Class notes for Randomized Algorithms

Sariel Har-Peled[①]

December 1, 2005

[②]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

Min Cut

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

> 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.

1.1 Min Cut

1.1.1 Problem Definition

Let $G = (V, E)$ be undirected graph with n vertices, and m edges. We are interested in the notion of a cut in a graph.

Definition 1.1.1 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*.

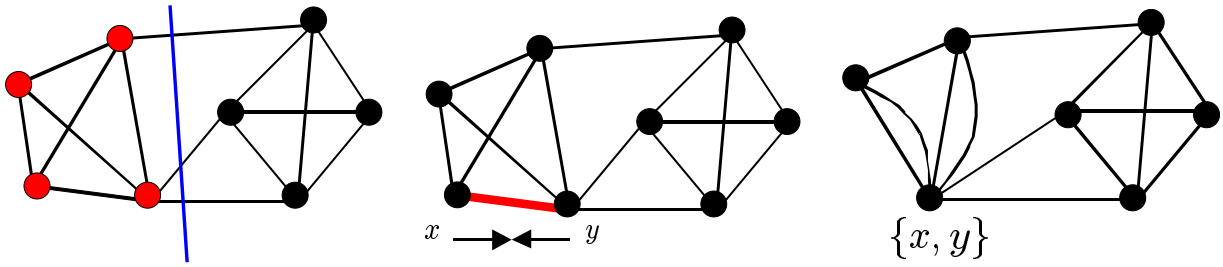
For an example of a cut, see Figure 1.1.

We are interested in the problem of computing the *minimum cut*, that is, the cut in the graph with minimum cardinality. Compute the cut with minimum number of edges in the graph. Namely, find $S \subseteq V$ such that $(S, V \setminus S)$ is as small as possible, and S is neither empty nor is $V \setminus S$.

1.1.2 Some Definitions

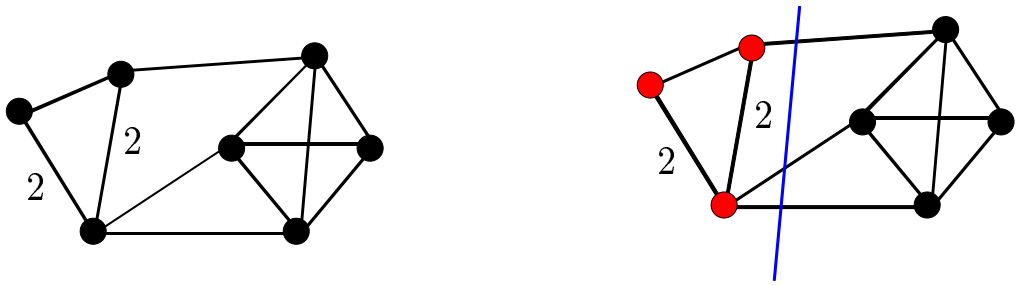
Definition 1.1.2 The conditional probability of X given Y is

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



(a) (b) (c)

Figure 1.1: (a) A cut in the graph. (b) A contraction of an edge. (c) The resulting graph.



(a) (b)

Figure 1.2: (a) A multi-graph. (b) A minimum cut in the resulting multi-graph.

An equivalent, useful statement of this is that

$$\mathbf{Pr}[(X = x) \cap (Y = y)] = \mathbf{Pr}[X = x \,|Y = y] * \mathbf{Pr}[Y = y]\,.$$

Definition 1.1.3 Two events X and Y are *independent*, if $\mathbf{Pr}[X = x \cap Y = y] = \mathbf{Pr}[X = x] \cdot \mathbf{Pr}[Y = y]$. In particular, if X and Y are independent, then

$$\mathbf{Pr}\Big[X = x \;\Big|\; Y = y\Big] = \mathbf{Pr}[X = x]\,.$$

The following is easy to prove by induction.

Lemma 1.1.4 *Let $\eta_1, \ldots, \eta_n$ be n events which are not necessarily independent. Then,*

$$\begin{aligned}\mathbf{Pr}[\cap_{i=1}^{n} \eta_i] \;=\; & \mathbf{Pr}[\eta_1] * \mathbf{Pr}[\eta_2 \,|\eta_1\,] * \\ & \mathbf{Pr}[\eta_3 \,|\eta_1 \cap \eta_2\,] * \ldots * \mathbf{Pr}\Big[\eta_n \;\Big|\; \eta_1 \cap \ldots \cap \eta_{n-1}\Big]\end{aligned}$$

1.2 The Algorithm

The basic operation used by the algorithm is edge contraction, depicted in Figure 1.1. We take an edge $e = xy$ and merge the two vertices into a single vertex. What we get is depicted in Figure 1.1 (c). The new graph is denoted by G/xy. Note, that we remove self loops. However, 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 1.2.

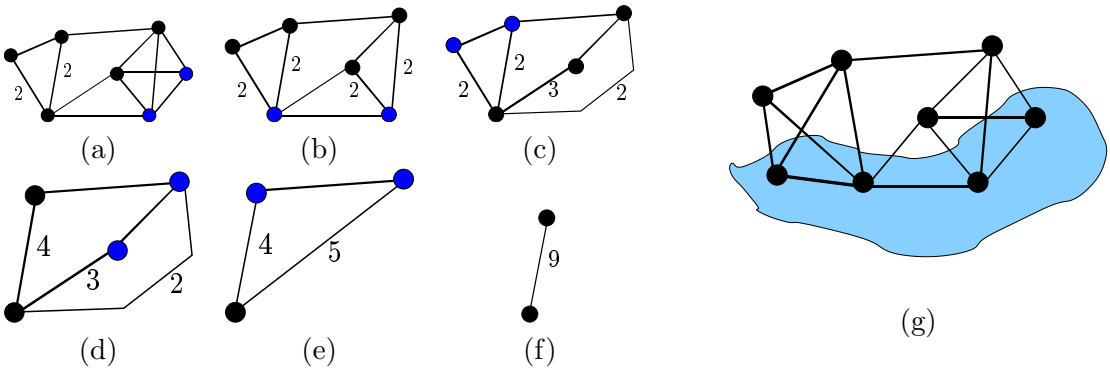


Figure 1.3: (a)-(f) a sequence of contractions in the graph, and (g) the cut in the original graph, corresponding to the single edge in (f).

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 an edge is in the cut, and it has weight w we add w to the weight of the cut.

Observation 1.2.1 *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.*

So, the main idea of our algorithm is to repeatedly perform contraction, which is beneficial since it shrinks the graph. And we would like to compute the cut in the resulting (smaller) graph. An "extreme" example of this, is shown in Figure 1.3, 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 1.3 also demonstrate the problem with taking this approach. Indeed, the resulting cut is not the minimum cut in the graph. So, why we did not find the minimum cut?

Observation 1.2.2 *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 $G' = G/\{e_1, \ldots, e_{n-2}\}$ is a single multi-edge. Then, this multi-edge correspond to the minimum cut in G.*

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.

Using Observation 1.2.2 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 cut the Gordian know here, is to randomly contract an edge.

Lemma 1.2.3 *If a graph G has a minimum cut of size k, and it 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. ■

Algorithm MinCut(G)
  $G_0 \leftarrow G$
  $i = 0$
  while G_i has more than two vertices **do**
    Pick randomly an edge e_i from the edges of G_i
    $G_{i+1} \leftarrow G_i / e_i$
    $i \leftarrow i + 1$
  Let $(S, V - S)$ be the cut in the original graph
    corresponding to the single edge in G_i

Figure 1.4: The minimum cut algorithm.

Lemma 1.2.4 *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 resulting algorithm is depicted in Figure 1.4.

Observation 1.2.5 MinCut *runs in $O(n^2)$ time.*

Observation 1.2.6 *The algorithm always outputs a cut, and the cut is not smaller than the minimum cut.*

Lemma 1.2.7 *MinCut outputs the min cut in probability $\geq \dfrac{2}{n(n-1)}$.*

Proof: Let η_i be the event that e_i is not in the minimum cut of G_i. By Observation 1.2.2, MinCut outputs the minimum cut if the events $\eta_0, \ldots, \eta_{n-3}$ all happen (namely, all edges picked are outside the minimum cut).

By Lemma 1.2.4, it holds $\mathbf{Pr}[\eta_i \,|\eta_1 \cap \ldots \cap \eta_{i-1}] \geq 1 - \dfrac{2}{|V(G_i)|} = 1 - \dfrac{2}{n-i}$. Implying that

$$\begin{aligned}
\mathbf{Pr}[\eta_0 \cap \ldots \cap \eta_{n-2}] &= \mathbf{Pr}[\eta_0] \cdot \mathbf{Pr}[\eta_1 \,|\eta_0] \cdot \mathbf{Pr}[\eta_2 \,|\eta_0 \cap \eta_1] \\
&\cdot \quad \ldots \quad \cdot \mathbf{Pr}[\eta_{n-3} \,|\eta_0 \cap \ldots \cap \eta_{n-4}] \\
&\geq \prod_{i=0}^{n-3} \left(1 - \frac{2}{n-i}\right) = \prod_{i=0}^{n-3} \frac{n-i-2}{n-i} \\
&= \frac{n-2}{n} * \frac{n-3}{n-1} * \frac{n-4}{n-2} \cdot \ldots \cdot \frac{2}{4} \cdot \frac{1}{3} \\
&= \frac{2}{n \cdot (n-1)}.
\end{aligned}$$

∎

Definition 1.2.8 (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 1.2.9 *The probability that* MinCutRep *fails to return the minimum cut is < 0.14.*

Proof: The probability of failure 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 1.2.10 *One can compute the minimum cut in $O(n^4)$ time with constant probability to get a correct result. In $O\big(n^4 \log n\big)$ time the minimum cut is returned with high probability.*

1.3 A faster algorithm

The algorithm presented in the previous section is extremely simple. Which raises the question of whether we can complicate things, and get a faster algorithm?

So, why is the algorithm needs so many executions? Well, the probability of success in the first l iterations, is

$$\begin{aligned}\mathbf{Pr}[\eta_0 \cap \ldots \cap \eta_{l-1}] \geq& \prod_{i=0}^{l-1}\left(1-\frac{2}{n-i}\right) = \prod_{i=0}^{l-1}\frac{n-i-2}{n-i} \\ =& \frac{n-2}{n} * \frac{n-3}{n-1} * \frac{n-4}{n-2}\cdots = \frac{(n-l)(n-l-1)}{n\cdot(n-1)}. \end{aligned} \tag{1.1}$$

Namely, this probability deteriorates very quickly toward the end of the execution, when the graph become small enough.

Observation 1.3.1 *As the graph get smaller, the probability to make a bad choice increases. So, run the algorithm more times when the graph is smaller.*

The basic new operation we use is Contract, depicted in Figure 1.5, which also depict the new algorithm FastCut.

Lemma 1.3.2 *The running time of* FastCut(G) *is $O\big(n^2 \log n\big)$, where $n = |V(G)|$.*

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:

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

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

Exercise 1.3.3 Show that one can modify FastCut so that it uses only $O(n^2)$ space.

Lemma 1.3.4 *The probability that $Contract(G, n/\sqrt{2})$ had NOT contracted the minimum cut is at least $1/2$.*

Proof: Just plug in $l = n - t = n - \left\lceil 1 + n/\sqrt{2} \right\rceil$ into Eq. (1.1). We have

$$\mathbf{Pr}[\eta_0 \cap \ldots \cap \eta_{n-t}] \geq \frac{t(t-1)}{n\cdot(n-1)} = \frac{\left\lceil 1+n/\sqrt{2}\right\rceil\left(\left\lceil 1+n/\sqrt{2}\right\rceil - 1\right)}{n(n-1)} \geq \frac{1}{2}.$$

