Fast Discrepancy Minimization with Hereditary Guarantees

Kasper Green Larsen*
Aarhus University
larsen@cs.au.dk

Abstract

Efficiently computing low discrepancy colorings of various set systems, has been studied extensively since the breakthrough work by Bansal (FOCS 2010), who gave the first polynomial time algorithms for several important settings, including for general set systems, sparse set systems and for set systems with bounded hereditary discrepancy. The hereditary discrepancy of a set system, is the maximum discrepancy over all set systems obtainable by deleting a subset of the ground elements. While being polynomial time, Bansal's algorithms were not practical, with e.g. his algorithm for the hereditary setup running in time $\Omega(mn^{4.5})$ for set systems with m sets over a ground set of n elements. More efficient algorithms have since then been developed for general and sparse set systems, however, for the hereditary case, Bansal's algorithm remains state-of-the-art. In this work, we give a significantly faster algorithm with hereditary guarantees, running in $O(mn^2 \lg(2 + m/n) + n^3)$ time. Our algorithm is based on new structural insights into set systems with bounded hereditary discrepancy. We also implement our algorithm and show experimentally that it computes colorings that are significantly better than random and finishes in a reasonable amount of time, even on set systems with thousands of sets over a ground set of thousands of elements.

1 Introduction

In discrepancy minimization, we are given a set system $\mathcal{S} = \{S_1, \ldots, S_m\}$, with each set S_i being a subset of a universe $\mathcal{U} = \{u_1, \ldots, u_n\}$ of n elements. The goal is to find a red-blue coloring of the elements of $\mathcal{U}$ such that each set S_i is colored as evenly as possible. More formally, if we define the $m \times n$ incidence matrix A such that entry (i, j) is 1 if $u_j \in S_i$ and 0 otherwise, then we seek a "coloring" $x \in \{-1, 1\}^n$ for which the discrepancy $\mathrm{disc}(A, x) := \|Ax\|_\infty = \max_i |(Ax)_i|$ is as small as possible. Discrepancy minimization has been studied for decades and has found numerous applications in theoretical computer science, see e.g. the textbooks by Matousek [18] and Chazelle [8].

A natural generalization of discrepancy minimization is to allow arbitrary real matrices $A \in \mathbb{R}^{m \times n}$, rather than only incidence matrices corresponding to set systems. For a matrix $A \in \mathbb{R}^{m \times n}$, its discrepancy is defined as $\mathrm{disc}(A) = \min_{x \in \{-1,1\}^n} \mathrm{disc}(A, x)$, i.e. as the best possible coloring achievable. Understanding the discrepancy of various matrices has been the focus of much research. The three main directions pursued in the area can be roughly categorized as follows:

1. Set systems corresponding to geometric ranges. Here the matrices A of interest correspond to incidence matrices, where the elements $\mathcal{U} = \{u_1, \ldots, u_n\}$ typically are points in $\mathbb{R}^d$ and each set S_i corresponds to the subset of $\mathcal{U}$ that lies inside a geometric range such as a half-space or an axis-aligned rectangle [1, 17, 19, 13].

2. General set systems, where one only assumes that $|a_{i,j}| \le 1$ for all (i, j). The classic "Six Standard Deviations Suffice" result by Spencer [24] is one of the pillars of this line of research, showing that for any $n \times n$ matrix A with $|a_{i,j}| \le 1$, we have $\mathrm{disc}(A) \le 6\sqrt{n}$. This is highly surprising as a random coloring x for most matrices A would result in a discrepancy of $\mathrm{disc}(A, x) = \Omega(\sqrt{n \lg n})$ with high probability.

3. Sparse set systems, where each element of the universe $u_j \in \mathcal{U}$ is contained in at most t sets S_i. The Beck-Fiala conjecture [6], which is a special case of the Komlós conjecture, is central in this direction, asserting that any such set system can be colored to achieve a discrepancy of only $O(\sqrt{t})$ (the Komlós conjecture states that any sequence of vectors with unit ℓ_2 norm can be assigned signs such that their sum has ℓ_∞ norm bounded by a constan). While not yet being resolved, the best upper bound, due to Banaszczyk [3] comes close by guaranteeing the existence of a coloring of discrepancy $O(\sqrt{t \lg n})$.

Constructive Discrepancy Minimization. Much of the early work on discrepancy minimization focused only on showing the existence of low-discrepancy colorings [24, 3, 25] and it was initially unclear whether there exist polynomial time algorithms for computing near-optimal colorings. Indeed, a hardness result by Charikar et al. [7] showed that it is NP-hard to distinguish whether $\mathrm{disc}(A)$ is 0 or $\Omega(\sqrt{n})$ for matrices A corresponding to set systems. This rules out any polynomial time multiplicative approximation algorithm for discrepancy minimization. While this might seem like an insurmountable obstacle for interesting algorithmic results, Bansal's [4] seminal paper presented a sequence of polynomial time algorithms with different exciting approximation guarantees. With respect to the main directions mentioned above, Bansal gave an algorithm for general set systems guaranteeing to find a coloring of discrepancy $O(\sqrt{n})$ for $n \times n$ set systems. In some sense, this can be thought of as an *additive* $O(\sqrt{n})$ approximation. For the sparse set system case, he gave an algorithm guaranteeing a discrepancy of $O(\sqrt{t} \lg n)$, getting close, but not quite matching the non-constructive bound by Banaszczyk [3]. Later work [5] has since then closed the gap and given a polynomial time algorithm matching Banaszczyk's bound. Bansal's final algorithmic result gives a coloring with discrepancy related to the so-called *hereditary discrepancy* of A [15]. The hereditary discrepancy of A, denoted $\mathrm{herdisc}(A)$, is a classic measure defined as the maximum discrepancy of any submatrix A' obtained by deleting a subset of the columns of A (equivalently, deleting a subset of the points in $\mathcal{U}$). With this definition, his algorithm finds a coloring x such that $\mathrm{disc}(A, x) = O(\lg n \cdot \mathrm{herdisc}(A))$. So while Charikar et al.'s NP-hardness result shows that it is generally hard to find colorings of discrepancy $o(\sqrt{n})$, even when such colorings exist, Bansal's result shows that this is in fact possible if all sub-matrices of A have low-discrepancy colorings. All of Bansal's algorithms are based on semi-definite programming (SDP) and are

not as such practically efficient. Concretely, his algorithm with hereditary guarantees needs to solve $\Omega(n^2)$ semi-definite programs, all consisting of m constraint matrices of size $n \times n$. With the current state-of-the-art SDP solvers [12], this requires time $\Omega(mn^{4.5})$, making it practically infeasible even for modest size matrices. Since Bansal's celebrated work, follow-up work by Lovett and Meka [16] gave an algorithm for the general set system case, running in time $O((m + n)n^2)$ and Alweiss et al. [2] recently gave an even faster algorithm for sparse set systems, guaranteeing $\mathrm{disc}(A, x) = O(\sqrt{t} \lg n)$ in time $O(mn)$, i.e. linear time in the input size. However, for the hereditary setup, nothing faster than Bansal's result is known. Giving such an algorithm is the main focus of this work.

Our Contribution.   In this work, we give a fast algorithm for finding low-discrepancy colorings when the hereditary discrepancy of the input matrix A is small. Concretely, we show the following:

Theorem 1. *There is a randomized algorithm that on any $m \times n$ input matrix $A \in \mathbb{R}^{m \times n}$, finds a coloring $x \in \{-1, 1\}^n$ for which $\mathrm{disc}(A, x) = O(\lg n \cdot \lg^{3/2} m \cdot \mathrm{herdisc}(A))$ in expected $O(mn^2 \lg(2 + m/n) + n^3)$ time.*

Since the input size is mn, the running time is only a factor $n \lg(2 + m/n)$ more than linear (at least for $m \geq n$). Comparing this to the $\Omega(mn^{4.5})$ running time of Bansal's algorithm, this is the first practical algorithm with hereditary guarantees. On the downside, we lose a factor $\lg^{3/2} m$ in the quality of the coloring found compared to Bansal's algorithm. To test the practicality of our algorithm, we implemented it and ran experiments on several types of matrices. The algorithm finishes in reasonable time (less than 30 minutes on a standard laptop) even on matrices of size 4000×4000 and 10000×2000. Moreover, despite the logarithmic factors in the theoretical analysis, it returns colorings that have far lower discrepancy than random colorings have (around a factor 10 smaller discrepancy on structured matrices), even when compared to repeatedly testing random colorings for the same amount of time as spent by our algorithm (around a factor 4 smaller discrepancy). See Section 4 for details.

