Analysis and Design of Algorithms

Wenda Chu

Contents

1 Probability Bounds

Markov Inequality

For a nonnegative random variable X, for any $k > 0$,

$$\Pr(X \geq k) \leq \frac{\mathbb{E}(X)}{k} \tag{1}$$

If we have the variance of a random variable, we can apply more detailed analyses:

Chebyshev Inequality

Let X be a random variable with $\mathbb{E}(X) = \mu$ and $Var(X) = \sigma^2$, then

$$\Pr(|X - \mu| \geq t) \leq \frac{\sigma^2}{t^2}. \tag{2}$$

The proof is straightforward after applying Markov inequality to $|X - \mu|$.

Chernoff Bound

Let W_i be independent random variables. $W = W_1 + \cdots + W_n$. Chernoff bound takes advantage of higher-order conditions of independent variables. For Bernoulli random variables with probability p,

$$\Pr(W \geq (1+\delta)\mathbb{E}[W]) \leq \left(\frac{e^\delta}{(1+\delta)^\delta}\right)^\mu \tag{3}$$

For $X_i \in [0, 1]$

$$[Pr(X \geq (1+\delta)\mu)] \leq \exp^{-\frac{\delta^2\mu}{3}},\ [Pr(X \leq (1-\delta)\mu)] \leq \exp^{-\frac{\delta^2\mu}{2}},\ \Pr[X \geq R] \leq \exp^{-R},\ \text{for } R \geq 5\mu \tag{4}$$

The proof is by considering the random variable e^{tW}, which contains higher-order information of W.

1.1 An Example: Median Finding

Goal:

- Input a set S of integers
- Find the median $m \in S$: at least $\lceil \frac{n}{2} \rceil$ elements $\leq m$ and at least $\lfloor \frac{n}{2} \rfloor + 1$ elements $\geq m$.

We want to find $d, u \in S$, such that $d \le m \le u$, with $C = \{s \in S, d \le s \le u\}$ small enough. Specifically, we do

- Count number of elements in S larger than u, and smaller than d.
- Sort C in $O(|C| \log |C|) = o(n)$ time.
- Deduce the median.

How to choose C? We first select a multi-set R of $n^{3/4}$ elements iid from S, and sort R.

- Let d be the $\frac{1}{2}n^{3/4} - \sqrt{n}$ smallest element and u be the $\frac{1}{2}n^{3/4} + \sqrt{n}$ smallest one.
- Compute $C, L_d = \{s \in S, s < d\}, L_u = \{s \in S, s > u\}$. If $|L_d|$ or $|L_u| \ge \frac{n}{2}$, we fail. If $|C| > 4n^{3/4}$, we fail.
- Otherwise, we sort C and return the $\frac{n}{2} - |L_d|$ smallest element in C.

The runtime is $O(n)$ indeed, but we have to bound the probability of failure.

$Y_1 = \mathbb{1}\{|\{r \in R : r \le m\}| < \frac{1}{2}n^{3/4} - \sqrt{n}\}$. If $Y_1 = 0$, then $|L_d| \le \frac{1}{2}$.

$Y_2 = \mathbb{1}\{|\{r \in R : r \ge m\}| < \frac{1}{2}n^{3/4} - \sqrt{n}\}$. If $Y_2 = 0$ then $|L_u| \le \frac{1}{2}$.

$Y_3 = \mathbb{1}\{|C| > 4n^{3/4}\}$.

$\Pr(Y_1 = 1) \le \frac{1}{4n^{1/4}}$ by applying Chebyshev bound. (Could get better bound if using Chernoff bound?)

$\Pr(Y_3 = 1) \le \frac{1}{2n^{1/4}}$. Either $2n^{3/4}$ elements larger than m or $2n^{3/4}$ elements smaller than m. Suppose the first happens, then R has at least $\frac{1}{2}n^{3/4} - \sqrt{n}$ samples that are among the $\frac{1}{2}n - 2n^{3/4}$ largest elements in S. (Consider the rightmost interval in R.) Let $R = \{a_i\}_{i=1}^{n^{3/4}}$, and $X_i = 1$ if a_i is larger than the $\frac{1}{2}n + 2n^{3/4}$ element. $\mathbb{E}(X_i) = \frac{1}{2} - 2n^{-1/4}$. $\mathbb{E}(\sum X_i) = \frac{1}{2}n^{3/4} - 2\sqrt{n}$. Then

$$\Pr\left(\sum_{i=1}^{n^{3/4}} X_i \ge \frac{1}{2}n^{3/4} - \sqrt{n}\right) \le \frac{1}{2}n^{-1/4} \tag{5}$$

By union bound, we fail with probability less than $n^{-1/4}$.

2 Flows

A graph $G = (V, E)$. Compute its maximal flow.

Ford-Fulkerson algorithm $O(mF)$, where $F = C \cdot \Delta_s$, C is the largest flow capacity.

Edmands Karp $O(m^2 n)$.

Theorem 2.1. *The maximal flow from s to t equals the min-cut that separates s, t.*

Note: Recursively reducing paths from s to t gives the maximal flow. In the end s and t will be disconnected, which naturally yields the solution for a Min-Cut.

2.1 Applications

Bipartite Matching Using max-flow algorithms to find maximum matching for bipartite matching.

Theorem 2.2 (Hall)**.** *A bipartite graph has a perfect matching iff $\forall S \subseteq L$ or $S \subseteq R$, $|\Gamma(S)| \geq |S|$, where $\Gamma(S)$ is the set of neighbors of S.*

Image Segmentation Let f_i be the cost of assigning pixel i to the foreground. b_i is the cost of assigning it to the background. And s_{ij} is the cost of separating i and j.

$$\min_S \text{cost(S)} = \sum_{i \in S} f_i + \sum_{i \in S^C} b_i + \sum_{i \in S, j \in S^C} s_{ij} \tag{6}$$

This can be translated into a s, t Min-cut problem.

2.2 Min-Cut

Instead of finding an (s, t)- Min-Cut, we find an (S, S^C) partition of vertices V.

Karger's randomized algorithm

Repeatedly contract two nodes together, randomly, until only two supernodes are left. The runtime would be $O(n^2)$. ($O(n)$ for each edge contraction.)

This randomized algorithm returns the correct answer with probability

$$p \geq \frac{1}{\binom{n}{2}} \tag{7}$$

Repeat $n^2 \log n$ time, so the overall runtime would be $O(n^4 \log n)$ time to achieve a $O(1/n^c)$ failure probability.

Proof Sketch. When there are t supernodes left, we have

$$\Pr\Big[e \in \delta(S, S^C)\Big] \leq \frac{2}{t} \tag{8}$$

This is because each node must have degree $\geq k$ if the minimum cut is k.

Then we multiply the probability together, which yields the final bound. □

Corollary 2.3. *There are at most $\binom{n}{2}$ number of min-cuts in a graph.*

2.3 Complexity classes of BPP: randomized poly time

A randomized algorithm that runs in $poly(n)$ time, with correct probability $\geq \frac{2}{3}$ for any input x.

Boosting: Run an algorithm in BPP n times independently and take a majority vote, then by Chernoff bound,

$$\Pr[Correct] \geq 1 - \exp\Big(-\frac{n}{48}\Big). \tag{9}$$

3 Streaming Algorithms

Inputs coming in consistently. Cannot store all data. Need an online algorithm to deal with incoming data.

Setup.

Inputs: $\sigma = (s_1, s_2, \dots s_m)$, where $s_j \in \{1, \dots, n\}$.

Assumption: The length of input $Len(m)$ is known, and the range is known.

Let $f_i = |\{1 \le j \le m\} \mid s_j = i|$.

$F_0 = \sum_{i=1}^{n} (f_i)^0 =$ # of distinct elements, where $0^0 = 0$.

$F_1 = \sum_i f_i = m$, space complexity: $\log m$.

What about $F_2 = \sum_i f_i^2$? Can we find an algorithm with logarithmic space complexity?

3.1 Deterministic algorithm fails for F_2

Assume there exists a deterministic algorithm to compute F_2 exactly. We construct a stream:

$$\Big((1, x_1), \dots, (n, x_n)\Big), \text{ where } x \in \{0,1\}^n \tag{10}$$

Run the algorithm. Assume the memory at the end is $m(x) \in \{0,1\}^S$.

We then initialize the memory to $m(x)$ and feed and $(i, 0)$ to the algorithm. There are two possibilities:

- F_2 increases by 1, if $x_i = 1$,
- F_2 increases by $2^2 - 1 = 3$, if $x_i = 0$.