∎

```
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 1.5: CONTRACT(G,t) shrinks G till it has only t vertices. FASTCUT computes the minimum cut using CONTRACT.

Theorem 1.3.5 FASTCUT *finds the minimum cut with probability larger than* $\Omega(1/\log n)$.

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 1.3.4), 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 $\leq 1 - \frac{1}{2}P\Big(\frac{n}{\sqrt{2}}\Big)$.

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

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

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. Divide both sides of the equation by $P\big(n/\sqrt{2}\big)$ 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,

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

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

Let $\Delta = \log n$

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

and

$$4(\Delta - \log\sqrt{2})^2 \geq 4\Delta(\Delta - \log\sqrt{2}) - c\Delta.$$

Which implies

$$-8\Delta\log\sqrt{2} + 4\log^2\sqrt{2} \geq -4\Delta\log\sqrt{2} - c\Delta$$

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

which clear 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 1.3.6 Prove, that running FastCut $c\cdot\log^2 n$ times, guarantee that the algorithm outputs the minimum cut with probability $\geq 1 - 1/n^2$, say, for c a constant large enough.

1.4 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 min-cut is described in [MR95, pages 7–9], the faster algorithm is described in [MR95, pages 289–295].

Chapter 2

Complexity, the Changing Minimum and Closest Pair

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

2.1 Las Vegas and Monte Carlo algorithms

Definition 2.1.1 A *Las Vegas algorithm!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 2.1.2 *Monte Carlo algorithm!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.

2.1.1 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 2.1.3 The class **P** consists of all languages L that have a polynomial time algorithm A, such that for any input Σ^*,

- $x \in L \Rightarrow A(x)$ accepts.
- $x \notin L \Rightarrow A(x)$ rejects.

Definition 2.1.4 The class **NP** consists of all languages L that have a polynomial time algorithm A, such that for any input Σ^*,

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

- $x \notin L \Rightarrow$ then $\forall y \in \Sigma^* A(x, y)$ rejects.

Definition 2.1.5 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 $\mathbf{P} = \text{co}-\mathbf{P}$ and $\mathbf{P} \subseteq \mathbf{NP} \cap \text{co}-\mathbf{NP}$. (It is currently unknown if $\mathbf{P} = \mathbf{NP} \cap \text{co}-\mathbf{NP}$ or whether $\mathbf{NP} = \text{co}-\mathbf{NP}$, although both statements are believed to be false.)

Definition 2.1.6 The class **RP** (for Randomized Polynomial time) consists of all languages L that have a randomized algorithm A with worst case polynomial running time such that for any input $x \in \Sigma^*$,

- $x \in L \Rightarrow \mathbf{Pr}[A(x) \text{ accepts}] \geq 1/2$.
- $x \notin L \Rightarrow \mathbf{Pr}[A(x) \text{ accepts}] = 0$.

An **RP** algorithm is Monte Carlo, but the mistake can only be if $x \in L$. 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 $\mathbf{RP} \cap \text{co}-\mathbf{RP}$ has an algorithm that does not make a mistake, namely a Las Vegas algorithm.

Definition 2.1.7 The class **ZPP** (for Zero-error Probabilistic Polynomial time) is the class of languages that have Las Vegas algorithms in expected polynomial time.

Definition 2.1.8 The class **PP** (for Probabilistic Polynomial time) is the class of languages that have a randomized algorithm A with worst case polynomial running time such that for any input $x \in \Sigma^*$,

- $x \in L \Rightarrow \mathbf{Pr}[A(x) \text{ accepts}] > 1/2$.
- $x \notin L \Rightarrow \mathbf{Pr}[A(x) \text{ accepts}] < 1/2$.

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

Definition 2.1.9 The class **BPP** (for Bounded-error Probabilistic Polynomial time) is the class of languages that have a randomized algorithm A with worst case polynomial running time such that for any input $x \in \Sigma^*$,

- $x \in L \Rightarrow \mathbf{Pr}[A(x) \text{ accepts}] \geq 3/4$.
- $x \notin L \Rightarrow \mathbf{Pr}[A(x) \text{ accepts}] \leq 1/4$.

2.2 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 appears in the sequence $c_1, \ldots, c_n$. What is the expectation of X?

Lemma 2.2.1 *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 $\mathbf{E}[X] = \sum_i \mathbf{E}[X_i] = \sum_{i=1}^{n} \frac{1}{i} = O(\log n)$. ■

2.3 Closest Pair

Assumption 2.3.1 *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 r a real positive number and a point $p = (x, y)$ in $\mathbb{R}^2$, define $\mathrm{G}_r(p)$ to be the point $(\lfloor x/r \rfloor r, \lfloor y/r \rfloor r)$. 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 parameter r, the partition of P into subsets by the grid G_r, is denoted by $\mathrm{G}_r(P)$. More formally, two points $p, q \in P$ belong to the same set in the partition $\mathrm{G}_r(P)$, 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 $p = (x, y)$ be any point in C, and consider the pair of integer numbers $\mathrm{id}_C = \mathrm{id}(p) = (\lfloor x/r \rfloor, \lfloor y/r \rfloor)$. Clearly, only points inside C are going to be mapped to id_C. This is very useful, since we store a set P of points inside a grid efficiently. Indeed, given a point p, compute its $\mathrm{id}(p)$. 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 $\mathrm{id}(p)$, 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 2.3.2 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 $\mathcal{CP}(P) = \min_{p,q \in P} \|pq\|$.

Lemma 2.3.3 *Given a set P of n points in the plane, and a distance r, one can verify in linear time, whether or not $\mathcal{CP}(P) < r$ or $\mathcal{CP}(P) \geq r$.*

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 $\mathrm{G}_r(P)$ contains more than, say, 9 points of p, then it must be that the $\mathcal{CP}(P) < r$. 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 $C' = \operatorname{diam}(C)/3 = \sqrt{r^2 + r^2}/3 < r$. Thus, the (at least) two points of P in C' are 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 $\mathbf{d}(p, S) < r$, we stop, otherwise, we continue to the next point, where $\mathbf{d}(p, S) = \min_{s \in S} \|ps\|$.

Overall, this takes $O(n)$ time. As for correctness, first observe that if $\mathcal{CP}(P) > r$ then the algorithm would never make a mistake, since it returns '$\mathcal{CP}(P) < r$' 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 $\|pq\| = \mathcal{CP}(P) < r$. 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 '$\mathcal{CP}(P) < r$'. ∎

Lemma 2.3.3 gives a natural way of computing $\mathcal{CP}(P)$. Indeed, permute the points of P in arbitrary fashion, and let $P = \langle p_1, \ldots, p_n \rangle$. Next, let $r_i = \mathcal{CP}(\{p_1, \ldots, p_i\})$. We can check if $r_{i+1} < r_i$, by just calling the algorithm for Lemma 2.3.3 on P_{i+1} and r_i. In fact, if $r_{i+1} < r_i$, the algorithm of Lemma 2.3.3, 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 check for awhile. In this case, we do not need to rebuild the data structure of Lemma 2.3.3, 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 $P_i = \langle p_1, \ldots, p_i \rangle$ into the new grid $\mathrm{G}_{r_i}(P_i)$. 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 2.3.4 *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 2.3.3. 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 $\mathrm{G}_{r_i}(P_i)$.

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 indicator variable X_i, we have $\mathbf{E}[X_i] = \mathbf{Pr}[X_i = 1]$.

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 $q \in P_i$ is called *critical*, if $\mathcal{CP}(P_i \setminus \{q\}) > \mathcal{CP}(P_i)$. 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 $\mathcal{CP}(P_i)$. The quantity r_i is smaller than r_{i-1}, one 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 realizing the closest distance, than if there is a third critical point r, then $\mathcal{CP}(P_i \setminus \{r\}) = \|pq\|$, and r is not critical.

We conclude that

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

We have that the expected running time is $O(\mathbf{E}[R]) = O(n)$. ■

Theorem 2.3.4 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 2.3.4 is considerably stronger, using hashing, randomization, and the floor function.

2.4 Bibliographical notes

Section 2.1 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 3

The Occupancy and Coupon Collector problems

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

3.1 Preliminaries

Definition 3.1.1 (Variance and Standard Deviation) For a random variable X, let $\mathbf{V}[X] = \mathbf{E}\big[(X - \mu_X)^2\big] = \mathbf{E}\big[X^2\big] - \mu_X^2$ 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 3.1.2 *(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 3.1.3 (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 3.1.4 (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

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

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

Observation 3.1.5 *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 3.1.6 *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 $\frac{n}{k} \leq n - 1k - 1$, as can be easily verified. As such, $\frac{n}{k} \leq \frac{n-i}{k-i}$, for $1 \leq i \leq k-1$. As such,

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

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). ∎

3.2 Occupancy Problems

Problem 3.2.1 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 3.1.6 (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,

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

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 3.2.2 *With probability at least $1 - 1/n$, no bin has more than $k^* = \lceil (3\ln n)/\ln\ln n \rceil$ balls in it.*

Exercise 3.2.3 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 3.2.4 *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 $\ln\ln n/lnd + O(1)$, 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].

3.2.1 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.

3.3 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 $\mathbf{E}[xf(Y)] = \sum_y f(y) \cdot \mathbf{Pr}[Y = y]$.

Theorem 3.3.1 (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, to see that:

Exercise 3.3.2 Define a random positive variable X, such that $\mathbf{Pr}[X \geq k\,\mathbf{E}[X]] = \frac{1}{k}$.

Theorem 3.3.3 (Chebyshev inequality) $\mathbf{Pr}[|X - \mu_X| \geq t\sigma_X] \leq \dfrac{1}{t^2}$.

Proof: Note that

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

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

3.4 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 i-th trial. The j-th 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 i-th success, till after the $(i + 1)$-th success. Clearly, the number of trials performed is

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

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

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 $H_n = \sum_{i=1}^{n} \frac{1}{i}$ is the n-th 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 $\lim_{n\to\infty} \frac{\mathbf{V}[X]}{n^2} = \frac{\pi^2}{6}$.

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.

3.5 Notes

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

Chapter 4

The Occupancy and Coupon Collector problems - part II

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

4.1 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 i-th coupon was not picked in the first r trials. Clearly,

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

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

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

This is quite strong, but still not as strong as we can do.

Lemma 4.1.1 *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 4.3.1, we have

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

Observe also that $\lim_{n\to\infty}\left(1-\frac{k^2m}{n}\right)=1$, 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 4.1.2 *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*

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

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

Proof: We have $\mathbf{Pr}[X>m]=\mathbf{Pr}[\cup_i Z_i^m]$. 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

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

Let $S_k^n=\sum_{i=1}^{k}(-1)^{i+1}P_i^n$. We know that $S_{2k}^n\le\mathbf{Pr}[\bigcup_i Z_i^m]\le S_{2k+1}^n$.

By symmetry,

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

Thus, $P_k=\lim_{n\to\infty}P_k^n=\exp(-ck)/k!$, by Lemma 4.1.1.

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

$$\lim_{n\to\infty}\mathbf{Pr}[X>m]=\lim_{n\to\infty}\mathbf{Pr}[\cup_{i=1}^{n}Z_i^m]=\lim_{n\to\infty}\lim_{k\to\infty}S_k^n=\lim_{k\to\infty}S_k=1-\exp\left(-e^{-c}\right).$$ ■

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

Figure 4.1: The LazySelect algorithm.

4.2 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 i-th element in the sorted list of S.

Given k, we would like to compute S_k (i.e., select the k-th element). The code of LazySelect is depicted in Figure 4.1.

Exercise 4.2.1 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

$$\mathcal{I}(j) = \left[R_{(\alpha(j))}, R_{(\beta(j))} \right],$$

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

Lemma 4.2.2 *For a fixed j, we have that* $\mathbf{Pr}\left[S_{(j)} \in \mathcal{I}(j)\right] \geq 1 - 1/(4n^{1/4})$.

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

By Chebyshev inequality

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

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

$$\begin{aligned}\Pr\Big[X < (jn^{-1/4} - \sqrt{n}) \text{ or } X > (jn^{-1/4} - \sqrt{n})\Big] &= \Pr\left[|X - jn^{-1/4}| \geq 2n^{1/8} \cdot \frac{n^{3/8}}{2}\right] \\ &\leq \frac{1}{\left(2n^{1/8}\right)^2} = \frac{1}{4n^{1/4}}.\end{aligned}$$

■

Lemma 4.2.3 LazySelect *succeeds with probability* $\geq 1 - O(n^{-1/4})$ *in the first iteration. And it performs only* $2n + o(n)$ *comparisons.*

Proof: By Lemma 4.2.2, we know that $S_{(k)} \in \mathcal{I}(k)$ with probability $\geq 1 - 1/(4n^{1/4})$. 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 $\mathcal{I}(k^-) \cap \mathcal{I}(k) = \emptyset$ and $\mathcal{I}(k) \cap \mathcal{I}(k^+) = \emptyset$. As such, we know by Lemma 4.2.2, that $S_{(k^-)} \in \mathcal{I}(k^-)$ and $S_{(k^+)} \in \mathcal{I}(k^+)$, and this holds with probability $\geq 1 - \frac{2}{4n^{1/4}}$. As such, the set P, which is by definition contained in the range $\mathcal{I}(k)$, 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 $\geq 1 - \frac{3}{4n^{1/4}}$.

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).

4.3 A technical lemma