1.1   Technical Contributions and Algorithm Overview

The main obstacle faced in the design of the new algorithm, is the current lack of structural understanding of matrices with low hereditary discrepancy. In more detail, the previous algorithm by Bansal only uses that A has low hereditary discrepancy to guarantee the feasibility of a carefully designed SDP. As such, the insight that a concrete SDP is feasible, does not seem to provide any line of attack beyond solving an SDP. Our first technical contribution is thus to demonstrate that matrices A with low hereditary discrepancy have very stringent requirements on the eigenvalues of $A^T A$. A connection between eigenvalues and hereditary discrepancy was earlier observed by Larsen [14] (other, slightly weaker/incomparable connections were also known [20, 9]) who proved:

Theorem 2 (Larsen [14])**.** *For an $m \times n$ real matrix A, let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ denote the eigenvalues of $A^T A$. For all positive integers $k \leq \min\{n, m\}$, we have*

$$\mathrm{herdisc}(A) \geq \frac{k}{2e} \sqrt{\frac{\lambda_k}{mn}}.$$

In Larsen's work, the same lower bound was also proved for the ℓ_2-version of hereditary discrepancy. Said briefly, the ℓ_2-discrepancy of a matrix $A \in \mathbb{R}^{m \times n}$ is defined as $\mathrm{disc}_2(A) = \min_{x \in \{-1, 1\}^n} \|Ax\|_2 / \sqrt{m}$ and the ℓ_2-hereditary discrepancy is again defined as the maximum over submatrices of A. For ℓ_2-discrepancy, this connection showed a clear path towards finding low discrepancy colorings. At a high level, the idea is to find a coloring $x \in \{-1, 1\}^n$ that is orthogonal to the $n/2$ eigenvectors corresponding to the largest eigenvalues of $A^T A$. For such x, it holds that $\|Ax\|_2^2 = x^T A^T A x \leq \lambda_{n/2} \cdot n$, which by the above relation implies $\|Ax\|_2 = O(\sqrt{\lambda_{n/2} \cdot n}) = O(\mathrm{herdisc}_2(A)\sqrt{m})$. By the definition $\mathrm{disc}_2(A, x) = \|Ax\|_2 / \sqrt{m}$, this gives a discrepancy of $O(\mathrm{herdisc}_2(A))$. This argument assumed that it was possible to find a coloring x orthogonal to the top $n/2$ eigenvectors. This is not quite the case, resulting in an additional $\sqrt{\lg n}$ factor in the ℓ_2-discrepancy of Larsen's algorithm by using the classic *partial coloring* technique. We will not discuss this further here.

2

For the more central ℓ_∞-discrepancy, the above connection to eigenvalues and eigenvectors seems very hard to exploit. Concretely, the mere fact that $\|Ax\|_2^2$ is small, says nothing interesting about $\|Ax\|_\infty$, and it is completely unclear that simple bounds on the eigenvalues of $A^T A$ may be useful for finding an $x \in \{-1,1\}^n$ with small $\mathrm{disc}(A,x)$. One of our main technical contributions, is to show that the connection to eigenvalues in Theorem 2 may be exploited for ℓ_∞-discrepancy as well. Concretely, we first observe that if we define V as the matrix having the $n/2$ eigenvectors of $A^T A$ with largest eigenvalues as rows, then $A(I - V^T V)$ cannot have more than $m/2$ rows of norm exceeding $O(\mathrm{herdisc}(A))$. The reason for this, is that the rows of $B = A(I - V^T V)$ are the projections of the rows of A onto the orthogonal complement of the top $n/2$ eigenvectors of $A^T A$ and therefore it must be the case that $B^T B$ has all eigenvalues bounded by $\lambda_{n/2}$. But $\mathrm{tr}(B^T B)$ is equal to the sum of squared row norms of B. If there were more than $m/2$ rows with norm more than $c \cdot \mathrm{herdisc}(A)$ for a large constant c, then we would have $\mathrm{tr}(B^T B) \geq (c/2)m\,\mathrm{herdisc}^2(A)$. But $\mathrm{tr}(B^T B)$ is the sum of the eigenvalues of $B^T B$. Since $B^T B$ has rank at most n, this implies that $B^T B$ has an eigenvalue of at least $cm\,\mathrm{herdisc}^2(A)/(2n)$. By Theorem 2, this is greater than $\lambda_{n/2}$ for c big enough, i.e. a contradiction.

With this observation in mind, a natural idea would be to find a coloring x in the orthogonal complement of the span of the rows of V. Such an x would behave well for the rows of norm $O(\mathrm{herdisc}(A))$, but not for the remaining up to $m/2$ rows of larger norm. To get around this, we instead pick only the top $n/(8\lg(8m/n))$ eigenvectors of $A^T A$ when forming V. Then we can again argue that there is at most $m/2$ rows of $B = A(I - V^T V)$ with norm exceeding $O(\mathrm{herdisc}(A)\lg(8m/n))$. We then focus on the submatrix B' of B obtained by deleting all but the $m/2$ rows of largest norm. We then find the top $n/(8\lg(8m/n))$ eigenvectors of $B'^T B'$ and add those as rows of V. We can then argue that this reduces the number of rows of $A(I - V^T V)$ of norm more than $O(\mathrm{herdisc}(A)\lg(8m/n))$ to $m/4$. Repeating this, where we delete all but the $m/2^i$ rows of largest norm from B, for $\lg(8m/n)$ rounds, leaves less than $n/8$ rows of norm exceeding $O(\mathrm{herdisc}(A)\lg(8m/n))$. We can finally add a basis for the subspace spanned by those $n/8$ rows to V. This results in a matrix V with $n/4$ rows, such that every row of $B = A(I - V^T V)$ has norm at most $O(\mathrm{herdisc}(A)\lg(8m/n))$. That is, we have shown the following theorem capturing a key structural property of matrices with low hereditary discrepancy:

Theorem 3. *For any $m \times n$ real matrix A with $m \geq n$, there is an $(n/4) \times n$ matrix V, having unit length orthogonal rows, such that all rows of $A(I - V^T V)$ have norm at most $O(\mathrm{herdisc}(A)\lg(2m/n))$. Moreover, such a matrix V can be computed in $O(mn^2\lg(2 + m/n))$ time.*

We remark that a similar result could be derived from the γ_2 norm characterization of hereditary discrepancy [21]. Concretely, it can be shown that a matrix A has a factorization $A = BC$ so that all columns of C have ℓ_2 norm at most 1 and all rows of B have ℓ_2 norm at most $\gamma_2(A) = O(\mathrm{herdisc}(A)\lg\mathrm{rank}(A))$. Then $C^T C$ has at most $n/2$ eigenvalues larger than 2. We could then let V have the top $n/2$ eigenvectors of $C^T C$ has rows, implying that $C(I - V^T V)$ has operator norm at most $\sqrt{2}$ and therefore all rows of $A(I - V^T V) = BC(I - V^T V)$ have norm bounded by $\sqrt{2}\gamma_2(A) = O(\mathrm{herdisc}(A)\lg\mathrm{rank}(A))$. Computing the factorization can be done by solving an SDP, which would be slower than our approach.

Having computed the matrix V in Theorem 3, we now seek a "random" coloring x in the orthogonal complement of the rows of V. Our basic approach for finding such a coloring, is to run the Edge-Walk algorithm by Lovett and Meka [16]. Here we start out with $x = 0$ and then repeatedly sample a random vector g in the orthogonal complement and add it to x. When a coordinate x_i reaches 1, we add e_i as a row of V, ensuring that all further g have $g_i = 0$ and thus leaves the coordinate unchanged. Moreover, whenever $|\langle a_i, g\rangle|$ exceeds $O(\mathrm{herdisc}(A)\lg^{3/2}(8m/n))$, we add a_i as a row of V, ensuring that $\langle a_i, g\rangle$ remains unchanged. This is similar in spirit to the Edge-Walk algorithm, with the key new observation being that the change in $|\langle a_i, g\rangle|$ is proportional to the length of the projection of a_i onto the orthogonal complement of the rows of V, and thus proportional to $\mathrm{herdisc}(A)$. On top of this, we pay a logarithmic factor due to partial coloring, which is similar to most previous discrepancy minimization algorithms.

As a second technical contribution, we observe that the Edge-Walk algorithm would need $\Omega(n^2)$ steps of sampling a random g (for the familiar reader, the Edge-Walk algorithm would need to add e_i to V when $|x_i|$ exceeds $1 - 1/n$ and this would require a small step size to avoid $|x_i|$ exceeding 1). We carefully reduce

this to only $O(n)$ steps by modifying the update performed in each step of the algorithm. The Edge-Walk algorithm always adds ηg to x for an η small enough that no coordinate of x_i ever changes from less than $1-1/n$ in absolute value to above 1. Then upon termination, each coordinate of x may be rounded to either -1 or $+1$. This approach requires a very small step size η. What we do instead, is that we compute the largest possible $\mu > 0$, such that adding μg or $-\mu g$ to x ensures that no coordinate x_i exceeds 1 in absolute value. We then add $\min\{\eta, \mu\}g$ to x. At a high level, this allows the Edge-Walk algorithm to temporarily violate $|x_i| \leq 1$, but when it happens, the step size is slightly reduced in that iteration, making $|x_i| = 1$ instead. By a careful analysis, this reduces the number of iterations to just $O(n)$.

Let us finally comment on an alternative approach for obtaining a result similar to ours. In the work by Dadush et al. [10], it is shown that running an algorithm by Rothvoss [22] with the convex body $K = \{x : \|Ax\|_\infty \leq 1\}$ can obtain a coloring whose discrepancy is bounded by the hereditary discrepancy of A. To run Rothvoss' algorithm, one needs to projet vectors onto K. This can be done by solving a linear program [11].