It's possible to completely recover x by $m(x)$, so $m(x)$ must have at least 2^n possible values. The space complexity is $\Omega(n)$.

3.2 Randomized algorithm for F_2

Hash function $h : \{1, \dots, n\} \to \{1, -1\}$.

- Initialize: set $c = 0$
- Process s_j: Add $h(s_j)$ to c.

- Return c^2.

Space: $-m \le c \le m$, so it takes $O(\log m)$ space.

We show that $Z = c^2$ is an unbiased estimation of F_2. For $j \in \{1, \ldots, n\}$, let $Y_j = h(j) \in \{\pm 1\}$.

$$Z = (f_1 Y_1 + \cdots + f_n Y_n)^2 \tag{11}$$

so

$$\mathbb{E}[Z] = \sum_{i=1}^{n} f_i \mathbb{E}[Y_i]^2 + \sum_{i \neq j} f_i f_j \mathbb{E}(Y_i)\mathbb{E}(Y_j) = \sum_{i=1}^{n} f_i^2 = F_2. \tag{12}$$

We now bound the variance of Z to apply tail bounds.

$$\mathbb{E}[Z^2] = \sum_{i,j,k,l} f_i f_j f_k f_l \mathbb{E}[Y_i]\mathbb{E}[Y_j]\mathbb{E}[Y_k]\mathbb{E}[Y_l] \tag{13}$$

$$= \sum_{i=1}^{n} f_i^4 \mathbb{E}[Y_i^4] + 3 \sum_{i \neq j} f_i^2 f_j^2 \mathbb{E}[Y_i^2]\mathbb{E}[Y_j^2] \tag{14}$$

$$= F_4 + 3(F_2^2 - F_4), \tag{15}$$

so $Var(Z) = 2F_2^2 - 2F_4 \le 2F_2^2$.

$$\Pr[|Z - F_2| \ge \epsilon F_2] \le \frac{2F_2^2}{\epsilon^2 F_2^2} = \frac{2}{\epsilon^2}. \tag{16}$$

This bound is great, but not so satisfactory. Actually, using the median of means trick, we can get an exponential boost, i.e., a $(1 \pm \epsilon)$ w.p. at least $(1 - \delta)$ using space $O(\frac{1}{\epsilon^2} \log(1/\delta) \log m)$.

3.3 Derandomization

Definition 3.1. *A family of random varibles $(x_1, \ldots, x_N)$ is called k-wise independent if for every k-tuple $(i_1, \ldots, i_k) \in \{1, \ldots, N\}^k$, $(x_{i_1}, \ldots, x_{i_k})$ are independent.*

Specifically, when they are N-wise independent, it is called fully independent.

Remark. The randomized algorithm above does not need full independence. In fact, all we need is a 4-wise independence (required in the analysis of variance). This is also the reason why we can construct such a hash function in $O(\log n)$ but not $O(n)$.

Definition 3.2. *A family H of hash functions $h : A \to B$ is called k-wise independent if for any distinct points $x_1, \ldots, x_k \in A$ and $i_1, \ldots, i_k \in B$,*

$$\Pr_{h \in H}\Big(h(x_1) = i_1, \ldots, h(x_k) = i_k\Big) = \frac{1}{|B|^k}. \tag{17}$$

3.4 Construction of k-wise independent hash functions

Let $A = B = \mathbb{Z}_p := \{0, 1 \ldots, p-1\}$, where p is a prime.

$$H_2 = \{f_{a,b} : x \mapsto ax + b \bmod p, (a,b) \in \mathbb{Z}_p^2\}. \tag{18}$$

Then

$$\Pr_{a,b}(ax_1 + b = i_1 \wedge ax_2 + b = i_2) = \Pr_{a,b}(a = \frac{i_2 - i_1}{x_2 - x_1} \wedge b = i_2 - x_2 a) = \frac{1}{p^2}. \tag{19}$$

So this is a 2-wise independent hash function family. We get $h(0), \ldots, h(p)$ p random variables with $2 \log p$ space.

We can extend to $x \mapsto \sum_{i=0}^{k-1} a_i x^i \bmod p$ to create a k-wise independent hash function family, with space complexity of $k \log p$.

3.5 Algorithm for F_0.

Function $h : \{1, \ldots, n\} \to \{1, \ldots, n\}$.

- Initialize $c = 0$
- Process s_j: Let z be the largest power of 2 that divides $h(s_j)$. If $z > c$, $c \leftarrow z$.
- Return $2^{c+1/2}$.

Intuition:

If the stream has d distinct elements, there is a good chance that one of the d values of $h(s_j)$ will be divisible by d.

If there are no more than d distinct elements, it's unlikely that an $h(s_j)$ has more than $\log d$ zeros in a row.

$j = \{1, \ldots, n\}, r \geq 0$, $X_{r,j}$: indicator random variable that $h(j)$ is divisible by 2^r.

$Y_r = \sum_{j:f_j>0} X_{r,j}$. So if $Y_r = 0$, no elements have r or more zeros, so $c \leq r - 1$.

$$\mathbb{E}[X_{r,j}] = \Pr\big(h(s_j) \text{ has r zeros}\big) = \frac{1}{2^r}. \tag{20}$$

so

$$\mathbb{E}[Y_r] = \frac{F_0}{2^r}, Var(Y_r) = \sum_{j:f_j>0} Var(X_{r,j}) = \frac{F_0}{2^r}. \tag{21}$$

$$\Pr[Y_r > 0] \leq \frac{F_0}{2^r}, \Pr[Y_r = 0] \leq \frac{2^r}{F_0}. \tag{22}$$

4 Johnson-Lindenstrauss Lemma

JL Lemma states that we can map data that lie in a high-dimensional space to a low-dimensional one, preserving some of the structures in data.

Lemma 1 (JL Lemma)**.** *Let S be a set of n points in $\mathbb{R}^d$. For some dimension $k = O(\frac{\ln n}{\epsilon^2})$, there exists a matrix $A \in \mathbb{R}^{k\times d}$ such that $\forall u, v, \in S$,*

$$(1-\epsilon)\|u-v\|^2 \leq \|Au - Av\|^2 \leq (1+\epsilon)\|u-v\|^2. \tag{23}$$

4.1 Norm Preservation

Lemma 2. *For any integer $d > 0, 0, \epsilon, \delta < 1$, and integer $k > \frac{4\log\left(\frac{2}{\delta}\right)}{\epsilon^2-\epsilon^3}$, there exists a distribution on $k \times d$ real matrices, such that for any $x \in \mathbb{R}^d$,*

$$\Pr\left((1-\epsilon)\|x\|^2 \leq \|Ax\|^2 \leq (1+\epsilon)\|x\|^2\right) > 1-\delta. \tag{24}$$

Proof of JL Lemma by lemma 2. Let $\delta = \frac{1}{n^3}$. We apply union bound over u, v, and the proof is done. □

In fact $A_{ij} \sim \mathcal{N}(0, \sigma^2)$ is the construction we need for this lemma.

Proof of lemma 2. Random choose the entry of $A \in \mathbb{R}^{k\times d}$. $A_{ij} \sim \mathcal{N}(0, \frac{1}{k})$.

$$(Ax)_1 = \sum_{i=1}^{d} A_{1,i}x_i \tag{25}$$

This is a random variable $\mathcal{N}(0, \frac{\|x\|^2}{k})$.

$$\mathbb{E}[\|Ax\|^2] = k \cdot \mathbb{E}\|(Ax_1)^2\| = k\frac{\|x\|^2}{k} = \|x\|^2. \tag{26}$$

Now let $Z_i \sim \mathcal{N}(0, \frac{\sum_{i=1}^{d} x_i^2}{k})$

$$\Pr\left(\|Ax\|^2 \geq (1+\epsilon)\|x\|^2\right) = \Pr\left(\sum_{i=1}^{k} Z_i^2 \geq (1+\epsilon)\|x\|^2\right) \tag{27}$$

$$= \Pr\left(\sum_{i=1}^{k} Y_i^2 \geq (1+\epsilon)k\right), \text{ where } Z_i \sim \mathcal{N}(0,1) \tag{28}$$

$$= \Pr\left(\exp\left(\sum_{i=1}^{k} Y_i^2\right) \geq e^{(1+\epsilon)k}\right) \tag{29}$$

$$\leq \frac{\prod_{i=1}^{k} \mathbb{E}(e^{tY_i^2})}{e^{tk(1+\epsilon)}} \tag{30}$$

$$= \frac{1}{(1-2t)^{k/2} e^{tk(1+\epsilon)}} \tag{31}$$

The last equality holds because $\mathbb{E}(e^{tX^2}) = \frac{1}{\sqrt{1-2t}}$ for $X \sim \mathcal{N}(0,1), t < \frac{1}{2}$. Then applying the fact that $1 - x \leq e^{-x} \leq 1 - x + \frac{x^2}{2}$ and let $t = \frac{\epsilon}{2(1+\epsilon)}$, $k > \frac{4\log(2/\delta)}{\epsilon^2 - \epsilon^3}$,

$$\Pr\big(\|Ax\|^2 \geq (1+\epsilon)\|x\|^2\big) \leq \frac{\delta}{2}. \tag{32}$$

□

4.2 Packing

In $\mathbb{R}^k$ we can only have k perfectly orthogonal vectors. Let $e_1, \ldots, e_n \in \mathbb{R}^n$ be unit orthogonal vectors. After applying JL lemma, we have $e_1', \ldots, e_n' \in \mathbb{R}^k$. Then $\langle e_i, e_j \rangle \leq \epsilon$ for $i \neq j$.

