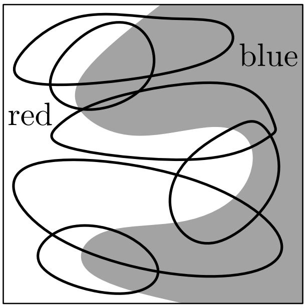
Chapter 11
Colorings with Low Discrepancy

11.1 Discrepancy of Set Systems

11.1.1 Let $V = \{1, 2, \ldots, n\}$ be a vertex set and let $\mathcal{F} = \{F_1, F_2, \ldots, F_m\}$ be a system of subsets of V. (We can also regard $(V, \mathcal{F})$ as a *hypergraph.*)

11.1.2 The basic problem in *combinatorial discrepancy theory* is to color each vertex $i \in V$ either red or blue, in such a way that each of the sets of $\mathcal{F}$ has roughly the same number of red points and blue points, as in the following schematic picture:



11.1.3 It is not always possible to achieve an exact splitting of each set. As an extreme example, if $\mathcal{F} := 2^V$ consists of all subsets of V, then there will always be a completely monochromatic set of size at least $\frac{n}{2}$.

11.1.4 The maximum deviation from an exact splitting, over all sets of $\mathcal{F}$, is the *discrepancy* of the set system $\mathcal{F}$. Formally, we let a *coloring* of $(V, \mathcal{F})$ be an arbitrary mapping $\chi: V \to \{-1, +1\}$.

B. Gärtner and J. Matousek, *Approximation Algorithms and Semidefinite Programming*, DOI 10.1007/978-3-642-22015-9_11,

The *discrepancy* of $\mathcal{F}$ is

$$\mathrm{disc}(\mathcal{F}) := \min_{\chi} \mathrm{disc}(\mathcal{F}, \chi),$$

where the minimum is over all colorings $\chi\colon V \to \{-1, +1\}$, and

$$\mathrm{disc}(\mathcal{F}, \chi) := \max_{F \in \mathcal{F}} |\chi(F)|,$$

where we use the shorthand $\chi(F)$ for $\sum_{j \in F} \chi(j)$.

If $+1$'s are red and -1's are blue, then $\chi(F)$ is the number of red points in F minus the number of blue points in F, which we call the *imbalance* of F under χ.

11.1.5 Discrepancy has been investigated extensively, and there are many upper and lower bounds known for various set systems; see, e.g., [Mat10, Spe87, AS08] for introductions. There are also close connections to the classical subject of uniformly distributed point sets and sequences in various geometric domains, such as the unit square (see, e.g., [Mat10, ABC97]).

11.1.6 We will consider the algorithmic problem of computing a low-discrepancy coloring for a given set system $\mathcal{F}$.

- This question, in addition to its intrinsic interest, also presents a prototype question in the more general and very basic problem of *simultaneous rounding under linear constraints*. In such a problem, we have a vector $\mathbf{x} \in [-1, 1]^n$, which satisfies some system of linear constraints $A\mathbf{x} = \mathbf{b}$, and we would like to "round" each component to either $+1$ or -1 so that the resulting vector $\mathbf{z}$ almost satisfies the constraints, i.e., the vector $A\mathbf{z} - \mathbf{b}$ is small in a suitable sense.
  - In the case of discrepancy, we have $\mathbf{x} = \mathbf{0}$, $\mathbf{b} = \mathbf{0}$, and A is the incidence matrix of the set system $\mathcal{F}$ – that is, $a_{ij} = 1$ if $j \in F_i$, and $a_{ij} = 0$ otherwise. The rounding error is measured as $\|A\mathbf{x} - \mathbf{b}\|_\infty$.
- We cannot expect a satisfactory general solution of the simultaneous rounding problem, e.g., because of the hardness results for discrepancy mentioned below, but it is interesting to study special cases where a good rounding is possible.

11.1.7 Specifically, we will discuss a recent breakthrough by Bansal [Ban10], an algorithm for computing colorings with "reasonably low" discrepancy. It is based on semidefinite programming and it introduces a very interesting rounding strategy. The coloring is obtained using a random walk, driven by optimal solutions of suitable semidefinite programs – quite different from the "usual" hyperplane cuts.

11.1.8 First, $\mathrm{disc}(\mathcal{F})$ in itself is very hard to approximate. Charikar et al. [CNN11] proved that it is NP-hard to distinguish between set systems of discrepancy 0 and those with discrepancy of order $\sqrt{n}$.