In Section 2, we introduce some inequalities and facts needed for the analysis of our algorithm and then proceed in Section 3 to present our new algorithm and its analysis.

2　Preliminaries

We will need a few inequalities regarding normal distributed random variables for the analysis of our algorithm. The first is a standard fact that we state without proof:

Claim 1. *Let $G \sim \mathcal{N}(0,1)$. Then, for any $\lambda > 0$, $\Pr[|G| \geq \lambda] \leq 2\exp(-\lambda^2/2)$.*

Secondly, we need the following fact regarding the distribution of the inner product between a vector a and the projection of a vector with i.i.d. normal distributed entries onto a subspace:

Claim 2. *Let $V \in \mathbb{R}^{\ell \times n}$ be a matrix with $\ell \leq n$ orthogonal unit length rows. Let $g \in \mathbb{R}^n$ be sampled with n i.i.d. $\mathcal{N}(0,1)$ distributed entries and let $a \in \mathbb{R}^n$ be an arbitrary vector. Then*

$$\langle a, (I - V^T V)g \rangle \sim \mathcal{N}(0, \|a(I - V^T V)\|^2).$$

A proof of Claim 2 can be found in Section 5.

Finally, we need a version of Azuma's inequality for Martingales with Subgaussian tails.

Azuma for Martingales with Subgaussian Tails. A sequence of random variables $Z_1, Z_2, \ldots$ is called a *martingale difference sequence* with respect to another sequence of random variables $X_1, X_2, \ldots$, if for any t, Z_{t+1} is measurable wrt. the sigma algebra generated by $X_1, \ldots, X_t$, and $\mathbb{E}[Z_{t+1} \mid X_1, \ldots, X_t] = 0$ with probability 1.

Theorem 4 (Shamir [23])**.** *Let $Z_1, \ldots, Z_T$ be a martingale difference sequence with respect to a sequence $X_1, \ldots, X_T$, and suppose there are constants $b > 1$, $c > 0$ such that for any t and any $a > 0$, it holds that*

$$\max\{\Pr[Z_t > a \mid X_1, \ldots, X_{t-1}], \Pr[Z_t < -a \mid X_1, \ldots, X_{t-1}]\} \leq b\exp(-ca^2).$$

Then for any $\delta > 0$, it holds with probability at least $1 - \delta$ that

$$\left| \frac{1}{T} \sum_{t=1}^{T} Z_t \right| \leq 2\sqrt{\frac{28b\lg(1/\delta)}{cT}}.$$

We remark that [23] proves the theorem without absolute values on the sum of Z_t's, and also without the factor 2 on the right hand side. However, defining $Z_i = -Z_i$ for all i, symmetry and a union bound over the original martingale difference sequence and the negated one implies the above.

4

3 Algorithm

In this section, we present our new algorithm with hereditary guarantees. As discussed in Section 1.1, one of its key ingredients is a new structural property of matrices with low hereditary discrepancy (Theorem 3). We start in Section 3.1 by proving Theorem 3. We then proceed in Section 3.2 to present our algorithm for discrepancy minimization with hereditary guarantees. Throughout this section, we assume $m \geq n$. The case of $m < n$ can be reduced to $m = n$ via standard techniques in time $O(mn^2 + n^3)$. For the interested reader, we have included a sketch of this reduction in Section 5. This reduction accounts for the $O(n^3)$ term in the running time stated in Theorem 1.

3.1 Structure of Matrices with Low Hereditary Discrepancy

The goal of this section is to prove Theorem 3, i.e. to give an algorithm that on an $m \times n$ matrix A (with $m \geq n$), outputs an $\ell \times n$ matrix V, with $\ell \leq n/4$, such that all rows of V are orthogonal unit length vectors and all rows of $A(I - V^T V)$ have norm $O(\mathrm{herdisc}(A) \lg(2m/n))$. The algorithm is based on the ideas outlined in Section 1.1 and uses as a subroutine the simple algorithm presented as Algorithm 1. Algorithm 1 takes as input an $\ell \times n$ matrix V assumed to have unit length orthogonal rows, as well as another vector $s \in \mathbb{R}^n$. It then computes $s(I - V^T V)$, which is the projection of s onto the orthogonal complement of the rows of V. If this vector is non-zero, it is scaled to unit length and added as a row of V. We will use Algorithm 1 as a subroutine to build matrices V with orthogonal unit length rows. Concretely, invoking Algorithm 1 with an input vector s, guarantees that s afterwards lies in the span of the rows of V and that V continues to have unit length orthogonal rows. It is nothing more than the Gram-Schmidt process.

Algorithm 1: Orthogonalize

    Input: Vector $s \in \mathbb{R}^n$, Matrix $V \in \mathbb{R}^{\ell \times n}$ with ℓ rows forming an orthonormal basis
    Result: Matrix $V' \in \mathbb{R}^{\ell' \times n}$ with at most $\ell + 1$ rows forming an orthonormal basis
1 Let $s' = s(I - V^T V)$.
2 **if** $s' \neq 0$ **then**
3   Add $s'/\|s'\|$ as a row of V.
4 **return** V

Using Algorithm 1 as a subroutine, we are ready to present our algorithm for computing the matrix V with the guarantees claimed in Theorem 3. The algorithm is presented as Algorithm 2. As discussed in Section 1.1, the idea is to build V iteratively. In the i'th step, all but the $m/2^i$ rows of $A(I - V^T V)$ of largest norm are deleted. The eigenvalues and eigenvectors of the resulting matrix are then computed and the top $O(n/\lg(m/n))$ eigenvectors are added to V.

We prove the following guarantees on the output of Algorithm 2 and remark that Theorem 3 is an immediate corollary of Lemma 1 and Lemma 2 below.

Lemma 1. *Let $V \in \mathbb{R}^{\ell \times n}$ be the matrix returned by Algorithm 2 on input matrix $A \in \mathbb{R}^{m \times n}$ with $m \geq n$. Then $\ell \leq n/4$ and every row a_i of $A(I - V^T V)$ has norm no more than $48e \lg_2(8m/n) \mathrm{herdisc}(A)$.*

Proof. We start by arguing that the rows of V form an orthonormal basis. This is clearly true initially as V has no rows. Now consider adding the eigenvectors $\{\eta_1, \ldots, \eta_{n/(8 \lg_2(8m/n))}\}$ as rows of V in line 6 of Algorithm 2. These are orthogonal to each other and have unit length. Moreover, they reside in the span of the rows of $\bar{B}$ and hence also in the span of the rows of B. Since all rows of B are orthogonal to all rows in V (since $B = A(I - V^T V)$), it follows that adding the eigenvectors to V maintains that V is an orthonormal basis. Finally, the last for-loop clearly preserves that the rows of V form an orthonormal basis.

We now prove by induction that upon completing the i'th iteration of the first for-loop (lines 2 to 6), there are no more than $m/2^i$ rows in $A(I - V^T V)$ of norm exceeding $48e \lg_2(8m/n) \mathrm{herdisc}(A)$. For $i = 0$ this trivially holds. For the inductive step, let $i \in \{1, \ldots, \lg_2(8m/n)\}$ and assume that after iteration $i - 1$, there are no more than $m/2^{i-1}$ rows of norm more than $48e \lg_2(8m/n) \mathrm{herdisc}(A)$. For the i'th iteration,

5

Algorithm 2: ProjectToSmallRows

Input: Matrix $A \in \mathbb{R}^{m \times n}$
Result: Matrix $V \in \mathbb{R}^{\ell \times n}$ with $\ell \le n/4$ rows forming an orthonormal basis
1 $V = 0 \in \mathbb{R}^{0 \times n}$ `// Initially empty matrix`
2 for $i = 1, \dots, \lg_2(8m/n)$ **do**
3   Let $B = A(I - V^T V)$.
4   Let $\bar{B}$ be the submatrix obtained from B by deleting all but the $m/2^{i-1}$ rows of largest norm.
5   Compute the eigenvalues $\mu_1 \ge \dots \ge \mu_n \ge 0$ and corresponding eigenvectors $\eta_1, \dots, \eta_n$ of $\bar{B}^T \bar{B}$.
6   Add $\{\eta_1, \dots, \eta_{n/(8\lg_2(8m/n))}\}$ as rows of V.
7 Let $B = A(I - V^T V)$.
8 Let $r_1, \dots, r_{n/8}$ be the $n/8$ rows of B with largest norm.
9 for $j = 1, \dots, n/8$ **do**
10   Orthogonalize(r_j, V)   `// Add `r_j` as row of `V` via Algorithm 1`
11 return V

---

this implies that all rows not in $\bar{B}$ have norm at most $48e \lg_2(8m/n) \operatorname{herdisc}(A)$. Since the norm of any row of $A(I - V^T V)$ may only decrease by adding more rows to V, we conclude that this remains the case for rows not in $\bar{B}$. What is left to show, is that adding $\{\eta_1, \dots, \eta_{n/(8\lg_2(8m/n))}\}$ as rows of V, leaves at most $m/2^i$ rows in $\bar{B}(I - V^T V) \in \mathbb{R}^{m/2^{i-1} \times n}$ with norm exceeding $48e \lg_2(8m/n) \operatorname{herdisc}(A)$ (before adding the new rows to V in line 6 of iteration i, it holds that $\bar{B} = \bar{B}(I - V^T V)$, as all rows of $\bar{B}$ are already orthogonal to all rows of V).