This means that for $k \geq 1$, $0 < \epsilon < 1$, $\exists n = e^{\Omega(\epsilon^2 k)}$ near-orthogonal vectors in $\mathbb{R}^k$.

Lower bound. Let e_i', e_j' be two ϵ-near orthogonal vectors in $\mathbb{R}^k$, then two unit balls with centers e_i', e_j' with radius $\frac{1}{2}\sqrt{2-2\epsilon}$ are disjoint. However, the total volume of the ball with radius $1 + \frac{\sqrt{2-2\epsilon}}{2}$ is $O(e^{O(\epsilon d)})$.

With a more fine-grained analysis, we can get a tighter lower bound $O(e^{\Omega(\epsilon^2 d)})$

4.3 Application: Approximate Nearest Neighbor

$x_1, \ldots, x_n \in \mathbb{R}^d$. Given a query $y \in \mathbb{R}^d$, find the closest x_i to y.

If d is small, we have efficient data structures with quasi-linear in space and in $\log n$ time.

If d is large, we need space $O(nd)$ and query $O(nd)$; or $n^{O(d)}$ in space and query time $O(d \log n)$.

Definition 4.1 (ϵ-Approximate NN)**.** *Given y, an ϵ-approximate nearest neighbor is an output i such that $\|y - x_i\| \leq (1+\epsilon) \min_j \|y - x_j\|$.*

Assume the closest point to y has distance 1.

- Preprocess:

  Fix a grid on $\mathbb{R}^d$ with side length $\epsilon/\sqrt{d}$. Let G_i be the set of grid cells that contain at least one point at a distance at most 1 from x_i.

  Store (c, i) for each $c \in G_i$.

- The space consumption for each i:

$$\frac{2^d / d^{d/2}}{(\frac{\epsilon}{\sqrt{d}})^d} \sim \epsilon^{-d}. \tag{33}$$

- The query time is $O(d)$.

How to do better using JL lemma?

In the preprocessing phase, project x_i to a space with dimension $d' = O(\frac{\log n}{\epsilon^2})$.

The space cost is then

$$n^{O(\log(1/\epsilon)/\epsilon^2)} \tag{34}$$

Time cost of projection during a query is $d\epsilon^{-2} \log n$ time, so the time is

$$d\epsilon^2 \log n \tag{35}$$

5 Linear Programming

Standard form

$$\max_x c^T x$$
$$\text{s.t. } Ax \leq b, x \geq 0. \tag{36}$$

Dual problem

$$\min_y b^T y \tag{37}$$
$$\text{s.t. } A^T y \geq c, y \geq 0. \tag{38}$$

5.1 Uncapacitated Facility Location

Let D be the set of clients and F be a set of facilities. There is a close f_i to open facility i.

Goal: choose a subset of $F' \subseteq F$ that minimizes $\sum_{i \in F'} f_i + \sum_{j \in D} \min_{i \in F'} c_{ij}$.

c_{ij} should satisfy the triangle quality.

We find an equivalent integer programming:

$$\begin{aligned} \min_{x,y} & \sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} c_{ij} x_{ij} \\ \text{s.t. } & \sum_{i \in F} x_{ij} = 1, \forall j \in D, \\ & x_{ij} \leq y_i, \forall i, j, \\ & x_{ij}, y_i \in \{0, 1\}. \end{aligned} \tag{39}$$

This problem is in general NP-hard. Hence we relax the integer constraint to $x_{ij}, y_i \in [0, 1]$.

How to round a solution of the linear program to an integer solution?

Fix an order $j_1, \ldots, j_k$ for clients. Let $N(j) = \{i : x_{ij} > 0\}$. We say j and j' are close if $N(j') \cap N(j) \neq \varnothing$.

Dual problem:

$$\max_v \sum_{j \in D} v_j \tag{40}$$
$$\text{s.t. } v_j \leq c_{ij} + w_{ij}, \forall i \in F, j \in D. \tag{41}$$
$$\sum_{j \in D} w_{ij} \leq f_i, \forall i \in F \tag{42}$$

$$w_{ij} \geq 0. \tag{43}$$

Greedy deterministic LP rounding

- Solve Primal and dual LPs to obtain x_{ij}^* and v_j^*
- Order the set D of users, according to increasing v_j^*.

  For a user in D,

- Let j be the user of smallest v_j^*, Let $i \in N(j)$ minimize f_i. Open facility i, assign it to user j. Remove j from D.
- $\forall j' D, N_j' \cap N_j \neq \varnothing$, assign j' to i. Remove j' from D.

Analysis.

Note that $\sum_{i \in N(j_k)} x_{ij_k^*} = 1$, we have

$$\begin{aligned}
\sum_k f_{i_k} &= \sum_k f_{i_k} \sum_{i \in N(j_k)} x_{ij_k}^* && (44)\\
&\leq \sum_k \sum_{i \in N(j_k)} f_i x_{ij_k}^* && (f_{i_k} \leq f_i, \forall I \in N(j_k)) \\
&\leq \sum_k \sum_{i \in N(j_k)} f_i y_i^* && \text{(LP constraint)} \\
&\leq \sum_i f_i y_i^*. && (45)
\end{aligned}$$

Suppose a user ℓ is assigned to facility i_k. Let $h \in N(l) \cap N(j_k)$.

$$\begin{aligned}
c_{i_k l} &\leq c_{ij_k} + c_{hj_k} + c_{hl} && (46)\\
&\leq v_{j_k}^* + v_{j_k}^* + v_l^*. && (47)\\
&\leq 3v_l^*. && (48)
\end{aligned}$$

This is because complementary slackness implies

$$\text{if } x_{ij}^* > 0 \rightarrow v_j^* = c_{ij} + w_{ij} \geq c_{ij}. \tag{49}$$

Therefore,

$$\sum_j c_{g(i)j} \leq 3 \sum_j v_j^* \leq 3Z_{LP}^* \tag{50}$$

$$cost = facility + clientconnection \leq \sum_i f_i y_i^* + 3Z_{LP}^* \leq 4Z_{IP}^*. \tag{51}$$

6 Semi-definite programming

We consider the Max-Cut problem on a graph $G = (V, E)$. We want to choose a subset $U \subseteq V$, to maximize $|E(U, V \backslash U)|$, i.e., to maximize the number of edges connecting a node in U to a node outside U.

Integer program for Max-Cut

$$\begin{aligned} &\max_{(u,v)\in E} z_{uv} \\ \text{s.t. } &z_{uv} \leq x_u + x_v \\ &z_{uv} \leq (1 - x_u) + (1 - x_v) \\ &z_{uv}, x_u \in \{0, 1\} \end{aligned} \tag{52}$$

Randomly cutting the graph gives us a $1/2$ approximation.

$$\max_{(u,v)\in E} \frac{1 - x_u x_v}{2} \tag{53}$$

$$s.t.\ x_i \in -1, 1, \forall i \in V, \ . \tag{54}$$

6.1 SDP definition

Definition 6.1 (Positive semi-definite matrix)**.** *Let $X \in \mathbb{R}^{n\times n}$ be symmetric. X is positive semi-definite or $X \succeq 0$, if the following equivalent statements are true:*

- $\forall a \in \mathbb{R}^n$, $a^T x a \geq 0$,
- $X = B^T B$ *for some* B.
- *All eigenvalues of X are nonnegative.*

Definition 6.2 (SDP standard form)**.**

$$\begin{aligned} &\min_{X\in\mathbb{R}^{n\times n}} tr(C^T X) \\ \text{s.t. } &tr(A_i^T X) = b_i, X \succeq 0. \end{aligned} \tag{55}$$

Dual problem

$$\max_y b^T y \tag{56}$$

$$\sum_{i=1}^{m} y_i A_i + S = C \tag{57}$$

$$S \succeq 0 \tag{58}$$

To derive this, remember that we want

$$\langle C, X \rangle \geq \sum_i y_i \langle A_i, X \rangle = b^T y. \tag{59}$$

so $\langle C - \sum_i y_i A_i, X \rangle \geq 0$ for all $X \succeq 0$, which means $C - \sum_i y_i A_i \succeq 0$.