- To appreciate this, one should know that $\mathrm{disc}(\mathcal{F}) = O(\sqrt{n \log(m/n)})$ for *every* system of $m \geq n$ sets on n points.
- It is easy to show that a *random* coloring has discrepancy $O(\sqrt{n \log m})$ with high probability, while the improvement of $\log m$ to $\log(m/n)$ is much harder; see [Mat10, Spe87, AS08].
- Thus, we have $\mathrm{disc}(\mathcal{F}) = O(\sqrt{n \log n})$ as long as m is bounded by a polynomial in n.

11.1.9 Discrepancy can be regarded as a measure of how complex a set system is. But it is not very well-behaved, since a system with almost the maximum possible discrepancy can be hidden in a system with zero discrepancy.

- Example: Take the complete set system $\mathcal{F} = 2^V$, make a disjoint copy V' of V, let F' be the clone of F in V'. The set system $(V \cup V', \{F \cup F' : F \in \mathcal{F}\})$ has discrepancy 0, yet we feel that it is as complex as $\mathcal{F}$.

11.1.10 A better behaved measure is the *hereditary discrepancy*, defined as

$$\mathrm{herdisc}(\mathcal{F}) := \max_{A \subseteq V} \mathrm{disc}(\mathcal{F}|_A).$$

Here $\mathcal{F}|_A$ denotes the *restriction* of the set system $\mathcal{F}$ to the ground set A, i.e., $\{F \cap A : F \in \mathcal{F}\}$.

- In other words, the enemy selects a subset $A \subseteq V$ and we must color its points red or blue so that each set in $\mathcal{F}$ is balanced; the uncolored points outside A do not count.
- Practically all known upper bounds on $\mathrm{disc}(\mathcal{F})$ also apply to $\mathrm{herdisc}(\mathcal{F})$.
- On the other hand, the question "Is $\mathrm{disc}(\mathcal{F}) \leq k$?" at least belongs to the class NP, while for "Is $\mathrm{herdisc}(\mathcal{F}) \leq k$?", membership in NP is open and maybe false.

11.1.11 Bansal's algorithm yields a coloring for a given $\mathcal{F}$ with discrepancy not much larger than $\mathrm{herdisc}(\mathcal{F})$:

> **Theorem** (Bansal [Ban10]). *There is a randomized polynomial-time algorithm which, for an input set system $\mathcal{F}$ on n points, with m sets, and with hereditary discrepancy at most H, computes a coloring χ with* $\mathrm{disc}(\mathcal{F}, \chi) = O(H \log(mn))$.

- It may be tempting to conclude that the algorithm approximates the hereditary discrepancy with $O(\log(mn))$ factor, but this is not necessarily the case! Paradoxically, the algorithm may err on the good side; it may possibly compute a coloring with discrepancy much *smaller* than $\mathrm{herdisc}(\mathcal{F})$. (So we never learn that $\mathrm{herdisc}(\mathcal{F})$ is large.)

- Bansal's paper also has additional, technically subtler results, which save logarithmic factors in the discrepancy bound in certain special settings.
  - Our presentation of Bansal's algorithm below is somewhat simplified compared to his original formulation. However, for some of the additional results in his paper, our simplifications do not seem applicable.

11.1.12 Bansal's algorithm has converted several famous *existential* proofs in discrepancy theory into *constructive* ones.

- For example, Spencer [Spe85] proved in 1986 that every system of n sets on n points has discrepancy $O(\sqrt{n}\,)$ (which is asymptotically tight in the worst case). The argument was existential. Bansal gave the first polynomial-time algorithm that finds a coloring with $O(\sqrt{n}\,)$ discrepancy in this setting (this requires adding some twists to the algorithm presented here).
- As another example, the system of all *arithmetic progressions* on the ground set $\{1, 2, \ldots, n\}$ was known to have discrepancy of order $n^{1/4}$, but Bansal's algorithm is the first polynomial-time algorithm that can compute a coloring with discrepancy close to $n^{1/4}$ (with an extra logarithmic factor in this case).

11.2 Vector Discrepancy and Bansal's Random Walk Algorithm

11.2.1 First we set up a semidefinite relaxation of discrepancy. Instead of coloring by ± 1's, we color by unit vectors. We will thus talk about *vector discrepancy* $\mathrm{vecdisc}(\mathcal{F})$, which is the smallest $D \geq 0$ for which the following vector program is feasible:

$$\boxed{\begin{aligned} \Big\|\sum_{j\in F_i} \mathbf{u}_j\Big\|^2 &\leq D^2, \quad i = 1, 2, \ldots, m,\\ \|\mathbf{u}_j\|^2 &= 1, \qquad j = 1, 2, \ldots, n. \end{aligned}}$$

- This is indeed a relaxation of disc, so $\mathrm{vecdisc}(\mathcal{F}) \leq \mathrm{disc}(\mathcal{F})$.
- We also introduce the *hereditary vector discrepancy* $\mathrm{hervecdisc}(\mathcal{F})$, as the maximum vector discrepancy of a restriction of $\mathcal{F}$ to a subset $A \subseteq V$.
- In the proof of Theorem 11.1.11, the algorithm will actually find a coloring with discrepancy at most $O(\mathrm{hervecdisc}(\mathcal{F}) \log(mn))$.
- $\mathrm{vecdisc}(\mathcal{F})$ can be computed (up to a prescribed error) in polynomial time; for $\mathrm{hervecdisc}(\mathcal{F})$ we do not know.