Assume for the sake of contradiction that there are more than $m/2^i$ rows in $\bar{B}(I - V^T V)$ with norm exceeding $48e \lg_2(8m/n) \operatorname{herdisc}(A)$ after adding $\{\eta_1, \dots, \eta_{n/(8\lg_2(8m/n))}\}$ as rows of V in line 6 of iteration i of the first for-loop. Pick an arbitrary orthonormal basis $u_1, \dots, u_{n-\ell}$ for the orthogonal complement C of the rows of V (where ℓ is the number of rows of V). Then all rows b_i of $\bar{B}(I - V^T V)$ lie in the span of $u_1, \dots, u_{n-\ell}$. It follows that $\sum_i \sum_j \langle b_i, u_j \rangle^2 = \sum_i \|b_i\|^2 > (m/2^i)(48e \lg_2(8m/n) \operatorname{herdisc}(A))^2$. Averaging over all u_j, there must exists a j with $\sum_i \langle b_i, u_j \rangle^2 > (m/2^i)(48e \lg_2(8n/m) \operatorname{herdisc}(A))^2/n$. Thus we have a unit vector $v = u_j$ in C with

$$\|\bar{B}v\|^2 > (m/2^i)(48e \lg_2(8m/n) \operatorname{herdisc}(A))^2/n.$$

Since v is orthogonal to $\eta_1, \dots, \eta_{n/(8\lg_2(8m/n))}$, we also have that $\|\bar{B}v\|^2 = v\bar{B}^T \bar{B}v \le \mu_{n/(8\lg_2(8m/n))}$. Combining the two yields

$$\mu_{n/(8\lg_2(8m/n))} > (m/2^i)(48e \lg_2(8m/n) \operatorname{herdisc}(A))^2/n.$$

Now let $\bar{A}$ be the submatrix of A obtained by deleting the same rows as when choosing $\bar{B}$. Then $\operatorname{herdisc}(A) \ge \operatorname{herdisc}(\bar{A})$. What remains is to relate $\operatorname{herdisc}(\bar{A})$ to the eigenvalues of $\bar{B}$. Recall that $\bar{B} = \bar{A}(I - V'^T V')$ where V' is the matrix V at the beginning of the loop iteration (before adding eigenvectors η_j in step 6). We may assume that all the eigenvectors η_j corresponding to non-zero eigenvalues of $\bar{B}^T \bar{B}$ are orthogonal to the rows of V'. Hence $\bar{B}\eta_j = \bar{A}\eta_j$. For any k, the vectors $\eta_1, \dots, \eta_k$ are thus orthogonal and have $\eta_j^T \bar{A}^T \bar{A}\eta_j \ge \mu_k$. It follows by the min-max characterization of eigenvalues that the k'th largest eigenvalue of $\bar{A}^T \bar{A}$ is at least μ_k. Moreover, Theorem 2 gives us, with $k = n/(8\lg_2(8m/n))$, that

$$\operatorname{herdisc}(A) \ge \operatorname{herdisc}(\bar{A}) > \frac{n}{16e \lg_2(8m/n)} \sqrt{\frac{(m/2^i)(48e \lg_2(8m/n) \operatorname{herdisc}(A))^2/n}{(m/2^{i-1})n}} = \sqrt{2}\operatorname{herdisc}(A),$$

which is a contradiction.

The induction proof implies that after completing the last iteration, $i = \lg_2(8m/n)$, of the first for-loop, there are no more than $m/2^i = n/8$ rows in $A(I - V^T V)$ of norm more than $48e \lg_2(8m/n) \operatorname{herdisc}(A)$. Since all such rows are in the span of the rows of V after completing the second for-loop, we conclude that the algorithm has the claimed guarantees. $\square$

Lemma 2. *Algorithm 2 can be implemented to run in $O(mn^2 \lg(2 + m/n))$ time.*

Proof. In each step of the first for-loop, computing the matrix B can be done in $O(mn^2)$ time. Constructing $\bar{B}$ takes $O(mn)$ time for computing norms (and $O(m)$ time for finding the largest $m/2^{i-1}$ of them). Computing $\bar{B}^T \bar{B}$ takes $O(n^2 m/2^i)$ time and computing an eigendecomposition of $\bar{B}^T \bar{B}$ takes $O(n^3)$ time. There are $O(\lg(2 + m/n))$ iterations of the for-loop, for a total running time of $O((mn^2 + n^3) \lg(2 + m/n))$. The second for-loop has $O(n)$ iterations, each taking $O(n^2)$ time when calling Orthogonalize. Hence we conclude that the total running time is $O((mn^2 + n^3) \lg(2 + m/n)) = O(mn^2 \lg(2 + m/n))$. $\qquad\square$

Lemma 1 and Lemma 2 together imply Theorem 3.

3.2 Discrepancy Minimization with Hereditary Guarantees

In this section, we give our new algorithm for discrepancy minimization with hereditary guarantees. The algorithm follows the outline in Section 1.1, combined with the *partial coloring* technique. In more detail, given an input $m \times n$ matrix A, we initialize a coloring $x = 0 \in \mathbb{R}^n$. We then repeatedly update x such that more and more entries become either -1 or 1. In each iteration, the number of coordinates x_i with $|x_i| < 1$ halves, while no coordinate ever exceeds 1 in absolute value. While changing the coordinates of x, we ensure that $\|Ax\|_\infty$ changes by as little as possible.

To implement the above strategy, the algorithm presented further below as Algorithm 4 (PartialColoring), takes as input a matrix $A \in \mathbb{R}^{m \times n}$ and a partial coloring $x \in (-1, 1)^n$. It then returns another vector $x' \in [-1, 1]^n$ such that at least $n/2$ coordinates x_i' are in $\{-1, 1\}$, and moreover, $|\|Ax'\|_\infty - \|Ax\|_\infty| \leq O(\lg^{3/2}(2m/n) \cdot \text{herdisc}(A))$. This is stated formally here:

Theorem 5. *For any input matrix $A \in \mathbb{R}^{m \times n}$ with $m \geq n$, and any partial coloring $x \in (-1, 1)^n$, it holds with probability at least 0.08 that Algorithm 4 (PartialColoring) on input (A, x), returns a vector x' for which $\|x'\|_\infty = 1$, $|\{i : |x_i'| = 1\}| \geq n/2$ and for all rows a_i of A, it holds that $|\langle a_i, x' \rangle - \langle a_i, x \rangle| \leq O(\lg^{3/2}(2m/n) \cdot \text{herdisc}(A))$. With the remaining probability, Algorithm 4 returns **Failure**. Moreover, Algorithm 4 can be implemented to run in expected $O(mn^2 \lg(2 + m/n))$ time.*

Using Algorithm 4 as a subroutine, our final discrepancy minimization algorithm is presented as Algorithm 3 (HereditaryMinimize).

Algorithm 3: HereditaryMinimize

   Input: Matrix $A \in \mathbb{R}^{m \times n}$
   Result: Coloring $x \in \{-1, 1\}^n$
1 Let $x \leftarrow 0 \in \mathbb{R}^n$.
2 **while** $x \notin \{-1, 1\}^n$ **do**
3   Let $S \subseteq [n]$ be the subset of indices i such that $|x_i| < 1$.
4   Let $\bar{x} \leftarrow x_S$ be the coordinates of x indexed by S.
5   Let $\bar{A} \leftarrow A_S$ be the columns of A indexed by S.
6   Finished $\leftarrow$ False
7   **while** *not Finished* **do**
8 $x' \leftarrow$ PartialColoring$(\bar{A}, \bar{x})$.   // Call Algorithm 4
9   **if** $x' \neq$ **Failure** **then**
10 $x_S \leftarrow x'$.   // Replace coordinates in x indexed by S, by the new vector x'
11   Finished $\leftarrow$ True
12 **return** x

Let us see that Algorithm 3 (HereditaryMinimize) gives the guarantees stated in our main result, Theorem 1. For the bound on the discrepancy of the returned coloring x, observe that Algorithm 3 has at most $\lg_2 n$ iterations of the outer while loop, since each iterations halves the number of entries x_i with $|x_i| \notin \{-1, 1\}$

(by Theorem 5). In the i'th iteration, we have $|S| \le n/2^i$. Thus replacing the entries in x_S by x' in line 10 increases $\|Ax'\|_\infty$ by at most an additive $O(\lg^{3/2}(m/|S|)\operatorname{herdisc}(\bar{A})) = O(\lg^{3/2} m \operatorname{herdisc}(A))$ compared to $\|Ax\|_\infty$ (here we use that $\operatorname{herdisc}(\bar{A}) \le \operatorname{herdisc}(A)$). Thus the final coloring x satisfies $\operatorname{disc}(A, x) = O(\lg n \cdot \lg^{3/2} m \cdot \operatorname{herdisc}(A))$ as claimed in Theorem 1.

