Chapter 19

Discrepancy and Derandomization

By Sariel Har-Peled, February 17, 2022

“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.”

Gustible’s Planet, Cordwainer Smith

19.1. Discrepancy

Consider a set system $(X, \mathcal{R})$, where $n = |X|$, and $\mathcal{R} \subseteq 2^X$. A natural task is to partition $X$ into two sets $S, T$, such that for any range $r \in \mathcal{R}$, we have that $\chi(r) = |S \cap r| - |T \cap r|$ is minimized. In a perfect partition, we would have that $\chi(r) = 0$ – the two sets $S, T$ partition every range perfectly in half. A natural way to do so, is to consider this as a coloring problem – an element of $X$ is colored by $+1$ if it is in $S$, and $-1$ if it is in $T$.

Definition 19.1.1. Consider a set system $S = (X, \mathcal{R})$, and let $\chi : X \rightarrow \{-1, +1\}$ be a function (i.e., a coloring). The discrepancy of $r \in \mathcal{R}$ is $\chi(r) = |\sum_{x \in r} \chi(x)|$. The discrepancy of $\chi$ is the maximum discrepancy over all the ranges – that is

$$\text{disc}(\chi) = \max_{r \in \mathcal{R}} \chi(r).$$

The discrepancy of $S$ is

$$\text{disc}(S) = \min_{\chi : X \rightarrow \{-1, +1\}} \text{disc}(\chi).$$

Bounding the discrepancy of a set system is quite important, as it provides a way to shrink the size of the set system, while introducing small error. Computing the discrepancy of a set system is generally quite challenging. A rather decent bound follows by using random coloring.

Definition 19.1.2. For a vector $v = (v_1, \ldots, v_n) \in \mathbb{R}^n$, $\|v\|_\infty = \max_i |v_i|$.

For technical reasons, it is easy to think about the set system as an incidence matrix.

Definition 19.1.3. For a $m \times n$ a binary matrix $M$ (i.e., each entry is either 0 or 1), consider a vector $b \in \{-1, +1\}^n$. The discrepancy of $b$ is $\|Mb\|_\infty \leq 4\sqrt{n \log m}$.

Theorem 19.1.4. Let $M$ be an $n \times n$ binary matrix (i.e., each entry is either 0 or 1), then there always exists a vector $b \in \{-1, +1\}^n$, such that $\|Mb\|_\infty \leq 4\sqrt{n \log n}$. Specifically, a random coloring provides such a coloring with high probability.

This work is licensed under the Creative Commons Attribution-Noncommercial 3.0 License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.
Proof: Let \( v = (v_1, \ldots, v_n) \) be a row of \( M \). Chose a random \( b = (b_1, \ldots, b_n) \in \{-1, +1\}^n \). Let \( i_1, \ldots, i_l \) be the indices such that \( v_{i_j} = 1 \), and let

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

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

\[
\mathbb{E}[Y] = \mathbb{E}[\langle v, b \rangle] = \mathbb{E}\left[ \sum_{i} v_i b_i \right] = \sum_{i} \mathbb{E}[v_i b_i] = \sum_{i} v_i \mathbb{E}[b_i] = 0.
\]

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

\[
\mathbb{P}[|Y| \geq \Delta] = 2 \mathbb{P}[\langle v, b \rangle \geq \Delta] = 2 \mathbb{P}\left[ \sum_{j=1}^{r} b_{i_j} \geq \Delta \right] \leq 2 \exp\left( -\frac{\Delta^2}{2\tau} \right) = 2 \exp\left( -8 \frac{n \ln m}{\tau} \right) \leq \frac{2}{m^8}.
\]

Thus, the probability that any entry in \( Mb \) exceeds \( 4\sqrt{n \ln} \), is smaller than \( 2/m^7 \). Thus, with probability at least \( 1 - 2/m^7 \), all the entries of \( Mb \) have absolute value smaller than \( 4\sqrt{n \ln m} \).

In particular, there exists a vector \( b \in \{-1, +1\}^n \) such that \( \| Mb \|_\infty \leq 4\sqrt{n \ln m} \). \( \blacksquare \)

We might spend more time on discrepancy later on – it is a fascinating topic, well worth its own course.

### 19.2. The Method of Conditional Probabilities

In previous lectures, we encountered the following problem.

**Problem 19.2.1 (Set Balancing/Discrepancy).** Given a binary matrix \( M \) of size \( n \times n \), find a vector \( v \in \{-1, +1\}^n \), such that \( \| M v \|_\infty \) is minimized.