11.2.2 In Bansal's algorithm we want to find a low-discrepancy coloring (by ± 1's). We will approach the desired coloring through a sequence of *semicolorings*, where a *semicoloring* is an arbitrary mapping $\xi\colon V \to [-1, 1]$.

- A semicoloring is like a coloring but by real numbers in $[-1, 1]$.
- The discrepancy $\mathrm{disc}(\mathcal{F}, \xi)$ for a semicoloring is defined in the same way as for a coloring, as $\mathrm{disc}(\mathcal{F}, \xi) = \max_{F \in \mathcal{F}} \left| \sum_{j \in F} \xi(j) \right|$.
- In the algorithm, we will represent a semicoloring by a point $\mathbf{x} \in [-1, 1]^n$.

11.2.3 Bansal's algorithm starts with the semicoloring $\mathbf{x}_0 := \mathbf{0}$, which has zero discrepancy but is rather useless as an "approximation" to a true coloring. Then it produces a sequence

$$\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\ell \in [-1, 1]^n,$$

of semicolorings. Here ℓ, the length of the sequence, is a suitable parameter to be determined later.

- The algorithm is randomized.
- As we will see, with probability close to 1, the final semicoloring $\mathbf{x}_\ell$ is actually a coloring, i.e., all coordinates are ± 1's. This is the output of the algorithm.
- If $\mathbf{x}_\ell$ is not a coloring, we restart the algorithm from scratch.

11.2.4 The algorithm can be regarded as a random walk in the cube $[-1, 1]^n$. In the t-th step, $\mathbf{x}_t$ is obtained from $\mathbf{x}_{t-1}$ by a (small) random step, as follows:

- First we generate an increment $\mathbf{\Delta}_t \in \mathbb{R}^n$. It is random but chosen from a carefully crafted distribution; we will discuss this later.
- A "tentative value" of $\mathbf{x}_t$ is $\tilde{\mathbf{x}}_t := \mathbf{x}_{t-1} + \mathbf{\Delta}_t$. But we still need to truncate each coordinate to the interval $[-1, 1]$:

$$(\mathbf{x}_t)_j := \begin{cases} +1 & \text{if } (\tilde{\mathbf{x}}_t)_j \geq 1, \\ -1 & \text{if } (\tilde{\mathbf{x}}_t)_j \leq -1, \text{ and} \\ (\tilde{\mathbf{x}}_t)_j & \text{otherwise.} \end{cases}$$

- The increments $\mathbf{\Delta}_t$ are generated in such a way that once a coordinate of $\mathbf{x}_t$ reaches $+1$ or -1, it will never change. We can think of the faces of the cube as being "sticky"; as soon as the walk hits a face, it will stay in that face until the end.
  - ◦ More formally, we let $A_t := \{j \in V : (\mathbf{x}_{t-1})_j \neq \pm 1\}$ be the set of coordinates that are still *active* in the t-th step. We will make sure that $(\mathbf{\Delta}_t)_j = 0$ for all $j \notin A_t$.
- Figure 11.1 shows a schematic illustration of the random walk.

11.2.5 It remains to discuss how the increment $\mathbf{\Delta}_t$ is generated. The idea is that each (active) coordinate of $\mathbf{\Delta}_t$ is random, but the various coordinates are correlated so that the contribution of $\mathbf{\Delta}_t$ to the discrepancy is small.

- First, via semidefinite programming, we compute a coloring of the current active set A_t by unit vectors witnessing the vector discrepancy of the set system $\mathcal{F}|_{A_t}$.