6.2 Strong duality

Theorem 6.3 (Slater's condition)**.** *Strong duality holds for SDP problems if the feasible region has an interior point.*

6.3 Max Cut

Now we relax the max cut problem to a SDP problem.

$$\max_{u,v \in E} 1 - \langle x_u, x_v \rangle \tag{60}$$

$$\text{s.t. } x_i \in \mathbb{S}^{n-1}. \tag{61}$$

Then let A be the adjacency matrix and $X = \sum_{ij} x_i x_j^T$ be the gram matrix. Then this is a SDP.

Theorem 6.4 (This is an SDP!)**.** *Let A be the adjacency matrix, $A_{ij} = 1$ if $(i,j) \in E$ and $A_{ij} = 0$ otherwise. We define X to be the gram matrix $X_{ij} = x_i^T x_j$. Then $MaxCut'$ is equivalent to the following SDP*

$$\begin{aligned} \min \langle X, A \rangle &= tr(A^T X) \\ \text{s.t.} X &\succeq 0, X \in \mathbb{S}_n \\ X_{ii} &= 1, \forall i. \end{aligned} \tag{62}$$

Theorem 6.5 (Goemans-Williamson)**.** $\alpha_{gw} = \min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \frac{\theta}{1 - \cos\theta} \sim 0.87856$.

Solve the SDP to obtain X. Then $X_{uv} = \langle x_u, x_v \rangle$. Find x_u by decomposing $X = U^T U$. Choose a vector a uniformly on the sphere $\mathbb{S}^{n-1}$.

Set $x_u = sign(\langle a, x_u \rangle)$.

6.4 Quadratic programs

$$OPT = \max_{x_i, y_i \in \{\pm 1\}} \sum_{ij} A_{ij} x_i y_j. \tag{63}$$

This problem is hard. Specifically, it is harder than Max-cut, i.e., $\max_{x_i \in \{\pm\}} \frac{1}{2}(1 - x_i x_j)$. To see this, let A_{ii} be extremely large, so $x_i = y_i$ must hold. Then Max-cut is reduced to this problem.

Relaxtion

$$OPT' = max_{\|u_i\|=1, \|v_i\|=1} A_{ij} \langle u_i, v_j \rangle = A_{ij} z_{ij}. \tag{64}$$

Let $B = \begin{pmatrix} C & Z \\ Z^T & D \end{pmatrix} \succeq 0$ where C, D has diagonal entries to be 0. Since $B = WW^T$, for $W = \begin{pmatrix} u \\ v \end{pmatrix}$, $B \succeq 0$.

This is an SDP program.

Theorem 6.6 (Grothendieck's inequality)**.**

$$OPT \leq OPT' \leq \frac{\pi}{2 \ln\left(1 + \sqrt{2}\right)} OPT. \tag{65}$$

We first try the hyperplane rounding we used in Goemans-Williamson,

SDP contributes: $A_{ij} \langle u_i, v_j \rangle = A_{ij} \cos \theta_{ij}$.

Rounding contributes: $A_{ij} \cdot \Big(-1 \cdot (\theta_{ij}/\pi) 1 \cdot (1 - \theta_{ij}/\pi) \Big) = \frac{2 A_{ij} \theta_{ij}}{\pi}$

However, A_{ij} could be negative, making this bound infeasible.

Fortunately, we have the following lemma, which directly proves the theorem. To see this, we first apply this transformation in the lemma and then do the hyperplane rounding.

Lemma 3. *For any unit vectors $u_1, \ldots, u_m$, $v_1, \ldots, v_n$, then there exists a new set of unit vectors $u'_1, \ldots, u'_m, v'_1, \ldots, v'_n$, s.t.*

$$\mathbb{E}_a[sign(a^T u'_i) sign(a^T v'_j)] = \frac{2}{\pi} \ln\left(1 + \sqrt{2}\right) \langle u_i, v_j \rangle. \tag{66}$$

Proof. Define tensor product $U^{\otimes 2} = [u_1^2, u_1 u_2, \ldots, u_n^2]$. Let $c = \sinh^{-1} 1 = \ln\left(1 + \sqrt{2}\right)$.

By Taylor's expansion,

$$\sin c\langle u, v\rangle = \sum_{k=0}^{\infty}(-1)^k \frac{c^{2k+1]}}{(2k+1)!}\langle u, v\rangle^{2k+1} \tag{67}$$

□

Let

$$u' = [\sqrt{u}, (-1)\sqrt{\frac{c^3}{3!}}u^{\otimes 3}, \ldots, (-1)^k\sqrt{\frac{c^{2k+1}}{(2k+1)!}}u^{\otimes 2k+1}] \tag{68}$$

and

$$v' = [\sqrt{v}, \sqrt{\frac{c^3}{3!}}v^{\otimes}3, \ldots, \sqrt{\frac{c^{2k+1}}{(2k+1)!}}v^{\otimes 2k+1}] \tag{69}$$

Then $\langle u', v' = \sin(c\langle u, v\rangle)$.

And $\langle u', u' = \sinh(c) = 1$. This is because removing the sign in the Taylor's expansion yields $\sinh(c\langle u, v\rangle)$.

Then

$$\mathbb{E}_a[sign(a^T u)sign(a^T v)] = \frac{2}{\pi}\arcsin\langle u', v'\rangle = \frac{2}{\pi}c\langle u, v\rangle \tag{70}$$

Lavasz theta function

Let $\alpha(G)$ be the size of the largest independent set of G.

Let $\chi(G)$ be the chromatic number of G. And let $\bar{G}$ be the complement of G.

Then

$$\chi(\bar{G}) \geq \alpha(G). \tag{71}$$

We define

$$\theta(G) := \min k, \tag{72}$$

$$\text{s.t. } \langle v_i, v_j\rangle = -\frac{1}{k-1}\ \forall (i,j) \notin E, \tag{73}$$

$$\langle v_i, v_i\rangle = 1. \tag{74}$$

Theorem 6.7. $\alpha(G) \leq \theta(G) \leq \chi(\bar{G})$.

Proof. First $\theta(G) \leq \chi(\bar{G})$.

Fact: $\exists k$ unit vectors $u_1, \dot{,} u_k$ such that $\langle u_i, u_j\rangle = -\frac{1}{k-1}$.

Then any k-coloring of $\bar{G}$ yields $u_1, \ldots, u_k$.

The other direction: $\theta(G) \geq \alpha(G)$. We solve the SDP to obtain v_i. For an independent set, consider $v_1, \ldots, v_s$ of the independent set

$$0 \leq (\sum_{i=1}^{s} v_i^T)(\sum_{i=1}^{s} v_i) \leq \sum_{i=1}^{s} v_i^T v_i + \sum_{i \neq j} v_i^T v_j \tag{75}$$

Then at least one term of $v_i^T v_j \geq -\frac{s}{2}\binom{s}{2} = -\frac{1}{s-1}$. However, $\langle v_i, v_j \rangle = -\frac{1}{\theta(G)-1}$. □

7 Online Algorithms

7.1 Multiplicative Weights

We wish to make predictions over T days, relying on n expects that we have. Define f_i^t to be the loss of expert i, on day t. Without loss of generality, $\|f^t\|)\infty \leq 1$.

Our goal is to minimize the regret

$$Regret = \sum_{t=1}^{T} \langle p^t, f^t \rangle - \min_u \sum_{t=1}^{T} f_i^t. \tag{76}$$

Naive idea: to pick the best expert so far. Then there exists a set of $f^{(t)}$ so that the regret $\sim T(1 - \frac{1}{n})$.

Weighted Majority. Assuming we are making binary choices. Assign each expect i with weight $w_i^{(1)} = 1$. We can make a majority vote and predict

$$x = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}. \tag{77}$$

Assumption: There is an expert who is always correct.

Algorithm: Remove the experts with the wrong predictions. Then each time we make a mistake, we halve the number of experts. Regret = $O(\log n)$. This is basically the idea of the multiplicative weights algorithm.