Lemma 4.3.1 *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, we prove it for $x \geq 0$. Let us first prove that

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

Dividing by $(1 + x)$, we get $(1 - x)e^x \leq 1$, which obviously holds by the Taylor expansion of e^x. Indeed,

$$\begin{aligned}(1-x)e^x = e^x - xe^x = 1 \quad + \quad & x/1! + x^2/2! + x^3/3! \ldots \\ & -x - x^2/1! - x^3/2! \ldots \leq 1.\end{aligned}$$

Next, observe that $(1 - x^2)^y \geq 1 - yx^2$, for $y \geq 1$. As such,

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

A similar argumentation works for $x \leq 0$. ■

Chapter 5

Sampling and other Stuff

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

5.1 Two-Point Sampling

5.1.1 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 a, b are equivalent modulo p, if $a \equiv p \mod p$; namely, the reminder of dividing a and b by p is the same.

Lemma 5.1.1 *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 5.1.2 *Given $y, z, x, w \in \mathbb{Z}_p$, such that $x \neq w$, and choosing a, 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 5.1.3 *Let i, j be two distinct elements of $\mathbb{Z}_p$. And choose a, 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 5.1.1, as $\mathbf{Pr}[Y_i = \alpha] = 1/p$, for any $\alpha \in \mathbb{Z}_p$. As for being pairwise independent, observe that

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

by Lemma 5.1.1 and Lemma 5.1.2. Thus, Y_i and Y_j are pairwise independent. ■

Remark 5.1.4 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 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 5.1.5 *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 $\mathbf{E}[XY] = \mathbf{E}[X]\, E\Big[Y\Big]$, as can be easily verified. Thus,

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

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

5.1.2 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 5.1.6 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 $\sigma_Y^2 = \mathbf{V}[Y] = \sum_{i=1}^{t} \operatorname{var}\Big[A(x, r_i)\Big] \leq t/4$, 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.

5.2 Chernoff Inequality - A Special Case

Theorem 5.2.1 *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 5.2.2 *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 5.2.3 *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 5.2.4 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.

5.2.1 Application – QuickSort is Quick

We revisit QuickSort. We remind the reader that the running time of QuickSort is proportional to the number of comparisions 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 involing 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 recurssion 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 indpendent 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 probablity 1/2. The question is how many elements in the sequence we need to read, till we get T ones.

Lemma 5.2.5 *Let $X_1, X_2, \ldots$ be an infinite sequence of indpendent 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 $2\exp(-t^2)$, for $t \leq \sqrt{M}$. If $t \geq \sqrt{M}$ then this probablity 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 5.2.3. For $t \leq \sqrt{M}$ we have that $L = 2M + 4t\sqrt{M} \leq 8M$, as such in this case $\mathbf{Pr}[Y \leq M] \leq 2\exp(-t^2)$.

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

Going back to the QuickSort problem, we have that if we sort n elements, the probablity 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 5.2.5. There are n elements being sorted, and as such the probablity that any element would partiticpate in more than $(4 + c + 1)\lceil \lg n \rceil$ recursive calls is smaller than $1/n^c$.

Lemma 5.2.6 *For any $c > 0$, the probablity that* QuickSort *performs more than $(6 + c)n \lg n$, is smaller than $1/n^c$.*

Chapter 6

Chernoff Inequality - Part II

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

6.1 Tail Inequalities

6.1.1 The Chernoff Bound — General Case

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

Question 6.1.1 *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 6.1.2 *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 $\delta \leq 2e - 1$,

$$\mathbf{Pr}[X > (1+\delta)\mu] < \exp\left(-\mu\delta^2/4\right), \tag{6.1}$$

and

$$\mathbf{Pr}[X > (1+\delta)\mu] < 2^{-\mu(1+\delta)}, \tag{6.2}$$

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,

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

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

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

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

For the proof of the simplified form, see Section 6.1.2. ■

Definition 6.1.3 $F^+(\mu,\delta) = \left[\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right]^\mu$.

Example 6.1.4 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 6.1.5 *Under the same assumptions as Theorem 6.1.2, we have:*

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

Definition 6.1.6 $F^-(\mu,\delta) = e^{-\mu\delta^2/2}$.

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

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

6.1.2 A More Convenient Form

Proof: **(of simplified form of Theorem 6.1.2)** Eq. (6.2) is easy. Indeed, we have

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

since $\delta > 2e - 1$.

Values	Probabilities	Inequality	Ref
$-1, +1$	$\mathbf{Pr}[X_i = -1] =$ $\mathbf{Pr}[X_i = 1] = \frac{1}{2}$	$\mathbf{Pr}[Y \geq \Delta] \leq e^{-\Delta^2/2n}$ $\mathbf{Pr}[Y \leq -\Delta] \leq e^{-\Delta^2/2n}$ $\mathbf{Pr}[\lvert Y\rvert \geq \Delta] \leq 2e^{-\Delta^2/2n}$	Theorem 5.2.1 Theorem 5.2.1 Corollary 5.2.2
$0, 1$	$\mathbf{Pr}[X_i = 0] =$ $\mathbf{Pr}[X_i = 1] = \frac{1}{2}$	$\mathbf{Pr}\left[\left\lvert Y - \frac{n}{2}\right\rvert \geq \Delta\right] \leq 2e^{-2\Delta^2/n}$	Corollary 5.2.3
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 6.1.2
	For $\delta \leq 2e-1$ $\delta \geq 2e-1$ For $\delta \geq 0$	$\mathbf{Pr}[Y > (1+\delta)\mu] < \exp\left(-\mu\delta^2/4\right)$ $\mathbf{Pr}[Y > (1+\delta)\mu] < 2^{-\mu(1+\delta)}$ $\mathbf{Pr}[Y < (1-\delta)\mu] < \exp\left(-\mu\delta^2/2\right)$	Theorem 6.1.2 Theorem 6.1.5

Table 6.1: Summary of Chernoff type inequalities covered. Here we have n variables $X_1, \ldots, X_n$, $Y = \sum_i X_i$ and $\mu = \mathbf{E}[Y]$.

As for Eq. (6.1), 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 6.1.2, 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 $\delta \leq 1/2$. ■

6.2 Application of the Chernoff Inequality – Routing in a Parallel Computer

The following is based on Section 4.2 in [MR95].

Let G be a graph of a network, where every node is a processor. The processor communicate by sending packets on the edges. Let $[1, \ldots, N]$ denote be vertices (i.e., processors) of G, where $N = 2^n$, and G is the hypercube. As such, each processes is a binary string $b_1 b_2 \ldots b_n$.

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.

(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 6.1: The routing algorithm

Theorem 6.2.1 *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 $\Omega(\sqrt{N/n})$ 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 6.1.

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 6.2.2 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 6.2.3 *Let the route of a message* $\mathbf{c}$ *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* $\mathbf{c}$ *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 if ready to follow edge e_j at time t, we define its *lag* at time t to be $t - j$. The lag of $\mathbf{c}$ is initially zero, and the delay incurred by $\mathbf{c}$ is its lag when it traverse e_k. We will show that each step at which the lag of $\mathbf{c}$ increases by one can be charged to a distinct member of S.

We argue that if the lag of $\mathbf{c}$ reaches $\ell + 1$, some packet in S leaves π with lag ℓ. When the lag of $\mathbf{c}$ increases from ℓ to $\ell + 1$, there must be at least one packet (from S) that wishes to traverse the same edge as $\mathbf{c}$ at that time step, since otherwise $\mathbf{c}$ 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 $\mathbf{d}$ 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 $\mathbf{c}$.

Since $\mathbf{d}$ is ready to follow e_μ at τ, some packet ω (which may be $\mathbf{d}$ 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 $\mathbf{c}$ 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.

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

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

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

By the Chernoff inequality, we have

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

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

6.3 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

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

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

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

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

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

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

Lemma 6.3.1 *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 6.3.1. A pair p, q of points of P have *Euclidean* distance at least $\sqrt{d_H(p,q)} = \sqrt{n}4 = \sqrt{n}/2$ from each other. We conclude:

Lemma 6.3.2 *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* $|Q| \geq \exp(n/16)$*.*

6.4 Bibliographical notes

The exposition here follows more or less the exposition in [MR95]. Exercise 6.5.1 (without the hint) is from [Mat99]. A similar result to Theorem 6.2.4 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].

6.5 Exercises

Exercise 6.5.1 [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 6.5.2 To some extent, Lemma 6.3.1 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 7

Martingales

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

'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

7.1 Martingales

7.1.1 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 7.1.1 The random variable $E\Big[X \;\Big|\; Y\Big]$ is the random variable $f(y) = E\Big[X \;\Big|\; Y = y\Big]$.

Lemma 7.1.2 $\mathbf{E}\Big[\mathbf{E}\Big[X \;\Big|\; Y\Big]\Big] = E\Big[Y\Big]$.

Proof:

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

■

Lemma 7.1.3 $\mathbf{E}\Big[Y\cdot\mathbf{E}\Big[X \;\Big|\; Y\Big]\Big] = \mathbf{E}[XY]$.

Proof:

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

■

7.1.2 Martingales

Definition 7.1.4 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 7.1.5 *Let $X_0, X_1, \ldots,$ be a martingale sequence. Then, for all $i\geq 0$, we have $\mathbf{E}[X_i] = \mathbf{E}[X_0]$.*

An example for martingales is the sum of money after participating in a sequence of fair bets.

Example 7.1.6 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 $1\leq j\leq m$, 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

$$X_i = \mathbf{E}\Big[f(G) \;\Big|\; I_1,\ldots,I_k\Big]\,,$$

while $X_0 = E(f(G)]$ 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 7.1.7 (Azuma's Inequzality) *Let $X_0, \ldots, X_m$ be a martingale with $X_0 = 0$, and $|X_{i+1} - X_i| \leq 1$ for all $0 \leq i < m$. Let $\lambda > 0$ be arbitrary. Then*

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

Proof: Let $\alpha = \lambda/\sqrt{m}$. Let $Y_i = X_i - X_{i-1}$, so that $|Y_i| \leq 1$ and $\mathbf{E}\Big[Y_i \;\Big|\; X_0, \ldots, X_{i-1}\Big] = 0$. We are interested in bounding $E\Big[e^{\alpha Y_i} \;\Big|\; X_0, \ldots, X_{i-1}\Big]$. Note that, for $-1 \leq x \leq 1$, we have

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

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,

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

Therefore, by Markov's inequality, we have

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

implying the result. ∎

Alternative form:

Theorem 7.1.8 (Azuma's Inequzality) *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 7.1.9 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 $X_i = E\Big[\chi(G) \;\Big|\; G_i\Big]$. Again, without proving it, we claim that $X_0, \ldots, X_n = X$ is a martingale, and as such, we have: $\mathbf{Pr}[|X_n - X_0| > \lambda\sqrt{n}] \leq e^{-\lambda^2/2}$. However, $X_0 = \mathbf{E}[\chi(G)]$, and $X_n = E\Big[\chi(G) \;\Big|\; G_n\Big] = \chi(G)$. Thus,

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

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

7.2 Even more probability

Definition 7.2.1 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 7.2.2 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 7.2.3 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 8

Martingales II

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

"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.

8.1 Filters and Martingales

Definition 8.1.1 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.

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

Example 8.1.2 Consider an algorithm A that uses n random bits, and let $\mathcal{F}_i$ be the σ-field generated by the partition of Ω into the blocks B_w, where $w \in \{0,1\}^i$. Then $\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_n$ form a filter.

Definition 8.1.3 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 8.1.4 Let $\mathcal{F}_0, \ldots, \mathcal{F}_n$ be the filter defined in Example 8.1.2. 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 $i < n$.

Namely, a random variable X is $\mathcal{F}_i$-measurable, only if it is a constant on the blocks of $\mathcal{F}_i$.

Definition 8.1.5 Let $(\Omega, \mathcal{F})$ be any σ-field, and Y any random variable that takes on distinct values on the elementary elements in $\mathcal{F}$. Then $\mathbf{E}\Big[X \;\Big|\; \mathcal{F}\Big] = \mathbf{E}\Big[X \;\Big|\; Y\Big]$.

8.2 Martingales

Definition 8.2.1 A sequence of random variables $Y_1, Y_2, \ldots,$ is said to be a *martingale difference* sequence if for all $i \geq 0$,

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

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

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

and a *sub martingale* sequence if

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

Example 8.2.3 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,

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

8.2.1 Martingales, an alternative definition

Definition 8.2.4 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 $X_0, \ldots, kX_n$ is a martingale provided, for all $i \geq 0$,

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

Lemma 8.2.5 *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: $\mathbf{E}\Big[\mathbf{E}\Big[X \;\Big|\; \mathcal{G}\Big] \;\Big|\; \mathcal{F}\Big] = \mathbf{E}\Big[\mathbf{E}\Big[X \;\Big|\; G = g\Big] \;\Big|\; F = f\Big]$

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

■