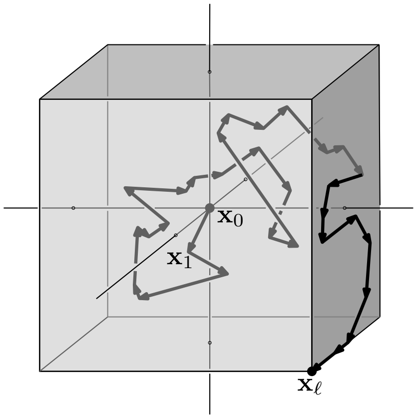


Fig. 11.1 Schematic illustration of the random walk

- ∘ More explicitly, we compute unit vectors $\mathbf{u}_{t,j}$, $j \in A_t$, so that
$$\Big\| \sum_{j \in F_i \cap A_t} \mathbf{u}_{t,j} \Big\|^2 \le D^2$$
for all i, with $D \ge 0$ as small as possible.
- ∘ For notational convenience, we also set $\mathbf{u}_{t,j} := \mathbf{0}$ for $j \notin A_t$.

- Next we generate a random vector $\boldsymbol{\gamma}_t$ from the n-dimensional standard normal (or Gaussian) distribution. That is, the coordinates of $\boldsymbol{\gamma}_t$ are independent $N(0,1)$ random variables (also independent of $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{t-1}$).
  - ∘ Actually, the Gaussian distribution of the $\boldsymbol{\gamma}_t$ is not crucial for the algorithm. For example, independent random ± 1 vectors work as well, but the analysis becomes somewhat more complicated.
- Then we set
$$(\boldsymbol{\Delta}_t)_j := \sigma \boldsymbol{\gamma}_t^T \mathbf{u}_{t,j}, \quad j = 1, 2, \ldots, n.$$
Here σ is a sufficiently small parameter. As we will see in due time,
$$\sigma := \frac{1}{C_0 n \sqrt{\log n}},$$
with a sufficiently large constant C_0, will work (and a smaller σ, say n^{-2}, would work as well – only the running time would suffer).
- The length ℓ of the random walk should be set to $C_1 \sigma^{-2} \log n$, with another suitable constant C_1.

This concludes the description of Bansal's algorithm.

11.2.6 The idea why the algorithm works is outlined in the following two items (whose formulation is intentionally imprecise):

- First, the projection of the random walk to a given coordinate axis behaves like a one-dimensional random walk with increments having the normal distribution $N(0, \sigma^2)$. Such a walk will typically cross the boundary of the interval $[-1, 1]$ in about σ^{-2} steps, and after $\sigma^{-2} \log n$ steps it is even *very* likely to have crossed the boundary. But this means that $\mathbf{x}_\ell$ is typically a ± 1 vector.
- Second, the imbalance of a fixed set F_i starts out at 0 under the zero semicoloring $\mathbf{x}_0$, and in the t-th step it is changed by $\sum_{j \in F_i} \sigma \boldsymbol{\gamma}_t^T \mathbf{u}_{t,j} = \sigma \boldsymbol{\gamma}_t^T \mathbf{v}_{t,i}$, where $\mathbf{v}_{t,i} := \sum_{j \in F_i} \mathbf{u}_{t,j}$. But the $\mathbf{u}_{t,j}$ were selected with the goal of making all $\|\mathbf{v}_{t,i}\|$ small, and so the imbalance of each F_i grows only slowly during the algorithm.

For proving Theorem 11.1.11, we will establish formal counterparts of the two intuitive claims above, as follows.

11.2.7 Claim. *The algorithm produces a coloring with probability close to* 1.

We will do this in Sect. 11.3.

11.2.8 Claim. *With probability close to* 1, *the discrepancy of the resulting (semi)coloring is of order* $O(H \log(mn))$, *where* H *is the hereditary vector discrepancy of* $\mathcal{F}$.

We will undertake this in Sect. 11.4.

11.3 Coordinate Walks

11.3.1 Let $\mathbf{x}_0, \ldots, \mathbf{x}_\ell$ be the sequence generated by the algorithm, and let $j \in \{1, 2, \ldots, n\}$ be a fixed index. We call the sequence $(\mathbf{x}_0)_j, (\mathbf{x}_1)_j, \ldots, (\mathbf{x}_\ell)_j$ the j-th *coordinate random walk*.

11.3.2 We say that the j-th coordinate random walk *terminates* if $(\mathbf{x}_\ell)_j \in \{\pm 1\}$. To prove Claim 11.2.7, it suffices to show that, for every j, the probability that the j-th coordinate random walk does *not* terminate is at most n^{-2}. Then, with probability at least $1 - \frac{1}{n}$, all of the coordinate walks terminate (by the union bound).

- We note that this argument does not use any kind of independence among the coordinate walks. Necessarily so, since the whole point of the algorithm is that the coordinate walks are highly correlated!

11.3.3 To simplify notation, let us fix j and write $X_t := (\mathbf{x}_t)_j - (\mathbf{x}_{t-1})_j$, $t = 1, 2, \ldots, \ell$.