Using random assignment and the Chernoff inequality, we showed that there exists \( v \), such that \( \| M v \|_\infty \leq 4\sqrt{n \ln n} \). 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 \( i \)th level we expose the \( i \)th coordinate of \( v \). 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 – here randomly assigning the remaining bits can be interpreted as a random walk down the tree to a leaf.

Formally, the algorithm is **successful** if ends up with a vector \( v \), such that \( \| M v \|_\infty \leq 4\sqrt{n \ln n} \).

Let \( v_l \) and \( v_r \) be the two children of \( v \). Clearly, \( P(v) = (P(v_l) + P(v_r))/2 \). In particular, \( \max(P(v_l), P(v_r)) \geq P(v) \). Thus, if we could compute \( P(\cdot) \) quickly (and deterministically), then we could derandomize the algorithm.

Let \( C^+_m \) be the bad event that \( r_m \cdot v > 4\sqrt{n \log n} \), where \( r_m \) is the \( m \)th row of \( M \). Similarly, \( C^-_m \) is the bad event that \( r_m \cdot v < -4\sqrt{n \log n} \), and let \( C_m = C^+_m \cup C^-_m \). Consider the probability, \( \mathbb{P}[C_m | v_1, \ldots, v_k] \) (namely, the first \( k \) coordinates of \( v \) are specified). Let \( r_m = (r_1, \ldots, r_n) \). We have that

\[
\mathbb{P}[C_m | v_1, \ldots, v_k] = \mathbb{P}\left[ \sum_{i=k+1}^{n} v_i r_i > 4\sqrt{n \log n} - \sum_{i=1}^{k} v_i r_i \right] = \mathbb{P}\left[ \sum_{i \geq k+1, r_i \neq 0} v_i r_i > L \right] = \mathbb{P}\left[ \sum_{i \geq k+1, r_i = 1} v_i > L \right],
\]

\( \mathbb{P}[C^-_m | v_1, \ldots, v_k] = \mathbb{P}\left[ \sum_{i=k+1}^{n} v_i r_i < -4\sqrt{n \log n} - \sum_{i=1}^{k} v_i r_i \right] = \mathbb{P}\left[ \sum_{i \geq k+1, r_i \neq 0} v_i r_i < L \right] = \mathbb{P}\left[ \sum_{i \geq k+1, r_i = 1} v_i < L \right].
\]

2
where $L = 4\sqrt{n \log n} - \sum_{i=1}^{k} v_i r_i$ is a known quantity (since $v_1, \ldots, v_k$ are known). Let $V = \sum_{i \geq k+1, r_i = 1} v_i$. We have,

$$P[C_m^+ | v_1, \ldots, v_k] = P\left[ \sum_{i \geq k+1, r_i = 1} (v_i + 1) > L + V \right] = P\left[ \sum_{i \geq k+1, r_i = 1} \frac{v_i + 1}{2} > \frac{L + V}{2} \right].$$

The last quantity is the probability that in $V$ flips of a fair 0/1 coin one gets more than $(L+V)/2$ heads.

Thus,

$$P_m^+ = P[C_m^+ | v_1, \ldots, v_k] = \sum_{i=[(L+V)/2]}^{V} \binom{V}{i} \left( \frac{1}{2^n} \right)^i = \frac{1}{2^n} \sum_{i=[(L+V)/2]}^{V} \binom{V}{i}.$$ 

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 $P_m^- = P[C_m | v_1, \ldots, v_k]$.

For a node $v \in T$, let $v_r$ denote the portion of $v$ that was fixed when traversing from the root of $T$ to $v$. Let $P(v) = \sum_{m=1}^{n} P[C_m | v_r]$. By the above discussion $P(v)$ can be computed in polynomial time. Furthermore, we know, by the previous result on discrepancy that $P(r) < 1$ (that 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 $v'$ such that $\|Av'\|_\infty \leq 4\sqrt{n \log n}$.

**Theorem 19.2.2.** Using the method of conditional probabilities, one can compute in polynomial time in $n$, a vector $v \in \{-1, 1\}^n$, such that $\|Av\|_\infty \leq 4\sqrt{n \log n}$.

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

19.3. Bibliographical Notes

There is a lot of nice work on discrepancy in geometric settings. See the books [Cha01, Mat99].

19.4. From previous lectures

**Theorem 19.4.1.** Let $X_1, \ldots, X_n$ be $n$ independent random variables, such that $P[X_i = 1] = P[X_i = -1] = \frac{1}{2}$, for $i = 1, \ldots, n$. Let $Y = \sum_{i=1}^{n} X_i$. Then, for any $\Delta > 0$, we have

$$P[Y \geq \Delta] \leq \exp(-\Delta^2/2n).$$
References