Theorem 8.2.6 *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 $X_i = \mathbf{E}\Big[X \;\Big|\; F_i\Big]$ then, the sequence $X_0, \ldots, X_n$ is a martingale.*

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

$$\mathbf{E}\Big[X_{i+1} \;\Big|\; F_i\Big] = \mathbf{E}\Big[\mathbf{E}\Big[X \;\Big|\; F_{i+1}\Big] \;\Big|\; F_i\Big] = \mathbf{E}\Big[X \;\Big|\; F_i\Big] = X_i,$$

by Lemma 8.2.5 and by definition of X_i. ■

Definition 8.2.7 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 8.2.8 Let $X_1, \ldots, X_n$ be a sequence of random variables, and a function $f(X_1, \ldots, X_n)$ defined over them that such that f satisfies the Lipschitz condition. The *Dobb 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 8.2.6.

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

8.3 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 $\mathbf{Pr}[|Z - \mathbf{E}[Z]| > \lambda\sqrt{m}] \leq 2e^{-\lambda^2/2}$.

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 8.3.1 (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 8.3.2 *Let $r = m/n$, and Z_m be the number of empty bins when m balls are thrown randomly into n bins. Then*

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

and for $\lambda > 0$,

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

Proof: Let $z(Y, t)$ be the expected number of empty bins, i 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_m be the end of empty balls at time m, and let $Z_t = \mathbf{E}\Big[Z_m \;\Big|\; F_t\Big]$. 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 9

The Probabilistic Method

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

"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

9.1 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".

9.1.1 Examples

Theorem 9.1.1 *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,

$$\mathbf{E}\Big[\Big|\Big\{uv \in E \;\Big|\; u \in A \text{ and } v \in B\Big\}\Big|\Big] = \sum_{e \in E(G)} \mathbf{E}[X_e] = \sum_{e \in E(G)} \frac{1}{2} = \frac{m}{2}.$$

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

Definition 9.1.2 For a vector $v = (v_1, \ldots, v_n) \in \mathbb{R}^n$, $\|v\|_\infty = \max_i |v_i|$.

Theorem 9.1.3 *Let A 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 $\|Ab\|_\infty \leq 4\sqrt{n \log n}$.*

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

$$\mathbf{E}[v \cdot b] = \sum_i \mathbf{E}[v_i b_i] = \sum_j v_i \, \mathbf{E}\big[b_{i_j}\big] = 0.$$

Let $X_j = 1$ if $b_{i_j} = +1$, for $j = 1, \ldots, m$. We have $\mathbf{E}\Big[\sum_j X_j\Big] = n/2$, and

$$\begin{aligned}
\mathbf{Pr}\Big[|v \cdot b| \geq 4\sqrt{n \ln n}\Big] &= 2\,\mathbf{Pr}\Big[v \cdot b \leq -4\sqrt{n \ln n}\Big] = 2\,\mathbf{Pr}\left[\sum_j X_j - \frac{n}{2} \leq -2\sqrt{n \ln n}\right] \\
&= 2\,\mathbf{Pr}\left[\sum_j X_j < \left(1 - 4\sqrt{\frac{\ln n}{n}}\frac{n}{m}\right)\frac{m}{2}\right] \\
&\leq 2\exp\left(-\frac{m}{2}\left(4\sqrt{\frac{\ln n}{n}}\frac{n}{m}\right)^2\right) = 2\exp\left(-\frac{m}{2}\left(16\frac{n \ln n}{m^2}\right)\right) \\
&= 2\exp\left(-\frac{8n \ln n}{m}\right) \\
&\leq 2\exp(-8 \ln n) = \frac{2}{n^8}
\end{aligned}$$

by the Chernoff inequality and symmetry. Thus, the probability that any entry in Ab 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 Ab have value smaller than $4\sqrt{n \ln n}$.

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

9.2 Maximum Satisfiability

Theorem 9.2.1 *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 even 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 A, let $m_A(I)$ denote the number of clauses satisfied computed by the algorithm A. The *approximation factor* of A, is $m_A(I)/m_{\text{opt}}(I)$. Clearly, the algorithm of Theorem 9.2.1 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 i-th 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:

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

We relax this into the following linear program:

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

Which can be solved in polynomial time. Let $\widehat{\cdot}$ denote the values assigned to the variables 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 called *randomized rounding*.

Lemma 9.2.2 *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,

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

The second part of the inequality,m 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,

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

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 9.2.3 *Given an instance of MAX-SAT, the expected number of clauses satisfied by linear programming and randomized rounding is at least* $(1 - 1/e)$ *times the maximum number of clauses that can be satisfied on that instance.*

Theorem 9.2.4 *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,* $\max(n_1, n_2) \geq \frac{3}{4} \sum_j \widehat{z}_j$.

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 9.2.2 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. [②] Thus, we have

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

■

[②] Indeed, by the proof of Lemma 9.2.2, 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 to check the inequality for $k = 1$ and $k = 2$, which can be done directly.

Chapter 10

The Probabilistic Method II

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

"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."

— – The tin drum, Gunter Grass

10.1 Expanding Graphs

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

Definition 10.1.1 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 radinality n, such that

1. Every vertex in L has degree at most d.
2. For any subset S of vertices from L , such that $|S| \leq \alpha n$, there are at least $c\,|S|$ neighbors in R.

(d should be as small as possible, and c as large as possible.)

Theorem 10.1.2 *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. ■

10.2 Probability Amplification

Let A be an algorithm in **RP**, such that given x, A 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 $A(x, r)$ with the following prpoerties:

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

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

Theorem 10.2.1 *For n large enough, there exists a bipartite graph $G(L, R, E)$ with $|L| = n$, $|R| = 2^{\log^2 n}$ such that:*

1. *Even subset of $n/2$ vertices of L has at least $\left(2^{\log^2 n} - n\right)$ neighbors in R.*
2. *No vertex of R has more than $12 \log^2 n$ neighbors.*

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

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

$$\binom{n}{n/2} \binom{2^{\log^2 n}}{n} \left(1 - \frac{n}{2^{\log^2 n}}\right)^{dn/2} \ll \frac{1}{2},$$

as can be easily verified.

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

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

There exist implictly represented such graphs as the graph required in Theorem 10.2.1. Namely, we can assume that given a vertex we can compute its neighbors, without computing the whole graph. We assume that we are given such an implicty representation of an expanding graph.

Use $\log^2 n$ bits to pick a vertex $v \in R$. We next identify the neighbors of v in L: $r_1, \ldots, r_k$. We then compute $A(x, r_i$ for $1 \leq i \leq k$. Note that $k = O(\log^2 n)$. If all k calls return 0, then we return that A is not in the language. Otherwise, we return that x belong to L.

Clearly, the probablity for our failure is $n/\log^2 n$, as n is the number of vertices on R which fail to be connected to one of the (at least $n/2$) witnesses of L.

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

10.3 Oblivious routing revisited

Theorem 10.3.1 *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 10.3.2 *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 10.3.3 *For every n, there exists a randomized oblivious scheme for permutation routing on a hypercube with $n = 2^n$ nodes that uses $3n$ rnadom bits and runs in expected time at most $15n$.*

Chapter 11

The Probabilistic Method III

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

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

11.1 The Lovász Local Lemma

Lemma 11.1.1 *(i)* $\mathbf{Pr}\Big[A \;\Big|\; B \cap C\Big] = \frac{\mathbf{Pr}\Big[A \cap B \;\Big|\; C\Big]}{\mathbf{Pr}\Big[B \;\Big|\; C\Big]}$

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

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

Proof:

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

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

Lemma 11.1.2 (Lovász Local Lemma) *Let $G(V, E)$ be a dependency graph for events $C_1, \ldots, C_n$. Suppose that there exist $x_i \in [0, 1]$, for $1 \leq i \leq n$ such that*

$$\mathbf{Pr}[C_i] \leq x_i \prod_{(i,j) \in E} (1 - x_j).$$

Then

$$\mathbf{Pr}\big[\cap_{i=1}^{n} \overline{C_i}\big] \geq \prod_{i=1}^{n} (1 - x_i).$$

Proof: Let S denote a subset of the vertices from $\{1,\ldots,n\}$. We first establish by induction on $k = |S|$ that for any S and for any i such that $i \notin S$,

$$\mathbf{Pr}\Big[C_i \;\Big|\; \cap_{j\in S}\overline{C_j}\Big] \leq x_i. \tag{11.1}$$

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

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 C_i is mutually independent of the events of $\mathcal{C}(R) = \Big\{C_j \;\Big|\; j \in R\Big\}$. Thus, $\mathbf{Pr}\Big[C_i \;\Big|\; \cap_{j\in S}\overline{C_j}\Big] = \mathbf{Pr}\Big[C_i \;\Big|\; \cap_{j\in R}\overline{C_j}\Big] = \mathbf{Pr}[C_i] \leq x_i$, by arguing as above.

By Lemma 11.1.1 (i), we have that

$$\mathbf{Pr}\left[C_i \;\middle|\; \bigcap_{j\in S}\overline{C_j}\right] = \frac{\mathbf{Pr}\Big[C_i \cap \big(\cap_{j\in N}\overline{C_j}\big) \;\Big|\; \cap_{m\in R}\overline{C_m}\Big]}{\mathbf{Pr}\Big[\cap_{j\in N}\overline{C_j} \;\Big|\; \cap_{m\in R}\overline{C_m}\Big]}.$$

We bound the numerator by

$$\mathbf{Pr}\Big[C_i \cap \big(\cap_{j\in N}\overline{C_j}\big) \;\Big|\; \cap_{m\in R}\overline{C_m}\Big] \leq \mathbf{Pr}\Big[C_i \;\Big|\; \cap_{m\in R}\overline{C_m}\Big] = \mathbf{Pr}[C_i] \leq x_i \prod_{(i,j)\in E}(1-x_j)\,,$$

since C_i is mutually independent of $\mathcal{C}(R)$. As for the denominator, let $N = \{j_1,\ldots,j_r\}$. We have, by Lemma 11.1.1 (ii), that

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

by Eq. (11.1) and induction, as every probability term in the above expression has less than $|S|$ items involved. It thus follows, that $\mathbf{Pr}\Big[C_i \;\Big|\; \bigcap_{j\in S}\overline{C_j}\Big] \leq x_i$.

Now, the proof of the lemma, follows from

$$\mathbf{Pr}\big[\cap_{i=1}^{n}\overline{C_i}\big] = (1-\mathbf{Pr}[C_1])\Big(1 - \mathbf{Pr}\Big[C_2 \;\Big|\; \overline{C_1}\Big]\Big)\cdots\Big(1 - \mathbf{Pr}\Big[C_n \;\Big|\; \cap_{i=1}^{n-1}\overline{C_i}\Big]\Big) \geq \prod_{i=1}^{n}(1-x_i)\,. \qquad \blacksquare$$

Corollary 11.1.3 *Let $C_1,\ldots,C_n$ be events, with $\mathbf{Pr}[C_i] \leq p$ 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 $\mathbf{Pr}\big[\cap_{i=1}^{n}\overline{C_i}\big] > 0$.*

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 11.1.2 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 fact that $\left(1-\frac{1}{d+1}\right)^d > 1/e$. To see that, observe that, observe that we need to show that $1/\left(1-\frac{1}{d+1}\right)^d < e$, which is equivalent to $((d+1)/d) < e^{1/d}$. However,

$$\frac{d+1}{d} = 1+\frac{1}{d} < 1+\left(\frac{1}{d}\right)+\frac{1}{2!}\left(\frac{1}{d}\right)^2+\frac{1}{3!}\left(\frac{1}{d}\right)^3+\cdots = e^{1/d},$$

establishing the claim. ■

11.2 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 C_i be the event that the ith clause was not satisfied. We know that $p = \mathbf{Pr}[C_i] = 2^{-k}$, and furthermore, C_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 11.1.3, that

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

11.2.1 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 11.2.1 A clause C_i is *dangerous* if both the following conditions hold:

1. $k/2$ literals of C_i have been fixed.
2. C_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.

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 11.2.2 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 11.2.3 *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 11.2.4 *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 C_i and C_j in U is an independent event, as no *neighbor* of C_i shares a variable with a neighbor of C_j (because of the distance 4 requirement). We conclude, that the probability that all the vertices of U appear in G' is bounded by

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

On the other hand, 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 11.2.3. 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 11.2.5 *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 12

The Probabilistic Method IV

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

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

12.1 The Method of Conditional Probabilities

In previous lecture, we encountered the following problem:

Problem 12.1.1 (Set Balancing) Given a binary matrix A of size $n \times n$, find a vector $\vec{v} \in \{-1, +1\}^n$, such that $\|A\vec{v}\|_\infty$ is minimized.

Using random assignment and the Chernoff inequality, we showed that there exists $\vec{v}$, such that $\|A\vec{v}\|_\infty \leq 4\sqrt{n \ln n}$. 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 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 $v_m = (\alpha_1, \ldots, \alpha_n)$. 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 $L = 4\sqrt{n \log n} - \sum_{i=1}^{k} \vec{v}_i \alpha_i$. Let $V = \sum_{i \geq k+1, \alpha_i = 1} 1$. We have,

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

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

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

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 $P_m^- = \mathbf{Pr}\Big[C_m^- \;\Big|\; \vec{v}_1, \ldots, \vec{v}_k\Big]$.

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 $P(v) = \sum_{m=1}^{n} \mathbf{Pr}\Big[C_m \;\Big|\; \vec{v}_v\Big]$. 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 $\|A\vec{v}'\|_\infty \leq 4\sqrt{n \log n}$.

Theorem 12.1.2 *Using the method of conditional probabilities, one can compute in polynomial time in n, a vector $\vec{v} \in \{-1, 1\}^n$, such that $\|A\vec{v}\|_\infty \leq 4\sqrt{n \log n}$.*

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

12.2 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.

12.2.1 High Girth and High Chromatic Number

Definition 12.2.1 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 girth(G) denote the length of the shortest circle in G.

Theorem 12.2.2 *For all K, L there exists a graph G with* 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 as an edge 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 have

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

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. ∎