11.3.4 Let $t_0 \le \ell$ be the last step of the j-th coordinate walk for which $(\mathbf{x}_{t_0})_j \in (-1, 1)$.

- By the rules of the algorithm, for $t \le t_0$ we have $X_t = (\boldsymbol{\Delta}_t)_j = \sigma \boldsymbol{\gamma}_t^T \mathbf{u}_{t,j}$ for some unit vector $\mathbf{u}_{t,j}$, where $\boldsymbol{\gamma}_t$ is n-dimensional Gaussian, independent of $\mathbf{u}_{t,j}$. Thus, as was mentioned in 9.3.3, X_t has the one-dimensional normal distribution $N(0, \sigma^2)$.
- We want to claim that $X_1, \ldots, X_{t_0}$ are independent random variables. But one has to be careful:
  - First, t_0 itself is not independent of $X_1, X_2, \ldots$, so even the formulation of such a claim may not be clear.
  - Moreover, the vector $\mathbf{u}_{t,j}$ depends on the previous history of the algorithm (more precisely, it depends on the set A_t, and through it on the earlier random choices made by the algorithm).
- Thus, we formulate our claim of independence in the following way. Let $Z'_1, \ldots, Z'_\ell$ be a new sequence of independent random variables, each with the $N(0, \sigma^2)$ distribution, also independent of everything in the algorithm. We define another sequence $Z_1, Z_2, \ldots, Z_\ell$ of random variables by

$$Z_t := \begin{cases} X_t & \text{for } t \le t_0 \\ Z'_t & \text{for } t > t_0. \end{cases}$$

  We claim that $Z_1, \ldots, Z_\ell$ are independent.

  - Indeed, if we fix the values of $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{t-1}$ in the algorithm and also the values of the auxiliary variables $Z'_1, \ldots, Z'_{t-1}$ arbitrarily, the values of $Z_1, \ldots, Z_{t-1}$ are determined uniquely, while Z_t has the $N(0, \sigma^2)$ distribution.
  - This easily implies the claimed independence of $Z_1, \ldots, Z_\ell$; the reader is invited to give a formal argument in Exercise 11.4.

11.3.5 According to the way the Z_t were defined, if the j-th coordinate walk does *not* terminate, then all the partial sums $Z_1 + Z_2 + \cdots + Z_t$ belong to $(-1, 1)$, $t = 1, 2, \ldots, \ell$. Thus we are left with the task of proving the following.

Lemma. *Let $Z_1, Z_2, \ldots, Z_\ell$ be independent random variables, each with the $N(0, \sigma^2)$ distribution. Then the probability that all of the partial sums $\sum_{i=1}^{t} Z_i$, $t = 1, 2, \ldots, \ell$, belong to the interval $(-1, 1)$ is at most $e^{-c_1 \lfloor \sigma^2 \ell \rfloor}$, for a suitable constant $c_1 > 0$.*

11.3.6 Here we can see the reason for choosing the walk length ℓ as we did, namely, $\ell := C_1 \sigma^{-2} \log n$. For this ℓ we get $e^{-c_1 \lfloor \sigma^2 \ell \rfloor} = e^{-c_1 \lfloor C_1 \log n \rfloor} \le n^{-2}$ (for $C_1 := 3/c_1$, and n sufficiently large). So Claim 11.2.7 follows from the lemma.

11.3.7 It remains to prove the lemma. The asymptotic value of the probability in question is known quite precisely in the theory of random walks.

Here is a quick proof, which gives only a rough bound, but sufficient for our purposes.

- Let $k := \sigma^{-2}$ (assuming for convenience that this is an integer). Let us partition the sequence $Z_1, Z_2, \ldots$ into contiguous blocks of length k, and let S_j be the sum of the j-th block. Formally, $S_j := \sum_{i=(j-1)k+1}^{jk} Z_i$. The number of full blocks is $\lfloor \ell/k \rfloor$.
- **Fact:** if X, Y are independent $N(0,1)$ random variables, and $a, b \in \mathbb{R}$, then $aX + bY \sim N(0, a^2 + b^2)$.
  - This is called the *2-stability* of the normal distribution.
  - **Sketch of a proof:** We may assume $a^2+b^2 = 1$ (re-scaling). The vector (X, Y) has the 2-dimensional standard normal distribution, rotationally symmetric. Thus, its scalar product with an arbitrary unit vector has the 1-dimensional $N(0,1)$ distribution. It remains to observe that $aX + bY$ is the scalar product of (X, Y) with (a, b).
- So each S_j has the standard normal distribution $N(0, k\sigma^2) = N(0,1)$. Thus, $\mathrm{Prob}[|S_j| \geq 2] \geq c_0$ for a suitable positive c_0 (by looking at a table of the normal distribution we can find that $c_0 \approx 0.0455$).
- If $\sum_{i=1}^{t} Z_i \in (-1, 1)$ for all $t = 1, 2, \ldots, \ell$, then necessarily $|S_j| < 2$ for all j. The S_j are independent, and thus the probability of the latter is at most $(1-c_0)^{\lfloor \ell/k \rfloor} = e^{-c_1 \lfloor \sigma^2 \ell \rfloor}$. The lemma is proved, and so is Claim 11.2.7.