If we remove the assumption that there is an expert always correct, we can restart after removing all experts. The bound is $O((M+1)\log n)$ if the best expert makes M mistakes.

The idea of multiplicative weight on binary outcomes

- Set $w_i^{(1)} = 1$.
- After observing the outcome at day t, set $w_i^{(t+1)} = \frac{w_i^{(t)}}{2}$ if i makes a wrong prediction.

Analysis: let $\Phi^{(t)} = \sum_{i=1}^{n} w_i^{(t)}$. Then

$$\Phi^{(T+1)} \geq (\frac{1}{2})^{\#i's \text{ mistakes}} \tag{78}$$

$$\Phi^{(t+1)} \leq \frac{3}{4}\Phi^{(t)}, \text{ (when the algorithm makes a mistake)} \tag{79}$$

Therefore,

$$\#\text{ mistakes} \leq \frac{1}{\log 4/3}(\#i's \text{ mistakes} + \log n) \tag{80}$$

However, this does not guarantee a bound on the regret, due to the $\log 4/3$ constant. We now state the full version of multiplicative weights.

Multiplicative weights algorithm

- Set $w_i^{(1)} = 1$.
- For $t = 1, \ldots, T$,
  - Follow expert i with probability $p_i^t = \frac{w_i^t}{\sum_i w_i^t}$.
  - $w_i^{(t+1)} \leftarrow w_i^{(t)}(1 - \epsilon f_i^t), \forall i$.

Theorem 7.1. *If $0 < \epsilon \leq \frac{1}{2}$, then the multiplicative weight gives the following bound on regret*

$$\textit{regret} \leq \frac{\log n}{\epsilon} + \epsilon T. \tag{81}$$

If T is known, $\epsilon = \sqrt{\frac{\log n}{T}}$, regret $\leq 2\sqrt{T \log n}$.

Proof. Define a potential function $\Phi^t = \sum_i w_i^{(t)}$. Then

$$\Phi^{(t+1)} = \sum_i w_i^{(t)}(1 - \epsilon f_i^t) = \sum_i p_i^t \Phi^t (1 - \epsilon f_i^t). \tag{82}$$

Therefore,

$$\mathbb{E}\Phi^{(t+1)} = \mathbb{E}\Phi^{(t)}(1 - \epsilon \mathbb{E}\ell_t) \leq \Phi^t \exp(-\epsilon \mathbb{E}\ell_t). \tag{83}$$

Hence $\mathbb{E}\Phi^{(T)} \leq n \exp(-\epsilon \mathbb{E} L_{1:t})$.

On the other hand, let i be any expert,

$$\mathbb{E}\Phi^{(T)} \geq w_i^T \geq \prod_{t=1}^{T} \exp\left(-\epsilon f_i^t - \epsilon^2 (f_i^t)^2\right). \tag{84}$$

Then let i be the optimal expert,

$$\mathbb{E}[L_{1:t}] - OPT \leq \frac{1}{\epsilon}(\log n + \epsilon^2 \sum_{t=1}^{T} (f_i^t)^2) \leq \frac{\log n}{\epsilon} + \epsilon T. \tag{85}$$

□

Remark. When $\|f^t\|_\infty \leq \rho$, regret $\leq \frac{\rho^2 \log n}{\epsilon} + \epsilon T$.

7.2 Application of Multiplicative Weight

Zero-sum Games.

$$v = \min_x \max_y c(x, y) \tag{86}$$

Theorem 7.2 (Von Neumann)**.** *Let x, y be mixed strategies of players A and B, then*

$$\min_x \max_y c(x, y) = \max_y \min_x c(x, y). \tag{87}$$

Another way of phrasing: $\exists (x, y), x, y \geq 0, \sum_i x_i = \sum_i y_i = 1$, then there is a value v,

$$x^T M \geq v, My \leq v, \text{ where } M_{ij} = c(i, j). \tag{88}$$

We try to prove this theorem by multiplicative weights. Let $(p^1, q^1), \ldots, (p^T, q^T)$ be the strategies over T days.

Then

$$-(\sum_{t=1}^{T} \langle p^t, Aq^t \rangle - \max_i \sum_{t=1}^{T} (Aq^t)_i) \leq \frac{\ln m}{\epsilon} + \epsilon T \tag{89}$$

and

$$\sum_{t=1}^{T} \langle q^t, Ap^t \rangle - \max_i \sum_{t=1}^{T} (Ap^t)_i \leq \frac{\ln m}{\epsilon} + \epsilon T \tag{90}$$

Adding them together,

$$\max_i \sum_{t=1}^{T} (Aq^t)_i - \min_j \sum_{t=1}^{T} ({p^t}^T A)_j \leq \frac{2 \ln m}{\epsilon} + 2\epsilon T = 2\sqrt{\frac{logn}{T}}. \tag{91}$$

Let $\bar{P} = \frac{1}{T} \sum_{i=1}^{T} p^t, \bar{q} = \frac{1}{T} \sum_{i=1}^{T} q^t$, then

$$\max_i (A\bar{q})_i - \min_j (\bar{q}^T A) \leq \delta = 2\sqrt{\frac{\log n}{T}}. \tag{92}$$

This means

$$\min_j (\bar{p}^T A)_j \leq \bar{P}^T A \bar{q} \leq \max_i (A\bar{q})_i \leq \min_j (\bar{p}^t A)_j + \delta \tag{93}$$

When T goes to infinity, the two bounds are equal, which proves our theorem.

7.3 Application: Max Flow

Consider an unweighted graph $G = (V, E)$. The max flow problem can described by a linear program:

Let $P_{s,t}$ be the set of all paths from s to t, then

$$\max_{p \in P_{s,t}} x(p). \tag{94}$$

$$\text{s.t.} \sum_{p: e \in p} x(p) \leq 1, \forall e \tag{95}$$

$$x(p) \geq 0. \tag{96}$$

The dual problem:

$$\min_{e} \sum_{e} \ell(e) \tag{97}$$

$$\text{s.t} \sum_{e \in p} \ell(e) \geq 1, \forall p \in P_{s,t} \tag{98}$$

$$\ell(e) \geq 0. \tag{99}$$

Zero sum game conversion:

Let γ be the optimal flow, a player P chooses a path, and a player D chooses an edge (might be a mixed strategy). Payoff for D is 1 if $e \in P$ and 0 otherwise.

Lemma 4. *Let v be the value of the game. Then $v = \frac{1}{\gamma}$.*

Proof. Given an optimal solution $\ell(e)$ to the dual, D plays edge e with probability $\frac{\ell(e)}{\sum_e \ell(e)} = \frac{\ell(e)}{\gamma}$, then for all paths, the payoff for D is

$$\sum_{e \in P} \frac{\ell(e)}{\gamma} = \frac{1}{\gamma} \sum_{e \in P} \ell(e) \geq \frac{1}{\gamma}. \tag{100}$$

Let $x(p)$ be the optimal solution to the primal problem. P chooses a path p with probability $\frac{x(p)}{\gamma}$. For any edge,

$$\Pr[e \in p] = \sum_{p: e \in p} = \frac{1}{\gamma} \sum_{p: e \in p} x(p) \leq \frac{1}{\gamma}. \tag{101}$$

□

Remark. We now want to run a multiplicative weights algorithm on this zero-sum game, but the problem is that the primal problem has exponentially many variables. We have make some modification to fix this issue.

- For each $t = 1, \ldots, T$, use multiplicative weights to choose a distribution w_t on edges. Let P^t be the best response to w_t, which is to find the shortest path algorithm with weights w_t.
- Set the reward vector to be: $r^t(e) = \mathbb{1}[e \in P^t]$.
- Suppose the solution is $p_1, \ldots, p_t$, we route $\frac{\gamma}{T}$ units of flow on each p^i. Denote this route as f.

Lemma 5. *This algorithm routes $\leq 1 + \delta$ units on each edge, for $T = \frac{4\gamma^2 \ln m}{\delta^2}$.*

Proof. Suppose for contradiction that $\exists e \in f$ routes more than $(1 + \delta)$ on e, then there is more than $\frac{(1+\delta)T}{\gamma}$ of the path $p^1, \ldots, p^T$ use edge e.

If the D player players edge e in hindsight, then the payoff is more than $\frac{(1+\delta)}{\gamma}$.

In each step, player D gets at most $\frac{1}{\gamma}$ in expectation, because P^t is the best response.