12.2.2 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* $\mathrm{cr}(G)$ of G is the minimum possible crossing number in an embedding of it in the plane.

Theorem 12.2.3 *The crossing number of any simple graph $G = (V, E)$ with $|E| \geq 4|V|$ is at least $\frac{|E|^3}{64|V|^2}$.*

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 12.2.4 *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 12.2.3, 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 12.2.5 *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 $m = |L| = s^2$. Furhtermore, 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 12.2.4, we have

$$s^3 \leq 4(m^{2/3}n^{2/3} + m + n) = 4\left(\left(s^2\right)^{2/3}(sr)^{2/3} + s^2 + sr\right) = 4\left(s^2 r^{2/3} + s^2 + sr\right).$$

For $r < s^3$, we have that $sr \leq s^2 r^{2/3}$. Thus, for $r < s^3$, we have $s^3 \leq 12 s^2 r^{2/3}$, 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 13

Random Walks I

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

"A drunk man will find his way home; a drunk bird may wander forever."

13.1 Definitions

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.

Example 13.1.1 In the completely graph K_n, visiting all the vertices takes in expectation $O(n \log n)$ time, as this is just the coupon collector problem with $n-1$ coupons. Similarly, arriving from u to v, takes in expectation $n-1$ steps of a random walk.

13.1.1 Walking on grids and lines

Lemma 13.1.2 *Consider the infinite random walk on the integer line, starting from* 0*. 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 $\frac{2^{2i}}{2\sqrt{i}} \leq \binom{2i}{i} \leq 2^{2i}\sqrt{2i}$, as can be verified from 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 13.1.3 *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 $\mathbf{Pr}[x'_{2i} = 0 \cap y'_{2i} = 0] = \mathbf{Pr}[x'_{2i} = 0]^2 = \left(\frac{1}{2^{2i}}\binom{2i}{i}\right)^2 \geq 1/4i$. We conclude, that the infinite random walk on the grid $\mathbb{Z}^2$ visits the origin in expectation

$$\sum_{i=0}^{\infty} \mathbf{Pr}\left[x'_i = 0 \cap y'_i = 0\right] \geq \sum_{i=0}^{\infty} \frac{1}{4i} = \infty,$$

as this sequence diverges. ∎

In the following, let $\binom{i}{a\ b\ c} = \dfrac{i!}{a!\,b!\,c!}$.

Lemma 13.1.4 *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 14

Random Walks II

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

"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." – The tin drum, Gunter Grass

14.1 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.

14.1.1 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 14.1.1 *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

$$\sum_{m=n/2}^{i} \frac{1}{2^i} \binom{i}{i-m} > 1/3,$$

for $i > \mu = c'n^2$, where c' is a large enough constant.

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, using those phases. 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. ■

14.2 Markov Chains

Let $\mathtt{S}$ 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 $\mathbf{P} = \{\mathtt{P}_{ij}\}_{ij}$ 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 $i, j \in \mathtt{S}$, 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 14.2.1 A state $i \in \mathtt{S}$ 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 i persistent and $\mathtt{h}_{ii} \neq \infty$ then it is *non null persistent*.

In finite Markov chains, there are no null persistent states (this required 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, j). Note that we include only edges with $\mathtt{P}_{ij} > 0$.

Definition 14.2.2 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 14.2.3 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 14.2.4 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 14.2.5 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 full defined by $\mathbf{q}^{(0)}$ and $\mathbf{P}$.

Definition 14.2.6 A *stationary distribution* for the Markov chain with 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 14.2.7 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 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 near trick that forces a Markov chain to be aperiodic, is to shrink all the probabilities by a factor of 2, and every state has transition probability to itself which is 1/. Clearly, the resulting Markov chain is aperiodic.

Definition 14.2.8 An *ergodic* state is aperiodic and (non-null) persistent.

An *ergodic* Markov check 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 14.2.9 (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 15

Random Walks III

" 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. " – Waiting for the Barbarians, J. M. Coetzee.

15.1 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 15.1.1 *For 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 15.1.2 *For all $v \in V$, $\mathtt{h}_{vv} = 1/\pi_v = 2m/d(v)$.*

Definition 15.1.3 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 15.1.4 Let L_n be the n-vertex *lollipop graph*, this graph consists of a clique on $n/2$ vertices, and a path on the remaining vertices. There is a vertex u in the clique which where the path is attached to it. Let v denote the end end of the path.

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 $(n/2-1)/(n/2)$ we enter $K_{n/2}$. 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 $\Omega(n^3)$.

Definition 15.1.5 A $n \times n$ matrix $\mathbf{M}$ is *stochastic* if all its entries are non-negative and for each row i, it holds $\sum_k \mathbf{M}_{ik} = 1$. It is *doubly stochastic* if in addition, for any i, it holds $\sum_k \mathbf{M}_{ki} = 1$.

Lemma 15.1.6 *Let $\mathcal{MC}$ 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 $\mathcal{MC}$.*

Proof: $[u\mathbf{P}]_i = \sum_{k=1}^{n} \frac{\mathtt{P}_{ki}}{n} = \frac{1}{n}$. ∎

Lemma 15.1.7 *For any edge $(u, v) \in E$, $\mathtt{h}_{uv} + \mathtt{h}_{vu} \leq 2m$.*

(Note, that the fact that (u, v) is an edge in the graph is crucial. Indeed, without it a worst bound holds, see Theorem 15.2.1.)

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 $Q_{(u,v),(v,w)} = \mathtt{P}_{vw} = \frac{1}{d(v)}$. 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, w) we have

$$\sum_{x \in V, y \in \Gamma(x)} Q_{(x,y),(v,w)} = \sum_{u \in \Gamma(v)} Q_{(u,v),(v,w)} = \sum_{u \in \Gamma(v)} \mathtt{P}_{vw} = d(v) \times \frac{1}{d(v)} = 1.$$

Thus, the stationary distribution for this Markov chain is uniform, by Lemma 15.1.6. Namely, the stationary distribution of $e = (u, v)$ is $\mathtt{h}_{ee} = \pi_e = 1/(2m)$. Thus, the expected time between successive traversals of e is $1/\pi_e = 2m$, by Theorem 15.3.1 (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, u) , we conclude that the expected time to go from there to v and then algorithm (v, 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 bac is at most $2m$. ∎

15.2 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 between the corresponding nodes in $\mathcal{N}(G)$.

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 15.2.1 *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 the edge uw is $\phi_{uw} = \phi_{uv} - \phi_{wv}$. Thus, using Kirchhoff's Law and Ohm's Law, we obtain that, for all ., we have

$$u \in V \setminus \{v\} \qquad d(u) = \sum_{w \in \Gamma(u)} \text{current}(uw) = \sum_{w \in \Gamma(u)} \frac{\phi_{uw}}{\text{resistance}(uw)} = \sum_{w \in \Gamma(u)} (\phi_{uv} - \phi_{wv}) .$$

By the definition of expectation we have

$$u \in V \setminus \{v\} \qquad \mathtt{h}_{uv} = \sum_{w \in \Gamma(u)} (1 + \mathtt{h}_{wv}) .$$

The last two displays show two systems of linear inequalities that have both a unique solution. However, if we identify $\mathtt{h}_{uv}$ with ϕ_{uv}. This implies, that $\phi_{uv} = \mathtt{h}_{uv}$, for all u, 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 $\widehat{\phi} = \phi_{uv} + \phi'_{uv} = \mathtt{h}_{uv} + \mathtt{h}_{vu} = \mathcal{C}uv$. 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. ∎

Example 15.2.2 Recall the lollipop from Exercise 15.1.4 L_n. Let u be the connecting vertex between the clique and the path. We inject $d(u)$ units of flow for each vertex u of L_n, and collect $2m$ units at u. Next, let $u = u_0, u_!, \ldots, u_{n/2} = v$ be the vertices of the path. Clearly, there are

$n/2-i$ units of electricity flowing on the edge (u_{i+1}, u_i). Thus, the resistance of this edge is $n/2-i$, 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 15.2.3 *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 15.2.1, yields the bound, since $m < n^2$. ■

15.3 Tools from previous lecture

Theorem 15.3.1 (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 $1 \le i \le n$, $\pi_i > 0$.

(iii) For $1 \le i \le n$, $\mathtt{f}_{ii} = 1$ 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.

15.4 Notes

A nice survey of the material covered here, is available at `http://arxiv.org/abs/math.PR/0001057`.

Chapter 16

Random Walks IV

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

" 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

16.1 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 $\mathcal{C}(G)$.

Theorem 16.1.1 *Let G be an undirected connected graph, then $\mathcal{C}(G) \leq 2m(n-1)$, where $n = |V(G)|$ and $m = |E(G)|$.*

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 15.1.7. 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 16.1.2 The *resistance* of G is $\mathbf{R}(G) = \max_{u,v \in V(G)} \mathbf{R}_{uv}$; namely, it is the maximum effective resistance in G.

Theorem 16.1.3 $m\mathbf{R}(G) \leq \mathcal{C}(G) \leq 2e^3 m\mathbf{R}(G) \ln n + 2n$.

Proof: Consider the vertices u and v realizing $\mathbf{R}(G)$, 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 15.2.1. Thus, $\mathcal{C}(G) \geq \mathbf{CT}_{uv}/2 \geq m\mathbf{R}(G)$.

As for the upper bound. Consider a random walk, and divide it into *epochs*, where a epoch is a random walk of length $2e^3 m\mathbf{R}(G)$. For any vertex v, the expected time to hit u is $\mathtt{h}_{vu} \leq 2m\mathbf{R}(G)$, by Theorem 15.2.1. 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 16.1.1. Thus, expected length of the walk is $\leq 2e^3 m\mathbf{R}(G) \ln n + 2n^3(1/n^2)$. ■

Rayleigh's Short-cut Principle. Observe that effective resistance is never raised by lowering the resistance on an edge, nd 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 16.1.4 *For a graph with minimum degree d, we have $\mathbf{R}(G) \geq 1/d$ (collapse all vertices except the minimum-degree vertex into a single vertex).*

Lemma 16.1.5 *Suppose that g contains p edge-disjoint paths of length at most ℓ from s to t. Then $\mathbf{R}_{st} \leq \ell/p$.*

16.2 Graph Connectivity

Definition 16.2.1 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 $\mathcal{RLP}$, 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 16.2.2 *Let $\mathcal{USTCON}$ denote the problem of deciding if a vertex s is connected to a vertex t in an undirected graph. Then $USTCON \in \mathcal{RLP}$.*

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 16.2.3 A graph d-regular, if all its vertices are of degree d.

A d-regular graph is *labeled* if at each vertex of the graph, ach 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.

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 graph of labeled graphs 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 $\mathcal{USTCON}$ 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 16.2.4 $U(\mathcal{F}) \leq 5m\mathbf{R}(\mathcal{F}) \lg(n\,|\mathcal{F}|)$.

Let $U(d, n)$ denote the length of the shortest universal traversal sequence of connected, labeled n-vertex, d-regular graphs.

Lemma 16.2.5 *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. Eery vertex has $d!$ possible labeling of the edges adjacent to it. ■

Lemma 16.2.6 $U(d,n) = O(n^3 d log n)$.

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 16.2.5 and Theorem 16.2.4. ■

This is, as mentioned before, not uniform solution. There is by now a known log-space deterministic algorithm for this problem, which is uniform.

16.2.1 Directed graphs

Theorem 16.2.7 *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*.*

16.3 Graphs and Eigenvalues

Consider an undirected graph $G = G(V,E)$ with n vertices. The adjacency matrix $A(G)$ of G is the $n \times n$ symmetric matrix where $A_{ij} = A_{ji}$ is the number of edges between the vertices v_i and v_j. Where G is bipartite, we assume that its has two independent sets X and Y. In this case the matrix $A(G)$ can be written in block form.

Since $A(G)$ 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 16.3.1 (Fundamental theorem of algebraic graph theory) *Let* $G = G(V,E)$ *be an* n*-vertex, undirected (multi)graph with maximum degree* d*. Then, under the canonical labeling of eigenvalues* λ_i *and orthonormal eigenvectors* e_i *for the matrix* $A(G)$ *we have:*

(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* $\lambda_1 = d$ *ha 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* 1*s and* -1*s in* e_n.

16.4 Bibliographical Notes

A nice survey of algebraic graph theory appears in [Wes01] and in [Bol98].

Chapter 17

The Johnson-Lindenstrauss Lemma

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

17.1 The Johnson-Lindenstrauss lemma

17.1.1 Some Probability

Definition 17.1.1 Let $N(0,1)$ denote the one dimensional *normal distribution*. This distribution has density $n(x) = e^{-x^2/2}/\sqrt{2\pi}$.

Let $N^d(0,1)$ denote the d-dimensional *Gaussian distribution*, induced by picking each coordinate independently from the standard normal distribution $N(0,1)$.

Let $\mathrm{Exp}(\lambda)$ denote the *exponential distribution*, with parameter λ. The density function of the exponential distribution is $f(x) = \lambda \exp(-\lambda x)$.

Let $\Gamma_{\lambda,k}$ denote the *gamma distribution*, with parameters λ and k. The density function of this distribution is $g_{\lambda,k}(x) = \lambda \frac{(\lambda x)^{k-1}}{(k-1)!} \exp(-\lambda x)$. The cumulative distribution function of $\Gamma_{\lambda,k}$ is $\mathbf{\Gamma}_{\lambda,k}(x) = 1 - \exp(-\lambda x)\Big(1 + \frac{\lambda x}{1!} + \cdots + \frac{(\lambda x)^i}{i!} + \cdots + \frac{(\lambda x)^{k-1}}{(k-1)!}\Big)$. As we prove below, gamma distribution is how much time one has to wait till k experiments succeed, where an experiment duration distributes according to the exponential distribution.

A random variable X has the *Poisson distribution*, with parameter $\eta > 0$ (which is a discrete distribution) if $\mathbf{Pr}[X = i] = \frac{\eta^i}{i!} e^{-\eta}$.

Lemma 17.1.2 *If $X \sim \mathrm{Exp}(\lambda)$ then $\mathbf{E}[\lambda] = \frac{1}{\lambda}$.*

Proof: $$\int_{x=0}^{\infty} x \cdot \lambda e^{-\lambda x}\, dx = \left[-\frac{1}{\lambda} e^{-\lambda x} - x e^{-\lambda x}\right]_{x=0}^{\infty} = \frac{1}{\lambda}.$$ ■

Lemma 17.1.3 *The following properties hold for the d dimensional Gaussian distribution $N^d(0,1)$:*

(i) The distribution $N^d(0,1)$ is centrally symmetric around the origin.

(ii) If $X \sim N^d(0,1)$ and u is a unit vector, then $X \cdot u \sim N(0,1)$.