As an interesting remark, notice that if we define $\operatorname{herdisc}(A, k)$ is the maximum discrepancy over all $m \times k$ submatrices of A, then the discrepancy of the returned coloring is bounded by

$$O\left(\sum_{i=0}^{\lg_2 n} \lg^{3/2}(m2^{i+1}/n) \cdot \operatorname{herdisc}(A, n/2^i) \right).$$

This may be an improvement for some matrices A. For instance in the case of geometric set systems, it is often the case that $\operatorname{herdisc}(A, k)$ grows as a polynomial in k. In that case, the sum is dominated by the first term, $i = 0$, and the discrepancy improves to $O(\lg^{3/2}(2m/n) \operatorname{herdisc}(A))$.

For the running time, notice that if we compute the set S for each iteration of the outer while loop by examining only the coordinates x_i with $i \in S$ from the previous iteration, then S can be maintained in total time $O(n)$ throughout the execution of the algorithm ($|S|$ halves with each iteration). Extracting the submatrices $\bar{A}$ takes a total of $O(mn)$ time by the same argument. Finally, in the i'th iteration of the outer while loop, we have $|S| \le n/2^i$. Hence by Theorem 5, the call in line 8 to PartialColoring (Algorithm 4) takes $O(m(n/2^i)^2 \lg(2 + 2^i m/n)) = O(mn^2 \lg(2 + m/n)2^{-i})$ time in expectation. Moreover, the expected number of times it needs to be called before not returning **Failure**, is at most $1/0.08 = O(1)$. Hence the total running time for all calls to PartialColoring is $O(mn^2 \lg(2 + m/n))$ in expectation.

What remains is thus to specify the PartialColoring algorithm (Algorithm 4). As mentioned in Section 1.1, the basic idea is to use the structural properties of matrices with low hereditary discrepancy in order to find a matrix V for which $A(I - V^T V)$ has rows of small norm. From there on, we basically execute the Edge-Walk algorithm by Lovett and Meka, except that we optimized it a bit by taking large step sizes. This is accomodated by capping the magnitude of steps taken, such that no coordinate of x_i ever exceeds 1 in absolute value (see line 8-9). The algorithm is presented in details here:

First we show that Algorithm 4, on an input $x \in (-1, 1)^n$, rarely fails in line 14:

Lemma 3. *On any input $A \in \mathbb{R}^{m \times n}$ and any vector $x \in (-1, 1)^n$, the probability that Algorithm 4 returns* **Failure** *in line 14 is at most $1/86$.*

Proof. We need to bound $\Pr[|\langle a_i, v + g \rangle| > \tau + \eta]$. Examining Algorithm 4, we observe that if $|\langle a_i, v + g \rangle| \ge \tau$ in line 12, we add a_i as a row of V or abort with **Failure**. If we add a_i as a row, this implies that all subsequent sampled vectors g will be orthogonal to a_i due to line 7. Hence the only way it could happen that $|\langle a_i, v + g \rangle| \ge \tau + \eta$, is, if during some iteration of the for-loop, the final vector g in line 9 results in $|\langle a_i, v + g \rangle| > \tau + \eta$ while $|\langle a_i, v \rangle| < \tau$ (from the previous iteration). By linearity of inner product, this can only happen if $|\langle a_i, g \rangle| > \eta$. Consider line 9 of Algorithm 4. There we set g to $\min\{\varepsilon, \mu\}g$. Hence the probability that $|\langle a_i, g \rangle| > \eta$ is bounded by the probability that a vector $\tilde{g}$ sampled with n i.i.d. $\mathcal{N}(0, 1)$ entries and then projected to $\tilde{g} \leftarrow \varepsilon \tilde{g}(I - V^T V)$ satisfies $|\langle a_i, \tilde{g} \rangle| > \eta$. The distribution of $\langle a_i, \tilde{g} \rangle$ for such $\tilde{g}$ is $\mathcal{N}(0, \varepsilon^2 \|a_i(I - V^T V)\|^2)$ by Claim 2, i.e. normal distributed with mean 0 and variance at most $\sigma^2 \le \varepsilon^2 \eta^2$. This is equal in distribution to $\sigma \mathcal{N}(0, 1)$ and thus by Claim 1, we have $\Pr[|\langle a_i, \tilde{g} \rangle| > \eta] \le 2 \exp(-(\eta^2/\sigma^2)/2) \le 2 \exp(-\varepsilon^{-2}/2)$. We now define an event $E_{t,i}$ for every iteration t of the for-loop and every row i which occurs if the vector g sampled in lines 6-7 during iteration t and then scaled by ε satisfies $|\langle a_i, g \rangle| > \eta$. The event $E_{t,i}$ does not occur if Algorithm 4 already returned **Failure** in an iteration $t' < t$. By the previous arguments, we have $\Pr[E_{t,i}] \le 2 \exp(-\varepsilon^{-2}/2)$. Since $\varepsilon \le (4 \ln(mn) + 20)^{-1/2}$, this is less than $1/(nm)^3$. A union bound over all $E_{t,i}$ for $i = 1, \ldots, m$ and all $t = 1, \ldots, Q$ shows that the probability that Algorithm 4 returns a vector $x' = x + v$ such that there is a row a_i with $|\langle a_i, x' \rangle| - |\langle a_i, x \rangle| > \tau + \eta$ is at most $Qm/((nm)^2 e^{10}) \le 1/86$. $\square$

We next want to bound the probability that the algorithm fails in line 20. For this, we first need a handle on the number of rows a_i that are added to V in line 16:

Algorithm 4: PartialColoring

Input: Matrix $A \in \mathbb{R}^{m \times n}$

   Partial coloring $x \in (-1, 1)^n$

Result: Vector $x \in \mathbb{R}^n$

1 $v \leftarrow 0 \in \mathbb{R}^n$.

2 $V \leftarrow \text{ProjectToSmallRows}(A)$ `// Call Algorithm 2`

3 $\eta \leftarrow \max_i \|a_i(I - V^T V)\|$ `// Largest norm of a row in` $A(I - V^T V)$

4 $\varepsilon \leftarrow (\max\{4\ln(mn) + 20, 256n\})^{-1/2}$, $Q \leftarrow 16\varepsilon^{-2} + 256n$, $\tau \leftarrow 22\varepsilon\eta\sqrt{Q \lg(256m/n)}$

5 **for** $t = 1, \ldots, Q$ **do**

6 $\quad$ Sample g with n i.i.d. $\mathcal{N}(0,1)$ entries.

7 $\quad$ $g \leftarrow g(I - V^T V)$.   `// Sample g in orthogonal complement of rows of V`

8 $\quad$ Let $\mu > 0$ be maximal such that $\max\{\|x + v + \mu g\|_\infty, \|x + v - \mu g\|_\infty\} = 1$.

9 $\quad$ $g \leftarrow \min\{\varepsilon, \mu\}g$.

10 $\quad$ **for** *every* i *such that* $|x_i + v_i + g_i| = 1$ *and* $|x_i + v_i| < 1$ **do**

11 $\quad\quad$ $\text{Orthogonalize}(e_i, V)$ `// Add` e_i `as row of V via Algorithm 1`

12 $\quad$ **for** *every row* a_i *of* A *such that* $|\langle a_i, v + g\rangle| \geq \tau$ *and* $|\langle a_i, v\rangle| < \tau$ **do**

13 $\quad\quad$ **if** $|\langle a_i, v + g\rangle| > \tau + \eta$ **then**

14 $\quad\quad\quad$ **return** *Failure*

15 $\quad\quad$ **else**

16 $\quad\quad\quad$ $\text{Orthogonalize}(a_i, V)$ `// Add` a_i `as row of V via Algorithm 1`

17 $\quad$ $v \leftarrow v + g$.

18 $\quad$ **if** $|\{i : |x_i + v_i| = 1\}| \geq n/2$ **then**

19 $\quad\quad$ **return** $x + v$

20 **return** *Failure*

Lemma 4. *For any input $A \in \mathbb{R}^{m \times n}$ and any vector $x \in (-1, 1)^n$, let R denote the random variable giving the number of rows a_i added to V in line 16 throughout the execution of Algorithm 4. Then $\mathbb{E}[R] \leq n/256$.*

Proof. Fix a row a_i. Then a_i is added to V only if $|\langle a_i, v\rangle|$ reaches τ during some iteration of the for-loop. Also, since we then add it to V, it will remain in V for all remaining iterations. This implies that all subsequently sampled g's will leave $\langle a_i, v\rangle$ unchanged. Consider some iteration t of the for-loop and let g_t denote the vector formed in line 9 of Algorithm 4 in iteration t. If Algorithm 4 returns **Failure** before iteration t, define $g_t = 0$. Define also $v_t = \sum_{i=1}^{t} g_i$. Then $\langle a_i, v_t\rangle = \sum_{j=1}^{t}\langle a_i, g_j\rangle$. Since $\langle a_i, v\rangle$ remains unchanged if a_i is added to V or if we return **Failure**, it follows that $|\langle a_i, v_t\rangle| \geq \tau$ during some iteration t, only if $|\langle a_i, v_Q\rangle| \geq \tau$. Hence we bound the probability that $|\langle a_i, v_Q\rangle| \geq \tau$.

