

Discrepancy and Derandomization

“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

18.1. 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 18.1.1. 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 $\chi(r) = \left| \sum_{x \in r} \chi(x) \right|$. The *discrepancy of χ* 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 18.1.2. For a vector $\mathbf{v} = (v_1, \dots, v_n) \in \mathfrak{R}^n$, $\|\mathbf{v}\|_\infty = \max_i |v_i|$.

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

Definition 18.1.3. For a $m \times n$ binary matrix $\mathbf{M}$ (i.e., each entry is either 0 or 1), consider a vector $\mathbf{b} \in \{-1, +1\}^n$. The **discrepancy** of $\mathbf{b}$ is $\|\mathbf{M}\mathbf{b}\|_\infty$.

Theorem 18.1.4. *Let $\mathbf{M}$ be an $n \times n$ binary matrix (i.e., each entry is either 0 or 1), then there always exists a vector $\mathbf{b} \in \{-1, +1\}^n$, such that $\|\mathbf{M}\mathbf{b}\|_\infty \leq 4\sqrt{n \log n}$. Specifically, a random coloring provides such a coloring with high probability.*

Proof: Let $v = (v_1, \dots, v_n)$ be a row of $\mathbf{M}$. Chose a random

$$\mathbf{b} = (b_1, \dots, b_n) \in \{-1, +1\}^n.$$

Let $i_1, \dots, i_\tau$ be the indices such that $v_{i_j} = 1$, and let

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

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

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

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

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

since $\tau \leq n$. In words, the probability that any entry in Mb exceeds (in absolute values) $4\sqrt{n \ln m}$, 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 $\mathbf{b} \in \{-1, +1\}^n$ such that $\|M\mathbf{b}\|_\infty \leq 4\sqrt{n \ln m}$. 🐼

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

18.2. The Method of Conditional Probabilities

In previous lectures, we encountered the following problem.

Problem 18.2.1 (Set Balancing/Discrepancy). Given a binary matrix M of size $n \times n$, find a vector $\mathbf{v} \in \{-1, +1\}^n$, such that $\|M\mathbf{v}\|_\infty$ is minimized.

Using random assignment and the Chernoff inequality, we showed that there exists $\mathbf{v}$, such that $\|M\mathbf{v}\|_\infty \leq 4\sqrt{n \ln n}$. 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 $\mathbf{v}$. 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 $\mathbf{v}$, such that $\|M\mathbf{v}\|_\infty \leq 4\sqrt{n \ln n}$.

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 $\mathbf{r}_m \cdot \mathbf{v} > 4\sqrt{n \log n}$, where $\mathbf{r}_m$ is the m th row of M . Similarly, C_m^- is the bad event that $\mathbf{r}_m \cdot \mathbf{v} < -4\sqrt{n \log n}$, and let $C_m = C_m^+ \cup C_m^-$. Consider the probability, $\mathbb{P}[C_m^+ \mid \mathbf{v}_1, \dots, \mathbf{v}_k]$ (namely,

the first k coordinates of $\mathbf{v}$ are specified). Let $\mathbf{r}_m = (r_1, \dots, r_n)$. We have that

$$\begin{aligned}\mathbb{P}[C_m^+ \mid \mathbf{v}_1, \dots, \mathbf{v}_k] &= \mathbb{P}\left[\sum_{i=k+1}^n \mathbf{v}_i r_i > 4\sqrt{n \log n} - \sum_{i=1}^k \mathbf{v}_i r_i\right] \\ &= \mathbb{P}\left[\sum_{i \geq k+1, r_i \neq 0} \mathbf{v}_i r_i > L\right] = \mathbb{P}\left[\sum_{i \geq k+1, r_i=1} \mathbf{v}_i > L\right],\end{aligned}$$

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

$$\begin{aligned}\mathbb{P}[C_m^+ \mid \mathbf{v}_1, \dots, \mathbf{v}_k] &= \mathbb{P}\left[\sum_{i \geq k+1, \alpha_i=1} (\mathbf{v}_i + 1) > L + V\right] \\ &= \mathbb{P}\left[\sum_{i \geq k+1, \alpha_i=1} \frac{\mathbf{v}_i + 1}{2} > \frac{L + V}{2}\right],\end{aligned}$$

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,

$$P_m^+ = \mathbb{P}[C_m^+ \mid \mathbf{v}_1, \dots, \mathbf{v}_k] = \sum_{i=\lceil (L+V)/2 \rceil}^V \binom{V}{i} \frac{1}{2^V} = \frac{1}{2^V} \sum_{i=\lceil (L+V)/2 \rceil}^V \binom{V}{i}.$$

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 $P_m^- = \mathbb{P}[C_m^- \mid \mathbf{v}_1, \dots, \mathbf{v}_k]$.

For a node $v \in T$, let $\mathbf{v}_v$ denote the portion of $\mathbf{v}$ that was fixed when traversing from the root of T to v . Let $P(v) = \sum_{m=1}^n \mathbb{P}[C_m \mid \mathbf{v}_v]$. 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 $v = \text{root}(T)$. 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 $\mathbf{v}'$ such that $\|A\mathbf{v}'\|_\infty \leq 4\sqrt{n \log n}$.

Theorem 18.2.2. *Using the method of conditional probabilities, one can compute in polynomial time in n , a vector $\mathbf{v} \in \{-1, 1\}^n$, such that $\|A\mathbf{v}\|_\infty \leq 4\sqrt{n \log n}$.*

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

18.3. Bibliographical Notes

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

18.4. From previous lectures

Theorem 18.4.1. *Let $X_1, \dots, X_n$ be n independent random variables, such that $\mathbb{P}[X_i = 1] = \mathbb{P}[X_i = -1] = \frac{1}{2}$, for $i = 1, \dots, n$. Let $Y = \sum_{i=1}^n X_i$. Then, for any $\Delta > 0$, we have*

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

References

- [Cha01] B. Chazelle. *The Discrepancy Method: Randomness and Complexity*. New York, NY, USA: Cambridge University Press, 2001.
- [Mat99] J. Matoušek. *Geometric Discrepancy: An Illustrated Guide*. Vol. 18. Algorithms and Combinatorics. Berlin, Heidelberg: Springer, 1999.