Note that $\frac{\delta}{\gamma} \leq \frac{Regret}{T} \leq \frac{2\sqrt{T \log n}}{T}$. □

7.4 Application: Adaboost

Goal: to learn an unknown function $X \to \{0, 1\}$, given a sequence of training samples $(x, c(x))$, $x \sim D$. We want to minimize

$$\mathbb{E}_{x \sim D}[|h(x) - c(x)|]. \tag{102}$$

Weak learner: does slightly better than randomly guessing, with loss $\frac{1}{2} - \gamma$.

We consider samples in the training set as experts.

For hypothesis h, the penalty for expert x is 0 if $h(x) \neq c(x)$ and 1 otherwise. (**Intuition**: we want to sample more on hard samples).

In each round, the algorithm gives a distribution D^t over experts and obtains a hypothesis h^t which is a weak learner for D^t, i.e., the penalty $M(D^t, h^t) \geq \frac{1}{2} + \gamma$.

The final hypothesis h_{final} labels x according to a majority vote over $h^1(x), \ldots, h^T(x)$.

Analtsis:

Let S be the set of training samples labeled incorrectly by the final hypothesis.

Penalty for each $x \in S$, then since $h_{final}(x)$ is a majority vote,

$$\sum_t M(x, h^t) \leq \frac{T}{2}. \tag{103}$$

Let the potential function be defined by $\Phi^t = \sum_x w_t$.

Then

$$\Phi^T \leq \Phi^1 e^{-\alpha \sum_t M(D^t, h^t)} \leq n e^{0\alpha T(\frac{1}{2}+\gamma)} \tag{104}$$

$$\Phi^T \geq \sum_{x \in S} w_x \geq \sum_{x \in S} (1-\alpha) \sum_t M(x, h^t) \geq |S|(1-\alpha)^{T/2} \tag{105}$$

Then we derive $T = \frac{2}{\gamma^2} \log\left(\frac{1}{\epsilon'}\right)$, where $\epsilon' = \frac{|S|}{n}$

8 Spectral Method

8.1 Sparsest cut

Definition 8.1 (Edge expansion). *Let $G = (V, E)$ be an undirected graph, and $S \subseteq V$ be a subset of vertices. The edge expansion of S is*

$$\phi(S) = \frac{E(S, V - S)}{d(S)} \tag{106}$$

where $d(S) = \sum_{v \in S} d(v)$ be the total degree of the vertices in S.

The edge expansion of a cut $(S, V - S)$ is $\max\{\phi(S), \phi(V - S)\}$.

The edge expansion of a graph is

$$\phi(G) = \min_{S:0<|S|<|V|} \phi(S, V - S) = \min_{S:d(S)\leq \frac{d(V)}{2}, |S|\neq 0} \phi(S). \tag{107}$$

Sparsest cut: Compute $\phi(G)$.

Example.

- Cycle: $2/n$
- Clique: $1/2$.
- Barbell: $1/n^2$

Let A be the adjacency matrix of G. Suppose G is a d-regular graph. The normalized Laplacian is defined as

$$L = I - \frac{1}{d}A. \tag{108}$$

We compute the eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, then the Cheeger inequality states that

$$\frac{\lambda}{2} \leq \phi(G) \leq \sqrt{2\lambda_2}. \tag{109}$$

Lemma 6. *Let $M \in \mathbb{R}^{n\times n}$ be a symmetric matrix and $\lambda_1 \leq \cdots \leq \lambda_n$ be eigenvalues of M. Then*

$$\lambda_k = \min_{k-dim\ V} \max_{x \in V - \{0\}} \frac{x^T M x}{x^T x}. \tag{110}$$

Fact. If x_1 is an eigenvector of λ_1, then

$$\lambda_2 = \max_{x \neq 0, x \perp x_1} \frac{x^T M x}{x^T x} \tag{111}$$

Note that the Laplacian of a d-regular matrix is $L = dI - A$.

$$x^T L x = \sum_{(u,v) \in E} (x_u - x_v)^2. \tag{112}$$

Let $\bar{L} = L/d$.

Theorem 8.2. *Let G be a d-regular, undirected graph, let $\lambda_1 \leq \cdots \leq \lambda_n$ be the eigenvalues of $\bar{L}$. Then*

- *$\lambda_1 = 0$ and $\lambda_n \leq 2$.*
- *$\lambda_k = 0$ if and only if G has at least k connected components.*
- *$\lambda_n = 2$ if and only if at least one of the connected components of G is bipartite.*

Proof. i. Trivial.

ii. $\lambda_k = \min_{k-dim\ S} \max_{x \in S - \{0\}} \frac{\sum_{(u,v) \in E} (x_u - x_v)^2}{d \sum_v x_v^2}$. If $\lambda_k = 0$.

If $\lambda_k = 0$, there exists a k-dimensional S, $\forall x \in S$, $x_u = x_v$ for all $(u, v) \in E$. This means x must be constant within each connected component. Therefore, the dimension of S can be at most the number of connected components. The reverse direction can be proved by letting S be the space of vectors that are constant within each connected component.

iii. $\lambda_n = \max_{x \neq 0} \frac{x^T \bar{L} X}{x^T x}$. Note that

$$2x^T x - x^T \bar{L} x = \frac{1}{d} \sum_{(u,v) \in E} (x_u + x_v)^2. \tag{113}$$

Therefore,

$$\lambda_n = 2 - \min_{x \neq 0} \frac{\sum_{(u,v) \in E} (x_u + x_v)^2}{d \sum_v x_v^2}. \tag{114}$$

If $\lambda_n = 2$, then there is a vector $x \neq 0$, with $\sum_{(u,v) \in E} (x_u + x_v)^2$, so for all $(u, v) \in E$, $x_u = -x_v$. Let $A = \{v : x_v < 0\}$, $B = \{u : x_u > 0\}$. Then $A \cup B$ must be disconnected from $\{v : x_v = 0\}$. Moreover, $A \cup B$ is bipartite. □

8.1.1 Generalize to general graphs

$L = D - A$ where $D = diag(d_1, \ldots, d_n)$. $\bar{L} = I - D^{-1/2}AD^{-1/2}$.

Then we generalize

$$\frac{\sum_{(u,v)\in E}(x_u - x_v)^2}{d} \to \sum_{(u,v)\in E}\Big(\frac{x_u}{\sqrt{d_u}} - \frac{x_v}{\sqrt{d_v}}\Big)^2. \tag{115}$$

8.1.2 Proof of Cheeger's theorem

We want to prove $\frac{\lambda_2}{2} \le \phi(G) \le \sqrt{2\lambda_2}$.

The easy part: $\frac{\lambda_2}{2} \le \phi(G)$.

Note that λ_2 (To fill.)

The hard part: $\phi(G) \le \sqrt{2\lambda_2}$.

Lemma 7. *Given any $y \in \mathbb{R}^v$, there exists an algorithm to find $S \subseteq supp(y)$ with*

$$\phi(S) \le \sqrt{2\frac{y^T L y}{y^T D y}}. \tag{116}$$

Proof. We can solve the $\min_{D^{1/2}y \perp v_1} \frac{y^T L y}{y^T D y}$ problem efficiently when $y \in [-1, 1]$. The problem is how should we round the solution.

Algorithm: sweep cut. Choose a threshold $t \in [0, 1]$ at random, and output $S_+ = \{i \in V \mid y_i^2 > t\}$. This procedure can be simply derandomized by enumerating all $n - 1$ possible cuts.

$$\begin{aligned}
\mathbb{E}_t|E(S_t, \bar{S}_t)| &= \sum_{(i,j)\in E} \Pr[(i,j) \text{ cut by } S_t] && (117)\\
&= \sum_{(i,j)\in E} |y_i - y_j|(y_i + y_j) && (118)\\
&\le \sqrt{\sum_{(i,j)\in E}(y_i - y_j)^2} \cdot \sqrt{\sum_{(i,j)\in E}(y_i + y_j)^2} && (119)\\
&\le \sqrt{\sum_{(i,j)\in E}(y_i - y_j)^2} \cdot \sqrt{2\sum_{i\in V} d(i) y_i^2} && (120)
\end{aligned}$$

The first term is just the numerator. Note that $\mathbb{E}_t[d(S_t)] = \sum_{i\in V} d(i)\mathbb{E}_t[\mathbb{1}(i \in S_t)] = \sum_{i\in V} d(i)y_i^2$.