Observe now that if we condition on $v_{t-1}, \ldots, v_1$, and let V_t denote the set V at the beginning of iteration t (which is completely determined from $v_{t-1}, \ldots, v_1$ and the input A, x), and then draw a vector $\tilde{g}$ with each coordinate i.i.d. $\mathcal{N}(0,1)$ distributed, then for any $t > 0$, we have $\Pr[|\varepsilon\langle a_i, \tilde{g}^T(I - V_t^T V_t)\rangle| > t] \geq \Pr[|\min\{\varepsilon, \mu\}\langle a_i, \tilde{g}^T(I - V_t^T V_t)\rangle| > t] \geq \Pr[|\langle a_i, g_t\rangle| > t]$. We also have that $\varepsilon\langle a_i, \tilde{g}^T(I - V_t^T V_t)\rangle$ is $\mathcal{N}(0, \varepsilon^2\|a_i(I - V_t^T V_t)\|^2)$ distributed. The variance σ^2 is at most $\varepsilon^2\eta^2$, hence by Claim 1, it holds that $\Pr[|\langle a_i, g_t\rangle| > t] \leq \Pr[|\varepsilon\langle a_i, \tilde{g}(I - V_t^T V_t)\rangle| > t] \leq 2\exp(-(\varepsilon\eta)^{-2}t^2/2)$. Finally, observe that $\mathbb{E}[\langle a_i, g_t\rangle \mid v_{t-1}, \ldots, v_1] = 0$. This follows by symmetry of the distribution of g_t and crucially uses that we define μ in line 8 to be such that $\max\{\|x + v + \mu g\|_\infty, \|x + v - \mu g\|_\infty\} = 1$. If we had only defined μ such that $\|x + v + \mu g\|_\infty = 1$ (we add μg to v, not $-\mu g$), then the distribution of g would not be symmetric. Thus $\langle a_i, g_1\rangle, \ldots, \langle a_i, g_t\rangle$ forms a martingale difference sequence with respect to $v_1, \ldots, v_t$, where the parameter c in Theorem 4 is at least $(\varepsilon\eta)^{-2}/2$ and the parameter b is 2. Theorem 4 implies that with probability at least $1 - \delta$, we have

$$|\langle a_i, v_Q\rangle| = \left|\sum_{t=1}^{Q}\langle a_i, g_t\rangle\right| \leq 2\varepsilon\eta\sqrt{112Q \lg(1/\delta)} \leq 22\varepsilon\eta\sqrt{Q \lg(1/\delta)}.$$

9

Setting $\delta = n/(256m)$, it follows by linearity of expectation that the expected number of rows added to V throughout the execution of Algorithm 4 is $n/256$, i.e. $\mathbb{E}[R] \leq n/256$. $\qquad\square$

With Lemma 4 established, we are finally ready to bound the probability of returning **Failure** in line 20:

Lemma 5. *On any input $A \in \mathbb{R}^{m \times n}$ and any vector $x \in (-1, 1)^n$, the probability that Algorithm 4 returns* **Failure** *in line 20 is at most $9/10$.*

Proof. The basic idea in the proof is to show that $\mathbb{E}[\|x + v\|^2]$ is large after the Q iterations and thus many coordinates of $x + v$ must reach 1 in absolute value. With every iteration, we add either εg or μg to v. Since μ may be the minimum, and thus result in a smaller increase in the norm of v, we first bound the probability that μ is the minimum in line 9. Consider line 8-9 of Algorithm 4. If $\mu < \varepsilon$, we have that μ is maximal for at least one of the constraints $\|x + v + \mu g\|_\infty = 1$ and $\|x + v - \mu g\|_\infty = 1$. In particular, for line 10 of Algorithm 4, this implies that at least one index i must satisfy $|x_i + v_i + \sigma g| = 1$ and $|x_i + v_i| < 1$ for a sign $\sigma \in \{-1, 1\}$. Let ρ_t denote the probability that Algorithm 4 reaches iteration t and that $\mu < \varepsilon$ in line 9 during iteration t. Due to the symmetry of g (g and $-g$ are equally likely conditioned on V), we get that the expected number of indices i such that $|x_i + v_i + \mu g| = 1$ (in iteration t) and $|x_i + v_i| < 1$, is at least $\rho_t/2$. Thus the expected number of vectors e_i added to V in line 10-11 throughout the execution of Algorithm 4 is at least $\sum_{t=1}^{Q} \rho_t/2$. Let X_t give the number of e_i's added to V during iteration t. Then $\mathbb{E}[\sum_t X_t] \geq \sum_t \rho_t/2$. On the other hand, the sum of X_t's can never exceed n, thus we conclude $\sum_t \rho_t \leq 2n$.

Next, define g_t as the vector g added to v in iteration t of Algorithm 4 (define it to $g_t = 0$ if the algorithm terminates with **Failure** before iteration t). Define $v = \sum_{i=1}^{Q} g_t$. If Algorithm 4 does not fail, then v is the returned vector. We have:

$$
\begin{aligned}
\mathbb{E}[\|x + v\|^2] &= \mathbb{E}\left[\left\|x + \sum_{t=1}^{Q} g_t\right\|^2\right] \\
&= \|x\|^2 + \sum_{i=1}^{Q} \mathbb{E}[\|g_t\|^2].
\end{aligned}
$$

Here the second equality holds because $\mathbb{E}[g_t \mid g_1, \ldots, g_{t-1}] = 0$ by symmetry of the distribution of g_t. To bound $\mathbb{E}[\|g_t\|^2]$, let V_t be the random variable giving the matrix V at the beginning of iteration t of the for-loop. If the algorithm fails to reach iteration t, we define V_t to equal V at the end of the last reached iteration. Let $\tilde{g}_t$ be the random variable giving the value of g during line 6 of iteration t, i.e. before projection and scaling by $\min\{\varepsilon, \mu\}$. If the algorithm terminates before iteration t, we still define $\tilde{g}_t$ as a random variable with n i.i.d. $\mathcal{N}(0, 1)$ distributed entries. Define F_t as an indicator taking the value 1 if the algorithm reaches iteration t. Define Y_t as the indicator taking the value 1 if the algorithm reaches iteration t and $\mu < \varepsilon$ in line 9 during iteration t. Then

$$
\begin{aligned}
\mathbb{E}[\|g_t\|^2] &= \mathbb{E}[F_t Y_t \mu^2 \|\tilde{g}_t(I - V_t^T V_t)\|^2 + F_t(1 - Y_t)\varepsilon^2 \|\tilde{g}_t(I - V_t^T V_t)\|^2] \\
&\geq \mathbb{E}[F_t(1 - Y_t)\varepsilon^2 \|\tilde{g}_t(I - V_t^T V_t)\|^2] \\
&\geq \varepsilon^2 \mathbb{E}[F_t \|\tilde{g}_t(I - V_t^T V_t)\|^2] - \varepsilon^2 \mathbb{E}[Y_t \|\tilde{g}_t(I - V_t^T V_t)\|^2].
\end{aligned}
$$

By Cauchy-Schwartz, we have $\mathbb{E}[Y_t \|\tilde{g}_t(I - V_t^T V_t)\|^2] \leq \sqrt{\mathbb{E}[Y_t^2]\mathbb{E}[\|\tilde{g}_t(I - V_t^T V_t)\|^4]}$. Since Y_t is an indicator, we have $\mathbb{E}[Y_t^2] = E[Y_t]$. We also have $\|\tilde{g}_t\| \geq \|\tilde{g}_t(I - V_t^T V_t)\|$. Therefore, it holds that $\mathbb{E}[Y_t \|\tilde{g}_t(I - V_t^T V_t)\|^2] \leq \sqrt{\mathbb{E}[Y_t]\mathbb{E}[\|\tilde{g}_t\|^4]} = \sqrt{\rho_t \mathbb{E}[\|\tilde{g}_t\|^4]}$. Let c_i denote the i'th coordinate of $\tilde{g}_t$. We have that the c_i's are i.i.d. $\mathcal{N}(0, 1)$ distributed. By linearity of expectation:

$$
\begin{aligned}
\mathbb{E}[\|\tilde{g}_t\|^4] &= \mathbb{E}\left[\left(\sum_{i=1}^{n} c_i^2\right)^2\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}[c_i^2 c_j^2].
\end{aligned}
$$

10

For $i \neq j$, we have $\mathbb{E}[c_i^2 c_j^2] = \mathbb{E}[c_i^2]\mathbb{E}[c_j^2] = 1$ by independence. For $i = j$, we have $\mathbb{E}[c_i^4] = 3$ (by well-known bounds on the 4'th moment of the normal distribution). Thus $\mathbb{E}[\|\tilde{g}_t\|^4] \leq n(n-1) + 3n \leq 2n^2$. We thus have $\mathbb{E}[Y_t \|\tilde{g}_t(I - V_t^T V_t)\|^2] \leq \sqrt{2\rho_t}n$.