(iii) If $X, Y \sim N(0,1)$ are two independent variables, then $Z = X^2 + Y^2$ follows the exponential distribution with parameter $\lambda = \frac{1}{2}$.

(iv) Given k independent variables $X_1, \ldots, X_k$ distributed according to the exponential distribution with parameter λ, then $Y = X_1 + \cdots + X_k$ is distributed according to the Gamma distribution $\Gamma_{\lambda,k}(x)$.

Proof: (i) Let $x = (x_1, \ldots, x_d)$ be a point picked from the Gaussian distribution. The density $\phi_d(x) = \phi(x_1)\phi(x_2) \cdot \phi(x_d)$, where $\phi(x_i)$ is the normal distribution density function, which is $\phi(x_i) = \exp(-x_i^2/2)/\sqrt{2\pi}$. Thus $\phi_d(x) = (2\pi)^{-n/2} \exp(-(x_1^2 \cdots + x_d^2)/2)$. Consider any two points $x, y \in \mathbb{R}^n$, such that $r = \|x\| = \|y\|$. Clearly, $\phi_d(x) = \phi_d(y)$. Namely, any two points of the same distance from the origin, have the same density (i.e., "probability"). As such, the distribution $N^d(0,1)$ is centrally symmetric around the origin.

(ii) Consider $e_1 = (1, 0, \ldots, 0) \in \mathbb{R}^n$. Clearly, $x \cdot e_1 = x_1$, which is distributed $N(0,1)$. Now, by the symmetry of of $N^d(0,1)$, this implies that $x \cdot u$ is distributed $N(0,1)$. Formally, let R be a rotation matrix that maps u to e_1. We know that Rx is distributed $N^d(0,1)$ (since $N^d(0,1)$ is centrally symmetric). Thus $x \cdot u$ has the same distribute as $Rx \cdot Ru$, which has the same distribution as $x \cdot e_1$, which is $N(0,1)$.

(iii) If $X, Y \sim N(0,1)$, and consider the density function $g(x,y) = \frac{1}{2\pi} \exp\Big(-\frac{x^2+y^2}{2}\Big)$ and the associated integral $\int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} g(x,y)\, dx\, dy$. We would like to change the integration variables to $x(r,\alpha) = \sqrt{r}\sin\alpha$ and $y(r,\alpha) = \sqrt{r}\cos\alpha$. The Jacobian of this change of variables is

$$I(r,\alpha) = \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \alpha} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \alpha} \end{vmatrix} = \begin{vmatrix} \frac{\sin\alpha}{2\sqrt{r}} & \sqrt{r}\cos\alpha \\ \frac{\cos\alpha}{2\sqrt{r}} & -\sqrt{r}\sin\alpha \end{vmatrix} = -\frac{1}{2}\big(\sin^2\alpha + \cos^2\alpha\big) = -\frac{1}{2}.$$

As such, we have

$$\begin{aligned} \mathbf{Pr}[Z = z] &= \int_{x^2+y^2=z} \frac{1}{2\pi} \exp\left(-\frac{x^2+y^2}{2}\right) dx\, dy \\ &= \int_{\alpha=0}^{2\pi} \frac{1}{2\pi} \exp\left(-\frac{x(\sqrt{z},\alpha)^2 + y(\sqrt{z},\alpha)^2}{2}\right) \cdot |I(z,\alpha)|\, d\alpha \\ &= \frac{1}{2\pi} \cdot \frac{1}{2} \cdot \int_{\alpha=0}^{2\pi} \exp\Big(-\frac{z}{2}\Big) = \frac{1}{2}\exp\Big(-\frac{z}{2}\Big). \end{aligned}$$

As such, Z has an exponential distribution with $\lambda = 1/2$.

(iv) For $k = 1$ the claim is trivial. Otherwise, let $g_{k-1}(x) = \lambda \frac{(\lambda x)^{k-2}}{(k-2)!} \exp(-\lambda x)$. Observe that

$$\begin{aligned} g_k(t) &= \int_0^t g_{k-1}(t-x) g_1(x)\, dx = \int_0^t \left(\lambda \frac{(\lambda(t-x))^{k-2}}{(k-2)!} \exp(-\lambda(t-x))\right)(\lambda \exp(-\lambda x))\, dx \\ &= \int_0^t \lambda^2 \frac{(\lambda(t-x))^{k-2}}{(k-2)!} \exp(-\lambda t)\, dx \\ &= \lambda \exp(-\lambda t) \int_0^t \lambda \frac{(\lambda x)^{k-2}}{(k-2)!}\, dx = \lambda \exp(-\lambda t) \frac{(\lambda t)^{k-1}}{(k-1)!} = g_k(x). \end{aligned}$$

■

17.1.2 Proof of the Johnson-Lindenstrauss Lemma

Lemma 17.1.4 *Let u be a unit vector in $\mathbb{R}^d$. For any even positive integer k, let $U_1, \ldots, U_k$ be random vectors chosen independently from the d-dimensional Gaussian distribution $N^d(0,1)$. For $X_i = u \cdot U_i$, define $W = W(u) = (X_1, \ldots, X_k)$ and $L = L(u) = \|W\|^2$. Then, for any $\beta > 1$, we have:*

1. $\mathbf{E}[L] = k$.
2. $\mathbf{Pr}[L \geq \beta k] \leq \frac{k+3}{2} \exp\big(-\frac{k}{2}(\beta - (1 + \ln\beta))\big)$.

3. $\mathbf{Pr}[L \leq k/\beta] < O(k) \times \exp(-\frac{k}{2}(\beta^{-1} - (1 - \ln \beta)))$.

Proof: By Lemma 17.1.3 (ii) each X_i is distributed as $N(0,1)$, and $X_1, \ldots, X_k$ are independent. Define $Y_i = X_{2i-1}^2 + X_{2i}^2$, for $i = 1, \ldots, \tau$, where $\tau = k/2$. By Lemma 17.1.3 (iii) Y_i follows the exponential distribution with parameter $\lambda = 1/2$. Let $L = \sum_{i=1}^{\tau} Y_i$. By Lemma 17.1.3 (iv), the variable L follows the Gamma distribution $(k/2, 1/2)$, and its expectation is $\mathbf{E}[L] = \sum_{i=1}^{k/2} \mathbf{E}[Y_i] = \tau \times 2 = k$, since $\mathbf{E}[Y_i] = 2$ by Lemma 17.1.2.

Now, let $\eta = \lambda\beta k = \beta k/2 = \beta\tau$, we have

$$\mathbf{Pr}[L \geq \beta k] = 1 - \mathbf{Pr}[L \leq \beta k] = 1 - \mathbf{\Gamma}_{1/2,\tau}(\beta k) = \sum_{i=0}^{\tau-1} e^{-\eta}\frac{\eta^i}{i!} \leq (\tau+1)e^{-\eta}\frac{\eta^\tau}{\tau!},$$

since $\eta = \beta\tau > \tau$, as $\beta > 1$ and $\mathbf{\Gamma}_{\lambda,k}(x) = 1 - \exp(-\lambda x)\big(1 + \frac{\lambda x}{1!} + \cdots + \frac{(\lambda x)^i}{i!} + \cdots + \frac{(\lambda x)^{k-1}}{(k-1)!}\big)$. Now, since $\tau! \geq (\tau/e)^\tau$, and thus

$$\begin{aligned}
\mathbf{Pr}[L \geq \beta k] &\leq (\tau+1)e^{-\eta}\frac{\eta^\tau}{\tau^\tau/e^\tau} = (\tau+1)e^{-\eta}\Big(\frac{e\eta}{\tau}\Big)^\tau = (\tau+1)e^{-\beta\tau}\left(\frac{e\beta\tau}{\tau}\right)^\tau \\
&= (\tau+1)e^{-\beta\tau} \cdot \exp(\tau \ln(e\beta)) = (\tau+1)\exp(-\tau(\beta - (1 + \ln\beta))) \\
&\leq \frac{k+3}{2}\exp\left(-\frac{k}{2}(\beta - (1+\ln\beta))\right).
\end{aligned}$$

Arguing in a similar fashion, we have, for a large constant $\rho \gg 1$

$$\begin{aligned}
\mathbf{Pr}[L \leq k/\beta] &= \mathbf{\Gamma}_{1/2,\tau}(k/\beta) = 1 - \sum_{i=0}^{\tau-1} e^{-\tau/\beta}\frac{(\tau/\beta)^i}{i!} = e^{-\tau/\beta}\sum_{i=0}^{\infty}\frac{(\tau/\beta)^i}{i!} - \sum_{i=0}^{\tau-1} e^{-\tau/\beta}\frac{(\tau/\beta)^i}{i!} \\
&= \sum_{i=\tau}^{\infty} e^{-\tau/\beta}\frac{(\tau/\beta)^i}{i!} \leq e^{-\tau/\beta}\sum_{i=\tau}^{\infty}\left(\frac{e\tau}{i\beta}\right)^i \\
&= e^{-\tau/\beta}\left[\sum_{i=\tau}^{\rho e\tau/\beta}\left(\frac{e\tau}{i\beta}\right)^i + \sum_{i=\rho e\tau/\beta+1}^{\infty}\left(\frac{e\tau}{i\beta}\right)^i\right]
\end{aligned}$$

The second sum is very small for $\rho \gg 1$ and we bound only the first one. As the sequence $(\frac{e\tau}{i\beta})^i$ is decreasing for $i \geq \tau/\beta$, we can bound the first sum by

$$\frac{\rho e\tau}{\beta} \cdot e^{-\tau/\beta}\left(\frac{e}{\beta}\right)^\tau = O(\tau)\exp\big(-\tau(\beta^{-1} - (1 - \ln\beta))\big).$$

Since $\tau = k/2$, we obtain the desired result. ∎

Next, we show how to interpret the above inequalities in a somewhat more intuitive way. Let $\beta = 1 + \varepsilon$, $\varepsilon > 0$. From Taylor expansion we know that $\ln\beta \leq \varepsilon - \varepsilon^2/2 + \varepsilon^3/3$. By plugging it into the upper bound for $\mathbf{Pr}[L \geq \beta k]$ we get

$$\begin{aligned}
\mathbf{Pr}[L \geq \beta k] &\leq O(k) \times \exp\left(-\frac{k}{2}\big(1 + \varepsilon - 1 - \varepsilon + \varepsilon^2/2 - \varepsilon^3/3\big)\right) \\
&\leq O(k) \times \exp(-\frac{k}{2}(\varepsilon^2/2 - \varepsilon^3/3))
\end{aligned}$$

On the other hand, we also know that $\ln \beta \geq \varepsilon - \varepsilon^2/2$. Therefore

$$\begin{aligned}
\mathbf{Pr}[L \leq k/\beta] &\leq O(k) \times \exp(-\frac{k}{2}(\beta^{-1} - 1 + \varepsilon - \varepsilon^2/2)) \\
&\leq O(k) \times \exp(-\frac{k}{2}(\frac{1}{1+\varepsilon} - 1 + \varepsilon - \varepsilon^2/2)) \\
&\leq O(k) \times \exp(-\frac{k}{2}(\frac{\varepsilon^2}{1+\varepsilon} - \varepsilon^2/2)) \\
&\leq O(k) \times \exp(-\frac{k}{2} \cdot \frac{\varepsilon^2 - \varepsilon^3}{2(1+\varepsilon)})
\end{aligned}$$

Thus, the probability that a given unit vector gets distorted by more than $(1+\varepsilon)$ in any direction[②] grows roughly as $\exp(-k\varepsilon^2/4)$, for small $\varepsilon > 0$. Therefore, if we are given a set P of n points in l_2, we can set k to roughly $8\ln(n)/\varepsilon^2$ and make sure that with non-zero probability we obtain projection which does not distort distances[②] between *any* two different points from P by more than $(1+\varepsilon)$ in each direction.

Theorem 17.1.5 *Given a set P of n points in $\mathbb{R}^d$, and parameter ε, one can compute a random projection R into $k = 8\varepsilon^{-2}\ln n$ dimensions, such that the distances between points are roughtly preserved. Formally, with constant probablity, for any $p, q \in P$, we have*

$$(1-\varepsilon)\|p-q\| \leq \|R(p) - R(q)\| \leq \|p-q\|.$$

The probability of success improves to high probability, if we use, say, $k = 10\varepsilon^{-2}\ln n$ dimensions.

17.2 Bibliographical notes

The probability review of Section 17.1.1 can be found in Feller [Fel71]. The proof of the Johnson-Lindenstrauss lemma of Section 17.1.2 is due to Indyk and Motwani [IM98]. The original proof of the Johnson-Linenstrauss lemma is from [JL84].

It exposes the fact that the Johnson-Lindenstrauss lemma is no more than yet another instance of the concentration of mass phenomena (i.e., like the Chernoff inequality).

Interestingly, it is enough to pick each entry in the dimension reducing matrix randomly out of $-1, 0, 1$. This requires more involved proof [Ach01]. This is useful when one care about storing this dimension reduction transformation efficiently.

Magen [Mag01] observed that in fact the Johnson-Lindenstrauss 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 17.3.1 is taken from there.

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 axises. 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), which 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].

[②] Note that this implies distortion $(1+\varepsilon)^2$ if we require the mapping to be a contraction.

[②] In fact, this statement holds even for the *square* of the distances.

17.3 Exercises

Exercise 17.3.1 [10 Points] 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.]

Chapter 18

Finite Metric Spaces and Partitions

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

18.1 Finite Metric Spaces

Definition 18.1.1 A *metric space* is a pair $(\mathcal{X}, \mathbf{d})$ where $\mathcal{X}$ is a set and $\mathbf{d} : \mathcal{X} \times \mathcal{X} \to [0, \infty)$ is a *metric*, satisfying the following axioms: (i) $\mathbf{d}(x, y) = 0$ iff $x = y$, (ii) $\mathbf{d}(x, y) = \mathbf{d}(y, x)$, and (iii) $\mathbf{d}(x, y) + \mathbf{d}(y, z) \geq \mathbf{d}(x, z)$ (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 $\mathbf{d}$ can be specified by $\binom{n}{2}$ real numbers. Alternatively, one can think about $(\mathcal{X}, \mathbf{d})$ 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 $\mathbf{d}_G(x, y)$ be the length of the shortest path between x and y in G. It is easy to verify that $(\mathcal{X}, \mathbf{d}_G)$ is a finite metric space. As such if the graph G is sparse, it provides a compact representation to the finite space $(\mathcal{X}, \mathbf{d}_G)$.

Definition 18.1.2 Let $(\mathcal{X}, d)$ be an n-point metric space. We denote the *open ball* of radius r about $x \in \mathcal{X}$, by $\mathbf{b}(x, r) = \left\{ y \in \mathcal{X} \;\middle|\; \mathbf{d}(x, y) < r \right\}$.

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 18.1.3 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 18.7.3 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 18.1.4 Let $(\mathcal{X}, \mathbf{d}_\mathcal{X})$ and $(Y, \mathbf{d}_Y)$ be metric spaces. A mapping $f : \mathcal{X} \to Y$ is called an *embedding*, and is *C-Lipschitz* if $\mathbf{d}_Y(f(x), f(y)) \leq C \cdot \mathbf{d}_\mathcal{X}(x, y)$ for all $x, y \in \mathcal{X}$. The mapping f is

called *K-bi-Lipschitz* if there exists a $C > 0$ such that

$$CK^{-1} \cdot \mathbf{d}_{\mathcal{X}}(x, y) \leq \mathbf{d}_Y(f(x), f(y)) \leq C \cdot \mathbf{d}_{\mathcal{X}}(x, y),$$

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 $c_Y(\mathcal{X})$.

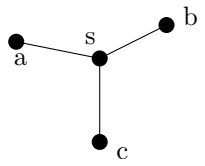
There are several powerful results in this vain, that show the existence of embeddings with low distortion that would be presented:

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 $k = O(\varepsilon^{-2} \log n)$.

18.2 Examples

What is distortion? When considering a mapping $f : \mathcal{X} \to \mathbb{R}^d$ of a metric space $(\mathcal{X}, \mathbf{d})$ 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 $\mathbf{d}(x, y) = \|x - y\|$. As such, we have $\operatorname{dist}(f) = \max_{x,y} \frac{\|x-y\|}{\mathbf{d}(x,y)}$.

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 $(V, \mathbf{d}_G)$ 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 $\mathbf{d}_G(a, b) = 2$. 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].

18.2.1 Hierarchical Tree Metrics

The following metric is quite useful in practice, and nicely demonstrate why algorithmically finite metric spaces are useful.

Definition 18.2.1 *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 $\Delta_{\operatorname{lca}(x,y)}$, where $\operatorname{lca}(x, y)$ 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 $\mathcal{H}$. 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 $\mathcal{H}$. Since $\mathcal{H}$ 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.

18.2.2 Clustering

One natural problem we might want to solve on a graph (i.e., finite metric space) $(\mathcal{X}, \mathbf{d})$ 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 $\nu_C(\mathcal{X}, \mathbf{d}) = \sum_{p \in \mathcal{X}} \mathbf{d}(p, C)$ is minimized, where $\mathbf{d}(p, C) = \min_{c \in C} \mathbf{d}(p, c)$ is the distance of p 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 $(\mathcal{X}, \mathbf{d})$ 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 $(\mathcal{X}, \mathbf{d})$ is a HST. Is not too hard to prove the following lemma. See Exercise 18.7.1.

Lemma 18.2.2 *Let $(\mathcal{X}, \mathbf{d})$ 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 $\mathcal{H}$, 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 $\mathcal{H}$.

18.3 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 $\mathcal{D}_{\mathcal{X}}$.

18.3.1 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.

18.3.2 Properties

Lemma 18.3.1 *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 $\mathbf{d}(x, w_k) < R - t$ then $\mathbf{b}(w_k, R)$ contains $\mathbf{b}(x,t)$ and as such $\mathcal{E}_k$ can not happen. Similarly, if $\mathbf{d}(x, w_k) > R + t$ 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 $R - t \leq \mathbf{d}(x, w_k) \leq R + t$. Namely, if $\mathcal{E}_k$ happen then $R \in I_k$. Namely, $\mathbf{Pr}[\mathcal{E}_k] = \mathbf{Pr}[\mathcal{E}_k \cap (R \in I_k)] = \mathbf{Pr}[R \in I_k] \cdot \mathbf{Pr}[\mathcal{E}_k \mid R \in I_k]$. 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 $\mathbf{Pr}[\mathcal{E}_k \mid R \in I_k]$, 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, $\mathbf{Pr}[\mathcal{E}_k \mid R \in I_k]$ 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 $\mathbf{Pr}[\mathcal{E}_k \mid R \in I_k] \leq 1/k$.

