

[KKMO04] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for max cut and other 2-variable csps. In *Proc. 45th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 146–154, 2004. To appear in SICOMP.

[KLMN05] R. Krauthgamer, J. R. Lee, M. Mendel, and A. Naor. Measured descent: A new embedding method for finite metric spaces. *Geom. funct. anal. (GAFA)*, 15(4):839–858, 2005.

[Mag07] A. Magen. Dimensionality reductions in ℓ_2 that preserve volumes and distance to affine spaces. *Discrete Comput. Geom.*, 38(1):139–153, 2007.

[Mat90] J. Matoušek. Bi-Lipschitz embeddings into low-dimensional Euclidean spaces. *Comment. Math. Univ. Carolinae*, 31:589–600, 1990.

[Mat99] J. Matoušek. *Geometric Discrepancy*. Springer, 1999.

[Mat02] J. Matoušek. *Lectures on Discrete Geometry*. Springer, 2002.

[MN98] J. Matoušek and J. Nešetřil. *Invitation to Discrete Mathematics*. Oxford Univ Press, 1998.

[MN08] M. Mendel and A. Naor. Towards a calculus for non-linear spectral gaps. manuscript, 2008.

[MOO05] E. Mossel, R. O'Donnell, and K. Oleszkiewicz. Noise stability of functions with low influences invariance and optimality. In *Proc. 46th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 21–30, 2005.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[MU05] M. Mitzenmacher and U. Upfal. *Probability and Computing – randomized algorithms and probabilistic analysis*. Cambridge, 2005.

[Nor98] J. R. Norris. *Markov Chains*. Statistical and Probabilistic Mathematics. Cambridge Press, 1998.

[Rab76] M. O. Rabin. Probabilistic algorithms. In J. F. Traub, editor, *Algorithms and Complexity: New Directions and Recent Results*, pages 21–39. Academic Press, 1976.

[RVW02] O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. *Annals Math.*, 155(1):157–187, 2002.

[Smi00] M. Smid. Closest-point problems in computational geometry. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 877–935. Elsevier, 2000.

[Tót03] C. D. Tóth. A note on binary plane partitions. *Discrete Comput. Geom.*, 30(1):3–16, 2003.

[VC71] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.*, 16:264–280, 1971.

[Wes01] D. B. West. *Intorudction to Graph Theory*. Prentice Hall, 2ed edition, 2001.

[WG75] H. W. Watson and F. Galton. On the probability of the extinction of families. *J. Anthrop. Inst. Great Britain*, 4:138–144, 1875.

Index