At the same time, we have $\mathbb{E}[F_t \|\tilde{g}_t(I - V_t^T V_t)\|^2] = \Pr[F_t = 1]\mathbb{E}[\|\tilde{g}_t(I - V_t^T V_t)\|^2] = \Pr[F_t = 1](\mathbb{E}[n - \dim(V_t)]) = \Pr[F_t = 1](n - \mathbb{E}[\dim(V_t)])$. Now recall the definition of the random variable R as the number of rows added to V in line 16 throughout the execution of Algorithm 4. Define also X as the random variable giving the total number of e_i's added to V throughout the execution of Algorithm 4. Then $\dim(V_t) \leq R + X + n/4$ for all $t = 1, \ldots, Q$. Hence $\mathbb{E}[\dim(V_t)] \leq \mathbb{E}[R] + \mathbb{E}[X] + n/4$. By Lemma 4, we have $\mathbb{E}[R] \leq n/256$, thus $\mathbb{E}[\dim(V_t)] \leq (65/256)n + \mathbb{E}[X]$. Thus we conclude $\mathbb{E}[\|g_t\|^2] \geq \Pr[F_t = 1]\varepsilon^2(n - (65/256)n - \mathbb{E}[X]) - \varepsilon^2\sqrt{2\rho_t}n$. By Lemma 3, we have $\Pr[F_t = 1] \geq 85/86$. Hence $\mathbb{E}[\|g_t\|^2] \geq (85/86)\varepsilon^2(n - (65/256)n - \mathbb{E}[X]) - \varepsilon^2\sqrt{2\rho_t}n$.

Assume for now that $\rho_t \leq 1/128$. Then this is at least $(85/86)\varepsilon^2(n - (65/256)n - \mathbb{E}[X] - (86/85)n/8) \geq (85/86)\varepsilon^2(n - (65/256)n - \mathbb{E}[X] - (33/256)n) = (85/86)\varepsilon^2((158/256)n - \mathbb{E}[X])$. If $\rho_t > 1/128$, we simply lower bound $\mathbb{E}[\|g_t\|^2]$ by 0. Since $\sum_t \rho_t \leq 2n$, we can have at most $256n$ iterations t for which $\rho_t \geq 1/128$. For the remaining $Q - 256n$ iterations, we have $\mathbb{E}[\|g_t\|^2] \geq (85/86)\varepsilon^2((158/256)n - \mathbb{E}[X])$. Therefore $\mathbb{E}[\|x + v\|^2] \geq \|x\|^2 + (Q - 256n)(85/86)\varepsilon^2((158/256)n - \mathbb{E}[X]) \geq (Q - 256n)(85/86)\varepsilon^2((158/256)n - \mathbb{E}[X]) = 16(85/86)((158/256)n - \mathbb{E}[X])$. Notice now that Algorithm 4 can never produce an output v for which $\|x + v\|_\infty > 1$. Hence it must be the case that $\mathbb{E}[\|x + v\|^2] \leq n$. This implies $n \geq 16(85/86)((158/256)n - \mathbb{E}[X])$. Rewriting in terms of $\mathbb{E}[X]$, we get $\mathbb{E}[X] \geq (158/256 - (86/85)(1/16))n > 0.553n$. Now define $Z = n - X$. Then Z is a non-negative random variable and $\mathbb{E}[Z] = n - \mathbb{E}[X] \leq 0.447n$. By Markov's inequality, we have $\Pr[Z > n/2] < 2 \cdot 0.447 = 0.894 < 0.9$. Conversely, $\Pr[Z \leq n/2] > 0.1$. This implies $\Pr[X \geq n/2] > 0.1$. When $X \geq n/2$, Algorithm 4 is guaranteed to terminate in line 19 and thus we conclude that the probability of returning failure in line 20 is at most $9/10$. $\square$

With these lemmas established, we may finally conclude the proof of Theorem 5.

Proof of Theorem 5. By Lemma 3, Lemma 5 and a union bound, it holds with probability at least $1/10 - 1/86 > 0.08$ that Algorithm 4 does not return **Failure**, and at the same time, every row a_i satisfies $|\langle a_i, x'\rangle - \langle a_i, x\rangle| = |\langle a_i, x' - x\rangle| = |\langle a_i, v\rangle| \leq \tau + \eta$, where x' is the returned vector. We have $Q = O(n + \max\{\ln(nm), n\}) = O(n + \ln m)$ and $\varepsilon = O(\min\{\ln(nm)^{-1/2}, n^{-1/2}\})$. Hence

$$\tau = O(\eta \min\{\ln(nm)^{-1/2}, n^{-1/2}\}\sqrt{(n + \ln m)}\sqrt{\lg(2m/n)}) = O(\eta\sqrt{\lg(2m/n)}).$$

The conclusion follows by invoking Lemma 1 to conclude that $\eta = O(\lg(2m/n) \operatorname{herdisc}(A))$.

For the running time, notice that the call to Algorithm 2 in line 2 takes $O(mn^2 \lg(2 + m/n))$ time by Lemma 2. Line 3 takes $O(mn^2)$ time. There are $Q = O(\varepsilon^{-2} + n) = O(\max\{\ln(nm), n\} + n) = O(n + \lg m)$ iterations of the for-loop. In each iteration, line 6-7 takes $O(n^2)$ time. Line 8 takes $O(mn)$ time. Checking the conditions in line 10, 12 and 18 takes $O(mn)$ time per iteration. Thus ignoring the calls to Orthogonalize, each iteration of the for-loop takes $O(mn + n^2) = O(mn)$ time. Given that there are $O(n + \lg m)$ iterations, the total running time is $O(mn(n + \lg m))$. The calls to Orthogonalize in line 11 and 16 are executed a total of $O(n)$ times in expectation (by Lemma 4), and any such call takes $O(n^2)$ time. This adds n^3 to the running time, which is dominated by mn^2 for $m \geq n$. The total running time is thus $O(mn(n + \lg m) + mn^2 \lg(2 + m/n)) = O(mn^2 \lg(2 + m/n))$ in expectation. $\square$

4 Experiments

In this section, we present a number of experiments to test the practical performance of our discrepancy minimization algorithm. We denote the algorithm by HEREDITARYMINIMIZE in the following. We compare it to two base line algorithms SAMPLE and SAMPLEMANY. SAMPLE simply picks a uniform random $\{-1, +1\}$ vector as its coloring. SAMPLEMANY repeatedly samples a uniform random $\{-1, +1\}$ vector and runs for the same amount of time as HEREDITARYMINIMIZE. It returns the best vector found within the time limit. For SAMPLE, we ran multiple experiments and took the median result (in fact, we use the median across all the runs of SAMPLEMANY when stating the discrepancy of SAMPLE).

The algorithms were implemented in Python, using NumPy and SciPy for linear algebra operations. All tests were run on a MacBook Pro (16-inch, 2019) running macOS Monterey 12.4. The machine has a 2.6 GHz 6-Core Intel Core i7 and 16GB 2667 MHz DDR4 RAM.

We tested the algorithms on three different classes of matrices:

- **Uniform** matrices: Each entry is uniform random and independently chosen among -1 and $+1$.

- **2D Corner** matrices: Obtained by sampling two sets $P = \{p_1, \ldots, p_n\}$ and $Q = \{q_1, \ldots, q_m\}$ of n and m points in the plane, respectively. The points are sampled uniformly in the $[0,1] \times [0,1]$ unit square. The resulting matrix has one column per point $p_j \in P$ and one row per point $q_i \in Q$. The entry (i,j) is 1 if p_j is *dominated* by q_i, i.e. $q_i.x > p_j.x$ and $q_i.y > p_j.y$ and it is 0 otherwise. Such matrices are known to have hereditary discrepancy $O(\lg^{2.5} n)$ [25, 13].

- **2D Halfspace** matrices: Obtained by sampling a set $P = \{p_1, \ldots, p_n\}$ of n points in the unit square $[0,1] \times [0,1]$, and a set Q of m halfspace. Each halfspace in Q is sampled by picking one point a uniformly on either the left boundary of the unit square or on the top boundary, and another point b uniformly on either the right boundary or the bottom boundary of the unit square. The halfspace is then chosen uniformly to be either everything above the line through a, b or everything below it. The resulting matrix has one column per point $p_j \in P$ and one row per halfspace $h_i \in Q$. The entry (i,j) is 1 if p_j is in the halfspace h_i and it is 0 otherwise. Such matrices are known to have hereditary discrepancy $O(n^{1/4})$ [17].

The running times of the algorithms varied exclusively with the matrix size and not the type of matrix, thus we only show one time column which is representative of all types of matrices. The results are shown in Table 1.