We are now ready for the kill. Indeed,

$$\begin{aligned}
\mathbf{Pr}[\mathbf{b}(x,t) \not\subseteq P(x)] &= \sum_{k=1}^{b'} \mathbf{Pr}[\mathcal{E}_k] = \sum_{k=a+1}^{b'} \mathbf{Pr}[\mathcal{E}_k] = \sum_{k=a+1}^{b'} \Pr[R \in I_k] \cdot \Pr[\mathcal{E}_k \mid R \in I_k] \\
&\leq \sum_{k=a+1}^{b'} \frac{8t}{\Delta} \cdot \frac{1}{k} \leq \frac{8t}{\Delta} \ln \frac{b'}{a} \leq \frac{8t}{\Delta} \ln \frac{b}{a},
\end{aligned}$$

since $\sum_{k=a+1}^{b} \frac{1}{k} \leq \int_a^b \frac{dx}{x} = \ln \frac{b}{a}$ and $b' \leq b$. ∎

18.4 Probabilistic embedding into trees

In this section, given n-point finite metric $(\mathcal{X}, \mathbf{d})$. 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* $\mathbf{d}_T(x, y)$. We constructed the tree T such that distances never shrink; i.e. $\mathbf{d}(x, y) \leq \mathbf{d}_T(x, y)$. The *probabilistic distortion* of this embedding is $\max_{x,y} \mathbf{E}\left[\frac{\mathbf{d}_T(x,y)}{\mathbf{d}(x,y)}\right]$. Somewhat surprisingly, one can find such an embedding with logarithmic probabilistic distortion.

Theorem 18.4.1 *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 18.3.1. 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 $\mathbf{d}_T(x, y) \leq 2^{\operatorname{level}(v)}$. 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 $\mathbf{b} = \mathbf{b}(x, \mathbf{d}(x, y))$ 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, $\mathbf{d}_T(x, y) \leq 2^{\operatorname{level}(w)}$.

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, $\mathbf{d}_T(x, y) \leq \sum Y_i 2^i$.

Let $t = \mathbf{d}(x, y)$ and $j = \lfloor \lg \mathbf{d}(x, y) \rfloor$, and $n_i = \left|\mathbf{b}(x, 2^i)\right|$ for $i = 0, \ldots, -\infty$. We have

$$\mathbf{E}[\mathbf{d}_T(x, y)] \leq \sum_{i=j}^{0} \mathbf{E}[Y_i]\, 2^i \leq \sum_{i=j}^{0} 2^i \, \mathbf{Pr}\left[\mathcal{E}_i \cap \overline{\mathcal{E}_{i-1}} \cap \overline{\mathcal{E}_{i-1}} \cdots \overline{\mathcal{E}_0}\right] \leq \sum_{i=j}^{0} 2^i \cdot \frac{8t}{2^i} \ln \frac{n_i}{n_{i-3}},$$

by Lemma 18.3.1. Thus,

$$\mathbf{E}[\mathbf{d}_T(x, y)] \leq 8t \ln \left(\prod_{i=j}^{0} \frac{n_i}{n_{i-3}} \right) \leq 8t \ln(n_0 \cdot n_1 \cdot n_2) \leq 24t \ln n.$$

It thus follows, that the expected distortion for x and y is $\leq 24 \ln n$. ∎

18.4.1 Application: approximation algorithm for k-median clustering

Let $(\mathcal{X}, \mathbf{d})$ 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 $\nu_{C_{\text{opt}}}(\mathcal{X}, \mathbf{d})$ is minimized, see Section 18.2.2. To this end, we randomly embed $(\mathcal{X}, \mathbf{d})$ into a HST $\mathcal{H}$ using Theorem 18.4.1. Next, using Lemma 18.2.2, we compute the optimal k-median clustering of $\mathcal{H}$. 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 18.4.2 *Let $(\mathcal{X}, \mathbf{d})$ 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 $(\mathcal{X}, \mathbf{d})$.*

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 $\overline{c}(p)$ denote the closest point in C_{opt} to p according to $\mathbf{d}$, 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 $\mathcal{H}$ be the HST used by the algorithm. We have

$$\beta = \nu_C(\mathcal{X}, \mathbf{d}) \leq \nu_C(\mathcal{X}, \mathbf{d}_{\mathcal{H}}) \leq \nu_{C_{\text{opt}}}(\mathcal{X}, \mathbf{d}_{\mathcal{H}}) \leq \sum_{p \in \mathcal{X}} \mathbf{d}_{\mathcal{H}}(p, C_{\text{opt}}) \leq \sum_{p \in \mathcal{X}} \mathbf{d}_{\mathcal{H}}(p, \overline{c}(p)).$$

Thus, in expectation we have

$$\begin{aligned} \mathbf{E}[\beta] &= \mathbf{E}\left[\sum_{p \in \mathcal{X}} \mathbf{d}_{\mathcal{H}}(p, \overline{c}(p))\right] = \sum_{p \in \mathcal{X}} \mathbf{E}[\mathbf{d}_{\mathcal{H}}(p, \overline{c}(p))] = \sum_{p \in \mathcal{X}} O(\mathbf{d}(p, \overline{c}(p)) \log n) \\ &= O\left((\log n) \sum_{p \in \mathcal{X}} \mathbf{d}(p, \overline{c}(p))\right) = O\big(\nu_{C_{\text{opt}}}(\mathcal{X}, \mathbf{d}) \log n\big), \end{aligned}$$

by linearity of expectation and Theorem 18.4.1. ■

18.5 Embedding any metric space into Euclidean space

Lemma 18.5.1 *Let $(\mathcal{X}, \mathbf{d})$ be a metric, and let $Y \subset \mathcal{X}$. Consider the mapping $f : \mathcal{X} \to \mathbb{R}$, where $f(x) = \mathbf{d}(x, Y) = \min_{y \in Y} \mathbf{d}(x, y)$. Then for any $x, y \in \mathcal{X}$, we have $|f(x) - f(y)| \leq \mathbf{d}(x, y)$. Namely f is nonexpansive.*

Proof: Indeed, let x' and y' be the closet points of Y, to x and y, respectively. Observe that $f(x) = \mathbf{d}(x, x') \leq \mathbf{d}(x, y') \leq \mathbf{d}(x, y) + \mathbf{d}(y, y') = \mathbf{d}(x, y) + f(y)$ by the triangle inequality. Thus, $f(x) - f(y) \leq \mathbf{d}(x, y)$. By symmetry, we have $f(y) - f(x) \leq \mathbf{d}(x, y)$. Thus, $|f(x) - f(y)| \leq \mathbf{d}(x, y)$. ■

18.5.1 The bounded spread case

Let $(\mathcal{X}, \mathbf{d})$ be a n-point metric. The *spread* of $\mathcal{X}$, denoted by $\Phi(\mathcal{X}) = \frac{\operatorname{diam}(\mathcal{X})}{\min_{x,y \in \mathcal{X}, x \neq y} \mathbf{d}(x,y)}$, is the ratio between the diameter of $\mathcal{X}$ and the distance between the closest pair of points.

Theorem 18.5.2 *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 18.4.1, 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 $f_{i,j}(x) = \mathbf{d}(x, \mathcal{X} \setminus V_{i,j}) = \min_{v \in \mathcal{X} \setminus V_{i,j}} \mathbf{d}(x, v)$, 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 $F(x) = \big(f_{0,1}(x), f_{0,2}(x), \ldots, f_{0,\beta}(x), f_{1,1}(x), f_{0,2}(x), \ldots, f_{1,\beta}(x), \ldots, f_{m,1}(x), f_{m,2}(x), \ldots, f_{m,\beta}(x)\big)$.

Next, consider two points $x, y \in \mathcal{X}$, with distance $\phi = \mathbf{d}(x, y)$. Let k be an integer such that $r_u \leq \phi/2 \leq r_{u+1}$. 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 18.3.1, we have

$$\mathbf{Pr}[\mathbf{b}(x,\rho) \nsubseteq P_{u,j}(x)] \le \frac{8\rho}{r_u} \ln n \le \frac{\phi}{8 r_u} \le 1/2.$$

Thus,

$$\begin{aligned} \mathbf{Pr}[\mathcal{E}_j] &= \mathbf{Pr}\Big[\Big(\mathbf{b}(x,\rho) \subseteq P_{u,j}(x)\Big) \cap (x \in V_{u,j}) \cap (y \notin V_{u,j})\Big] \\ &= \mathbf{Pr}[\mathbf{b}(x,\rho) \subseteq P_{u,j}(x)] \cdot \mathbf{Pr}[x \in V_{u,j}] \cdot \mathbf{Pr}[y \notin V_{u,j}] \ge 1/8, \end{aligned}$$

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

$$\|F(x) - F(y)\| \ge \sqrt{\sum_{j=1}^{\beta} (f_{u,j}(x) - f_{u,j}(y))^2} \ge \sqrt{\rho^2 \frac{\beta}{16}} = \sqrt{\beta}\frac{\rho}{4} = \phi \cdot \frac{\sqrt{\beta}}{256 \ln n}.$$

On the other hand, $|f_{i,j}(x) - f_{i,j}(y)| \le \mathbf{d}(x,y) = \phi \le 64\rho \ln n$. 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 $\Big[\phi, \phi \cdot \sqrt{\alpha\beta} \cdot \frac{256 \ln n}{\sqrt{\beta}}\Big]$. 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. ∎

18.5.2 The unbounded spread case

Our next task, is to extend Theorem 18.5.2 to the case of unbounded spread. Indeed, let $(\mathcal{X}, d)$ be a n-point metric, such that $\operatorname{diam}(\mathcal{X}) \le 1/2$. 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 $\le r_i/n^2$). 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

$\mathbf{d}(x,y) \leq r_i/n^2$. Consider a connected component $C \in G_i$. For any two points $x, y \in C$, we have $\mathbf{d}(x,y) \leq n(r_i/n^2) \leq r_i/n$. 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 $\mathbf{d}_i(C,C') = \mathbf{d}(C,C') = \min_{c\in C, c'\in C'} \mathbf{d}(c,c')$.

It is easy to verify that $(\mathcal{X}_i, \mathbf{d}_i)$ is a metric space (see Exercise 18.7.2). Furthermore, we can naturally embed $(\mathcal{X}, \mathbf{d})$ into $(\mathcal{X}_i, \mathbf{d}_i)$ by mapping a point $x \in \mathcal{X}$ to its connected components in $\mathcal{X}_i$. Essentially $(\mathcal{X}_i, \mathbf{d}_i)$ is a snapped version of the metric $(\mathcal{X}, d)$, with the advantage that $\Phi((\mathcal{X}, \mathbf{d}_i)) = O(n^2)$. 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 $f_{i,j}(x) = \min(\mathbf{d}_i(x, \mathcal{X}\setminus V_{i,j}), 2r_i)$. 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

$$|f_{i,j}(x) - f_{i,j}(y)| \leq |\mathbf{d}_i(x, V_{i,j}) - \mathbf{d}_i(y, V_{i,j})| \leq \mathbf{d}_i(x,y) \leq \mathbf{d}(x,y),$$

as a simple case analysis[3] shows.

For a pair $x, y \in \mathcal{X}$, and let $\phi = \mathbf{d}(x,y)$. To see that $F(\cdot)$ is the required embedding (up to scaling), observe that, by the same argumentation of Theorem 18.5.2, we have that with high probability

$$\|F(x) - F(y)\| \geq \phi \cdot \frac{\sqrt{\beta}}{256 \ln n}.$$

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 $r_i/n^2 \leq \mathbf{d}(x,y) \leq r_i n^2$. 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:

[3] Indeed, if $f_{i,j}(x) < \mathbf{d}_i(x, V_{i,j})$ and $f_{i,j}(y) < \mathbf{d}_i(x, V_{i,j})$ then $f_{i,j}(x) = 2r_i$ and $f_{i,j}(y) = 2r_i$, which implies the above inequality. If $f_{i,j}(x) = \mathbf{d}_i(x, V_{i,j})$ and $f_{i,j}(y) = \mathbf{d}_i(x, V_{i,j})$ then the inequality trivially holds. The other option is handled in a similar fashion.

Theorem 18.5.3 (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.

18.6 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 18.3.1 (which is originally from [FRT03]) is taken from [KLMN04]. The proof of Theorem 18.5.3 is by Gupta [Gup00].

A good exposition of metric spaces is available in Matoušek [Mat02].

18.7 Exercises

Exercise 18.7.1 (Clustering for HST.) Let $(\mathcal{X}, \mathbf{d})$ 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 18.7.2 (Partition induced metric.)

(a) Give a counter example to the following claim: Let $(\mathcal{X}, \mathbf{d})$ be a metric space, and let P be a partition of $\mathcal{X}$. Then, the pair $(P, \mathbf{d}')$ is a metric, where $\mathbf{d}'(C, C') = \mathbf{d}(C, C') = \min_{x \in C, y \in C'} \mathbf{d}(x, y)$ and $C, C' \in P$.

(b) Let $(\mathcal{X}, \mathbf{d})$ be a n-point metric space, and consider the set $U = \left\{ i \;\middle|\; 2^i \leq \mathbf{d}(x, y) \leq 2^{i+1}, \text{ for } x, y \in \mathcal{X} \right\}$. Prove that $|U| = O(n)$. Namely, there are only n different resolutions that "matter" for a finite metric space.

Exercise 18.7.3 (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 $\mathbf{d}(f(x), f(y)) \geq (1-\varepsilon)\mathbf{d}(x, y)$.

(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 $\mathbf{d}(x, y) \geq (1-\varepsilon)\mathrm{diam}(P)$, where $\mathrm{diam}(P) = \max_{z,w \in P} \mathbf{d}(z, w)$ 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 $\mathrm{diam}(Q) \geq (1-\varepsilon/2)\mathrm{diam}(P)$. (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 19

VC Dimension, ε-nets and ε-approximation

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

> "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."
>
> – Momo, Emile Ajar

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.

19.1 VC Dimension

Definition 19.1.1 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 $\mathrm{VC}(S)$, is the maximum cardinality of a shattered subset of X. It there are arbitrarily large shattered subsets then $\mathrm{VC}(S) = \infty$.

19.1.1 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 19.1 (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 $\mathrm{VC}(S) = \infty$. 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 19.1.2 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 19.1.3 (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. ■

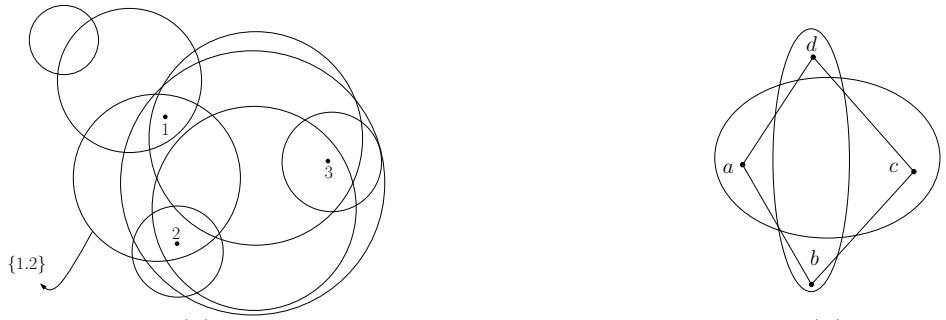
In particular, this implies that if a set Q of $d+2$ points is being shattered by S, we can partition this set Q into two disjoint sets A and B such that $\mathcal{CH}(A) \cap \mathcal{CH}(B) \neq \emptyset$. It should now be clear that any halfspace h^+ containing all the points of A, must also contain 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 $\mathrm{VC}(S) < d+2$. It is also easy to verify that the regular simplex with $d+1$ vertices is being shattered by S. Thus, $\mathrm{VC}(S) = d+1$.

19.2 VC-Dimensions and the number of different ranges

Let

$$g(d, n) = \sum_{i=0}^{d} \binom{n}{i}.$$

Note that for all $n, d \geq 1$, $g(d, n) = g(d, n-1) + g(d-1, n-1)$



(a) (b)

Figure 19.1: Disks in the plane can shatter three points, but not four.

Lemma 19.2.1 (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 19.2.1, to a finite subset of X, we get:

Corollary 19.2.2 *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 19.2.3 *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 $\mathrm{VC}\left(\widehat{S}\right) = O((d + d') \log(d + d'))$*

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 $n \leq O((d + d') \log(d + d'))$. ∎

19.3 On ε-nets and ε-sampling

Definition 19.3.1 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 19.3.2 *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 19.3.3 (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*

$$m \geq \max\left(\frac{4}{\varepsilon} \log \frac{2}{\delta}, \frac{8d}{\varepsilon} \log \frac{8d}{\varepsilon}\right). \tag{19.1}$$

Then N is an ε-net for A with probability at least $1 - \delta$.

19.4 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. (19.1). 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 19.4.1 $\mathbf{Pr}[E_2] \leq \mathbf{Pr}[E_1] \leq 2\,\mathbf{Pr}[E_2]$.

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 $\mathbf{Pr}\Big[E_2 \;\Big|\; E_1\Big] = \mathbf{Pr}[E_2 \cap E_1] / \mathbf{Pr}[E_1] = \mathbf{Pr}[E_2] / \mathbf{Pr}[E_1]$. It is thus enough to show that $\mathbf{Pr}\Big[E_2 \;\Big|\; E_1\Big] \geq 1/2$.

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 3.3.3),

$$\mathbf{Pr}\Big[|r \cap T| < \frac{\varepsilon m}{2}\Big] \leq \mathbf{Pr}\Big[\big||r \cap T| - \varepsilon m\big| > \frac{\varepsilon m}{2}\Big] \mathbf{Pr}\left[\big||r \cap T| - \varepsilon m\big| > \frac{\sqrt{\varepsilon m}}{2}\sqrt{\varepsilon m}\right] \leq \frac{4}{\varepsilon m} \leq \frac{1}{2},$$

by Eq. (19.1). Thus, $\mathbf{Pr}[E_2] / \mathbf{Pr}[E_1] = \mathbf{Pr}\big[|r \cap T| \geq \frac{\varepsilon n}{2}\big] = 1 - \mathbf{Pr}\big[|r \cap T| < \frac{\varepsilon m}{2}\big] \geq \frac{1}{2}$. ∎

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 19.4.2 $\mathbf{Pr}[E_2] \leq \mathbf{Pr}[E_2'] \leq g(d, 2m)2^{-\varepsilon m/2}$.

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,

$$\mathbf{Pr}\big[E_2'\big] = \sum_Z \mathbf{Pr}\Big[E_2' \;\Big|\; Z\Big] \mathbf{Pr}[Z].$$

Thus, from this point on, we fix the set Z, and we bound $\mathbf{Pr}\Big[E_2' \;\Big|\; Z\Big]$.

It is now enough to consider the ranges in the projection space $P_R(Z)$. By Lemma 19.2.1, 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

$$\begin{aligned}
\mathbf{Pr}[E_r] &\leq \mathbf{Pr}\Big[r \cap N = \emptyset \;\Big|\; |r \cap (N \cup T)| > \frac{\varepsilon m}{2}\Big] = \frac{\binom{2m-k}{m}}{\binom{2m}{m}} \\
&= \frac{(2m-k)(2m-k-1)\cdots(m-k+1)}{2m(2m-1)\cdots(m+1)} \\
&= \frac{m(m-1)\cdots(m-k+1)}{2m(2m-1)\cdots(2m-k+1)} \leq 2^{-k} \leq 2^{-\varepsilon m/2}.