Therefore,

$$\frac{\mathbb{E}_t[|E(S_t, \bar{S}_t)|]}{\mathbb{E}_t[d(S_t)]} \leq \sqrt{2\frac{y^T L y}{y^T D y}}. \tag{121}$$

This means

$$\mathbb{E}_t[|E(S_t, \bar{S}_t)| - \sqrt{2\frac{y^T L y}{y^T D y}}|d(S_t)|] \leq 0. \tag{122}$$

It means there exists a t^* such that

$$\frac{|E(S_{t^*}, \bar{S}_{t^*})|}{[d(S_{t^*})]} \leq \sqrt{2\frac{y^T L y}{y^T D y}}. \tag{123}$$

□

Now we only need to show a construction of z from y, which satisfies $R(z) \leq R(y)$, and at the same time

$$\sum_{i:z_i\geq 0} d(i) \leq \frac{d(V)}{2}. \tag{124}$$

Since $D^{1/2}y \perp v_1$ we know $\sum_i d(u)y_i = 0$. Suppose $y_1 \leq y_2 \leq \ldots y_n$ WLOG. Find the smallest j such that

$$\sum_{1\leq i\leq j} d(i) \geq \frac{d(V)}{2}. \tag{125}$$

and let $z = y - y_j$. Let $z^+ = \max(0, z)$ and $z^- = \min(0, z)$. Then we know

$$\sum_{i:z_i^-\neq 0} d(i) \leq \frac{d(V)}{2} \text{ and } \sum_{i:z_i^+\neq 0} d(i) \leq \frac{d(V)}{2} \tag{126}$$

Lemma 8. $\min\{R(z^-), R(z^+)\} \leq R(z) \leq R(y)$.

Proof. First, $R(z) \leq R(y)$ because their numerators $\sum(z_i - z_j)^2 = \sum(y_i - y_j)^2$ and their denominators $\sum_i d(i)z_i^2 = \sum_i d(i)y_i^2 + \sum_i d(i)y_j^2 - 2y_i\sum_i y_i d(i) = \sum_i d(i)y_i^2 + \sum_i d(i)y_j^2 \geq \sum_i d(i)y_i^2$. Note that $\sum_i y_i d(i) = 0$ is derived from $D^{1/2}y \perp v_1$.

Moreover,

$$z^T D z = {z^+}^T D z^+ + {z^-}^T D z^- \tag{127}$$

and

$$z^T L z \geq {z^+}^T L z^+ + {z^-}^T L z^- \tag{128}$$

Then

$$\frac{{z^+}^T L z^+ + {z^-}^T L z^-}{{z^+}^T D z^+ + {z^-}^T D z^-} \leq R(z). \tag{129}$$

which means one of $R(z^-)$ and $R(z^+)$ must be less than $R(z)$. □

Finally, we conclude that $\phi(G) \leq \sqrt{2\lambda_2}$ since

$$\phi(G) = \min_{S: d(S) \leq \frac{d(V)}{2}, |S| \neq 0} \phi(S). \tag{130}$$

8.2 Spectal Clustering

Problem: input a set of points a_i with a measure of similarity $w_{ij} \geq 0$. How do we cluster them?

- Embed into $\mathbb{R}^d$ by using eigenvectors of $\bar{L}$. Suppose $(v_1, \ldots, v_{d+1})$ are the d smallest eigenvalues. We map a_i to $((v_2)_i, \ldots, (v_{d+1})_i)$.
- When $d = 2$, it is equivalent to our sweep cut algorithm.

9 Random Walk

2-SAT problem.

Start with a random assignment. At each step, choose a clause that is not satisfied, and flip a variable in the clause.

Analysis.

Consider the Hamming distance $r \in [0, n]$ between the current assignment to the best. (n is the number of variables). The hitting time to $r = 0$ should be $O(n^2)$.

9.1 Random walk on graphs

Let $G = (V, E)$ be an undirected graph.

At each time, a random walk is at some node $i \in V$. At time $t + 1$, the random walk chooses a neighbor of i at random and moves to that neighbor.

A "lazy" random walk: stays at i with probability $1/2$.

Stationary distribution.

- Does there exist one?
- If exists, how long does it take to converge? (mixing time)

Let $p^{(t)}$ be the probability distribution at time t.

$$p_{t+1}(i) = \sum_{j:(i,j)\in E} p_t(j) \frac{1}{d(j)}. \tag{131}$$

We can write this as a matrix multiplication:

$$p_{t+1} = AD^{-1}p_t, \tag{132}$$

where $D = \text{diag}(d_1, \ldots, d_n)$, and A is the adjacency matrix. AD^{-1} is called the transition matrix.

Definition 9.1 (Stationary distribution)**.** *A probability distribution π over V is a stationary distribution if*

$$\pi = (AD^{-1})\pi. \tag{133}$$

We can directly write out a stationary distribution

$$\pi = \frac{(d_1, \ldots, d_n)}{2m}, \text{ where } m = |E|. \tag{134}$$

Definition 9.2 (ergodic). *A random walk is called ergodic if there is a distribution π, such that for any initial distribution p_0,*

$$\lim_{t\to\infty} p_t = \pi. \tag{135}$$

Theorem 9.3. *A random walk is ergodic if and only if it is connected and not bipartite.*

Lazy random walk:

$$p_{t+1}(i) = \frac{1}{2}p_t(i) + \frac{1}{2}\sum_{(i,j)\in E}\frac{p_t(j)}{d(j)}. \tag{136}$$

and in matrix forms:

$$p_{t+1} = (\frac{1}{2}I + \frac{1}{2}AD^{-1})p_t. \tag{137}$$

Theorem 9.4. *A lazy random walk is ergodic if and only if G is connected.*

Proof of theorem 9.3 and 9.4 for d-regular graphs. Let $\alpha_1 \geq \cdots \geq \alpha_n$ be the eigenvalues of $\bar{A} = AD^{-1}$, and $x_1, \ldots, x_n$ be the corresponding eigenvectors.

From theorem 8.2,

- $\alpha_1 = 1, x_1 = 1/\sqrt{n}$
- $\alpha_2 < 1$ if G is connected.
- $\alpha_n \geq -1$. If G is not bipartite if and only if $\alpha_n > -1$.

Let $p_0 = \sum_{i=1}^n c_i x_i$, where $c_i = \langle x_i, p_0\rangle$. Then

$$p_t = \sum_{i=1}^{t} c_i \alpha_i^t x_i. \tag{138}$$

If G is connected and not bipartite, then $|\alpha_i| < 1$, for all $i \neq 1$. Thus, $\alpha_i^t \to 0$ as $t \to \infty$.

$$\lim_{t\to\infty}\sum_{i=1}^{n} c_i\alpha_i^t x_i = c_1 x_1 = \frac{\mathbf{1}}{n}. \tag{139}$$

For lazy random graph, $w_i = \frac{1}{2}(1+\alpha_i) \in [0,1)$ when G is connected. So $w_i^t \to 0$ as $t \to \infty$. □

9.2 Mixing time

Definition 9.5 (Mixing time). *The smallest time t such that $D_{TV}(p_t, \pi) \leq 1/4$, where π is the stationary distribution.*

Definition 9.6 (Spectral gap). *We define the spectral gap of a graph by*

$$\alpha = \min\{1 - \alpha_2, 1 - |\alpha_n|\}. \tag{140}$$

For regular random walks,

$$p_t = \sum_{i=1}^{n} c_i \alpha_i^t x_i = \frac{1}{n} + \sum_{i=2}^{n} c_i \alpha_i^t x_i. \tag{141}$$

Then

$$D_{TV}(p_t, \pi) = \| \sum_{i=2}^{n} c_i \alpha_i^t x_i \|_1 \tag{142}$$

$$\leq \sqrt{n} \| \sum_{i=2}^{n} c_i \alpha_i^t x_i \|_2 \tag{143}$$

$$\leq \sqrt{n}(1 - \alpha^t)(\sum_{i=2}^{n} c_i^2)^{1/2} \leq \sqrt{n}(1 - \alpha^t). \tag{144}$$

10 Expander Graph

Definition 10.1. *A two-sided spectral expander (d, γ) is a graph, such that*

- *G is d-regular*
- $\forall i \geq 2 |\lambda_i(\bar{L}) - 1| \leq \gamma$
- *Explicit: If it takes $poly(n)$ time to produce A.*
- *Strongly explicit: If it takes time $\log n$ to produce the neighbor of a given vertex.*

Remark. For complete graphs, $\lambda_2(\bar{L}) = 1/2$. For a path, $\lambda_2(\bar{L}) = 1/n^2$.

Proposition 10.2 (Expanders are reliable networks.)**.** *For expander graphs, if k edges are removed, the largest connected component is $\sim n - \frac{k}{d\phi}$.*

10.1 Random walks on expander graphs

On a complete graph, it takes $O(k \log n)$ bits to represent a k-steps random walk. On an expander, it takes $\log n + kd = \log n + O(k)$.