11.4 Set Walks

11.4.1 It remains to prove Claim 11.2.8; concretely, we will prove

$$\mathrm{Prob}[\mathrm{disc}(\mathcal{F}, \mathbf{x}_\ell) > D_{\max}] \leq \frac{1}{n},$$

where $D_{\max} = O(H \log(mn))$ is the desired bound on the discrepancy.

11.4.2 Let us fix a set $F_i \in \mathcal{F}$, and let $D_i := \sum_{j \in F_i} (\mathbf{x}_\ell)_j$ be its imbalance in the final (semi)coloring $\mathbf{x}_\ell$. We will prove that $\mathrm{Prob}[|D_i| > D_{\max}] \leq \frac{1}{mn}$ for every i, and Claim 11.2.8 will follow by the union bound.

11.4.3 We recall how the j-th coordinate of the current semicoloring $\mathbf{x}_t$ develops as t goes from 0 to ℓ.

- It starts with $(\mathbf{x}_0)_j = 0$, then it changes by the random increments $(\boldsymbol{\Delta}_t)_j$, $t = 1, 2, \ldots, t_0$, then at some step $t_0 + 1$ it gets truncated to $+1$ or -1, and then it stays fixed until the end (it may also happen, with some small probability, that $t_0 = \ell$ and no truncation occurs).
- Since $(\boldsymbol{\Delta}_t)_j = 0$ for $t > t_0 + 1$, we can write

$$(\mathbf{x}_\ell)_j = \sum_{t=1}^{\ell} (\mathbf{\Delta}_t)_j + T_j,$$

where T_j is a "truncation effect," reflecting the fact that $(\mathbf{x}_{t_0+1})_j$ equals ± 1 and not $(\mathbf{x}_{t_0} + \mathbf{\Delta}_{t_0+1})_j$.

- We have $|T_j| \le |(\mathbf{\Delta}_{t_0+1})_j|$, and as we know, $(\mathbf{\Delta}_{t_0+1})_j \sim N(0, \sigma^2)$.
- Here is where our choice $\sigma := 1/(C_0 n \sqrt{\log n}\,)$ comes from: we will now show that for σ this small, all truncation effects are negligible with probability close to 1. Quantitatively, we claim that, for each j,

$$\mathrm{Prob}\left[|T_j| > \tfrac{1}{n}\right] \le \tfrac{1}{n^3}$$

(both $\frac{1}{n}$ and $\frac{1}{n^3}$ are chosen somewhat arbitrarily here; we could as well take $\frac{1}{n^{10}}$).

- **Proof:**
  - We just employ a tail bound for the standard normal distribution, e.g., Lemma 9.3.2. For a standard normal random variable Z, that formula gives $\mathrm{Prob}[|Z| \ge \lambda] \le e^{-\lambda^2/2}$ for all $\lambda \ge 1$.
  - In our situation,

$$\begin{aligned}\mathrm{Prob}\left[|T_j| > \frac{1}{n}\right] &\le \mathrm{Prob}\left[|\sigma Z| \ge \frac{1}{n}\right] = \mathrm{Prob}\left[|Z| \ge \frac{1}{\sigma n}\right] \\ &\le e^{-\sigma^{-2} n^{-2}/2} = e^{-(C_0^2 \log n)/2} \le \frac{1}{n^3}\end{aligned}$$

  for a suitable C_0.

11.4.4 Thus, with probability at least $1 - \frac{1}{n^2}$, the total contribution of the truncation effects T_j to the discrepancy of each set F_i is at most 1. So instead of D_i, it suffices to bound the "pure random walk" quantity

$$\begin{aligned}\tilde{D}_i &:= \sum_{j \in F_i} \sum_{t=1}^{\ell} (\mathbf{\Delta}_t)_j = \sum_{t=1}^{\ell} \sum_{j \in F_i} (\mathbf{\Delta}_t)_j \\ &= \sum_{t=1}^{\ell} \sum_{j \in F_i} \sigma \boldsymbol{\gamma}_t^T \mathbf{u}_{t,j} = \sum_{t=1}^{\ell} \sigma \boldsymbol{\gamma}_t^T \mathbf{v}_{t,i},\end{aligned}$$

where $\mathbf{v}_{t,i} = \sum_{j \in F_i} \mathbf{u}_{t,j}$.