\end{aligned}$$

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 $\mathbf{Pr}[E_2'] \leq g(d, 2m)2^{-\varepsilon m/2}$. ∎

Proof of Theorem 19.3.3. By Lemma 19.4.1 and Lemma 19.4.2, 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. (19.1), then $2g(d, 2m)2^{-\varepsilon m/2} \leq \delta$. One can verify that this inequality is implied by Eq. (19.1).

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. (19.1)), 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. ∎

19.5 Exercises

Exercise 19.5.1 (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 19.3.2 using (B).

Exercise 19.5.2 (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 $p \in A$, let $\mathcal{R}(p)$ denote the set of all the ranges of $P_R(A)$ that contains is, and consider the *dual range space* $\mathsf{D} = \left(P_R(A), \left\{ \mathcal{R}(p) \;\middle|\; p \in A \right\}\right)$.

Prove that the VC dimension of D is at most 2^d.

Exercise 19.5.3 (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 19.5.4 (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 $p \in A$, let $\mathcal{R}(p)$ denote the set of all the ranges of $P_R(A)$ that contains is, and consider the *dual range space* $\mathsf{D} = \left(P_R(A), \left\{ \mathcal{R}(p) \;\middle|\; p \in A \right\}\right)$.

Prove that the VC dimension of D is at most 2^d.

Exercise 19.5.5 (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 $p \in A$ has an associated weight w_p, 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_p 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 19.5.6 *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 $P = NP$.)

19.6 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 19.5.1 is from Anthony and Bartlett [AB99].

Chapter 20

Approximate Max Cut

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

20.1 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:

$$\begin{aligned} (P) \quad \text{Maximize} \quad & \tfrac{1}{2} \textstyle\sum_{i<j} w_{ij}(1 - \langle v_i, v_j \rangle) \\ \text{subject to:} \quad & v_i \in \mathbb{S}^{(n)} \qquad \forall i \in V, \end{aligned}$$

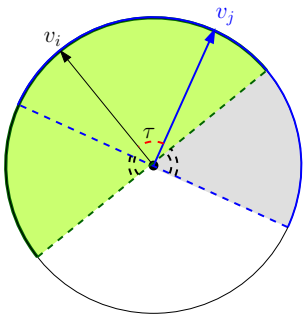
where $\mathbb{S}^{(n)}$ is the n dimensional unit sphere in $\mathbb{R}^{n+1}$. This is an instance of semi-definite programming, which is a special case of convex programming, which can be solved in polynomial time (solved here means approximated within arbitrary constant in polynomial time). Observe 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 $\overline{S}$.

20.1.1 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 20.1.1 *We have* $\mathbf{Pr}[\operatorname{sign}(\langle v_i, \vec{r}\,\rangle) \neq \operatorname{sign}(\langle v_j, \vec{r}\,\rangle)] = \frac{1}{\pi}\arccos(\langle v_i, v_j\rangle)$.



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, $\operatorname{sign}(\langle v_i, \vec{r}\,\rangle) \neq \operatorname{sign}(\langle v_j, \vec{r}\,\rangle)$ 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 $\langle v_i, v_j\rangle = \cos(\tau)$, where $\tau = \angle v_i v_j$. Thus, $\mathbf{Pr}[\operatorname{sign}(\langle v_i, \vec{r}\,\rangle) \neq \operatorname{sign}(\langle v_j, \vec{r}\,\rangle)] = 2\tau/(2\pi) = \frac{1}{\pi}\cdot\arccos(\langle v_i, v_j\rangle)$, as claimed. ■

Theorem 20.1.2 *Let W be the random variable which is the weight of the cut generated by the algorithm. We have*

$$\mathbf{E}[W] = \frac{1}{\pi}\sum_{i<j} w_{ij}\arccos(\langle v_i, v_j\rangle)\,.$$

Proof: Let X_{ij} be an indicator variable which is 1 if ij is in the cut. We have $\mathbf{E}[X_{ij}] = \mathbf{Pr}[\operatorname{sign}(\langle v_i, \vec{r}\,\rangle) \neq \operatorname{sign}(\langle v_j, \vec{r}\,\rangle)] = \frac{1}{\pi}\arccos(\langle v_i, v_j\rangle)$, by Lemma 20.1.1.

Clearly, $W = \sum_{i<j} w_{ij}X_{ij}$, and by linearity of expectation, we have

$$\mathbf{E}[W] = \sum_{i<j} w_{ij}\,\mathbf{E}[X_{ij}] = \sum_{i<j}\arccos(\langle v_i, v_j\rangle)\,.$$

■

Lemma 20.1.3 *For $-1 \le y \le 1$, we have* $\dfrac{\arccos(y)}{\pi} \ge \alpha\cdot\dfrac{1}{2}(1-y)$, *where* $\alpha = \min\limits_{0\le\psi\le\pi}\dfrac{2}{\pi}\dfrac{\psi}{1-\cos(\psi)}$.

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 20.1.4 $\alpha > 0.87856$.

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 20.1.5 *The above algorithm computes in expectation a cut of size* $\alpha\mathrm{Opt} \ge 0.87856\mathrm{Opt}$, *where* Opt *is the weight of the maximal cut.*

Proof: Consider the optimal solution to (P), and lets its value be $\gamma \ge \mathrm{Opt}$. We have

$$\mathbf{E}[W] = \frac{1}{\pi}\sum_{i<j} w_{ij}\arccos(\langle v_i, v_j\rangle) \ge \sum_{i<j} w_{ij}\alpha\frac{1}{2}(1-\langle v_i, v_j\rangle) = \alpha\gamma \ge \alpha\mathrm{Opt},$$

by Lemma 20.1.3. ■

20.2 Semi-definite programming

Let us define a variable $x_{ij} = \langle v_i, v_j \rangle$, 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

$$\begin{aligned} (SD) \quad \text{Maximize} \quad & \tfrac{1}{2} \textstyle\sum_{i<j} w_{ij}(1 - x_{ij}) \\ x_{ii} = 1 \quad & \text{for } i = 1, \ldots, n \\ \text{subject to:} \quad & \left(x_{ij}\right)_{i=1,\ldots,n, j=1,\ldots,n} \text{ is semi-definite.} \end{aligned}$$

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 20.2.1 *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, GLS88] for more details.

20.3 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 $\mathbf{P} = \mathbf{NP}$). 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 21

Entropy, Randomness, and Information

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

> "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.

21.1 Entropy

Definition 21.1.1 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 $\mathbb{H}(') (p) = -\lg p + \lg(1-p) = \lg \frac{1-p}{p}$ and $\mathbb{H}'(p) = \frac{p}{1-p} \cdot \left(-\frac{1}{p^2}\right) = -\frac{1}{p(1-p)}$. Thus, $\mathbb{H}''(p) \leq 0$, 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 21.1.2 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 21.1.3 *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 21.1.4 *Suppose that nq is integer in the range $[0,n]$. Then $\dfrac{2^{n\mathbb{H}(q)}}{n+1} \le \binom{n}{nq} \le 2^{n\mathbb{H}(q)}$.*

Proof: This trivially holds if $q=0$ or $q=1$, so assume $0<q<1$. We know that $\binom{n}{nq} q^{nq}(1-q)^{n-nq} \le (q+(1-q))^n = 1$. As such, since $q^{-nq}(1-q)^{-(1-q)n} = 2^{n(-q\lg q-(1-q)\lg(1-q))} = 2^{n\mathbb{H}(q)}$, we have

$$\binom{n}{nq} \le q^{-nq}(1-q)^{-(1-q)n} = 2^{n\mathbb{H}(q)}.$$

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, $\Delta_k \ge 0$ when $k \ge nq+q-1$, and $\Delta_k < 0$ otherwise. Namely, $\mu(k) < \mu(k+1)$, for $k < nq$, and $\mu(k) \ge \mu(k+1)$ for $k \ge nq$. 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 $\mu(nq) = \binom{n}{nq} q^{nq}(1-q)^{n-nq} \ge \frac{1}{n+1}$, which implies

$$\binom{n}{nq} \ge \frac{1}{n+1} q^{-nq}(1-q)^{-(n-nq)} = \frac{1}{n+1} 2^{n\mathbb{H}(q)}.$$

■

Lemma 21.1.4 can be extended to handle non-integer values of q. This is straightforward, and we omit the easy details.

Corollary 21.1.5 *We have:* *(i)* $q \in [0,1/2] \Rightarrow \binom{n}{\lfloor nq \rfloor} \le 2^{n\mathbb{H}(q)}$. *(ii)* $q \in [1/2,1]$ $\binom{n}{\lceil nq \rceil} \le 2^{n\mathbb{H}(q)}$. *(iii)* $q \in [1/2,1] \Rightarrow \frac{2^{n\mathbb{H}(q)}}{n+1} \le \binom{n}{\lfloor nq \rfloor}$. *(iv)* $q \in [0,1/2] \Rightarrow \frac{2^{n\mathbb{H}(q)}}{n+1} \le \binom{n}{\lceil nq \rceil}$.

The bounds of Lemma 21.1.4 and Corollary 21.1.5 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 21.1.2, a fact that we already know from Lemma 21.1.3.

21.1.1 Extracting randomness

Entropy can be interpreted as the amount of unbiased random coin flips can be extracted from a random variable.

Definition 21.1.6 An extraction function $\texttt{Ext}$ takes as input the value of a random variable X and outputs a sequence of bits y, such that $\mathbf{Pr}\Big[\texttt{Ext}(X) = y \;\Big|\; |y| = k\Big] = \frac{1}{2^k}$, 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 21.1.6 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 21.1.6 holds for this extraction function, since $\mathbf{Pr}\Big[\texttt{Ext}(X) = 00 \;\Big|\; |\texttt{Ext}(X)| = 2\Big] = \frac{1}{4}$, as required. This scheme can be of course extracted for any range.

Theorem 21.1.7 *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 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 21.1.6, 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 $u = \big\lfloor \lg(m - 2^k) \big\rfloor < k$. 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, $\frac{m - 2^k}{m} \leq \frac{2^{u+1}}{2^{u+1} + 2^k}$. Thus,

$$\begin{aligned}
\mathbb{H}(X) &\geq \frac{2^k}{m} k + \frac{m - 2^k}{m}\Big(\Big\lfloor \lg(m - 2^k) \Big\rfloor - 1\Big) = k + \frac{m - 2^k}{m}(u - k - 1) \\
&\geq k + \frac{2^{u+1}}{2^{u+1} + 2^k}(u - k - 1) = k - \frac{2^{u+1}}{2^{u+1} + 2^k}(1 + k - u)\,.
\end{aligned}$$

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

$$\mathbb{H}(X) \geq k - \frac{2^{u+1}}{2^k}(1 + k - u) \geq k - \frac{k - u + 1}{2^{k-u-1}} \geq k - 1,$$

since $\frac{2+i}{2^i} \leq 1$ for $i \geq 2$. ∎

Theorem 21.1.8 *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 21.1.7. 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 21.1.7, 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 21.1.5 (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 $\mathbf{Pr}\Big[|Z-np| \geq \frac{\varepsilon}{p}pn\Big] \leq 2\exp\left(-\frac{np}{4}\left(\frac{\varepsilon}{p}\right)^2\right) = 2\exp\left(-\frac{n\varepsilon^2}{4p}\right)$, 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 $2\exp\left(-\frac{n\varepsilon^2}{4p}\right) \leq \frac{\delta}{10}$. 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,

$$\mathbf{E}[B] = \sum_x \mathbf{Pr}[X=x]\,|\mathtt{Ext}(x)| \leq \sum_x \mathbf{Pr}[X=x] \lg \frac{1}{\mathbf{Pr}[X=x]} = \mathbb{H}(X).$$

■

21.2 Bibliographical Notes

The presentation here follows [MU05, Sec. 9.1-Sec 9.3].

Chapter 22

Entropy II

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

> 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.

22.1 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 22.1.1 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 22.1.2 *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 22.1.3 *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 $\mathbf{Pr}[X < (p-\varepsilon)n] \leq \exp\bigl(-n\varepsilon^2/2p\bigr)$.

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 21.1.5. 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

$$(n+1)\exp\bigl(-n\varepsilon^2/2p\bigr) + (1 + \lfloor \lg n + n\mathbb{H}(p-\varepsilon)\rfloor) \leq (1+\delta)n\mathbb{H}(p)\,,$$

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 21.1.5. Now, any input string with less than τ heads has lower probability to be generated (since $1-p < p$. As such, Lemma 22.1.2 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 $\mathbf{Pr}[X \leq \tau] \geq 1 - \exp\bigl(-n\varepsilon^2/12p\bigr)$. Implying that an optimal compresser outputs on average at least $\bigl(1 - \exp\bigl(-n\varepsilon^2/12p\bigr)\bigr)\mu$. Again, by carefully choosing ε and n sufficiently large, we have that the average output length of an optimal compressor is at least $(1-\delta)n\mathbb{H}(p)$. ∎

22.2 Bibliographical Notes

The presentation here follows [MU05, Sec. 9.1-Sec 9.3].

Chapter 23

Entropy III - Shannon's Theorem

598 - Class notes for Randomized Algorithms
Sariel Har-Peled
December 1, 2005

> 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.

23.1 Coding: Shannon's Theorem

Definition 23.1.1 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.

Definition 23.1.2 A (k, n) *encoding function* $\mathtt{Enc} : \{0,1\}^k \to \{0,1\}^n$ takes as input a sequence of k bits and outputs a sequence of n bits. A (k, n) decoding function $\mathtt{Dec} : \{0,1\}^n \to \{0,1\}^k$ takes as input a sequence of n bits and outputs a sequence of k bits.

Theorem 23.1.3 (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 $k \geq n(1 - \mathbb{H}(p) + 1)$ such that the probability of decoding correctly is at least γ for a k-bit input message chosen uniformly at random.

23.1.1 The encoder/decoder

We will provide encoding and decoding functions for the case $k \leq n(1 - \mathbb{H}(p) - \delta)$ by using the probabilistic method. For $i = 0, \ldots, 2^{k-1}$, let X_i be a random string of length 2^n. The encoder would map the binary string of length k corresponding to number i to the binary string X_i.

The decoder when receiving a message

23.2 Bibliographical Notes

The presentation here follows [MU05, Sec. 9.1-Sec 9.3].

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.

[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, New York, NY, USA, 2004. ACM Press.

[Aro98] S. Arora. Polynomial time approximation schemes for euclidean tsp and other geometric problems. *J. Assoc. Comput. Mach.*, 45(5):753–782, Sep 1998.

[AS00] N. Alon and J. H. Spencer. *The probabilistic method*. Wiley Inter-Science, 2nd edition, 2000.

[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 metrices by tree metrics. In *Proc. 30th Annu. ACM Sympos. Theory Comput.*, pages 161–168. ACM Press, 1998.

[Bol98] B. Bollobas. *Modern Graph Theory*. Springer-Verlag, 1998.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge, 2004.

[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 *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 8–16. Society for Industrial and Applied Mathematics, 2001.

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press / McGraw-Hill, Cambridge, Mass., 2001.

[Fel71] W. Feller. *An Introduction to Probability Theory and its Applications*, volume II. John Wiley & Sons, NY, 1971.

[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.

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin Heidelberg, 2nd edition, 1988. 2nd edition 1994.

[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.

[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.

[JL84] W. B. Johnson and J. Lindenstrauss. Extensions of lipschitz mapping into hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[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.

[KLMN04] R. Krauthgamer, J. R. Lee, M. Mendel, and A. Naor. Measured descent: A new embedding method for finite metric spaces. In *Proc. 45th Annu. IEEE Sympos. Found. Comput. Sci.*, page to appear, 2004.

[Mag01] A. Magen. Dimensionality reductions that preserve volumes and distance to affine spaces, and its algorithmic applications. Submitted to STOC 2002, 2001.

[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 Pr, 1998.

[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, New York, NY, 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, New York, NY, 1976.

[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 Science Publishers B. V. North-Holland, Amsterdam, 2000.

[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.

Index