10.2 Application: Pseudo-random number generation

Syooise we are given an algorithm with a constant error rate. We can decrease its error rate to exponentially small by repeating polynomial times.

Using Chernoff bound and repeat t times, the error rate is C^{-t}. Suppose the algorithm requires r random bits. The repeat-Chernoff requires rt random bits. However, with expanders, we reduce the number of random bits to $r + 10t$.

Algorithm.

- Choose $v \in \{0,1\}^r$ at random.
- Take $t-1$ steps running a random walk on $(V = \{0,1\}^r, E)$ to generate $v_2, \ldots, v_t$.

Note that a strongly explicit graph is required, as the number of vertices is exponentially large.

Expander graph on V, $d = 400$, $\forall i \geq 2$, $\frac{|\mu_i||}{d} \leq \frac{1}{10}$. ($\mu_i$ is the adjacency).

Let $|X| \subseteq \{0,1\}^r$ be the set that the algorithm is incorrect on. $|X| \leq \frac{2^r}{100}$. Let $S = \{i : v_i \in X\}$. Let $Y = V - X$.

$p_0 = \frac{1}{n}$. The characteristic vectors of X and Y is defined χ_x and χ_y.

Let $D_x = \text{diag}(\chi_x)$ and $D_y = \text{diag}(\chi_y)$. $W = \frac{A}{d}$ is the transition matrix. The probability that a vertex chosen according to p in X is

$$\chi_x^T p = 1^T D_x p. \tag{145}$$

$q = WD_x p_0$ is the probability that a walk starts at a vertex in X and goes to q. The probability that the walk is in X at time $i \in R$ is (R is an arbitrary subset of time in $1, \ldots, t$) We want to show

$$1^T D_{Z_t} W D_{Z_{t-1}} W \ldots D_{Z_1} W D_{Z_0} p_0 \leq \frac{1}{5}^{|} R|. \tag{146}$$

where $Z_i = X$ is $i \in R$ and Y otherwise.

Lemma: $\|D_X W\| \leq 1/5$.

Then

$$1^T D_{Z_t} W D_{Z_{t-1}} W \ldots D_{Z_2} W D_{Z_0} p_0 = 1^T (D_{Z_t} W)(D_{Z_{t-1}} W) \ldots (D_{Z_1} W)(D_{Z_0} W) p_0 \leq (1/5)^R \|p_0\| \tag{147}$$

Then,

$$\Pr\left[|S| > \frac{t}{2}\right] \leq \sum_{|R| > \frac{t}{2}} \Pr[\text{walks is in } X \text{ at times } R] \leq (2/\sqrt{5})^{t+1} \tag{148}$$

For lemma: Let $v = c_1 \mathbf{1} + y$, with $1^T y = 0$. Then $\|v\| \geq \max\{c_1\sqrt{n}, \|y\|\}$.

$$\frac{\|D_x W v\|}{\|v\|} \tag{149}$$

11 Hardness Assumption

11.1 Encryption

Alice sends a message b to Bob. Bob should be the only person who can decode the message.

Private-key encryption

- Shared key: random bit $c \in \{0,1\}$.
- To send a message b, Alice sends $b \oplus c$. (One time pad).

Public-Key encryption

- Key generation: Generate a public key and a secret key. Publish pk.
- Encryption: take randomness r, $Enc(b, pk, r) = c$
- Decryption: $Dec(pk, sk, c) = b$.

Definition 11.1 (Computationally indistinguishable)**.** *We say* $(pk, Enc(pk, r, 0)) \sim (pk, Enc(pk, r, 1))$ *computationally indistinguishable, if:*

If there exists an efficient algorithm that distinguishes $Enc(0), Enc(1)$*, then some underlying hardness assumption is broken.*

11.2 Hardness

Definition 11.2 (Learning with errors)**.** *The LWE problems are defined by*

- $A \in \mathbb{Z}_q^{m \times n}$ *with* $m \gg n \log q$.
- A *is sampled uniformly at random,*
- *Consider* $s \in \mathbb{Z}_q^n$*. Given* As*, solving* s *is easy.*
- *Given* $As + e$, $e \in \mathbb{Z}_q^m$ *and* $e = O(\sqrt{n})$.

Assumption 11.3 (Decisional LWE assumption)**.** *Let* e *sampled from a truncated Gaussian,* $|e| \leq \sigma$ *and* $\sigma \geq 2\sqrt{n}$.

$$(A, As + e) \sim (A, Unif(\mathbb{Z}_q^m)) \tag{150}$$

11.3 Encryption scheme from LWE.

Private-key scheme:

- Shared key $s \in \mathbb{Z}_q^n$.
- To encrypt a message b, pick A at random and pick e.

$$c = As + e + (0, \ldots, 0, b \cdot \lfloor q/2 \rfloor .) \tag{151}$$

- Send (A, c).
- To decrypt, output $rounding(c - As)$.

Public-key scheme:

The only difference is that we treat A, $As + e$ as the public-key, and treat s as the secret key.

- Randomness $r \in \{0,1\}^m$. Encrypt: $(r^T A, r^T(As + e) + b\lfloor q/2 \rfloor)$
- Decrypt (c_1, c_2): round $c_2 - c_1^T s$.

Security of the Public-key scheme.

We can show $r^T[A\ u] + (0, \ldots, 0, b\lfloor q/2 \rfloor)$ is indistinguishable to uniform by the following lemma.

Lemma 9 (Leftover Hash)**.** *For $m \gg n \log n$, and r uniformly at random, then the following two distributions are statistically close:*

$$(A, r^T A), (A, u). \tag{152}$$

11.4 Worst-case to average-case reduction

Start with a distribution D over s, such that $(A, As + e)$ is hard.

Sample $\hat{s}$ uniformly at random, $(A, As + e + A\hat{s})$.

12 Quantum Computing

Consider qubit $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$, and $|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$.

Definition 12.1 (Quantum circuits.)**.** *A quantum circuit is defined by a unitary matrix U ($U^TU = I$), so that*

$$\langle\psi|U^TU|\psi\rangle = \langle\psi|\psi\rangle = 1. \tag{153}$$

Definition 12.2 (Universal quantum gates)**.**

- *Toffoli: $T|a,b,c\rangle = |a,b,ab\oplus c\rangle$.*
- *Hadamard: $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle$.*

One can check that both gates are unitary.

12.1 Efficient period finding

Simon's problem. Input a black-box function $f : \{0,1\}^n \to \{0,1\}^n$. Promise that there exists $s \in \{0,1\}^n$, $f(x) = f(x+s)$. How to find s?

Classical algorithms take exponential time.

Quantum speedup.

- Create a uniform superposition: $\frac{1}{\sqrt{2^n}}\sum_x |x\rangle$ by $H|0\rangle$.
- Apply f in superposition:

$$\frac{1}{\sqrt{2^n}}\sum_x |x\rangle|0^n\rangle \to \frac{1}{\sqrt{2^n}}\sum_x |x\rangle|f(x)\rangle. \tag{154}$$

  Note that this operation is unitary (reversible) as we can subtract $f(x)$ to recover.

- Measure the second register. If we measure $y \in \{0,1\}^n$, the probability of y is $2\cdot\frac{1}{\sqrt{2^n}}$, and the remaining state

$$\frac{1}{\sqrt{2}}(|x\rangle + |x\oplus s\rangle), s.t. f(x) = f(x+\oplus s) = y. \tag{155}$$

- Apply $H^{\oplus n}$.

$$\frac{1}{\sqrt{2\cdot 2^n}}(\sum_y (-1)^{x\cdot y}(1+(-1)^{y\cdot s})|y\rangle) \tag{156}$$

  Notice that $1+(-1)^{y\cdot s} = 0$ if $y\cdot s = 1$.

- Sample many y to get linear equations $y\cdot s = 0$. Solve s.

12.2 Reduction of factorization to period finding

Factoring: $N = pq$, find p, q.

- Find $a < N$ with $(a, N) = 1$.
- Construct function $f(x) = a^x \pmod N$. So $f(x) = f(x + r)$ if and only if $x^r = 1 \pmod N$. We can thus find r.

  Lemma 10. *If $N = pq$, $a < N$ chosen uniformly at random and $(a, N) = 1$, then with probability $\geq 1/2$, r is even and $a^{r/2} \neq \pm 1 \pmod N$.*

- $(a^{r/2} + 1)(a^{r/2} - 1) = 0 \pmod N$. Then $gcd(N, a^{r/2} + 1)$ is a nontrivial factor of N.