11.4.5 Now, finally, the careful choice of the $\mathbf{u}_{t,j}$ (see 11.2.5) comes into play: we know that $\|\mathbf{v}_{t,i}\| \le H$ for all t and i. Writing $Y_t := \sigma \boldsymbol{\gamma}_t^T \mathbf{v}_{t,i}$ (we recall that i is considered fixed), we get that $Y_t \sim N(0, \beta_t^2)$, where $0 \le \beta_t \le \sigma H$.

- Intuitively, things should be simple here: the sum $\tilde{D}_i = Y_1 + \cdots + Y_\ell$ should have a distribution like $N(0, \ell\sigma^2 H^2)$, and hence we should get an exponential tail bound; the probability of $|\tilde{D}_i|$ exceeding $\sigma H\sqrt{\ell}$, the standard deviation, λ times, should behave like $e^{-\lambda^2/2}$.

- However, unlike in the case of the coordinate walks, we cannot claim that the Y_t are independent, because the variance β_t^2 of Y_t does depend on the random choices made by the algorithm in step 1 through $t-1$.
- So we need a more sophisticated and technical tool, such as the following lemma.

11.4.6 Lemma. *Let $W_1, \ldots, W_\ell$ be independent random variables on some probability space, and let Y_t be a function of $W_1, \ldots, W_t$, $t = 1, 2, \ldots, \ell$. Suppose that, conditioned on $W_1, \ldots, W_{t-1}$ attaining some arbitrary values $w_1, \ldots, w_{t-1}$, the distribution of Y_t is $N(0, \beta_t^2)$, where β_t may depend on $w_1, \ldots, w_{t-1}$, but we always have $\beta_t \le \beta$. Then $Y := Y_1 + \cdots + Y_\ell$ satisfies the tail bound*

$$\mathrm{Prob}\left[|Y| > \lambda\beta\sqrt{\ell}\right] \le 2e^{-\lambda^2/2}, \quad \textit{for all } \lambda \ge 0.$$

- We will use the lemma with $W_t := \boldsymbol{\gamma}_t$. Our λ is $\frac{D_{\max}}{\sigma H\sqrt{\ell}}$, where $D_{\max} = C_2 H \log(mn)$, $\sigma = 1/(C_0 n\sqrt{\log n},)$, and $\ell = C_1\sigma^{-2}\log n$. We thus calculate that $\lambda \ge C_3\sqrt{\log(mn)}$, with a constant C_3 that can be made as large as needed by adjusting the constant C_2 from $D_{\max}$. Then

$$\mathrm{Prob}\left[|\tilde{D}_i| > D_{\max}\right] \le 2e^{-\lambda^2/2} \le \frac{1}{n^2m^2},$$

  and, as announced, Claim 11.2.8 follows by the union bound.
- Some readers may have recognized that we are really talking about a *martingale* in the lemma (while those not familiar with martingales may ignore this remark). They may also know *Azuma's inequality*, which gives a tail bound for martingales. That inequality assumes pointwise bounded martingale differences, and thus is not directly applicable in our setting. However, its *proof* is applicable with only a minor adjustment, and this is how we prove the lemma.

11.4.7 Proof of the lemma:

- We will prove the upper tail bound, $\mathrm{Prob}\left[Y > \lambda\beta\sqrt{\ell}\right] \le e^{-\lambda^2/2}$; the lower one follows by applying the upper tail to $-Y$.
- As in the usual proof of the Chernoff inequality, the main trick is to bound the *moment generating function* $G(u) := \mathbf{E}\left[e^{uY}\right]$, where u is a real parameter.
- By induction on t, we will show that $\mathbf{E}\left[e^{u(Y_1+\cdots+Y_t)}\right] \le e^{u^2 t\beta^2/2}$ (the basis is for $t = 0$).
- The expectation is over a random choice of $W_1, \ldots, W_t$. We can do the expectation over W_t first, regarding $W_1, \ldots, W_{t-1}$ fixed, and then the expectation of the result over $W_1, \ldots, W_{t-1}$ (this is Fubini's theorem).