Algorithm	Matrix Size	Disc Uniform	Disc 2D Corner	Disc 2D Halfspace	Time (s)
HEREDITARYMINIMIZE	200×200	24	3	4	< 1
SAMPLE	200×200	42	17	19	< 1
SAMPLEMANY	200×200	28	6	6	< 1
HEREDITARYMINIMIZE	1000×1000	56	6	8	24
SAMPLE	1000×1000	108	40	47	< 1
SAMPLEMANY	1000×1000	80	15	18	24
HEREDITARYMINIMIZE	4000×4000	122	8	11	1637
SAMPLE	4000×4000	238	84	99	< 1
SAMPLEMANY	4000×4000	186	32	36	1637
HEREDITARYMINIMIZE	10000×2000	124	9	11	1479
SAMPLE	10000×2000	178	61	71	< 1
SAMPLEMANY	10000×2000	142	25	29	1479

Table 1: Results of experiments with our HEREDITARYMINIMIZE algorithm. The Matrix Size column gives the size $m \times n$ of the input matrix. The Disc columns show $\text{disc}(A, x) = \|Ax\|_\infty$ for the coloring x found by the algorithm on the given type of matrix. Time is measured in seconds.

The table clearly shows that HEREDITARYMINIMIZE gives superior colorings for all types of matrices and all sizes. The tendency is particularly clear on the structured matrices **2D Corner** and **2D Halfspace** where the coloring found by HEREDITARYMINIMIZE on 4000×4000 matrices is a factor roughly 10 smaller than a single round of random sampling (SAMPLE) and a factor roughly 4 better than random sampling for as long time as HEREDITARYMINIMIZE runs (SAMPLEMANY).

5 Deferred Proofs

In this section, we give the deferred proof from Section 2 and also the reduction from $n > m$ to $m \geq n$ case of discrepancy minimization.

Proof of Claim 2. Let $v_1, \ldots, v_\ell$ denote the rows of V and let $u_1, \ldots, u_{n-\ell}$ be an orthonormal basis for the orthogonal complement of $v_1, \ldots, v_\ell$. By the rotational invariance of the gaussian distribution, we have that g is distributed as $\sum_{i=1}^{\ell} X_i v_i + \sum_{j=1}^{n-\ell} Y_j u_j$ where all X_i and Y_i are independent $\mathcal{N}(0,1)$ random variables. Write $a = \sum_{i=1}^{\ell} \alpha_i v_i + \sum_{j=1}^{n-\ell} \beta_j u_j$. Then

$$
\begin{aligned}
\langle a, (I - V^T V)g \rangle &= \sum_{i=1}^{\ell} \alpha_i v_i^T (I - V^T V)g + \sum_{j=1}^{n-\ell} \beta_j u_j^T (I - V^T V)g \\
&= \sum_{i=1}^{\ell} \alpha_i (v_i^T - v_i^T)g + \sum_{j=1}^{n-\ell} \beta_j u_j^T g \\
&= \sum_{j=1}^{n-\ell} \beta_j u_j^T \left(\sum_{i=1}^{\ell} X_i v_i + \sum_{h=1}^{n-\ell} Y_h u_h \right) \\
&= \sum_{j=1}^{n-\ell} \beta_j Y_j \\
&\sim \mathcal{N}\left(0, \sum_{j=1}^{n-\ell} \beta_j^2 \right) \\
&\sim \mathcal{N}\left(0, \|a(I - V^T V)\|^2 \right).
\end{aligned}
$$

$\square$

Reducing to $m \geq n$. Assume the input matrix A has $m < n$. Initialize an empty matrix V $(0 \times n)$ and add all rows of A to V using Orthogonalize (Algorithm 1). Also initialize a coloring $x = 0 \in \mathbb{R}^n$. While the number of rows in V is less than n, there is a non-zero vector g in the orthogonal complement of the rows of V. Such a vector g can e.g. be found by sampling each coordinate as an independent $\mathcal{N}(0,1)$ distributed random variable and then updating $g \leftarrow g(I - V^T V)$. Given such a g, observe that $\langle a_i, g \rangle = 0$ for all rows a_i of A. Now compute the largest coefficient ε such that $\|x + \varepsilon g\|_\infty = 1$, i.e. ε is the largest scaling such that at least one coordinate of $x + g$ becomes 1 in absolute value. Update x by adding εg to it. For every entry x_i that becomes 1 in absolute value, add e_i to V using Algorithm 1. Repeat until there are n rows in V.

We observe that when the above process terminates, we still have $Ax = 0$. Moreover, we must have added $n - m$ vectors e_i to V using Algorithm 1. Hence there are at most $n - (n - m) \leq m$ coordinates left that are not in $\{-1, 1\}$. We may now use PartialColoring starting from this vector x.

For the running time, first notice that adding all rows of A to V takes time $O(mn^2)$ since a call to Algorithm 1 takes $O(n^2)$ time. Next, observe that each iteration adds at least one e_i to V. Thus we have at most n iterations. In any iteration, computing g takes $O(n^2)$ time. Computing the coefficient ε takes $O(mn)$ time and adding any e_i to V takes $O(n^2)$ time. Thus the total running time is $O(mn^2 + n^3)$.

References

[1] R. Alexander. Geometric methods in the study of irregularities of distribution. *Combinatorica*, 10(2):115–136, 1990.

[2] R. Alweiss, Y. P. Liu, and M. Sawhney. Discrepancy minimization via a self-balancing walk. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, 2021*, pages 14–20. ACM, 2021.

[3] W. Banaszczyk. Balancing vectors and gaussian measures of n-dimensional convex bodies. *Random Structures & Algorithms*, 12:351–360, July 1998.

[4] N. Bansal. Constructive algorithms for discrepancy minimization. In *Proc. 51th Annual IEEE Symposium on Foundations of Computer Science (FOCS'10)*, pages 3–10, 2010.

[5] N. Bansal, D. Dadush, and S. Garg. An algorithm for komlós conjecture matching banaszczyk's bound. In *Proc. 57th IEEE Annual Symposium on Foundations of Computer Science (FOCS'16)*, pages 788–799, 2016.

[6] J. Beck and T. Fiala. Integer-making theorems. *Discrete Applied Mathematics*, 3:1–8, February 1981.

[7] M. Charikar, A. Newman, and A. Nikolov. Tight hardness results for minimizing discrepancy. In *Proc. 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 1607–1614, 2011.

[8] B. Chazelle. *The Discrepancy Method: Randomness and Complexity.* Cambridge University Press, 2000.

[9] B. Chazelle and A. Lvov. A trace bound for the hereditary discrepancy. In *Proc. 16th Annual Symposium on Computational Geometry*, SCG '00, pages 64–69, 2000.

[10] D. Dadush, A. Nikolov, K. Talwar, and N. Tomczak-Jaegermann. Balancing vectors in any norm. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1–10, 2018.

[11] R. Eldan and M. Singh. Efficient algorithms for discrepancy minimization in convex sets. *Random Structures & Algorithms*, 53:289 – 307, 2018.

[12] H. Jiang, T. Kathuria, Y. T. Lee, S. Padmanabhan, and Z. Song. A faster interior point method for semidefinite programming. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*, pages 910–918. IEEE, 2020.

[13] K. G. Larsen. On range searching in the group model and combinatorial discrepancy. *SIAM Journal on Computing*, 43(2):673–686, 2014.

[14] K. G. Larsen. Constructive discrepancy minimization with hereditary L2 guarantees. In *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019*, volume 126 of *LIPIcs*, pages 48:1–48:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[15] L. Lovász, J. Spencer, and K. Vesztergombi. Discrepancy of set-systems and matrices. *European Journal of Combinatorics*, 7(2):151 – 160, 1986.

[16] S. Lovett and R. Meka. Constructive discrepancy minimization by walking on the edges. *SIAM Journal on Computing*, 44(5):1573–1582, 2015.

[17] J. Matoušek. Tight upper bounds for the discrepancy of half-spaces. *Discrete and Computational Geometry*, 13:593–601, 1995.

[18] J. Matousek. *Geometric Discrepancy: An Illustrated Guide.* Algorithms and Combinatorics. Springer Berlin Heidelberg, 1999.

[19] J. Matoušek and A. Nikolov. Combinatorial Discrepancy for Boxes via the gamma_2 Norm. In *31st International Symposium on Computational Geometry (SoCG 2015)*, volume 34, pages 1–15, 2015.

[20] J. Matoušek, A. Nikolov, and K. Talwar. Factorization norms and hereditary discrepancy. *CoRR*, abs/1408.1376, 2014.

[21] J. Matoušek, A. Nikolov, and K. Talwar. Factorization Norms and Hereditary Discrepancy. *International Mathematics Research Notices*, 2020(3):751–780, 03 2018.

[22] T. Rothvoss. Constructive discrepancy minimization for convex sets. *SIAM Journal on Computing*, 46(1):224–234, 2017.

[23] O. Shamir. A variant of azuma's inequality for martingales with subgaussian tails. *CoRR*, abs/1110.2392, 2011.

[24] J. Spencer. Six standard deviations suffice. *Trans. Amer. Math. Soc.*, 289:679–706, 1985.

[25] A. Srinivasan. Improving the discrepancy bound for sparse matrices: Better approximations for sparse lattice approximation problems. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, 1997*, pages 692–701. ACM/SIAM, 1997.