- For $W_1, \ldots, W_{t-1}$ fixed, $Y_1, \ldots, Y_{t-1}$ are fixed, and so

$$\begin{aligned}\mathbf{E}_{W_t}\left[e^{u(Y_1+\cdots+Y_t)}\right] &= e^{u(Y_1+\cdots+Y_{t-1})}\mathbf{E}_{W_t}\left[e^{uY_t}\right] \\ &= e^{u(Y_1+\cdots+Y_{t-1})}e^{u^2\beta_t^2/2} \le e^{u^2\beta^2/2}e^{u(Y_1+\cdots+Y_{t-1})}.\end{aligned}$$

  - Here we have used that, with $W_1, \ldots, W_{t-1}$ fixed, we have $Y_t \sim N(0, \beta_t^2)$. Then $\mathbf{E}\left[e^{uY_t}\right] = e^{u^2\beta_t^2/2}$ is a standard fact about the normal distribution, which is also easy to check (a substitution converts the required integral to $\int_{-\infty}^{\infty} e^{-x^2/2}\mathrm{d}x$).
- Now we take expectation over $W_1, \ldots, W_{t-1}$ and use the inductive hypothesis, and thus finish the inductive step. We have shown that $\mathbf{E}\left[e^{uY}\right] \le e^{u^2\ell\beta^2/2}$.
- Next we use the Markov inequality for the random variable e^{uY}:

$$\begin{aligned}\mathrm{Prob}\left[Y > \lambda\beta\sqrt{\ell}\right] &= \mathrm{Prob}\left[e^{uY} > e^{u\lambda\beta\sqrt{\ell}}\right] \\ &\le \mathbf{E}\left[e^{uY}\right]/e^{u\lambda\beta\sqrt{\ell}} \le e^{u^2\ell\beta^2/2 - u\lambda\beta\sqrt{\ell}}.\end{aligned}$$

Substituting $u := \lambda/(\beta\sqrt{\ell})$ gives the required tail bound, and the lemma is proved. So are Claim 11.2.8 and Theorem 11.1.11.

Exercises

11.1 (a) Show that every set system on n points has vector discrepancy at most $\sqrt{n}$.

(b) Show that this bound is tight, possibly up to a multiplicative constant independent of n.

(c) Let $\mathcal{F}$ be a system of n sets on a set V of points, such that every point is contained in exactly r sets of $\mathcal{F}$, and for every two distinct points $i, j \in V$, there are exactly t sets $F \in \mathcal{F}$ with $\{i, j\} \subseteq F$. Prove that $\mathrm{vecdisc}(\mathcal{F}) \ge \sqrt{r-t}$.

Hint: Instead of $\max_{F\in\mathcal{F}}\left\|\sum_{j\in F}\mathbf{u}_j\right\|^2$, estimate $\sum_{F\in\mathcal{F}}\left\|\sum_{j\in F}\mathbf{u}_j\right\|^2$.

Remark: The so-called *Hadamard designs* provide set systems as above with $n = 4m+3$, $r = 2m+1$, and $t = m$ for infinitely many values of m, and thus they show that a system of n sets on n points can have vector discrepancy $\Omega(\sqrt{n})$.

11.2 Use the probabilistic method (and, in particular, the Chernoff bound for the sum of independent ± 1 random variables) to show that $\mathrm{disc}(\mathcal{F}) = O(\sqrt{n \log m})$ for every system of m sets on n points.

11.3 Use the probabilistic method to show the existence of set systems with n^2 sets on n points and with discrepancy $\Omega(\sqrt{n \log n}\,)$. (Together with Exercise 11.1(a) this shows that the gap between vecdisc and disc can be at least of order $\sqrt{\log m}$, for $m = n^2$. The complete set system 2^V exhibits a similar gap for $m = 2^n$.)

Hint: Fix an arbitrary coloring χ and show that the discrepancy of χ for a system of n^2 independent random sets is below $c\sqrt{n \log n}$ with probability smaller than 2^{-n}. A random set is obtained by including each point independently with probability $\frac{1}{2}$.

11.4 Let $Z_1, Z_2, \ldots, Z_n$ be real random variables, and suppose that there are distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ such that for each $t = 1, 2, \ldots, n-1$ and for every $z_1, \ldots, z_{t-1} \in \mathbb{R}$, the conditional distribution of Z_t given that $Z_1 = z_1$, …, $Z_{t-1} = z_{t-1}$ is $\mathcal{D}_t$. Prove rigorously that $Z_1, \ldots, Z_t$ are independent. (Recall that real random variables $X_1, \ldots, X_n$ are independent if for every index set $I \subseteq \{1, 2, \ldots, n\}$ and for every $a_1, \ldots, a_n \in \mathbb{R}$, we have $\mathrm{Prob}[X_i \le a_i \text{ for all } i \in I] = \prod_{i \in I} \mathrm{Prob}[X_i \le a_i]$.)

http://www.springer.com/978-3-642-22014-2

Approximation Algorithms and Semidefinite Programming
Gärtner, B.; Matousek, J.
2012, XI, 251 p., Hardcover
ISBN: 978-3-642-22014-2